UNIVERSITY OF ŽILINA
Faculty of Management Science and Informatics

BRAIN:IT

PROJECT

Multi-tenant Cybersecurity Operations Centre Delivered as an Open Cloud Service with Machine Learning Elements

V13 Detailed Solution Design

KPB4 – Solution

Architecture type –

documentation month of

submission – M22

Funded by the European Union
NextGenerationEU

RECOVERY AND RESILIENCE PLAN

GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC

Contents

1 Abbreviations and Terms Used 5
2 Executive Summary 8
2.1 Innovativeness of the Solution 8
2.2 ML, AI and Cybersecurity 10
2.3 Solution Intent 25
2.4 Solution Objectives 27
2.4.1 Main objective of the project 27
2.4.2 Specific objectives of the project 28
2.5 Work Packages, Outputs, Measurable and Performance Indicators (KPIs) 33
2.5.1 List of work packages 33
2.5.2 List of outputs 34
2.6 Solution Scope 36
2.7 Methodology 36
2.8 Project Impact and Benefits 39
2.9 Wider Impact of the Project 40
3 Requirements Analysis 42
3.1 Requirements Analysis Procedure 42
3.2 User Research 43
3.3 Motivation Architecture 43
3.3.1 Motivation Viewpoints 44
3.4 Requirements Catalogue and Its Mapping to the Design of Their Solution 44
3.5 Legislation, Directives and Methodological Materials Taken into Account in the Solution 45
3.6 Acceptance Criteria 47
4 Business Architecture 47
4.1 Business Viewpoints 47
4.1.1 Detection and Correlation of Security Events (SIEM) – Business Realisation of the Service 47
4.1.2 Response Automation Management (SOAR) – Business Realisation of the Service 53
4.1.3 Threat Response Management (EDR/XDR) – Business Realisation of the Service 58
4.1.4 Vulnerability Monitoring – Business Realisation of the Service 63
4.1.5 Threat Intelligence Management (CTI) – Business Realisation of the Service 68
4.1.6 Asset Inventory – Business View 73
4.1.7 Cybersecurity Incident Management – Business Realisation of the Service 75

4.1.8 Security Reporting Management – Business Realisation of the Service 78
4.1.9 Knowledge portal management for the SOC team – business realisation of the service 84
4.2 BPMN – Business process model 87
4.3 Use cases 88
4.3.1 Actors 88
4.3.2 General use cases 92
5 Data Architecture 206
6.1 Application Viewpoints 211
7 Technology Architecture 214
7.1 Design of the Technology Architecture 215
7.1.1 Technology Viewpoint CTI 215
7.1.2 Technology Viewpoint EDR/XDR 222
7.1.3 Technology Viewpoint INNOSOC Common 226
7.1.4 Technology Viewpoint Asset Inventory 229
7.1.5 Technology Viewpoint Kubernetes 232
7.1.6 Technology Viewpoint OpenStack 251
7.1.7 Technology Viewpoint OpenStack External 255
7.1.8 Technology Viewpoint Reporting 256
7.1.9 Technology Viewpoint SIEM 259
7.1.10 Technology Viewpoint Vulnerability Scanner 262
7.1.11 Technology Viewpoint SOAR 265
7.1.12 Technology Viewpoint Ticketing System / Incident Recording and Management 270
7.1.13 Technology Viewpoint Knowledge Portal for the SOC Team 274
8 Service Level (SLA) and Performance Requirements 276
8.1 Support Levels and Helpdesk 276
8.2 Proposed SLA Parameters and Incident Resolution 277
8.3 Proposed Planned Availability of the Infrastructure 279
9 Software Licences and Source Code 280
10 Ensuring Availability, Backup and Recovery of the Solution 280
10.1 Ensuring Availability 280
10.2 Backup 281
10.3 Recovery of the Solution 281
11 Security – Addressing Security Requirements 281
12 Data Migration 285

12.1 Data Migration Procedure 285
12.2 Verification of Data Migration 285
13 Implementation and Deployment Schedule, Dependencies 285
13.1 Work Packages 286
13.2 Schedule 288
13.3 Project Risks 288
15 Annexes 289

Funded by the European Union
NextGenerationEU
RECOVERY AND RESILIENCE PLAN
GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC

1 Abbreviations and Terms Used

Abbreviation	Full Name	Description
AI	Artificial Intelligence	Artificial intelligence used to support the analysis, automation and detection of security threats.
API	Application Programming Interface	A programming interface enabling integration between systems and solution components.
ArchiMate	Architecture Modeling Language	A modelling language used to describe enterprise architecture.
BPMN	Business Process Model and Notation	A notation for modelling business processes.
CI/CD	Continuous Integration / Continuous Deployment	Practices of continuous integration, testing and deployment of software.
CIA	Confidentiality, Integrity, Availability	The fundamental triad of information security and cybersecurity: confidentiality, integrity, availability.
CTI	Cyber Threat Intelligence	Intelligence on cyber threats, indicators of compromise and attacker behaviour.
DDoS	Distributed Denial of Service	A distributed attack aimed at making a service unavailable.
DNN	Deep Neural Network	A deep neural network used in detection, classification and prediction tasks.
DoS	Denial of Service	An attack aimed at limiting or preventing the availability of a service.
EDR	Endpoint Detection and Response	A tool for detecting and responding to threats on endpoint devices.
IOC	Indicator of Compromise	An indicator of compromise, e.g. a hash, IP address, domain or other artefact associated with an incident.
IPS	Intrusion Prevention System	A system for detecting and blocking security attacks in traffic.
IDS	Intrusion Detection System	A system for detecting suspicious or malicious activities in a network or system.
IAM	Identity and Access Management	Management of identities, roles and access permissions of users and systems.
KPI	Key Performance Indicator	A key performance indicator used to evaluate the success of a service or process.
ML	Machine Learning	Machine learning used for anomaly detection, classification and prediction.

Funded by the European Union NextGenerationEU
RECOVERY AND RESILIENCE PLAN
GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC

MLOps	Machine Learning Operations	Practices for the management, deployment, monitoring and continuous improvement of ML models.
MFA	Multi-Factor Authentication	Multi-factor verification of a user's identity.
NIS2	Network and Information Systems Directive 2	A European directive on measures for a high common level of cybersecurity.
OIDC	OpenID Connect	A protocol built on OAuth 2.0 used for federated authentication.
OAuth 2.0	OAuth 2.0 Authorization Framework	An authorisation framework for delegating access between systems.
RBAC	Role-Based Access Control	Management of access rights based on roles.
REST	Representational State Transfer	An architectural style for designing web API services.
SIEM	Security Information and Event Management	A system for the collection, correlation, analysis and visualisation of security events.
SLA	Service Level Agreement	An agreement on the level of services provided and guaranteed parameters.
SOC	Security Operations Center	A cybersecurity operations centre providing monitoring, detection and response to incidents.
SOCaaS	Security Operations Center as a Service	Provision of SOC functionality in the form of a service.
SOAR	Security Orchestration, Automation and Response	A platform for the orchestration, automation and management of response to security events and incidents.
TTP	Tactics, Techniques and Procedures	Tactics, techniques and procedures of attackers.
TRL	Technology Readiness Level	The technology readiness level of the solution.
UAT	User Acceptance Testing	Acceptance testing by users.
UC	Use Case	A use case describing system functionality from an actor's perspective.
UML	Unified Modeling Language	A modelling language used to design software and system solutions.
XDR	Extended Detection and Response	An extended approach to detection and response across multiple layers of the environment.

Term	Definition

Asset	Any element of value to the organisation, e.g. a device, account, application, database, service or information.
Alert	A system-generated notification of a suspicious, anomalous or risky event.
Anomaly	Behaviour, an event or a pattern that differs from the expected or normal state.
Audit trail	A record of operations performed, changes and accesses, enabling retrospective checking and traceability.
Security event	An observable phenomenon in a system or network that may be relevant to security.
Security incident	A security event or series of events that actually or potentially compromises the confidentiality, integrity or availability of assets.
Detection rule	Formally defined logic for identifying suspicious or undesirable behaviour in data and logs.
False negative	A case where the system fails to flag a real threat or incident.
False positive	A case where the system incorrectly flags legitimate activity as a threat or incident.
Incident response	A set of processes and activities for the analysis, handling, elimination and evaluation of a security incident.
Integration	The interconnection of systems, tools or services through interfaces and data exchange.
Event correlation	Combining multiple security events into a single context in order to identify relationships and incident risk.
Multi-tenancy	The ability of a single platform to securely serve multiple customers or organisations while logically separating their data and configuration.
Orchestration	Coordinated management of multiple security tools, processes and responses within a single workflow.
Playbook	A standardised procedure or scenario for responding to a specific type of security event or incident.
User role	A set of permissions and responsibilities assigned to a user according to their function in the system.
Use case	A structured description of an actor's interaction with the system in order to achieve a specific result.
Data retention	Rules determining the retention period of data, logs, incidents and audit records.
Tenancy / Tenant	A logically separated customer space in a multi-tenant solution, containing its own data, configuration and users.

Threat hunting	Proactive searching for threats and suspicious activities that have not been captured by standard detection mechanisms.
Threat intelligence	The collection, evaluation and use of information on threats, attackers, campaigns and their techniques.
Event	An individual recorded occurrence in a system or network, e.g. a login, a configuration change or network communication.
Vulnerability	A weakness in a system, process or configuration that can be exploited by an attacker.
Knowledge base	A central repository of recommendations, procedures, playbooks, experience and analytical insights of the SOC team.
Customer organisation	An entity using the SOCaaS platform as a separate tenant or as a recipient of a shared service.

2 Executive Summary

2.1 Innovativeness of the Solution

This project is submitted, on the one hand, as a response to the current EU challenges in the area of digitalisation and transformation of the European economy (the Digital Europe strategy) and, on the other hand, as a possible solution for organisations facing problems arising from the implementation of the requirements of the NIS2 Directive. With effect from 16 January 2023, the European directive on the security of network and information systems (**NIS2 Directive 2022/2555**) was introduced, with the ambitious goal of improving the security of network and information systems throughout the European Union. Just as in 2018 Parliament approved Act No. 69/2018 Coll. on Cybersecurity, which reflects the European directive on the security of network and information systems, the NIS1 Directive, similar local regulation reflecting the European NIS 2 Directive is expected in the near future. This directive focuses on operators of critical infrastructure and providers of essential services and mandates strict cybersecurity measures and immediate reporting of incidents to the competent authorities. The scope of the NIS 2 Directive is broader than that of NIS 1, both in terms of the range of organisations and of sectors. Several estimates (e.g. those of the Cybersecurity Competence and Certification Centre of the Slovak Republic) agree that the range of organisations affected by the subsequent local transposition of NIS2 into national legislation in Slovakia will increase approximately sixfold compared with the current situation, from the current one thousand five hundred to approximately **ten thousand organisations**. This range of organisations will also include, for example, many educational institutions, colleges, universities and secondary schools, as well as public or state administration organisations, various commercial entities and others, which will have to strengthen their cybersecurity, protect critical assets and actively contribute to building a secure digital environment. The obligations of each of these organisations will include not only introducing processes for cybersecurity management, but also implementing and auditing them. Introducing and subsequently managing these processes within an organisation's operations is and will be financially and time-consuming; therefore, within the INNOSOC project we come up with a solution in the form of a so-called security operations centre (SOC), built on an open system and as a scalable cloud service. In addition to the possibility of establishing such a centre for supervising cybersecurity management directly within the given organisation, there is also the option of ordering such a service from a provider. With regard to which

organisations the NIS2 Directive concerns, or rather which its transposition into Slovak legislation will concern, and especially in the case of organisations funded from state or public sources, the problem lies in the human and financial resources necessary to establish and operate such a security management supervision centre in-house. It can therefore be expected that there will be great demand for outsourcing such a service in the future. The current offer of similar solutions and services in the Slovak Republic is not large and typically focuses on expensive on-premise and turnkey solutions. From this perspective, their availability in this form is of little suitability or accessibility for some organisations, particularly in view of the price of such commercial services. We therefore consider our INNOSOC project, which will be able to offer an alternative solution for these organisations, to be necessary and highly topical.

At present, the number and diversity of cyber attacks is continuously increasing, as reported in their surveys by, for example, the Cybersecurity Competence and Certification Centre for 2023 (https://cybercompetence.sk/kniznica) or the specialist journal Cybercrime Magazine (https://cybersecurityventures.com/research/). It is becoming apparent that regulation in the area of cybersecurity is necessary but not sufficient in itself. It is also necessary to focus on supporting organisations which, after the transposition of the NIS2 Directive into national legislation, will belong to the set of organisations regulated by the state in terms of compliance with legal requirements in the area of cybersecurity. In the near term, these organisations (schools, local governments, etc.) will have neither a sufficient budget earmarked for the implementation and management of information security, nor the specialist human capital that is needed and necessary for it. The output of the project is the design of an innovative SOCaaS-type service solution that is suitable and helpful in addressing the expected situation of many public and state organisations.

The project is therefore directed towards procedures and designs of an innovative platform for creating SOCaaS-type services in such a form that interested parties will be able to implement them in their own organisational environment or will be able to use and share the services of such a centre with another similar organisation and thus speed up and make more efficient the processes of cybersecurity management. The entire solution will be built as an open, flexible and multi-tenant platform, and the description of the solution will be available to the general public. At the scientific and research level, the solution will be supplemented with modules under development for the detection and prediction of anomalies using ML and DNN.

Based on the analyses we have carried out so far, as well as an analysis of the availability of similar solutions, we are currently not aware of any existing solution similar to ours that would combine the use of virtualisation technologies as a flexible and scalable platform for running a multi-tenant shared SOCaaS service, both with elements of automation and the use of machine learning methods. We consider this to be the main contribution and originality of our solution and project.

The INNOSOC project and its activities reflect the requirements of today and are in harmony with the "**EU Recovery and Resilience Plan**" (EU Regulation 2021/241) and the "**Recovery and Resilience Plan of the Slovak Republic (SR)**" (Act No. 368/2021 Coll.). The project also reflects the needs of the digitalisation of the European economy as declared by the documents concerning a digital Europe and the EU calls for the digitalisation of industry. This is also in line with Component 17 of the Recovery and Resilience Plan of the Slovak Republic (known as Digital Slovakia).

The INNOSOC project is also in line with the objectives of the new European Research Area, in particular in the priority area Joint solution of the green and digital transformation it corresponds with **ERA Actions** such as No. 8:
Accelerate the green and digital transformation of Europe's key industrial ecosystems, or No. 15: Build research and innovation ecosystems in order to improve excellence and competitiveness.

The INNOSOC project is in line with the **HORIZON EUROPE** framework programme, specifically with **Pillar II and Cluster 4 – Digital, Industry and Space**. With particular emphasis on key digital technologies and **advanced computing and big data processing**.

The outputs of the INNOSOC project and the creation of a partnership between academic and commercial entities to address multidisciplinary problems are in line with the **Research and Innovation Strategy for Smart Specialisation of the Slovak Republic 2021-2027** (SK RIS3 2021+) in the domain **Digital Transformation of Slovakia in the priority area Cybersecurity and Cryptography**. Since the applicant, the University of Žilina (UNIZA), is a modern public educational and research institution which is one of the largest Slovak universities. UNIZA has many years of experience in implementing various types of projects and has experts who manage and implement such projects. The partner, brainit.sk s.r.o. (hereinafter BRAIN:IT), is a software company engaged in the design, development and administration of information systems and is also a provider of qualified services within the meaning of eIDAS.

In terms of readiness, the project implementers have extensive experience in the area of CC technologies. The applicant has already for *5* years been operating its own private CC solution with an IaaS service, which is primarily used for teaching and research purposes. It currently has one KEGA project and one structural funds project in this area. In its practice, BRIAN:IT uses CI/CD DevOps approaches with lightweight virtualisation. The experience gained will be used in addressing partial objectives 1 and 2. In the area of ML, the implementers from UNIZA FRI are leading figures in the development of ML at the faculty, with the application of knowledge to anomaly detection in high-speed flows. BRAIN:IT applies ML in its solutions. The knowledge acquired and the algorithms developed will be used for the development of ML methods within the intent of the INNOSOC project and partial objective 4. Finally, for the SOC topic area, or intrusion detection, UNIZA FRI offers several courses and has supervised several final theses. BRAIN:IT plans to develop a SOC solution to satisfy the demand of its customers. The implementation of the project is in line with the future R&D and commercial plans of the partners and will thus be used in addressing partial objective 5.

2.2 ML, AI and Cybersecurity

The rapid development of information and communication technologies has transformed everything into a digital era. The Internet has played a significant role for organisations and individuals turning to digital technologies and solutions. The development of advanced technologies such as machine learning (Machine Learning-ML) and artificial intelligence (Artificial Intelligence-AI) makes the digital revolution fascinating. AI is the intelligence demonstrated by machines that are aware of their environment and act accordingly in order to achieve something. There are various sub-domains of AI, such as neural networks, fuzzy systems, machine learning, deep learning, computer vision, natural language processing, cognitive computing, evolutionary computing and nature-inspired algorithms. AI is perceived as an essential element of the digital transformation of society. Future applications are expected to bring major changes. Practical examples of AI include autonomous cars, Alexa, robo-advisers, conversational bots, OTT platform recommendations, e-mail spam filters and the like. Essentially, there are three categories of AI: Artificial Narrow Intelligence, Artificial General Intelligence and Artificial Super Intelligence. Another way to categorise AI is by the value it brings to business.

We can distinguish several basic classes of AI:

- visual AI (computer vision, augmented reality)
- interactive AI (chatbots, intelligent personal assistants)
- text AI (text recognition, speech recognition)
- functional AI (robots, Internet of Things (IoT) solutions)
- analytical AI (preference/sentiment analysis, supplier risk assessment)

The advantages of AI include the following:

- automation
- intelligent decision-making (smart decision making)
- research

- accuracy
- data collection
- data analysis
- solving complex problems
- management of repetitive tasks
- error minimisation

The risks of AI include the following:

- high implementation costs
- human replication
- programmability
- lack of creativity
- acceleration of hacking
- no improvement with increasing experience
- unemployment
- global regulation
- AI terrorism

AI in cybersecurity is not merely an intelligent revolution; it has enormous potential. AI-enabled systems can be trained for secure human–machine communication, threat generation, recognition of new malware or protection of information sensitive to a particular organisation. The introduction of certain ethical practices in AI must be recommended so as to ensure that the technology will be a helper and not an enemy. AI-based cybersecurity systems have many capabilities for systems, data and applications. System security means security for the network, cloud, malware, IoT and the like. Data security is associated with the security of information in the cloud, threat detection and the recognition of new attacks. AI in cybersecurity helps organisations improve threat response time and breach response and reduce costs. AI has three key characteristics in the area of cybersecurity:

- Detection: Today's organisations make extensive use of AI for accurate threat detection. More than 50 % of organisations have implemented AI-enabled cybersecurity systems. AI-based detection is a unique behavioural analysis capability for regular monitoring and identification of irregular traffic. Machine learning and deep learning algorithms are used most frequently.

- Prediction: The rate of use of prediction functionality is the second highest. According to research, 35 % of organisations make intensive use of AI to predict cyber threats. AI helps with prediction over huge data based on system training. AI-based automation helps organisations recognise vulnerabilities and strengthens their networks so that they are able to defend themselves against any cyber attack.

- Response: In threat response, AI is still developing. Approximately 18 % of organisations use AI as a response to cyber attacks. Thanks to the automation of threat detection, a new real-time defence mechanism is emerging.

The fundamental goal of cybersecurity is to keep users and data safe. The 3 traditional goals of cybersecurity form the **CIA triad** .

1. Confidentiality

The goal of confidentiality is to keep data secret from unauthorised parties. Only authorised entities should have access to data. Confidentiality can be achieved by encrypting data. Encryption is the process by which plaintext data are encoded into ciphertext using an encryption key. Ciphertext is not human-readable; the corresponding

decryption key is required to decode the data. Encrypting information transmitted over networks prevents attackers from reading the content even if they intercept the communication. Encrypting data at rest ensures that adversaries will not be able to read your data even if the physical storage is compromised (for example, if an attacker breaks into an office and steals hard drives). Another approach to ensuring confidentiality is access control. Controlling access to information is the first step in preventing unauthorised exposure or sharing of data (whether intentional or not). Access to data should be granted in compliance with the principle of least privilege – an individual or application must have access to the minimum data necessary to perform its function. For example, only a company's finance department should have access to information on revenues and transactions, or only system administrators should be able to see the network and access logs.

ML can help detect abnormal access, suspicious behavioural patterns or traffic from malicious sources, thereby ensuring that confidentiality is preserved. For example, imagine that an administrator suddenly starts accessing confidential/privileged files outside their usual pattern or the pattern applicable to administrators in general. A machine learning model can flag this as anomalous and raise an alarm so that the administrator can be investigated.

2. Integrity

The goal of integrity ensures that data are trustworthy and free from tampering. If data can be modified, there is no guarantee of their authenticity and accuracy – such data cannot be trusted. Integrity can be ensured by means of hashing and checksums. A checksum is a numerical value calculated by applying a hash function to the data. Even if a single bit of the data were changed, the checksum would change. Digital signatures and certificates also support integrity – if an e-mail or a code library is digitally signed by a user, it can no longer be changed, and any change requires a new digital signature.

Attackers may want to compromise the integrity of a system or service to their advantage. For example, an attacker may intercept incoming requests to a banking server and change the target account for any money transfer. Malicious browser extensions may redirect users to a site for gaming traffic and advertising statistics – the original target URL entered by the user has been modified. By analysing patterns in the data together with other signals, ML models can detect integrity violations.

3. Availability

The first two goals ensure that data remain secret, free from tampering and protected from attackers. However, these guarantees are meaningless if authorised users cannot access the data as needed. The goal of availability ensures that information is always available to legitimate users of the system.

Attackers may attempt to compromise availability by carrying out a denial-of-service (DoS) attack, in which the target service is bombarded with incoming requests from unauthorised or fake nodes. An attacker may, for example, send millions of fake queries to a database server. Although the attacker may not be able to obtain any useful information, the server is so overloaded that legitimate queries from authorised users are never executed. Alternatively, an attacker may reduce availability by physically destroying data.

Availability can be guaranteed by implementing redundancy in data and services. Regular data backup ensures that data are still available even if one copy is destroyed or modified. If multiple API endpoints are present to achieve the same functionality, legitimate users can switch to another endpoint and availability is preserved. As with confidentiality and integrity, pattern analysis and classification algorithms can help detect DoS attacks. An emerging paradigm, graph neural networks, can help detect coordinated bot attacks known as distributed denial of service (DDoS).

The CIA triad focuses exclusively on the cybersecurity of data. However, cybersecurity today extends far beyond data and servers. Data stored on servers become information used by organisations, which is turned into

applications that are ultimately used by people. Cybersecurity encompasses all of these aspects, and there are different problem areas in each of them. The figure below shows how diverse and multifaceted the issue of cybersecurity is and the various elements it encompasses:

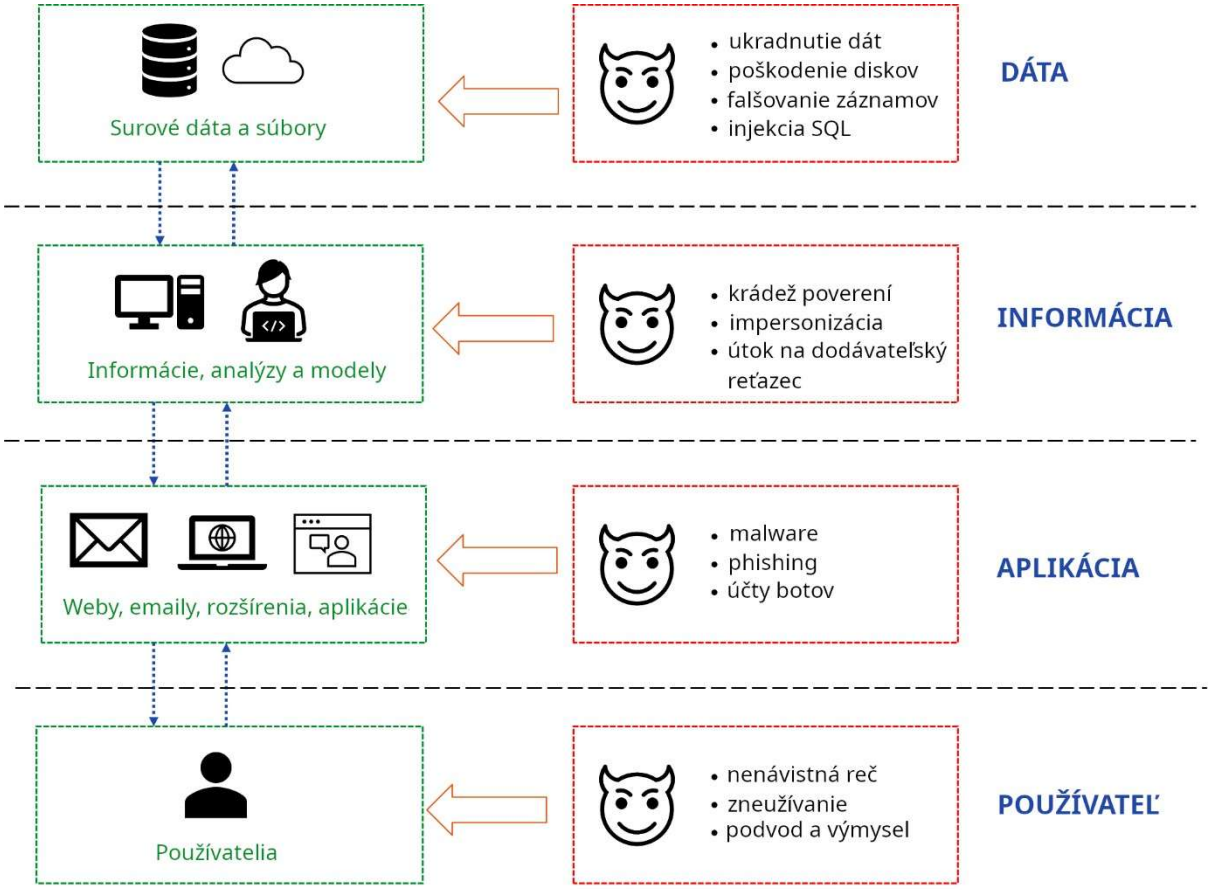


Fig. : Multiple problem areas in cybersecurity

Data security

Data in raw format are stored on hard drives in offices or in the cloud, which are ultimately stored in physical machines in data centres. At the data level, the task of cybersecurity is to keep data safe.

Emphasis is therefore placed on preserving the confidentiality, integrity and availability (the three goals of the CIA triad) of data. Cybersecurity problems at this level focus on new cryptographic schemes, lightweight encryption algorithms, fault-tolerance systems and compliance with data retention regulations.

Information security

Data from data centres and the cloud are turned into information that companies use to create various services. At the information level, the task of cybersecurity is to ensure that employees access and process information correctly. Problems at this level focus on network security management, detection of policy violations and insider threats, and ensuring that no unintentional leakage of private data occurs.

Application security

Funded by the European Union NextGenerationEU
RECOVERY AND RESILIENCE PLAN
GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC

Information is turned into a form suitable for consumers using various services. An example is information on Facebook users, which is turned into a list of top friend recommendations. At the application level, the task of cybersecurity is to ensure that the application cannot be compromised.

Problems at this level focus on malware detection, supply-chain attack detection, anomaly detection, detection of bots and automated accounts, and flagging of phishing e-mails and malicious URLs.

User security

Finally, applications are used by human end users, and the task of cybersecurity is to ensure the security of these users. Problems at this level include detection of hate speech, content moderation, flagging of fraudulent transactions, characterisation of abusive behaviour and protection of users against digital crime (identity theft, fraud and extortion). Please note that this aspect goes beyond the technical aspects of cybersecurity and extends into the fields of the humanities, law and the social sciences.

Privacy

The terms security and privacy are often confused and used interchangeably. However, the goals of security and privacy are very different. While security aims to secure data and systems, privacy means that individuals have full control over their data. With regard to privacy, every individual should know the following:

- what data are collected (location, application usage, web tracking and health metrics)
- how long they will be retained (deleted immediately or within a week/month/year)
- who has access to them (advertisers, research organisations and governments)
- how they can be deleted (how to request this from the application)

Security and privacy are interrelated. If an attacker compromises a hospital or medical database (a security breach), they may have access to sensitive patient data (a privacy breach). There are numerous laws around the world, for example the General Data Protection Regulation (GDPR) in Europe, which require strict security controls to ensure user privacy.

Since machine learning depends on large amounts of collected data, there has been pressure for the use of ML algorithms and methods in a privacy-preserving manner.

Overview of machine learning

The traditional computing paradigm defines an algorithm as having 3 elements: input, output and a process that determines how to derive the output from the input. For example, in a credit card detection system, a module for flagging suspicious transactions may take transaction metadata (location, amount, type) as input and a flag (suspicious or not) as output. The process defines a rule for setting the flag based on the input, as shown in the figure below.

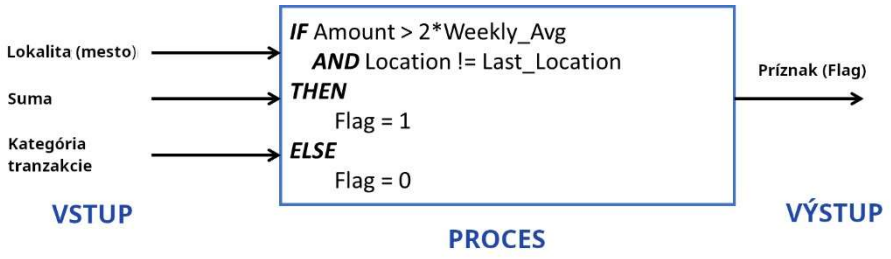


Fig.: Traditional “input-process-output” model for fraud detection

ML represents a fundamental change to the input-process-output philosophy. The traditional approach defined computation as deriving output by applying a process to input. In machine learning, we are given the input and the output, and the task is to derive the process that links them.

Continuing our analogy of a credit card fraud detection system, we are now given a dataset containing input features and output labels (flag). Our task is to learn how the label can be computed from the input. Once we have learned the process, we can generalise it to new data that come to us in the traditional input-process-output manner. Most security problems are classification problems. Classification is a task in which data points are assigned discrete, categorised labels (fraud/not fraud or low/medium risk/high risk).

ML is not a one-off process. It involves multiple steps, such as cleaning and preparing data in the right form, training and tuning models, deploying them and monitoring them.

Data preprocessing

The first step in any ML experiment is to process the data into a format suitable for ML. Real-world data are often not suitable for direct use by a model, for two main reasons.

The first reason is the variability of format. The data around us come in various formats, such as numbers, images, text, audio and video. All of these data need to be converted into numerical representations so that a model can work with them. Preprocessing would convert images into matrices, text into word embeddings and audio streams into time series. Some properties of data are discrete and represent categorical variables. For example, in a dataset about users, the field “country” would take text strings such as India, Canada, China and the like. Preprocessing converts such categorised variables into a numerical vector form with which we can feed the model.

The second reason is noise in real-world data, which represents incomplete or erroneous data. Measurement inaccuracies, processing errors and human errors can cause corrupted values to be recorded. For example, a data entry operator may enter your age as 233 instead of 23 by mistakenly pressing the 3 key twice. A web scraper collecting data may have network problems and fail to submit a request; some fields in the data will then be missing.

In short, preprocessing removes noise, handles missing data and transforms the data into a format suitable for an ML model.

Training phase

After appropriate data preprocessing, we train the model to learn from these data. The model expresses the relationship between the preprocessed features and some target variables. For example, a model for detecting phishing e-mails will produce as output the probability that an e-mail is malicious based on the defined input features. A model that classifies malware as one of 10 malware families will generate a 10-dimensional vector of the probability distribution over the 10 classes. In the training phase, the model learns the parameters of this relationship so as to minimise the error on the training data (i.e. so that it can predict the labels of the training data as accurately as possible). During the training phase, a subset of the data is set aside as validation data to examine the error of the trained model on unseen data.

One of the problems a model may encounter is overfitting (overfit). If the model learns parameters that are too specific to the training data, it cannot generalise well to newer data. This can be diagnosed by comparing the model's performance on the training and validation data. A model that is overfitted will show a decreasing loss or error on the training data but a non-decreasing loss on the validation data. There is also the opposite problem – a model may suffer from underfitting, when it is simply unable to learn from the training data.

Funded by the European Union NextGenerationEU
RECOVERY AND RESILIENCE PLAN
GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC

Inference phase

Once the model has been trained, we want to use it to make predictions for new, unseen data. In this phase, no parameter learning is performed any more; we simply feed the model with features from the data and examine the prediction produced by the model. Inferences often take place in real time. For example, every time you use a credit card, the parameters of your transaction (amount, location and category) are used to perform inferences on the fraud detection model. If the model flags the transaction as suspicious, the transaction is declined.

Maintenance phase

Machine learning models need constant monitoring and tuning so that their performance does not deteriorate. Over time, a model may become more prone to errors in its predictions because it was trained on older data. For example, a disinformation detection model trained in 2019 would never have been exposed to fake news about the COVID-19 pandemic. As a result, it never learned the characteristics of COVID-related fake news and may fail to recognise these articles as disinformation. To prevent this, the model must be retrained on newer data at regular intervals so that it continues to learn from them. In addition, new features may emerge that could improve the model's performance. In such a case, it would be necessary to train the model with the new feature set and check the increase in model performance. Moreover, there may be different segments of data in which different classification thresholds show higher accuracy. The model then needs to be tuned to use a different threshold in each data segment. Model monitoring is a continuous process, and there are automated tools (called MLOps tools) that offer features for continuous monitoring, training, tuning and alerting of models.

Basic ML paradigms

Supervised learning

Supervised learning methods or algorithms include learning algorithms that take data samples (known as training data) and the associated outputs (known as a label or response) for each data sample during the model training process. The main goal is to learn the mapping or association between input data samples x and their corresponding outputs y based on multiple instances or samples of training data. This learned knowledge can then be used in the future to predict the output y for any new input data sample x that was previously unknown or unavailable during the model training process (Fig.).



Fig. : Example of supervised learning – a labelled training set for a spam detection task.

These learning methods are called supervised learning because the model learns on data samples where the desired output labels are known in advance in the training phase. Supervised learning essentially tries to model the relationship between inputs and their corresponding outputs from the training data so that we are able to predict output responses for new data inputs based on the knowledge it has previously acquired with regard to the relationships and mappings between inputs and their target outputs. This is precisely why supervised learning methods are widely used in predictive analytics, where the main goal is to predict a certain response

to some input data, which are usually fed into a trained supervised ML model. Supervised learning methods are divided into two main classes according to the type of ML tasks they try to solve.

- Classification
- Regression

Classification

Classification-based tasks are a sub-field of supervised machine learning where the key goal is to predict output labels, which are categorical in nature, for input data based on what the model has learned in the training phase. The output labels here are also known as classes or class labels because they are categorical in nature, which means that they are unordered and discrete values. Each output response thus belongs to a specific discrete class or category. Let us take weather prediction as a real-world example. For simplicity, let us say that we are trying to predict whether the weather is sunny or rainy based on multiple samples of input data consisting of attributes or features such as humidity, temperature, pressure and precipitation. Since the prediction may be either sunny or rainy, there are two distinct classes in total; therefore this problem can also be referred to as a binary classification problem. Figure 2 depicts a binary weather classification task predicting the weather as sunny or rainy based on training a supervised model on input data samples with feature vectors (precipitation, humidity, pressure and temperature) for each data sample/observation and their corresponding class labels as sunny or rainy.

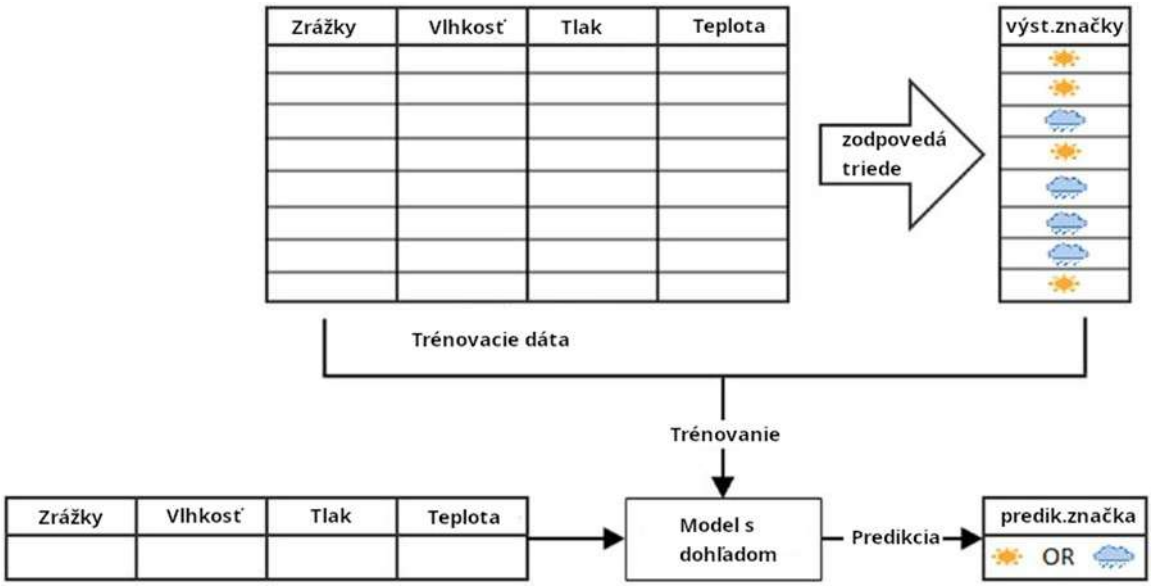


Fig.: Binary classification for weather prediction

A task where the total number of distinct classes is more than two becomes a multi-class classification problem, where each prediction response can be any of the probable classes from this set. A simple example would be an attempt to predict numerical digits from scanned handwritten images. In this case, it becomes a 10-class classification problem, because the output class label for any image can be any digit from 0 to 9. In both cases, the output class is a scalar value pointing to one specific class. Multi-label classification tasks are such that, based on any input data sample, the output response is usually a vector with one or more output class labels. A simple real-world problem would be trying to predict the category of a news article, which could have multiple output classes such as news, finance, politics, etc.

Regression

Machine learning tasks whose main goal is the estimation of a value can be called regression tasks. Regression-based methods are trained on input data samples that have output responses which are continuous numerical values, unlike classification, where we have discrete categories or classes. Regression models use the attributes or features of the input data (also called explanatory or independent variables) and their corresponding continuous numerical output values (also called the response, dependent or outcome, i.e. output, variable) to learn specific relationships and associations between inputs and their corresponding outputs. With this knowledge, the model can predict output responses for new, unseen data instances, similarly to classification but with continuous numerical outputs. One of the most common real-world examples of regression is house price prediction. A simple regression model can be built to predict house prices based on data concerning land areas in square metres.

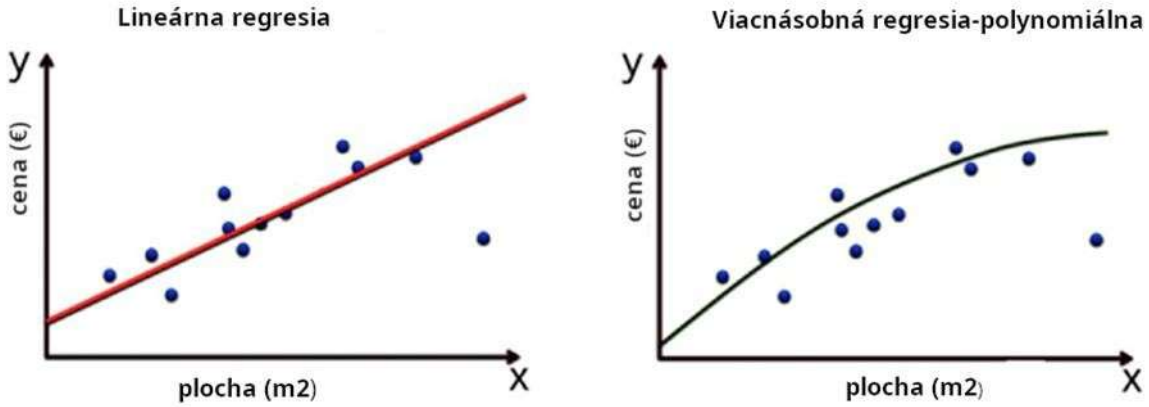


Fig. : Regression model for house price prediction

The figure above shows two possible regression models based on different methods of predicting house prices based on land area. The basic idea is that we try to determine whether there is any relationship or association between the data feature plotted in the graph and the outcome variable, which is the house price, and which we want to predict. Thus, once we learn this trend or relationship shown in Figure 3, we can predict house prices in the future for any given plot of land. If you have looked at this figure carefully, we have deliberately shown two types of models to demonstrate that there are multiple ways of building a model based on training data. The main goal is to minimise errors during training and validation of the model so that it generalises well, does not overfit or is not biased only towards the training data, and performs well in future predictions.

Simple linear regression models try to model relationships in data with one feature or explanatory variable x and one response/output variable y, where the goal is to predict y. Methods such as ordinary least squares (OLS) are usually used to achieve the best linear fit during model training. Multiple regression is also known as multivariate regression. These methods try to model data where we have one output response variable y in each observation but multiple explanatory variables in the form of a vector X instead of a single explanatory variable. The goal is to predict y based on the various features present in the vector X. A real-world example could be extending our house prediction model by building a more sophisticated model in which we predict the house price based on multiple features instead of only the land area in each data sample. The features can be represented in a vector containing the land area, number of bedrooms, number of bathrooms, total number of floors, and whether the house is furnished or unfurnished. Based on all these attributes, the model tries to learn the relationship between the individual feature vectors and the corresponding house price so that it can predict them in the future.

Funded by the European Union NextGenerationEU
RECOVERY AND RESILIENCE PLAN
GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC

Polynomial regression is a special case of multiple regression, where the output variable y is modelled as an nth-degree polynomial of the input feature x. Essentially, it is multiple regression where each element in the input feature vector is a multiple of x. The model on the right in Figure 3 for predicting house prices is a polynomial model of degree 2. Non-linear regression methods try to model relationships between input features and outputs based on a combination of non-linear features applied to the input attributes and the necessary model parameters.

Lasso regression is a special form of regression that performs normal regression and generalises the model well by performing regularisation as well as feature or variable selection. Lasso stands for least absolute shrinkage and selection operator. The L1 norm is usually used as the regularisation term in lasso regression.

Ridge regression is another special form of regression that performs normal regression and generalises the model by performing regularisation in order to prevent overfitting of the model. The L2 norm is usually used as the regularisation term in ridge regression.

Generalised linear models are general frameworks that can be used to model data predicting various types of output responses, including continuous, discrete and ordinal data. Algorithms such as logistic regression are used for categorical data and ordered probit regression for ordinal data.

The most important supervised learning algorithms include:

- K-Nearest Neighbours method – K-Nearest Neighbors (KNN)
- Linear regression – Linear Regression
- Logistic regression – Logistic Regression
- Support vector method – Support Vector Machines (SVMs)
- Decision trees – Decision Tree (DT) and random forests – Random Forest (RF)
- Neural networks – Neural Networks (NN). Some neural network architectures, such as the autoencoder and restricted Boltzmann machines (RBM), represent unsupervised learning. Some architectures may represent semi-supervised learning, e.g. in the case of using a Deep Belief Network with pre-training based on unsupervised learning.

A more detailed description of the individual algorithms can be found in the background documents.

Unsupervised learning

Supervised learning methods usually require some training data where the outcomes we are trying to predict are already available in the form of discrete labels or continuous values. In practice, however, we often do not have pre-labelled training data available and still want to extract useful knowledge or patterns from our data. In this scenario, unsupervised learning methods are extremely effective. These methods are called unsupervised because the model or algorithm tries to learn the inherent latent structures, patterns and relationships from the given data without any help or supervision, such as the provision of annotations in the form of labelled outputs or outcomes. Unsupervised learning is more concerned with trying to extract meaningful insights or information from data than with trying to predict some outcome based on previously available supervised training data. There is greater uncertainty in the results of unsupervised learning, but a lot of information can also be gained from these models that was previously not available when looking at the raw data. Unsupervised learning can often be one of the tasks involved in building a huge intelligence system. For example, we could use unsupervised learning to obtain possible outcome labels for tweet sentiment using knowledge of English vocabulary and then train a supervised model on similar data points and their outcomes, which we previously obtained through unsupervised learning. There is no fixed rule regarding the use of only one particular technique. Multiple methods can always be combined, as long as they are relevant to solving the problem.

Unsupervised learning methods can be categorised into the following broad areas of ML tasks relevant to unsupervised learning.

- clustering – Clustering
- dimensionality reduction – Dimensionality Reduction
- anomaly detection – Anomaly Detection
- association rule mining – Association Rule-Mining

Clustering

Clustering methods are machine learning methods that try to find patterns of similarity and relationships between data samples in our dataset and then group these samples into different groups, so that each group or cluster of data samples has a certain similarity based on inherent attributes or features. These methods are completely unsupervised, because they try to cluster data by looking at the data features without prior training, supervision or knowledge of the data attributes, associations and relationships. A real-world example is the problem of running multiple servers in a data centre and trying to analyse the logs for typical problems or errors. Our main task is to determine the different kinds of log messages that usually occur frequently every week. Simply put, we want to group the log messages into different clusters based on some inherent characteristics. A simple approach would be to extract features from the log messages, which would be in text format, apply clustering to them and group similar log messages based on content similarity. Figure 4 shows how clustering would solve this problem. Essentially, we start with raw log messages. Our clustering system would use feature extraction to extract features from the text, such as word occurrences, phrase occurrences, etc. Finally, a clustering algorithm such as K-means or hierarchical clustering would be used to group or cluster the messages based on the similarity of their inherent properties.

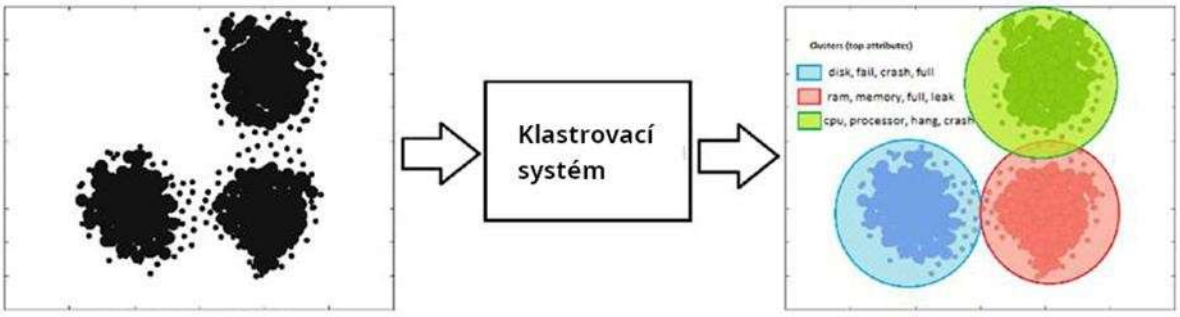


Fig. : Clustering

From the figure above, it is quite evident that our system has three distinct clusters of log messages, where the first cluster shows disk problems, the second cluster is about memory problems and the third cluster is about processor problems. The figure also shows the main words that helped to distinguish the clusters and group similar data samples (logs). Of course, sometimes some features may be present in multiple data samples, and therefore slight overlapping of clusters may also occur, since this is unsupervised learning. However, the main goal is always to create clusters such that the elements of each cluster are close to each other and far from the elements of other clusters. There are various types of clustering methods, which can be classified according to the following main approaches:

- Centroid-based methods – K-means, K-medoids
- Hierarchical clustering methods – Hierarchical Cluster Analysis (HCA)

Funded by the European Union NextGenerationEU
RECOVERY AND RESILIENCE PLAN
GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC

- Distribution-based clustering methods – Gaussian Mixture Model GMM)
- Density-based methods – DBSCAN, OPTICS

Dimensionality reduction

Once we start extracting attributes or features from raw data samples, the feature space can sometimes become bloated with a huge number of features. This poses multiple challenges, including the analysis and visualisation of data with thousands or millions of features, which makes this feature space extremely complex and poses problems with regard to training models, memory and space constraints. In fact, this is referred to as the "curse of dimensionality". In these scenarios, unsupervised learning methods can also be used, in which we reduce the number of features or attributes for each data sample. These methods reduce the number of feature variables by extracting or selecting a set of principal or representative features. Several popular algorithms are available for dimensionality reduction:

- principal component analysis PCA (Principal Component Analysis PCA)
- independent component analysis ICA (Independent Component Analysis ICA)
- discriminant analysis (Discriminant Analysis)

The figure below shows the output of a typical feature reduction process applied to the 3D Swiss Roll structure, which has three dimensions, to obtain a two-dimensional feature space for each data sample using PCA.

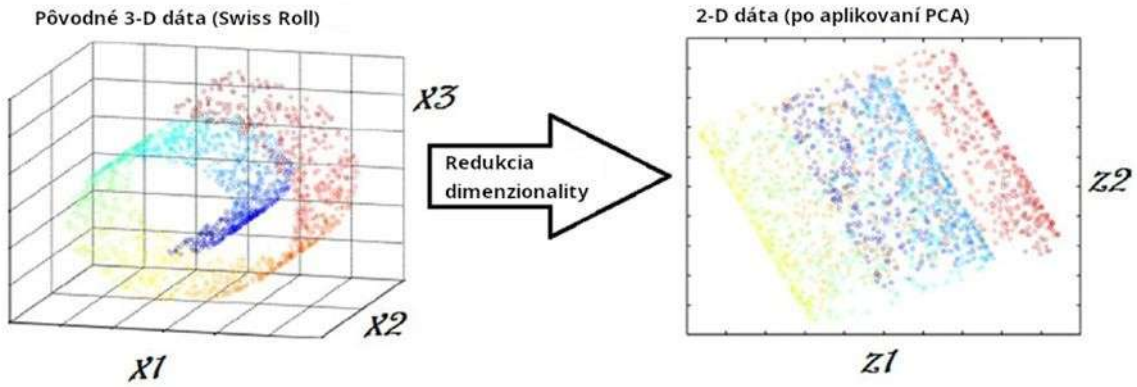


Fig. : Dimensionality reduction using PCA

From the figure above, it is quite clear that each data sample originally had three features or dimensions, namely D(x1, x2, x3), and after applying PCA we reduce each data sample from our dataset to two dimensions, namely D'(z1, z2). Dimensionality reduction techniques can be classified into two main approaches as follows:

- Feature selection methods: Specific features are selected for each data sample from the original list of features and the other features are discarded. No new features are generated in this process.
- Feature extraction methods: We design or extract new features from the original list of features in the data. The reduced subset of features will thus contain newly generated features that were not part of the original feature set. PCA belongs to this category.

Anomaly detection

The process of anomaly detection is also called outlier detection, where we are interested in detecting the occurrence of rare events or observations that usually do not occur normally based on samples of historical

data. Sometimes anomalies occur infrequently and are therefore rare, and in other cases anomalies may not be rare but may occur in very short bursts over time, so they have specific patterns. Unsupervised learning methods can be used for anomaly detection, whereby we train the algorithm on a training dataset that has normal, non-anomalous data samples. Once it has learned the necessary data representations, patterns and relationships between attributes in normal samples, for any new data sample it would be able to identify it as an anomalous or normal data point using its learned knowledge. The figure below shows some typical scenarios based on anomaly detection. The most common methods include:

- One-class SVM – One-class SVM
- Isolation forest – Isolation Forrest

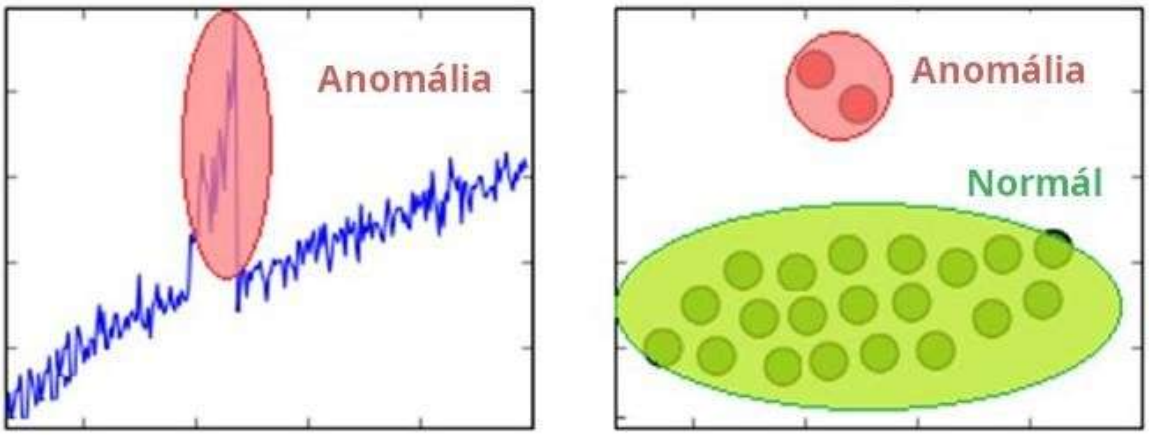


Fig. : Anomaly detection

Methods based on anomaly detection are extremely useful in various real-world cases, such as the detection of security attacks or intrusions, credit card fraud, manufacturing anomalies, network problems and many others.

Association Rule-Mining

Association rule mining is usually a data mining method used to explore and analyse large transactional datasets in order to find patterns and rules of interest. These patterns represent interesting relationships and associations between various items across transactions. Association rule mining is also often called market basket analysis, which is used to analyse customers' purchasing patterns. Association rules help to detect and predict transactional patterns based on the knowledge obtained from training transactions. Using this technique, we can answer questions about which items people tend to buy together, thereby indicating frequent itemsets. We can also associate or correlate products and items, i.e. insights such as that people who buy beer tend to buy chicken wings in a restaurant. Figure 8 shows how a typical association rule mining method should ideally work on a transactional dataset.

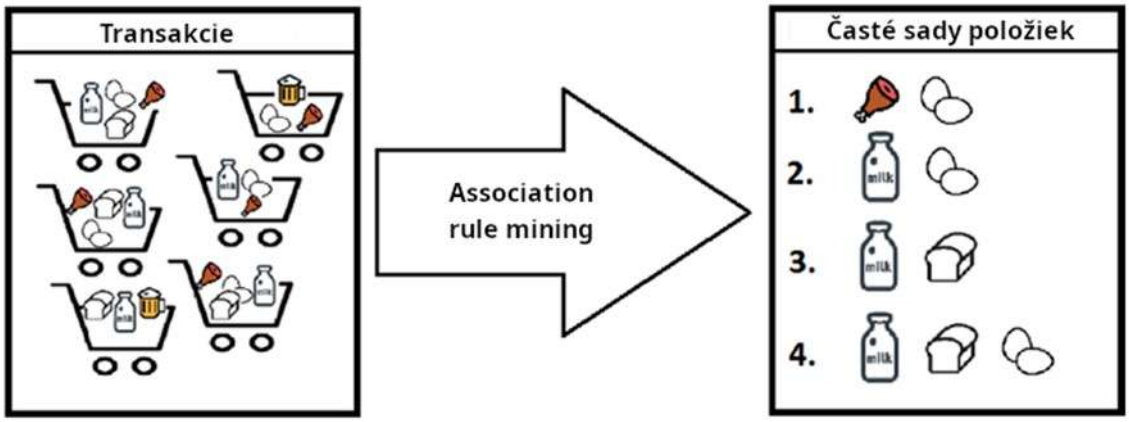


Fig. : Association rule-mining

From the figure above, it can clearly be seen that, based on various customer transactions over a certain period of time, we have obtained items that are closely associated and that customers tend to buy together. Some of these frequent itemsets are shown as {meat, eggs}, {milk, eggs} and so on. The criterion for determining good-quality association rules or frequent itemsets is usually applied using metrics such as support, confidence and lift. This is an unsupervised method, because we do not know in advance what the frequent itemsets are or which items are more strongly associated with which items. Only after applying decision algorithms can we discover and predict products or items that are closely associated with each other and find conditional probability dependencies. The most commonly used decision algorithms include:

- Apriori
- FP-growth
- Eclat

Semi-supervised learning (learning with partial supervision)

Since annotating or labelling data is usually time-consuming and costly, many unlabelled instances or examples and few labelled instances are often available. Some algorithms are able to work with data that are partially labelled. This is called semi-supervised learning, an example of which with two classes is shown in the figure below. The unlabelled examples (grey circles) help to classify a new instance (the cross) into the “triangle” class rather than the “square” class, even though it is closer to the labelled examples of the “square” class.

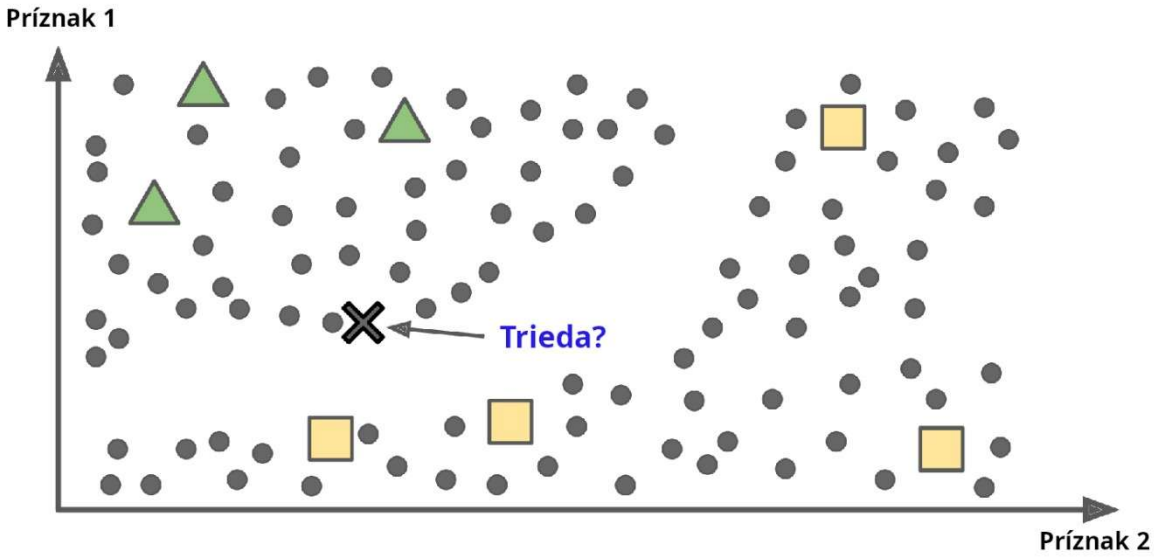


Fig. : Example of semi-supervised learning

Some photo-hosting services, such as Google Photos, are a good example of this. When you upload all your family photos to the service, it automatically recognises that the same person A appears in photos 1, 5 and 11, while another person B appears in photos 2, 5 and 7. This is the unsupervised part of the algorithm (clustering). Now all the system needs is for you to tell it who these people are. Just add one tag per person 4 and it is able to name everyone in every photo, which is useful when searching for photos. Most semi-supervised learning algorithms are a combination of unsupervised and supervised algorithms. For example, Deep Belief Networks (DBN) are based on unsupervised components called Restricted Boltzmann machines (RBM) stacked on top of each other. The RBMs are trained sequentially in an unsupervised manner, and then the whole system is fine-tuned using supervised learning techniques.

Reinforcement learning (reward-based learning)

Reinforcement learning methods differ somewhat from conventional supervised or unsupervised methods. The learning system, which in this context is called an agent, can observe the environment, select and perform actions and, based on these decisions, receive rewards (or penalties in the form of negative rewards, as shown in Figure 14). It must then learn by itself what the best strategy is, called a policy, in order to obtain the greatest possible reward over time. The policy defines which action the agent should choose when it is in a given situation. The main steps of the reinforcement learning method are listed below.

1. preparing the agent with a set of initial policies and strategies
2. observing the environment and the current state
4. selecting the optimal policy and performing the action
5. obtaining the corresponding reward/penalty
6. updating the policy if necessary
7. iteratively repeating steps 2–5 until the agent learns the most optimal policies

An example of a real problem could be a situation in which you are trying to get a robot or machine to learn to play chess. In this case, the agent would be the robot, and the environment and states would be the chessboard

and the positions of the chess pieces. A suitable reinforcement learning methodology is shown in the figure below.

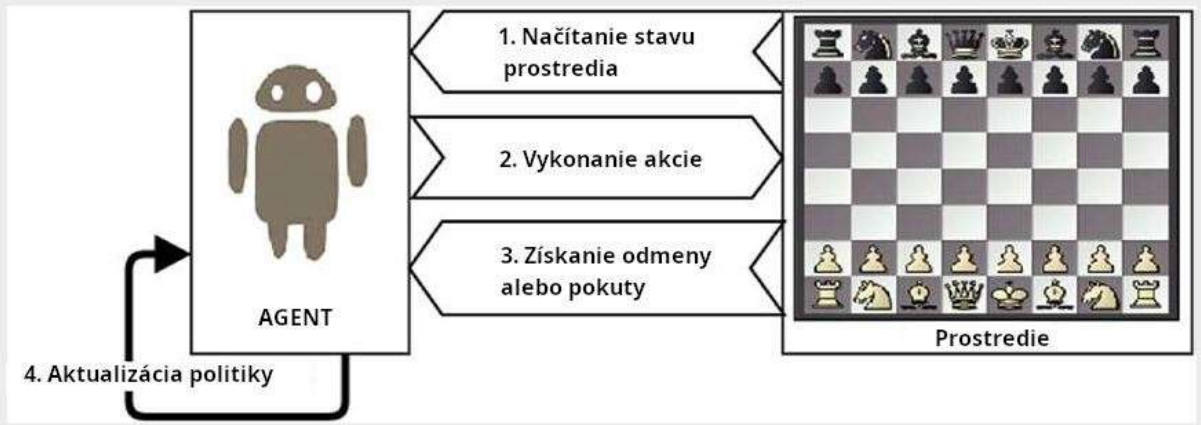


Fig. : Reinforcement learning – training an agent to play chess

Many robots, for example, implement reinforcement learning algorithms in order to learn to walk. DeepMind's AlphaGo program is also a good example of this type of learning. It made the newspaper headlines in May 2017 when it beat world champion Ke Jie at the game of Go. It learned its winning policy by analysing millions of games and then playing many games against itself. Note that learning was switched off during the matches against the champion – AlphaGo merely applied the policy it had learned.

Implementation

In our system, we initially decided to implement ML/AI tools for selected types of threats and, based on the tests performed (detailed results are given in the underlying documents), we decided to use the following algorithms and methods:

- spam/ham detection – XGBoost
- phishing/spoofing detection – HDBSCAN
- DoS/DDoS detection – a combination of detection-boundary and prediction-tunnel methods
- malware/Trojan horse detection – Random Forest
- detection of advanced persistent threats (APT) – Random Forest

A detailed description of the individual algorithms and methods is given in output V9 – Description of the selected algorithm meeting the specified criteria.

Security notifications and alerts sent by an IDS/IPS tool such as Suricata will be checked and evaluated in our system in terms of threat severity using the methods listed above.

2.3 Solution Intent

The INNOSOC project focuses on industrial applied research in the field of key digital technologies, namely cloud technologies and cybersecurity. The project thereby reflects the requirements of this call in the area of the

transition to a digital economy, as well as the current needs of the commercial sphere (the offer of modern IT services), the needs of society (a higher level of security and a sufficient number of IT professionals and specialists) and, finally, the linking of educational and research institutions with practice (education and applied research). The main focus and objective of the project is the application of innovative approaches and the use of open modern technologies to create a flexible and multi-tenant cybersecurity operations centre (SOC) as an open and accessible cloud service that will be available to diverse organisations. At the level of management of the virtual cloud environment and the service, as well as in anomaly detection at the level of SOC operations, the solution will use and contain elements of automation and deep machine learning DNN (Deep Neural Networks). Such a solution will reflect the diverse requirements of customers and will be built on the integration of available open technologies, which guarantees its openness into the future as well.

The development of new technologies and innovation bring changes that transform not only the economy but our entire society. Digitalisation, automation and the use of artificial intelligence AI (Artificial Intelligence) are changing established business models in individual sectors of the economy. On the other hand, however, they raise questions related to information security and cybersecurity. The response of the European Union (EU) to these dynamic changes is contained in Directive 2022/2555 (**Document 32022L2555**) on measures for a high common level of cybersecurity across the Union, also known as NIS2. The NIS2 Directive, and presumably also its national implementation, extends security requirements to a wider range of organisations, which will have to respond to its requirements by adopting appropriate security measures, for example in terms of the detection and reporting of security incidents. The project therefore focuses on implementing a cybersecurity service solution and places emphasis on developing an open solution in the form of an on-premise solution or service that will be available to diverse interested parties falling within the group of affected organisations under NIS/NIS2 and their national implementations, such as the **Cybersecurity Act No. 69/2018 Coll**.

The INNOSOC project sets itself the ambitious objective of creating an open cloud solution for a SOC (Security Operating Center) service to improve the cybersecurity of organisations, with elements of automation and deep machine learning (DML – Deep Machine Learning). Both areas – the application of ML methods and automation elements to the SOC solution and the overall appropriate integration of all components of the SOC solution – are currently a requirement and a response to the ever-increasing demands placed on SOC solutions. The use of AI and ML elements is, generally speaking, a very topical trend today. It is therefore natural to seek to integrate them into SOC solutions as well, where they will help to detect and respond to emerging security events and threats more quickly and effectively, ideally in real time. Their incorporation into SOC solutions is thus currently one of the most topical challenges. In the overall picture of the project's intent, which combines the complexity of building scalable on-premise cloud/container environments with building and operating a virtual SOC service with ML/DNN elements, this is a complex and innovative undertaking. When submitting the project, the project team did not find a similar comprehensive SOC solution combining AI/ML with deployment as a multi-tenant and open cloud service offered by commercial entities, nor were similar solutions in this field described in the relevant professional literature.

As a commercial company, BRAIN:IT has recorded a sharp increase in demand for an affordable security service or solution that does not require specialised human capital, in particular for micro, small and medium-sized enterprises, as well as smaller public administration institutions that fall within the group of affected organisations under the **Cybersecurity Act No. 69/2018 Coll**. (or NIS/NIS2) and do not have an adequate budget or specialised security experts to fulfil these requirements and obligations.

The project thus aims to provide the partner with a competitive advantage consisting in the development of an innovative security service offered in the form of a scalable virtualised service used by multiple customers, thereby contributing to its easier accessibility and to fulfilling the intent of the NIS2 Directive. The project assumes that the results of research into new procedures and concepts of original innovative security service solutions will have high potential applicability in socio-economic practice.

The implementation of the INNOSOC project activities envisages the preparation of technical analyses, conceptual designs, technical studies, plans for further development or evidence of technical viability, findings and recommendations. The aim is to obtain convincing evidence of the technical capability of the solution and

to ensure its readiness for further development. All the activities planned and described here, as set out in Chapter 3, are directed towards fulfilling the main objective of the project at TRL3 level. After further development, it will be suitable for deployment in the form of a commercial product. At the same time, it will be suitable for use as a platform supporting practical education and further development and research.

To ensure that the project objectives are achieved, the PRINCE2 project management methodology will be used, which both consortium members (UNIZA FRI and BRAIN:IT) use to manage their projects. The methodology is based on principles applicable to diverse types of projects regardless of their size. The PRINCE2 methodology is built on several key elements and components that reflect the needs of the project in terms of fulfilling the set objectives and outputs of the project. The methodology is flexible and adaptable with regard to the submitted project and the organisations of the applicant and the partner. An important aspect of the PRINCE2 methodology with regard to this project is the regular review of project implementation and adaptation to changes that may occur during the course of the project. We consider this very important given the field of the project, which in broader terms is information and communication technologies (ICT) and the related products and services, which change very rapidly and dynamically as a result of the informatisation of society as a whole. These changes need to be responded to continuously and within a short time, thereby ensuring the required level of sustainability and competitiveness of the project as well as its outputs.

The TOGAF methodology (Open Group Architecture Framewor) will be used in designing the solution architecture, and Archimate 3 and UML v2 will be used to model it. ISTQB (International Software Testing Qualifications Board) techniques will be used to verify and test the functional requirements, ensuring the required quality of the solution. In the proposals for system management and the formulation of recommendations, the project will rely on the ITILv4 methodology for designing complete service delivery management procedures and ensuring the specified operational parameters. Experiments and data sources will be implemented on the UNIZA FRI network infrastructure as well as on our own private cloud computing system built on OpenStack with extended support for Kubernetes/containers.

Within the INNOSOC project, we will comply with the rules as defined in the national strategy for open science approved by the Government of the Slovak Republic in 2021, which are also among the strategic objectives of UNIZA. The implementers will gradually publish all available information and results of the project in a relevant form in domestic and international proceedings and journals that provide open access. The outputs achieved within the project will thus be available in a form that enables their further use for scientific and research purposes, as well as the availability of the results to the general public, in line with the principles of "***open access***". In addition, our team will organise various dissemination activities, such as a specialist symposium for both the professional and the lay public, focused on disseminating the project results. Finally, an open-access website containing information on the subject matter addressed will be operated within the project.

2.4 Solution Objectives

2.4.1 Main objective of the project

In the current digital era, when organisations face increasingly sophisticated security threats, securing data and protecting against cyber attacks are key aspects of their success and competitiveness. The EU is also aware of this within the Digital Europe programme, which focuses on bringing digital technologies to businesses, citizens and public administration bodies. Research and development of new security solutions and procedures is therefore a constant imperative of our time and is in line with the objectives of the European Cybersecurity Competence Centre and Network. On the other hand, cloud technologies have become a driving force for the digital transformation of many processes in organisations, as also pointed out by The European Alliance for Industrial Data, Edge and Cloud, whose aim is to support the development and deployment of cutting-edge next-generation cloud technologies in practice. These technologies give individual organisations flexibility,

scalability and efficiency. This project focuses on the development of a flexible cloud service that combines these two key areas – security and cloud technologies.

The main objective of the INNOSOC project is the design of an open, multi-tenant security operations centre (SOC) solution that will be provided as a performance-flexible cloud/container service with modules for anomaly detection and prediction using machine learning and neural networks.

The innovative element of this project is the implementation of automation elements and advanced anomaly detection methods using ML, specifically deep neural networks (DNN). These will, on the one hand, be applied to monitoring and predicting anomalies in the operations of the cloud solution itself, thus contributing to smoother and more secure operation of the platform with a higher degree of availability, reliability and performance. Above all, however, the developed ML methods will be used to detect security anomalies in data obtained from the systems (components) of the SOC centre, complementing or increasing its efficiency and effectiveness. The INNOSOC project has the potential to deliver a real open platform where knowledge and new solutions in the field of cybersecurity and open cloud solutions will be applicable.

Research into open multi-tenant cloud-based SOC solutions itself is relevant because of the need for efficient, scalable and cost-effective security solutions in an increasingly complex and interconnected digital environment, for which demand will grow also owing to the national implementations of directives such as NIS2 and the requirements of a more secure Europe. At the same time, the innovative involvement of ML methods in the operation of the platform and the service follows the trend of developing multi-domain applications of ML methods, which supports powerful analytics and decision-making support, faster and more effective response, accuracy and scalability of the solution and, last but not least, develops automation elements as a technique that saves human resources. The project will thus contribute to addressing new challenges in the fields of cybersecurity, cloud computing and the practical education of cyber specialists.

An important aspect of openness is also associated with the project. The INNOSOC project will be built on open and freely available virtualisation technologies (cloud, alternatively container) with elements of management and monitoring automation as a technological means for flexible and scalable operation of the proposed service. The SOC service itself will also use the integration of open solutions supplemented by the implementation of AI/ML (DNN) elements, for example for rapid anomaly detection. This approach will contribute to the sustainability and availability of its outputs for a wide range of users. Our efforts are directed towards creating a more secure and reliable digital environment; they bring new knowledge and solutions, support economic growth and increase user trust in the use of digital services and in digital transformation.

The procedures and methods developed and the outputs achieved in the project will enable the development of an innovative cybersecurity service solution in the cloud. Given the state of the market for similar solutions and the requirements arising from new EU directives in the field of cybersecurity, the future commercial potential of this solution is very large, and not only on the Slovak market. It will enable customers of BRAIN:IT (current or potential) to provide protection for their users and assets. At the same time, it will improve the efficiency and competitiveness of the organisation concerned. It will enable UNIZA FRI to build a SOC solution that will contribute to the development of the practical skills of full-time students, supported by the project- and problem-oriented education implemented at UNIZA FRI, as well as contributing to the development of further scientific and research activities in the field of security and cloud computing (CC). By its content, the project thus meets all the requirements of a future solution with commercial potential.

2.4.2 Specific objectives of the project

Achieving the main objective is a comprehensive and complicated process, both in each of its individual components and as a whole. To ensure that the main objective of the project is achieved, the following **specific objectives of the project** were identified in line with the application of architectural approaches to the development of complex information and communication systems:

1. Design of a technological solution for operating the SOC service using virtualised solutions, developed up to TRL3 level.
2. Design of an intelligent system for monitoring technological resources and services.
3. Design of the architecture of the innovative elastic INNOSOC service itself as an integration of open solutions and components.
4. Design of suitable machine learning algorithms applying deep neural networks to the task of detecting and predicting security and operational anomalies.
5. Creation of a platform for sharing the knowledge acquired in the course of the project and in line with the principles of open science.

The individual specific objectives will be addressed in mixed teams of the partners UNIZA FRI and BRAIN:IT, using the SMART methodology, in order to achieve the main objective of the INNOSOC project.

(S) Definition of specific objective No 1: *Design of a technological solution for operating the SOC service using virtualised solutions, developed up to TRL3 level, according to the requirements defined for operating the SOC service.* The solution will be designed so that it can be easily extended and adapted to needs and load according to customer requirements. The solution design will include mechanisms for automating deployment and sustainability, infrastructure monitoring and management, updating and backup, reliability and availability, orchestration, automation, and the securing and handling of related processes. A systems approach will be used in the design for comprehensive treatment of the subject.

(M) To measure the achievement of the individual activities of specific objective 1, a range of suitable key metrics – KPIs (Key performance indicators) – will be used, such as MTTR (mean time to recovery) – the average time to restore the service, HA (High Availability) in terms of the availability of nodes that may be disabled while maintaining a functional system, the average system response time for different load levels, the degree of automation, the cost and efficiency of the solution, and others. Within the measurement of this objective, regular evaluation of the results achieved will also be carried out, along with the preparation of documentation in the individual phases of the solution's implementation, validation of whether the defined expectations have been met, verification of whether the requirements placed on the system have been met, comparisons, testing reports, etc.

(A) To achieve the implementation of the design, a holistic and systems approach will be used for comprehensive treatment of the subject, from the initial requirements through the selection of suitable components to the subsequent creation of the architecture design of the comprehensive virtualisation platform of the solution. The design will use modern open virtualisation technologies. The resulting design will meet the technical and security requirements associated with operating the innovative virtual INNOSOC service with a higher degree of automation of operational activities.

(R). We consider the proposed solution realistic and feasible, particularly in view of the project staffing, where the individual implementers have many years of extensive experience with modern virtualisation technologies, which include the use of on-premise cloud computing solutions such as OpenStack or solutions in the form of microservices on Kubernetes, etc. **(T)** The timeframe for implementing the specific objective is described in more detail in Chapter 3.

(S) Definition of specific objective No 2: *Design of an intelligent innovative system for monitoring technological resources and services.* The system design will take into account the selection of suitable parameters for tracking the current state of resources and performance, identification of

problems, rapid detection of outages and security threats, improvement of application performance, efficient use of resources, data security. rapid scalability. The results will be provided through the designed visualisations (dashboards), alerts and reporting. ML elements will be incorporated into the solution with the aim of detecting and predicting anomalous events in the operations of the virtualised solution based on collected attributes, logs and other information from the various sources used. The proposed system will provide extended options for configuring alarms and alerting.

(M) The measurability of specific objective No 2 will be based on information obtained from suitably chosen key metrics in relation to the objective. Such key metrics may include Outage detection accuracy, as a measurement of the system's accuracy in detecting outages and problems in real time; Speed of detection and response, focused on verifying how quickly the system identifies outages and responds; Scalability and efficiency of resource use, focused on tracking or monitoring the use of resources and services of the cloud platform. On the basis of this metric, optimal use of cloud resources can be ensured to prevent excessive costs; Configurability of alarms, as a measure of the adaptability of alerting. Performance of the system and applications in terms of response time and the volume of data, for example logs, and their processing.

(A) The achievability of specific objective No 2 is closely linked to the results of the individual defined key metrics. The data obtained will be compared within the individual control stages of the project and will also be compared against the definition of specific objective No 2 and the main objective of the project. Verification and evaluation will also include a performance evaluation, i.e. whether the objective was achieved based on the monitoring results. Any shortcomings identified will be reflected in corrective measures in the form of system updates, configuration changes, etc. The design will use modern open technologies. At the same time, as part of verifying and evaluating the objective, documentation will be prepared on all steps, monitoring results and updates of the summary output based on these findings. The documentation will also include comparisons, testing reports, a final report and other publications.

(R). To address such a complex topic, it is necessary to understand the views of the stakeholders, where the output leads to correctly determined monitoring parameters (metrics), reporting and the setting of a service level agreement (SLA). Into such a complex subject area, the project team intends to incorporate supervision elements applying machine learning and thus to design an **innovative monitoring system**. In terms of the solution, we expect that the intelligent monitoring system will help prevent problems, predict failures and optimise downstream processes. On the basis of the set partial objective, we propose the following realistic areas of the solution: design of monitoring metrics and SLA handling, design of the monitoring system architecture and monitoring of the computing and network infrastructure, selection and testing of solutions for the architecture components, visualisation, reporting, alarms and alerting, and the like. These areas of the solution should enable the creation of a comprehensive monitoring system capable of improving the operation and security of virtualised services in the cloud and providing valuable information to administrators and operators. On the basis of the above information concerning intelligent systems for monitoring the environment and services in the cloud, specific objective No 2 can be considered realistic.

(T) The timeframe for implementing specific objective 2 is described in Chapter 3.

(S) Definition of specific objective No 3: *Design of the architecture of the innovative elastic INNOSOC service itself as an integration of open solutions and components*, which takes into account a proactive approach to security, for the purpose of monitoring and protecting information assets and

implementing the overall security strategy of organisations – tenants in the cloud – to which the service will be provided in this way. The solution will include the design of the roles of the specialists needed in the SOC centre, and the design of the processes and technologies that will handle continuous monitoring, prevention, detection, investigation and response to cyber threats.

(**M**) The measurability of specific objective 3 is important for verifying the functionality of the entire SOCaaS solution and for the continuous improvement of the designed processes, human capital and technologies used in SOCaaS. During the project, various aspects of the performance of the SOCaaS service will be measured, with the following 5 metrics used as key performance indicators: dwell time, during which threat actors have access to the network before they are detected and their access is stopped; MTTR (Mean Time To Detect) – the average time needed to identify valid security incidents in the network; MTTR (Mean Time to Respond) – the average time needed to stop a security incident and subsequently remediate it; MTTC (Mean Time to Contain) – the time needed to resolve an incident such that it does not cause further damage to systems or data; and TTC (Time to Control) – the time needed to stop the spread of malware in the network. Further suitable metrics are Solution costs, i.e. the total costs of implementing and operating the SOCaaS solution. This metric will present information on the cost-effectiveness and efficiency of the SOCaaS architecture compared with commercially available alternative solutions. Implementation time, i.e. the time within which the solution will be successfully deployed, including its configuration. The faster the solution can be implemented, the more effectively it will be possible to respond to potential threats.

(**A**) In terms of processes, the solution will include the individual categories so that all basic processes are covered and achievable, including the classification of events and their triage, the ticketing process, the prioritisation and analysis of monitored events, the recovery and remediation process, and the evaluation and audit process. From a technological point of view, open solutions will be analysed, selected and used, which significantly reduce costs compared with commercial substitutes. From this perspective, such solutions can be considered attractive for various organisations and users with a limited budget. Another advantage of open source solutions is the high level of flexibility and adaptability of the code to the specific needs of the organisation. The SOCaaS service will include open source technological solutions that will provide the necessary functions of security monitoring, automated discovery of information assets, vulnerability assessment, behaviour monitoring, prevention and detection on endpoint devices and in the network, and collection and analysis of security events and logs from the entire network with the possibility of aggregation, correlation and analysis of alerts. On the basis of the above information concerning the design of the SOCaaS architecture based on open source solutions, specific objective No 3 can be considered realistic.

(**R**) The SOCaaS service will be designed so that, when recovering systems in the event of a cybersecurity incident, all processes and technologies are set up to take into account the main components of system recovery after a crisis event and of business continuity planning, such as the recovery time objective (RTO), which represents the maximum tolerated outage interval of a given device, system, network or application that may occur after a fault or critical event. Partial objective 3 will draw on and build upon the results of the two previous partial objectives 1 and 2 and, in monitoring and detecting incidents, will also use innovative machine-learning-based anomaly detection methods from partial objective 4.

(T) The timeframe for implementing specific objective 2 is specified in more detail in Chapter 3.

(S) Definition of specific objective No 4: *Design of suitable machine learning algorithms applying deep neural networks to the task of detecting and predicting security and operational anomalies.* Achieving this specific objective is intended to enable the functionality of the virtualisation platform, and in particular of the SOCaaS within it, to be extended with solutions for detecting and predicting security (operational) anomalies using advanced technologies in the sense of ML, specifically deep neural networks.

(M) Suitable key metrics for measuring the success of the algorithms may include, for example, anomaly detection accuracy, the rates of false positive and false negative results, the Number of identified security and operational anomalies, detection speed,, prediction accuracy, resource intensity and the like. Given the dynamic nature of the sector in which the project will be implemented, it is assumed that the above procedure and key metrics will be adjusted and possibly supplemented during the project, which will make it possible to achieve the best possible result in implementing specific objective 4.

(A) To achieve specific objective No 4, it is necessary to define procedures for obtaining suitable data that will serve as the basis for training ML algorithms to identify and predict anomalies. The implementers of the project will focus in particular on the Identification and acquisition of suitable data from the various components of the visualisation platform and the SOCaaS components (logs, reports on network flows, individual packets, security events, notifications and the like) suitable for building training and test datasets. On the basis of the set objectives, ML models will be designed, trained and tested with the task of identifying and predicting security and operational anomalies, with the aim of achieving adequate model accuracy with adequate resource consumption and sufficient detection speed. This stage will also address the selection of the most suitable algorithm (application of the ML model) based on the compared results and its verification on real data under laboratory conditions, followed by adjustment of the solution based on the information obtained.

(R) Given the growing number of complex cyber threats, the importance of developing and deploying sophisticated tools for their detection and prediction is increasing. Machine learning and neural networks can identify patterns and anomalies that could not be captured by traditional methods. This is caused mainly by the large volume of generated data that needs to be analysed in the field of cybersecurity and virtualised operations, moreover with the imperative of real-time detection conditions. It is precisely suitable machine learning methods, for example deep machine learning applied to the task of detecting security anomalies, that are able to work in this way and identify patterns in the large amount of available data. At the same time, their advantage is the ability to learn and adapt to new threats and patterns, on the basis of which they can be effective in identifying new and unknown (zero-day) threats. The possibility of training such a specialised neural network may lead to a lower frequency of false alarms compared with traditional systems, i.e. dealing with fewer unnecessary warnings. Given the project staffing, where many of the implementers have many years of extensive experience in designing suitable machine learning algorithms applying neural networks for the detection and prediction of security anomalies, specific objective No 4 can be considered realistic.

(T) The timeframe for implementing specific objective 1 is described in more detail in Chapter 3.

(S) Definition of specific objective No 5: *Creation of a platform for sharing the knowledge acquired in the course of the project and in line with the principles of open science.* This objective reflects the need to disseminate the knowledge and results of the INNOSOC project.

(M) The measurability of specific objective No 5 will be based on information obtained from suitably chosen KPI metrics in relation to the objective. Such KPIs may include the number of active users on the platform or the number of visits over a given period. This metric will present information on the extent to which the platform is used and how often. Another metric is the Number of shared documents, articles, datasets or other contributions on the platform. This metric identifies the activity of team members in sharing knowledge in line with the set objective of the project. A further metric is Feedback, i.e. the number of posts, discussions and shares of available materials and knowledge by users. This metric will provide important information on how users perceive the platform, its quality and the quality of the information shared in the context of the project. Last but not least, a metric will also be the number of scholarly articles on the subject published in journals and proceedings. The above key metrics will ensure the measurability of specific objective No 5 and, at the same time, will enable the creation of an online platform for sharing the knowledge acquired in the course of the project.

(A) Awareness-raising and dissemination activities relating to the subject matter of the project will be implemented with the support of a web platform/site that supports content sharing. Final reports, reports from experiments and testing, guides and the updated complete project documentation will be available there. The project outputs will also be published in the form of papers at domestic and international conferences and scholarly publications in indexed scientific journals and at international scientific conferences, preferably those supporting “open-access” principles.

(R). With the creation of a platform for sharing the knowledge acquired by the project partners and with the general public, using feedback options and sharing datasets with the public to support further research and development in the context of the project, specific objective No 5 can be considered realistic.

(T) The timeframe for implementing specific objective 5 is planned for is described in Chapter 3.

2.5 Work Packages, Outputs, Measurable and Performance Indicators (KPIs)

2.5.1 List of work packages

Work package code	**Work package name**	**Entity responsible for managing the work package**	**Start of activities** *(state the month of project implementation)*	**End of activities** *(state the month of project implementation)*
KPB1	Design of virt. solution	UNIZAFRI	M1	M16
KPB2	Design of SOCaaS service	UNIZAFRI	M6	M18
KPB3	Design of ML algorithms	UNIZAFRI	M5	M18
KPB4	Solution architecture	BRAIN:IT	M13	M22
KPB5	Solution implementation	BRAIN:IT	M19	M24
KPB6	Publication outputs	UNIZAFRI	M8	M24
KPB7	Awareness-raising activities	BRAIN:IT	M13	M24
KPB8	Proj. documentation	UNIZAFRI	M1	M24

KPB9	Final report	UNIZAFRI	M23	M24

2.5.2 List of outputs

Output number	Output	Work package code	Entity responsible for the output	Type	Access and dissemination	Method of verification	Delivery of output (project month)
V1	result of the analysis of available solutions	KPB1	UNIZAFRI	documentation	N	complete documentation	M3
V2	list of defined criteria	KPB1	UNIZAFRI	list	N	approved list	M5
V3	description of the selected solution meeting the specified criteria	KPB1	UNIZAFRI	documentation	N	complete documentation	M7
V4	result of the analysis of suitable components	KPB2	UNIZAFRI	documentation	N	complete documentation	M3
V5	list of defined criteria	KPB2	UNIZAFRI	list	N	approved list	M5
V6	description of dashboards and the monitoring and reporting system	KPB2	UNIZAFRI	documentation	N	complete documentation	M7
V7	result of the analysis of suitable algorithms	KPB3	UNIZAFRI	documentation	N	complete documentation	M10
V8	list of defined criteria	KPB3	UNIZAFRI	list	N	approved list	M12
V9	description of the selected algorithm meeting the specified criteria	KPB3	UNIZAFRI	documentation	N	complete documentation	M17
V10	List of functional and non-functional requirements	KPB4	BRAIN:IT	list	N	approved list	M14
V11	Business architecture	KPB4	BRAIN:IT	documentation	N	approved documentation	M16
V12	Technology architecture	KPB4	BRAIN:IT	documentation	N	approved documentation	M19

V13	DNR – Detailed Solution Design including annexes	KPB4	BRAIN:IT	documentation	N	complete documentation	M22
V14	Functional infrastructure	KPB5	BRAIN:IT	documentation	N	functional infrastructure	M19
V15	SW components	KPB5	BRAIN:IT	list	N	approved list	M20
V16	Functional PoC	KPB5	BRAIN:IT	documentation	N	stable functional cloud service	M22
V17	Test results and evaluation	KPB5	BRAIN:IT	protocol	N	approved protocol	M24
V18	published indexed articles in journals	KPB6	UNIZAFRI	articles	P	library registration, indexing	M24
V19	conference papers included in proceedings	KPB6	UNIZAFRI	proceedings	P	library registration, indexing	M24
V20	Creation of an online platform	KPB7	BRAIN:IT	information portal	P	online availability	M15
V21	Available online awareness-raising materials and outputs	KPB7	BRAIN:IT	educational materials	P	approved list	M22
V22	awareness-raising activities carried out	KPB7	BRAIN:IT	activities	P	list participants, photo documentation	M24
V23	updated project justification	KPB8	UNIZAFRI	documentation	N	approved by the project	M3
						board	
V24	updated logical framework of the project	KPB8	UNIZAFRI	documentation	N	approved by the project board	M3
V25	updated tools tracking the project time allocations and the fulfilment of objectives	KPB8	UNIZAFRI	documentation	N	approved by the project board	M5
V26	updated drawdown of the set budget	KPB8	UNIZAFRI	documentation	N	approved by the project board	M6
V27	risk register	KPB8	UNIZAFRI	register	N	approved by the project board	M3
V28	interim project report	KPB8	UNIZAFRI	report	P	approved by the project board	M12
V29	final project report	KPB9	UNIZAFRI	report	P	approved by the project board	M24

2.6 Solution Scope

The scope of the solution is defined by the work packages, outputs and the work schedule with durations. Further information is given in the chapter Implementation and deployment schedule, dependencies.

2.7 Methodology

To achieve the project objectives and the required outputs of the individual project activities, a systematic methodological procedure consisting of the following steps will be used: *defining the research objectives; identifying parameters for selecting relevant information sources; searching for relevant information sources; storing information sources; verifying information sources; classifying information; evaluating and using information; preparing documentation; checking the fulfilment of the defined research objective*.

A prerequisite for achieving the required results of the individual project activities is the use of relevant information sources, in particular: *specialist literature by domestic and foreign authors relating to the subject matter of the project; sources recognised by the scientific community (in particular the scientific publication databases Web of Science, Scopus, etc.); case studies; available secondary data sources (relevant databases, product sheets, documentation, etc.); the expertise and skills of the project implementers and interviews with experts in the field of the project's subject matter.*

Given the extent of the above information sources, it will be necessary to identify those that will be relevant to the project. For this purpose, the concept of the PRISMA methodology model will be used, i.e. a procedure for the systematic identification, selection, evaluation and synthesis of various types of sources, including publications, case studies, scientific articles and the other information sources mentioned. In this way, a structured and objective analysis will be ensured

In addition to the above procedure and the PRISMA concept, the project objectives will also be achieved by other methods, which can be divided according to the following classification: information-gathering methods; information-processing methods; problem-solving methods; solution evaluation methods.

Within the **information-gathering methods**, the *Document analysis method* will be used to analyse qualitative data, which are expected to occur mainly in connection with diverse documentation of technologies, architectures or open source solutions. The *Case study analysis method* will be used to process qualitative and quantitative data from examples of good practice relating to the subject matter, such as available solutions and the relationships between their individual elements, and the like.

Within the **data-processing methods**, the *Qualitative evaluation method* will be used to formulate conclusions from the analysed data sources, the experiments performed, etc. Another method is the *Comparative method*, which will be used to compare the results of the experiments performed, the training of ML algorithms and the tests carried out within the relevant project activities. The *Data generalisation method* will be used in drawing conclusions from the analyses performed, monitoring, defining the functional and non-functional requirements of the proposed project solution, etc. The *Data categorisation method* will be used to support the design of the SOCaaS service for the selected virtual. solution, etc. The *Holistic approach method* will be used to formulate summary conclusions for the ongoing updating of the project justification, the ongoing updating of the project logical framework, updating risks, preparing the interim project report, and obtaining and linking the materials needed to prepare the final project report, etc.

Within the **problem-solving methods**, the *Intuitive approach method* will be used to apply the knowledge and experience of the project team. The *Synthesis method* will be applied mainly in defining the solution at TRL3 level. The *Modelling method* will be used in the project for the activities of modelling diagrams in UML and Archimate and, overall, in preparing the DNR. *Logic*, as a system of logical means, will be applied in formulating

conclusions, defining the functional and non-functional requirements that the resulting solution must meet, defining constraints, defining relationships between elements, etc. *Creativity* will also be applied to a substantial extent, particularly in defining constraints, designing the individual solution elements contained in the architecture, etc.

As the **solution evaluation method**, the *Comparative method* will be used to compare the knowledge obtained by analysing various sources in connection with the selection of the open source solution and tools and the finalisation of the architecture design, the evaluation of proof-of-concept tests, the final design of the ML algorithm, the detailed solution design, etc.

The applicant UNIZA FRI participates in the implementation of tasks and twenty-two research projects (see Section 1.2). In implementing these projects, ensuring the above procedure and using the methods mentioned (and many others) is an essential part of achieving the required objectives and results of the project. The partner BRAIN:IT has many years of experience in software development and hardware administration (see Chapter 1) and has also long cooperated with the applicant, not only in the technical field but also in education, in supervising final theses and in other activities in which contact with the proposed procedures and methods is an integral part.

Given the rapidly evolving field of information and communication technologies in which the project is situated, challenges may arise from insufficient application of the above procedure and methods in terms of obtaining the information needed to carry out the activities within the individual KPBs. Should such a challenge materialise, we propose to overcome it by applying further methods beyond those listed. One of the proposed methods is the **Grounded theory method** (GTM – grounded theory method), which is suitable for rationalising and explaining various phenomena, particularly after obtaining data from observation and testing activities. Another proposed method is the **Scientific mapping method,** which will make it possible to create objective criteria for evaluating the results of the analyses performed, for example by visualising similarities, etc.

Using the above methodology, primary and secondary data will be obtained, which will be processed by the project team and published as part of preparing the interim and final project documentation. The integration of open science principles will be ensured in the project by applying open solutions down to the code level, and the results will be stored in databases accessible to the scientific community and the public (see Section 2.2). At the same time, the project outputs will be available on the online platform as part of the KPB7 awareness-raising activities for the dissemination of knowledge (see Section 3.1.1).

The project neither envisages nor **plans to make investments in research** infrastructure, and the existing infrastructure of the partners will be used.

To demonstrate compliance with the principle *of "do no significant harm"* , the checklist according to Annex I of the technical guidance on the application of the "do no significant harm" principle (2021/C58/01) was used. The checklist according to Annex I is shown in the following table.

Indicate which of the following environmental objectives require a substantive assessment of the measure in terms of compliance with the "do no significant harm" principle	Yes	No	If you selected "no", provide a justification

Climate change mitigation		X	The activity supported by the measure has an insignificant foreseeable impact on this environmental objective, because the proposed solution is oriented towards reaching TRL3 level using existing infrastructure.
Climate change adaptation		X	The activity supported by the measure has an insignificant foreseeable impact on this environmental objective, taking into account both direct and primary indirect effects throughout the entire life cycle.
Sustainable use and protection of water and marine resources		X	The activity supported by this measure has an insignificant foreseeable impact on this environmental objective. No risks of environmental degradation have been identified, as no water supply installations or water-using equipment will be installed.
Circular economy, including waste prevention and recycling		X	The activity supported by the measure has an insignificant foreseeable impact on this environmental objective. The measure will not lead to significant inefficiencies in the use of resources or to an increase in waste generation.
Pollution prevention and control of air, water or soil		X	The activity supported by the measure has an insignificant foreseeable impact on this environmental objective. No risks of environmental degradation have been identified, as no equipment using water or polluting the air or soil will be installed, nor will there be any interference with the soil.
Protection and restoration of biodiversity and ecosystems		X	The activity supported by the measure has an insignificant foreseeable impact on this environmental objective. The design of the INNOSOC service is not oriented towards any activity in biodiversity-sensitive areas or in the vicinity of such areas .

The proposed solution and the activities arising from it (see Section 3.1.1) are not among the activities and assets excluded from support (2021/C58/01):

i. Given the main objective (see Section 1.1), no activities and assets related to fossil fuels, including their downstream use, will be carried out within the project.
ii. The project activities *(see Section 3.1.1)* do not envisage emissions trading (ETS) iii. The main objective and the specific objectives of the project (see Section 1.1) demonstrably declare that no activities and assets related to landfills or incinerators will be carried out or planned.
iv. Likewise, the project is not focused on activities and assets related to mechanical biological waste treatment facilities.
v. The project activities will not lead to significant inefficiencies in the use of resources or to an increase in waste generation, i.e. no activities *and assets will be carried out in which the long-term disposal of waste may cause harm to the environment.*

BRAIN:IT has an approved and applied company policy for product development and operations in line with keeping the environmental burden as low as possible (environmental friendly). In addition to the Quality Management System STN EN ISO 9001:2016, the company also holds the certificates “The environmental management system STN EN ISO 14001:2016” and “The Information Security Management System STN ISO/IEC 27001:2014”, Service Management System STN ISO/IEC 20000-1:2021 and Business Continuity Management System EN ISO 22301:2019. In accordance with Slovak law, it holds Certificate of competence to carry out research and development activities number 2020/16471:2-D1230.

The excellence of the project applicant and the partner, in terms of sufficient expertise for the project and its implementation in the required quality and scope to cover all aspects (including a multi- and interdisciplinary approach, knowledge of open science, a FAIR approach to research data, gender equality in research, etc.), is presented in Section 1.2 and Section 2.2 of this project documentation.

2.8 Project Impact and Benefits

Applied research in the field of CC technologies and cybersecurity, extended with the use of ML methods, has potential with several aspects of impact. The development of new innovative cloud and security solutions will in general contribute to accelerating scientific and research cooperation between research institutions and the business environment. This is therefore expected to bring **an increase in innovation capacity** and excellence in research and innovation, the development and popularisation of science at national and international level, and the development of education and training of specialists in the two identified research areas. However, the primary impact of the INNOSOC project is to increase the level of cybersecurity and limit the impact of security threats on business processes by providing an open, multi-tenant and flexible security operations centre (SOC) solution. The solution will be developed and publicly available for future on-premise deployments or in the form of an efficient service offered to the general public, but above all to smaller public administration organisations as well as micro, small and medium-sized enterprises. At the same time, the development and operation of such a service will give the operator increased competitiveness in the relevant market segment, further commercial opportunities with the potential of a future commercial product and, finally, increased trust of future customers. For the user of the service, in addition to a direct increase in security and the fulfilment of legally prescribed requirements, it will also bring a reduction in the costs of operating new security solutions.

The significance of the impact on the **field of knowledge and the scientific community** lies mainly in the specific deployment of already widely used machine learning and neural network techniques to security issues and their usability in the given application area. The solution will thus cover the relevant aspects and ensure the availability of results such as access to training datasets, new methods for recognising anomalies in the operations of the selected virtualisation solution as well as in the task of detecting and predicting security anomalies with an emphasis on the explainability of decisions, and a new concept enabling real-time decision-making. Given the intent of the project and the complexity of the design, a benefit is the availability of technological procedures using systems and architectural approaches that can lead to the creation of comprehensive security solutions.

The **impact on the economy** is also important, in particular through making available an open solution that will be able to reduce the negative economic impacts of security incidents and cyber attacks. At the same time, an important fact for society is that the INNOSOC project can **increase the level of trust** of the general public in the secure environment of electronic government services by raising the level of security of smaller public institutions too, as well as increasing the sense of security in the online electronic space created precisely also by micro, small and medium-sized enterprises, which, in order to remain competitive, are encouraged to offer these electronic services but at the same time are obliged to protect them. The very essence of digitalisation, automation and above all the use of CC solutions with more efficient use of computing resources and the offer of this service to smaller entities can have a significantly positive effect on the environmental burden on our environment.

The project will bring qualitative and quantitative benefits:

Qualitative benefits	Quantitative benefits

• Increased trust in the electronic services of public institutions and private enterprises, • an increased level of security of the electronic services of the state and the private sector, • an increased level of digitalisation and the creation of a more secure online electronic environment, • an increased level of security awareness among the general public, • increasing innovation capacity, • increased competitiveness, • reduced environmental burden, • development of knowledge in the field and impact on the R&D community • development of knowledge in the field of implementing non-linear models in the form of neural networks to address anomaly tasks in cloud systems and security • development of scientific and research cooperation between research institutions and the business environment	• Reduction in the number of security threats and incidents • reduction of the negative economic impact as a consequence of attacks • an increase in the number of students in project-based and practical teaching in the field of cybersecurity and cloud technologies • an increase in the number of final theses on the project topic • announcement and supervision of doctoral thesis topics on the project topic • dissemination of new methods for detecting security threats • applying for new projects in the field in response to open project calls

During all phases of the project, project risks will be continuously evaluated and verified, and the application of mitigation measures will be overseen to ensure successful fulfilment of the measurable indicators and objectives of the project. Both UNIZA FRI and BRAIN:IT have many years of experience in the individual areas of which the INNOSOC project consists. Each of the partial areas is covered by the expertise of the Applicant or the Partner, or of both entities, and together they are able to cover all aspects arising from known as well as unknown risks. It is therefore expected that, on the basis of the proposed procedures, the consortium will achieve the envisaged outputs and the project will have the expected impacts.

The measures proposed in the project and their quality are fully in line with the current security strategy, strategic plans, digitalisation needs, the HORIZON framework programme and the **SK RIS3 2021+** Strategy, as stated in Section 1.2, part **Challenges and problems addressed by the project.** This clearly determines the suitability and correctness of the measures, which are generally regarded as the way to deal with cyber threats.

In terms of the strategy for managing intellectual property rights to the project results, the implementation of the main objective as well as the individual outputs of the INOSOC project, from the complete design through implementation to the conclusion itself, will be defined as **open and available to the general public**. Our aim is to achieve the highest possible transparency, which will culminate in making all intellectual property created during and at the end of the project available in the form of open software.

2.9 Wider Impact of the Project

In terms of medium-term impacts, an increase in the security of cyberspace as such can be considered. At the same time, from a medium-term perspective, this is mainly an economic impact, namely a reduction in the costs arising from cyber incidents and attacks. In the long term, the creation of a knowledge database with an obvious scientific and societal impact can be considered.

In the long term, the INNOSOC project will lay the foundation stone and provide a springboard for all other entities dealing with cybersecurity, which they will be able to use freely, develop the prepared designs in various further directions and thus ensure that they keep pace with the development of new and more sophisticated cyber threats.

In terms of linking cooperation between the two types of organisations, UNIZA FRI and BRAIN:IT jointly undertake that not only during but also after the end of the project they will continue to participate in the operation and further development of the outputs of the INNOSOC project and of topics related to cybersecurity. The joint effort will be to take the project forward in cooperation, on the basis of agreements, to third parties representing smaller public institutions and micro, small and medium-sized enterprises, which already in the initiation phase of the project demonstrably expressed, also in the form of agreements, considerable interest in using the project outputs in real operation.

The target groups with the potential to use the resulting project outputs are:

- Smaller public and state administration institutions
- Educational and research institutions
- Enterprises and organisations (micro, small and medium-sized enterprises)
- Non-profit organisations and civic associations
- End users and the wider public
- The professional community and also the general public

For all small entities, the INNOSOC project will bring an affordable open platform solution, or just a service, thanks to which they will be able to meet the security requirements placed on them arising from new legal norms.

Through the publicly available open platform, the professional community as well as the general public will gain access to the project results, thanks to which they will be able to pursue their partial individual objectives or even participate in the actual development, direction, maintenance and subsequent support after the end of the project.

On the other hand, among the identified risks, one negative impact has also been identified, namely excessive centralisation of SOCaaS, and the main mitigation measure is precisely the possibility of securing such a central point or points by expert groups with a higher level of expertise in the given field, as well as compliance with operational measures in line with ITILv4.

The expected results and impacts of the individual activities of the INNOSOC project will be reflected in the following measurable indicators:

- number of young scientific and research staff under 35 years of age working on the project: 14
- number of scientific and research staff working on the project: 36 (for the partners)
- number of publications: 20, of which 10 indexed in the WoS, SCOPUS or IEEE databases
- presentation of project results (poster, presentation) at domestic scientific conferences: 5
- presentation of project results (poster, presentation) at international conferences: 2
- organisation of one specialist symposium
- number of cooperations with the private sector: min. 2
- number of bachelor's or master's theses announced/completed within the subject matter of the project: 5
- 10 as the number of students involved in project-based teaching on topics related to security and CC
- 10 professional/popular-science texts on CC and security
- One university textbook in the field of CC
- Development of the existing private CC solution

The results as well as the impacts of the project were determined on the basis of the actual situation of the partners, which reflects the scientific, research and educational focus of UNIZA FRI and the commercial focus

of BRAIN:IT, and thus the actual demand for a service of a similar or identical nature as well as for the solution itself, from smaller public administration institutions as well as micro, small and medium-sized enterprises. BRAIN:IT, as a provider of trust services under eIDAS and thus acting as a certification authority, regularly receives suggestions and evaluates marketing strategies, which result in demonstrable demand. This is also supported by the fact that already in the initial phase of the project, cooperation agreements on supporting the use and further extension of the project outputs were concluded on the basis of negotiations.

Potential obstacles to the planned impact of the project

As the main obstacle under the conditions of the Slovak Republic, we identify the still limited use of advanced security solutions focused on collecting information on the operation of information networks and systems, which form an obvious prerequisite for an organisation to engage in using the proposed service. In order to eliminate project risks, the main mitigation measure was set as creating recommendations and procedures for introducing suitable technologies and solutions that will serve as a data source for the INNOSOC service. These include, for example, open-source IPS/IDS, other security devices such as NextGen FW, sources of network and application logs, identity IAM logs and the like, the deployment of which will make it possible to increase the usability of the entire solution for the entities.

3 Requirements Analysis

This chapter focuses in more detail on the analysis of requirements, which represent the essential basic input for the design and subsequent implementation of the solution. Its main aim is to thoroughly identify, gather and systematically describe user expectations as well as the functional needs that the system must meet. This analysis also serves as one of the key inputs for preparing the architectural design and for precisely defining the scope of implementation of the individual parts of the system. The requirements obtained are subsequently divided according to their nature, which makes them easier to navigate and process in the subsequent phases of the project. All requirements are processed in a clear, structured and uniform manner corresponding to a proven methodology. The chapter also creates a framework for further validation, ongoing review and efficient planning of the solution's development.

3.1 Requirements Analysis Procedure

In developing the architecture, this project uses the TOGAF framework and the Archimate language. The use of TOGAF framework components has been adapted for the purposes of this project. Above all, the part of the Architecture Development Method concerning the development of the architecture in all four architecture domains was taken from the framework. The Architecture Development Method represents a process with recommended steps and activities. Although the ADM is often associated with iterative and agile approaches, in view of the objectives and nature of this project it has been adapted to the Waterfall methodology, which is sequential and linear.

The project is developed in individual successive phases.

In the first phase, an analysis of the requirements that define the expectations of users and stakeholders is carried out.

During the second phase, the requirements are transformed into a system architecture design that defines how the system will work. The design is carried out at two levels. The high-level design determines the overall structure of the system, such as the main components and the way they are interconnected. The detailed specification describes each module or component in more detail, including data models, algorithms, interfaces and logical flows. This design is submitted for approval to ensure that it meets the requirements and respects technological, financial and legislative constraints.

After it has been reviewed and approved, the implementation phase begins. During this phase, the individual system modules are implemented according to the specification, and quality and security standards are applied when writing code. Continuous testing of each module will be ensured by UNIT testing. The implementation phase will take place in two steps. In the first step, a version of the application containing functionality covering the specified functional requirements will be deployed, and the first implementation phase will be reviewed and approved. In the second step, the remaining functionality will be implemented and integrated with the existing modules.

The next phase will ensure that the system is tested. It will be verified that all components work together correctly and that the system as a whole meets the requirements defined in the initial phase.

After successful testing, the system is deployed to the production environment. This process includes software installation and environment configuration. This phase also includes user training so that users can use the system effectively.

After deployment, the system is monitored and continuously updated according to user needs and technological changes. Maintenance includes bug fixing, implementation of new features and performance optimisation. The maintenance process must respect all legislative requirements, for example ensuring personal data protection and cybersecurity.

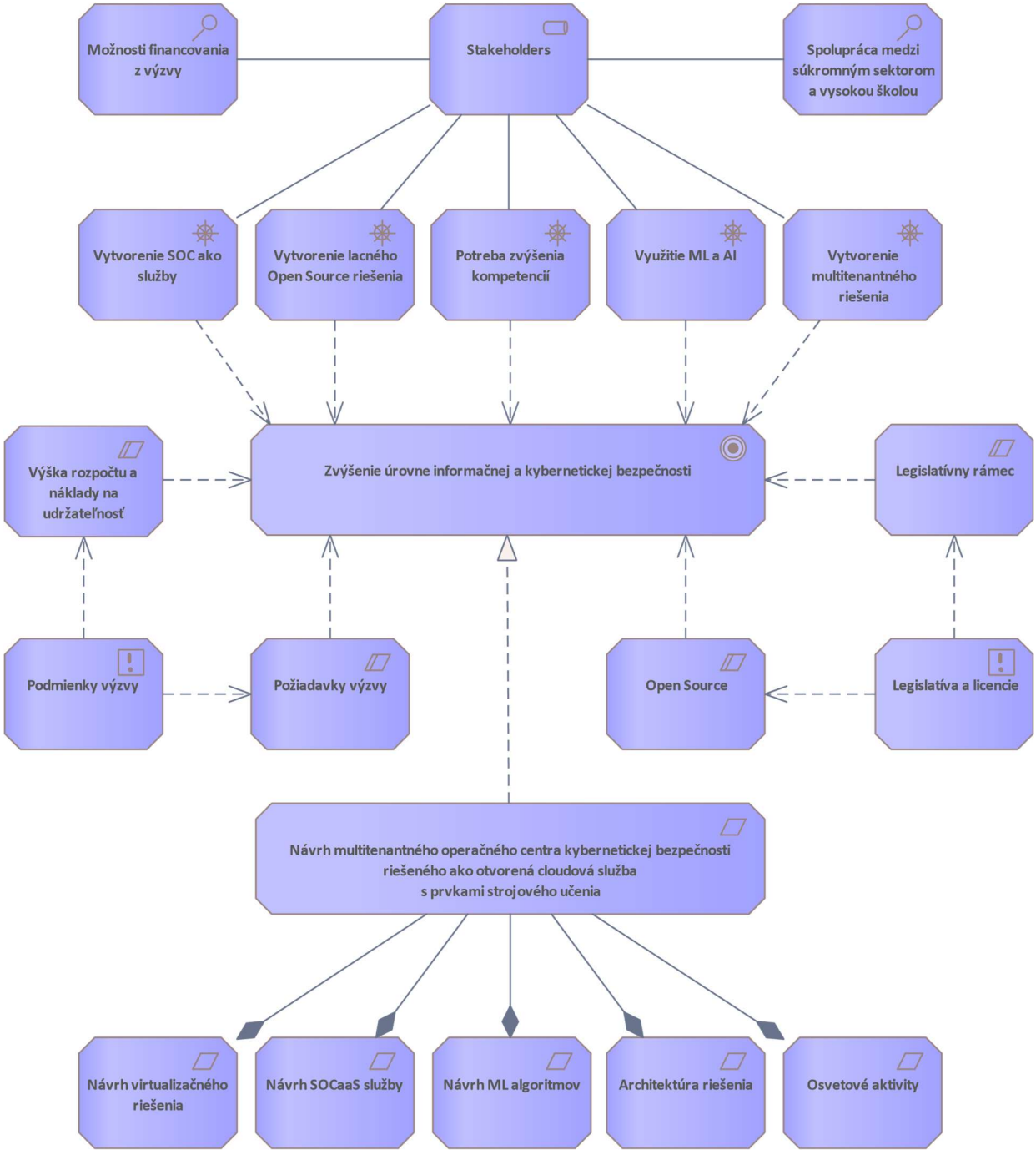
The Enterprise Architect tool is used for the design of the application components being developed.

3.2 User Research

No user research was carried out within the project. The system requirements were defined on the basis of the materials given in the project description at the time the application was submitted.

3.3 Motivation Architecture

3.3.1 Motivation Viewpoints



Motivation Viewpoint

3.4 Requirements Catalogue and Its Mapping to the Design of Their Solution

The requirements catalogue is defined as an annex to this document. The design of the solution to the requirements is the content of this document and is described in detail in the following chapters. Testing will be captured in output V17.

3.5 Legislation, Directives and Methodological Materials Taken into Account in the Solution

In designing the cybersecurity information system, binding legislation of the European Union and the Slovak Republic was taken into account, as well as selected technical standards, frameworks and methodological materials that constitute a relevant professional basis for the design, implementation, operation and development of a multi-tenant SOC/SOCaaS type solution. The aim was to ensure that the proposed solution meets the requirements for cyber risk management, personal data protection, auditability, operational continuity, incident management and appropriate technical and organisational security measures.

Binding Legislation

At European Union level, the design of the solution took into account in particular the legal framework arising from Directive (EU) 2022/2555 of the European Parliament and of the Council (NIS2), which lays down measures for a high common level of cybersecurity across the Union. This Directive extends the scope of regulated entities and strengthens the requirements for cyber risk management, incident reporting, business continuity, supply chain security and management accountability. The design of the solution is therefore based on the principle that the architecture, processes and security mechanisms must be designed so that they can support the fulfilment of these obligations on the part of both the operator and the customer organisations.

At national level, account was taken in particular of Act No. 69/2018 Coll. on Cybersecurity, as amended, which constitutes the basic cybersecurity framework in the Slovak Republic. This Act governs in particular the organisation of cybersecurity, the basic obligations of regulated entities, incident reporting, the performance of cybersecurity audits and related security measures. The design of the solution therefore reflects the requirements for security management, the recording and evaluation of events, incident recording, audit support and the preparation of security documentation.

Closely related to the solution are also the implementing regulations of the National Security Authority, in particular Decree No. 227/2025 Coll., which lays down the content of security measures and the scope of general security measures for areas of cybersecurity, and Decree No. 493/2022 Coll. on cybersecurity audit. These regulations were taken into account in particular in the design of the security architecture, the scope of logging, access control, vulnerability management, operations monitoring, incident management and support for audit and control activities.

To the extent that the solution processes personal data, the requirements of Regulation (EU) 2016/679 of the European Parliament and of the Council (GDPR) and Act No. 18/2018 Coll. on the Protection of Personal Data, as amended, were also taken into account. This is relevant in particular when processing user identities, audit records, security logs, IP addresses, device identifiers or data arising during incident investigations. For this reason, the design of the solution takes into account the principles of lawfulness of processing, data minimisation, access control, data retention, recording of operations and appropriate security of processing.

Where the solution monitors or processes traffic and communication data of electronic communications, Act No. 452/2021 Coll. on Electronic Communications must also be taken into account. In such a case, the mechanisms for monitoring, collecting and retaining data must be designed so as to respect the specific rules applicable to electronic communications, in particular in the areas of confidentiality of communications, processing of traffic data and the lawful scope of monitoring. This part applies depending on the specific manner of deployment of the solution and the type of integrated data sources.

Regulation (EU) 2019/881 of the European Parliament and of the Council (Cybersecurity Act) was also taken into account, in particular as a reference framework for European cybersecurity certification of ICT products,

services and processes and for the general direction of trustworthiness of security solutions in the EU environment. Although this Regulation does not in itself determine the detailed design of a SOCaaS platform, it is relevant when assessing the trustworthiness of the technologies used and for the future development of the solution towards a higher level of demonstrable security.

Standards, Frameworks and Methodological Materials

In designing the solution, internationally recognised standards and frameworks were also taken into account as professional and methodological starting points. The basic reference framework is the ISO/IEC 27000 family of standards, in particular ISO/IEC 27001 for the information security management system, ISO/IEC 27002 for the catalogue of security measures and controls and ISO/IEC 27005 for information security risk management. These standards were used as a methodological basis in the design of the governance model, security policies, asset classification, risk management, access rights, cryptographic measures, logging and internal controls.

For the area of operational continuity and post-incident recovery, the ISO 22301 standard was taken into account, which provides a framework for a business continuity management system. Its principles are reflected in the design of the solution mainly in the requirements for service availability, backup, recovery of operations, capacity planning, outage management and the design of contingency scenarios.

In view of the processing of data with a possible impact on privacy, the ISO/IEC 27701 standard, which extends the information security management system with privacy information management, was also taken into account as a recommended methodological framework. In the design of the solution, this is reflected mainly in the areas of recording processing operations, the roles of controller and processor, data minimisation, retention rules and the protection of personal data in logs and incident records.

For the design of cyber risk management processes, their mapping to organisational measures and communication with stakeholders, the NIST Cybersecurity Framework 2.0 was also taken into account. This framework was used as a supporting methodological model in structuring security areas and processes into the domains Govern, Identify, Protect, Detect, Respond and Recover. For the area of incident response, the document NIST SP 800-61 Rev. 3 was further taken into account, which provides current recommendations for incident response in line with CSF 2.0.

From the perspective of the SOC operating model, materials of the ENISA agency were also taken into account, in particular the document “How to set up CSIRT and SOC”, which provides practical recommendations for building and improving CSIRT and SOC capabilities. This material was used in the design of roles, processes, operational links, escalation mechanisms, reporting and the organisational arrangement of security monitoring. In connection with NIS2, current ENISA technical guidance for the implementation of cybersecurity risk management measures was also taken into consideration.

For the area of threat detection, event correlation, threat intelligence and the mapping of detection scenarios, the MITRE ATT&CK knowledge bases were taken into account. This framework was used as a reference model in the design of detection rules, the classification of attacker techniques, the coverage of use-case scenarios and in evaluating the effectiveness of detection in the SIEM, SOAR, EDR/XDR and CTI environment.

Manner of Reflection in the Solution Design

The above legal, normative and methodological sources were reflected mainly in the design of the identity and access model, segmentation and separation of tenants, recording and auditability of operations, log collection and correlation, incident management, vulnerability management, personal data protection, retention rules, operational continuity and support for audit and compliance reporting. In the case of a specific deployment of the solution in a regulated sector, this chapter needs to be supplemented with the sector-specific requirements

of the relevant domain, for example according to the type of customer, industry and the nature of the data processed.

3.6 Acceptance Criteria

Within the project, only the achievement of the TRL 3 phase is considered, which is also the maximum within the funding under the call. The acceptance criteria with a description of the testing and the test report will be available in output V17.

4 Business Architecture

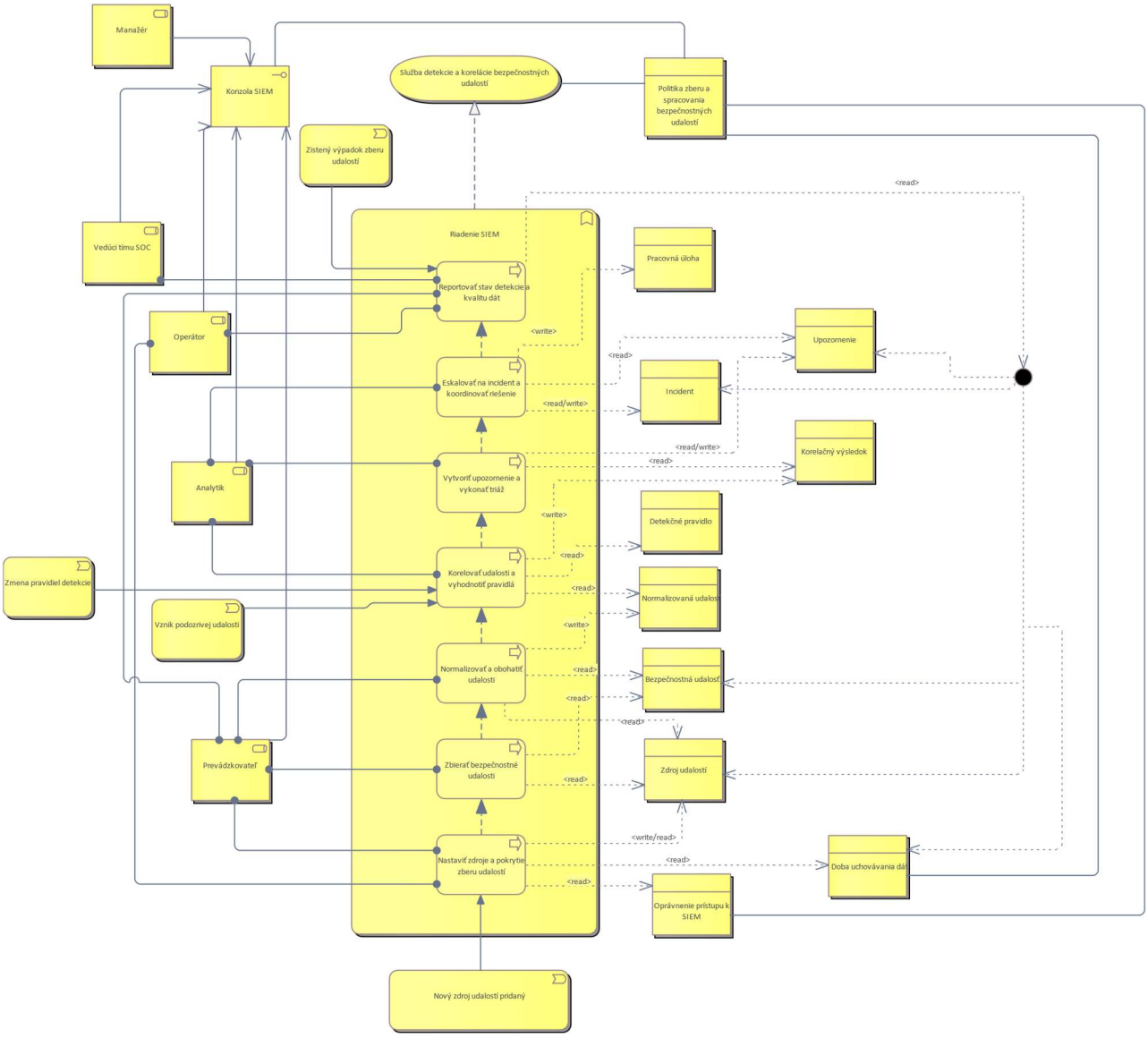
This chapter deals with the design of the business architecture as part of the organisation's overall architectural framework. The business architecture provides a logical and structured view of the capabilities, processes, services and organisational units that are necessary for achieving strategic and operational objectives. It creates a basis for aligning business needs with technological possibilities and supports transparent change management, process optimisation and performance improvement.

The aim of this chapter is to identify and describe the main business domains, capabilities and the links between them in a way that enables their use in the further design of the application, data and technology architecture. It is based on the principles of modularity, reusability and scalability, with particular emphasis placed on the interconnection between the individual architectural layers and their impact on the organisation's value-creating activities.

4.1 Business Viewpoints

The following section contains models that visualise the key business activities and relationships within the provided solution. The models were created in order to capture the way in which the individual services are provided from a business perspective, including the roles involved, process steps and data objects. Each model represents a specific view of a selected functionality or service in the context of the security centre. The aim is to create a clear basis for the further specification of the application and technology layers of the solution.

4.1.1 Detection and Correlation of Security Events (SIEM) – Business Realisation of the Service



Detection and Correlation of Security Events (SIEM) - business realisation of the service

The diagram describes the management of the collection of security events from customer systems, their processing, normalisation, correlation according to detection rules and the creation of alerts for triage and escalation. It includes roles, the processing policy, the SIEM console, business events as triggers and process access to business objects via read, write, update.

Description of the elements used:

New event source added - BusinessEvent

The event occurs when the customer adds a new system or service that is to deliver events to detection. It requires the collection coverage to be extended and the quality of the received data to be checked. It reduces the risk of blind spots in monitoring.

SIEM management - BusinessFunction

The business function covers event collection, data processing and correlation, alert creation, triage and escalation. It ensures that detection is consistent across customers and that rule changes are managed. The function creates the conditions for rapid response and traceability.

Security event detection and correlation service - BusinessService

The service ensures the collection of security events from customer systems, their processing and evaluation according to detection rules. The output consists of alerts ready for triage and possible escalation. The service is designed to be traceable, measurable and separated for each customer.

Occurrence of a suspicious event - BusinessEvent

The event represents the capture of an anomaly or behaviour pattern that may signal an attack. It triggers correlation and subsequent triage in order to confirm or rule out a security impact. It supports the rapid capture of incidents.

Event collection outage detected - BusinessEvent

The event occurs when the inflow of events from one or more sources is interrupted. It triggers verification of source availability and corrective steps to restore collection. It minimises the period during which detection works with incomplete data.

Change of detection rules - BusinessEvent

The event represents a change in the logic of rules, thresholds or correlation conditions. It triggers a check of the impact on false alerts and on threat coverage. It enables managed changes and retrospective verification of decisions.

Analyst - BusinessRole

Performs alert triage, verifies the context and decides on escalation to an incident. Uses the console to analyse events, timelines and relationships. Records the outcome of the assessment and prepares the basis for corrective steps.

Security event - BusinessObject

The object represents an individual activity record that may have security significance. It contains the time, origin, target and data needed for further processing. It is the basic input for normalisation and correlation.

Detection rule - BusinessObject

The object defines the conditions under which an alert or correlation result is created from events. It contains the evaluation logic, thresholds and the expected output for triage. It is subject to change management so that it can be verified retrospectively why a result was produced.

Data retention period - BusinessObject

The object determines how long events, correlation results and alerts are retained for the purposes of retrospective analysis and audit. It takes into account data protection requirements and operational capabilities. It supports traceability and comparison of trends over time.

Incident - BusinessObject

The object is used when an alert requires coordinated resolution, recording of steps and closure with an outcome. It makes it possible to assign responsibility, status, impact and related work tasks. The link to outputs supports audit and retrospective analysis.

Correlation result - BusinessObject

The object combines multiple events into a single interpretation with an explanation of the conditions met. It captures which rules and data were decisive for the severity. It serves as the basis for triage and for retrospective verification.

Manager - BusinessRole

Uses the outputs for an overview of risks, trends and the status of detection. Requires a clear summary of impacts and resolution priorities. Supports decision-making on changes to coverage, capacities and requirements on the customer.

Normalised event - BusinessObject

The object is an event after conversion into a uniform structure and after context enrichment. It enables comparison across sources and reduces errors caused by differing formats. It contains the assignment of the customer and related assets.

Operator - BusinessRole

Ensures operational coordination and oversight of compliance with settings and agreements. Monitors the processing status, the availability of outputs and handles routine operational requests. Supports escalation in the event of deterioration of quality or availability.

SIEM access permission - BusinessObject

The object defines who can view events, alerts and exports according to role and customer. It prevents data from being mixed between customers and reduces the risk of information leakage. It is used for every access via the console and is bound to the policy.

Security event collection and processing policy - Contract

The policy determines which sources are collected, how they are processed and what minimum requirements apply to data quality and completeness. It defines access rules, customer labelling and data retention rules. It ensures a uniform procedure across customers and defensibility during an audit.

Work task - BusinessObject

The object represents a remediation or improvement task arising from an incident or from the need to adjust detection. It makes it possible to track the deadline, responsibility, status and result of implementation. It ensures that recommendations are turned into completed steps.

Service operator - BusinessRole

Is responsible for the technical coverage of event collection, the availability of data processing and the stability of operations. Monitors the status of sources, resolves outages and sets the basic processing parameters. Ensures that data arrive in the required quality and within the required time.

Alert - BusinessObject

The object represents a finding that requires assessment by an analyst and possible further steps. It contains the severity, affected assets and references to related events. It makes it possible to record the triage status and the outcome of the assessment.

SOC team leader - BusinessRole

Manages the quality of detection and sets the direction for improving rules and processes. Approves fundamental changes that affect the number of alerts or threat coverage. Oversees the consistency of the team's work and the traceability of decisions.

Event source - BusinessObject

The object represents a system, device or service that produces security events. It carries information on the type, ownership, criticality and assignment to a customer. It makes it possible to manage collection coverage and identify missing parts of the infrastructure.

<read> - Junction

SIEM console - BusinessInterface

The interface provides a single place for working with alerts, correlation results and the status of event collection. It allows filtering by customer, time, severity and affected assets. It supports retrospective verification by displaying links to incidents and work tasks.

Escalate to an incident and coordinate resolution - BusinessProcess

The process ensures that an incident is created when an alert requires coordinated resolution and recording of steps. It assigns responsibility and status and creates work tasks for corrective measures. It supports traceability of resolution and retrospective verification.

Correlate events and evaluate rules - BusinessProcess

The process combines related events into meaningful units and evaluates detection rules. It creates an explanation of why a pattern was matched and which conditions were decisive. It reduces overload by grouping events into interpretable results.

Set up sources and event collection coverage - BusinessProcess

The process determines which systems and services are to deliver events and which types of events are needed for detection. It ensures customer labelling, asset mapping and minimum requirements for time synchronisation. The output is stable collection coverage without blind spots.

Normalise and enrich events - BusinessProcess

The process converts events into a uniform structure and adds the context needed for correlation. It assigns the customer, asset, criticality and other attributes that reduce the error rate of evaluation. It ensures the comparability of events across different sources.

Report detection status and data quality - BusinessProcess

The process evaluates the completeness of collection, the quality of processing and the stability of the number of alerts over time. It identifies source outages, increases in noise and the need to adjust rules or coverage. The outputs are used to plan improvements and to review the service.

Create an alert and perform triage - BusinessProcess

The process creates an alert from a correlation result and assesses its relevance. It adds severity, context and an initial recommendation of further steps. The aim is to quickly distinguish false alerts from real threats.

Collect security events - BusinessProcess

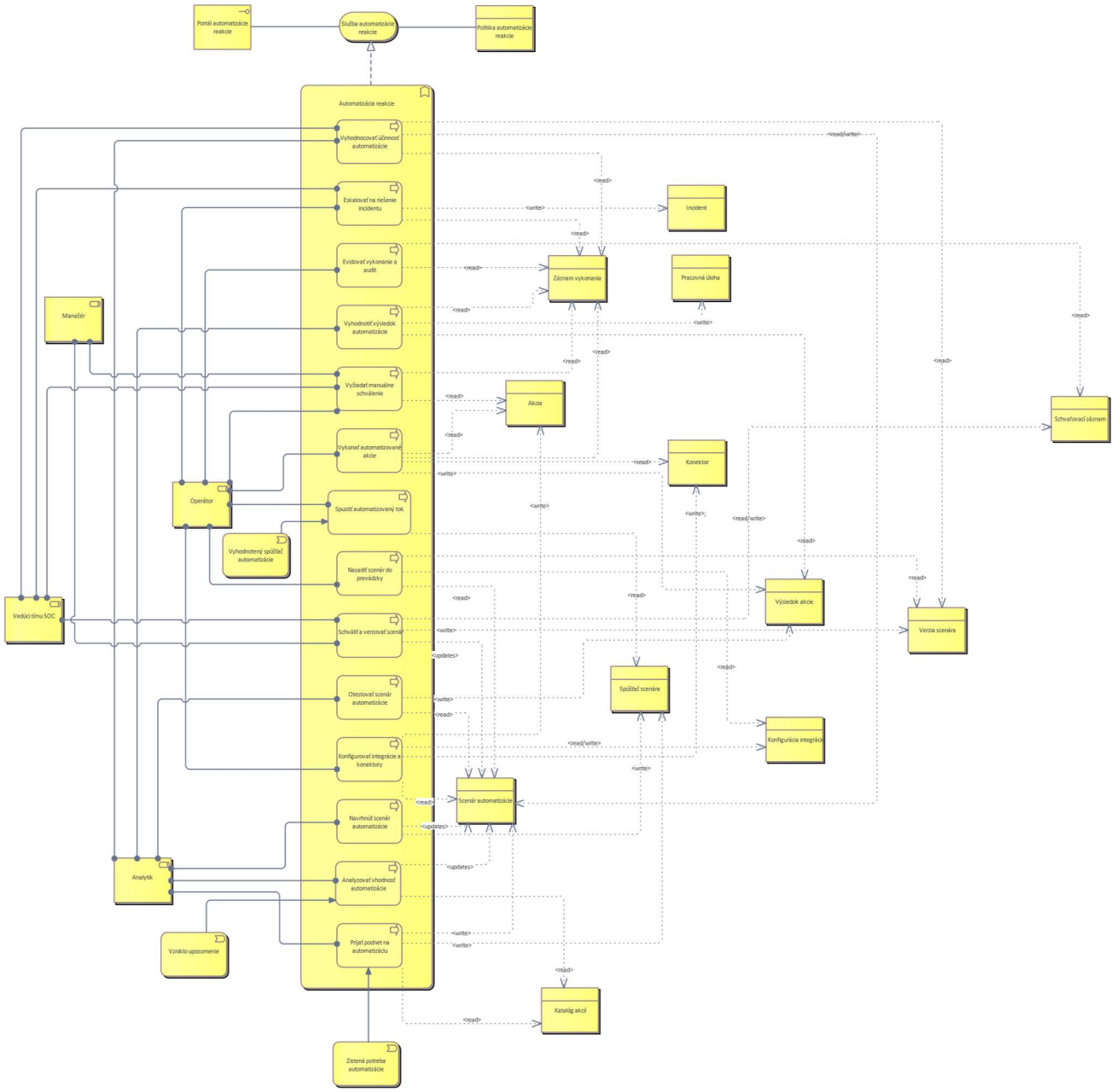
The process ensures the receipt of events from sources and their delivery to processing. It monitors availability, latency and basic transmission errors. The aim is for events to arrive completely, on time and without corruption.

Funded by the European Union
NextGenerationEU

RECOVERY AND RESILIENCE PLAN

GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC

4.1.2 Response Automation Management (SOAR) – Business Realisation of the Service



Response Automation Management (SOAR) – business realisation of the service

The diagram describes the business realisation of the service for automating the response to security events and incidents in SOCaaS. It captures the processes from receiving a request, designing a scenario and configuring integrations through to the approval, versioning and deployment of the scenario into operation. In

the operational part, it models the triggering of automated flows, the execution of actions, the evaluation of the result and the recording of execution for audit. The diagram includes the roles Analyst, Operator, SOC Team Leader and Manager together with their responsibilities in the automation life cycle. It also includes triggering events, business objects of scenarios, triggers, integrations and execution records, including process access to objects via read, write and update.

Description of the elements used:

Response automation - BusinessFunction

The function ensures the orchestration of response steps and the execution of automated actions across integrated tools. It covers the management of scenarios, triggers and integrations and the control of flow execution.

Response automation service - BusinessService

The service ensures the automated execution of response steps in the case of security events and incidents. It enables the triggering of scenarios, the coordination of manual confirmations and the recording of the results of the actions performed.

Automation trigger evaluated - BusinessEvent

The event confirms that the launch conditions for a specific scenario are met according to the defined trigger. It ensures that the automated flow is launched consistently and traceably. It triggers the process of launching the automated flow.

Alert created - BusinessEvent

The event occurs when a new alert is created in a detection or monitoring process. It signals that there is a specific case that can be assessed from the perspective of response automation. It triggers the process of analysing the suitability of automation.

Need for automation identified - BusinessEvent

The event arises on the basis of a recurring manual activity, an operational finding or a request to speed up the response. It expresses the need to create a new scenario or modify an existing one. It triggers the process of receiving an automation request.

Action - BusinessObject

The object represents a specific scenario step executed in an integrated tool. It contains parameters, execution conditions and the expected result.

Analyst - BusinessRole

The role designs and tunes automation scenarios and defines triggers according to detection requirements. It evaluates the results of flows and decides on escalation in the event of failure or an ambiguous result.

Incident - BusinessObject

The object represents a security incident handled in a coordinated manner in the SOC. It serves as the parent record for escalated cases and related tasks.

Action catalogue - BusinessObject

The object represents a list of the available types of actions that can be executed automatically. It helps to ensure uniform use and restrictions according to the policy.

Connector - BusinessObject

The object represents a connection to an external tool or service needed to execute actions. It records the connector type, status and the access credentials used.

Integration configuration - BusinessObject

The object contains the settings, mappings and parameters for a specific integration. It enables change management and compatibility checks during updates.

Manager - BusinessRole

The role sets the priorities and limits of automation where there is a significant impact on the customer's business and operations. It approves exceptions and decides on escalation in the event of critical events.

Operator - BusinessRole

The role oversees the operation of automation and ensures the correct functioning of integrations and flows. It intervenes when automation fails and confirms steps that require manual approval.

Response automation policy - Contract

The policy defines the rules for when an automated intervention may be performed and when manual approval is required. It determines the permitted types of actions, the scope of permissions and the requirements for audit recording.

Work task - BusinessObject

The object represents a task assigned to a person when automation requires a manual activity. It makes it possible to track status, responsibility and deadlines.

Automation scenario - BusinessObject

The object represents the definition of an automated flow, including steps, conditions and links to integrations. It also contains information on approval and deployment status.

Approval record - BusinessObject

The object records manual approvals and rejections of steps that require them. It contains who approved, when and under what conditions.

Scenario trigger - BusinessObject

The object defines the conditions under which a scenario is launched automatically or manually. It contains links to the event source and threshold values.

SOC team leader - BusinessRole

The role approves higher-impact scenarios and sets the rules for using automation in practice. It is responsible for ensuring that automated interventions comply with the policy and operational constraints.

Scenario version - BusinessObject

The object captures a specific approved version of a scenario and its parameters. It enables retrospective traceability of changes and controlled deployment into operation.

Action result - BusinessObject

The object contains the result of executing a specific action, including the return status and error messages. It enables decisions on continuing the flow or escalating.

Execution record - BusinessObject

The object captures the course of a scenario run, including the time, input data and status. It serves for audit and for evaluating the success rate of automation.

Response automation portal - BusinessInterface

The interface provides access to the management of scenarios, triggers and integrations. It makes it possible to launch and monitor automated flows and displays records of execution and results.

Analyse the suitability of automation - BusinessProcess

The process assesses whether the given case is suitable for automation and what the risk impact is. The result is a decision to proceed with the design of a scenario or to reject the request.

Escalate to incident handling - BusinessProcess

The process hands the case over to security incident management if automation is not sufficient or the situation is serious. It ensures the creation of an incident and the assignment of responsibilities.

Record execution and audit - BusinessProcess

The process supplements the execution record with the data needed for audit and traceability. It ensures the link to the alert or incident and closes the flow.

Configure integrations and connectors - BusinessProcess

The process prepares connections to the tools and services needed to execute actions. It includes setting up access and parameters and checking the availability of integrations.

Deploy the scenario into operation - BusinessProcess

The process activates the approved version of the scenario and sets up its triggering in the production environment. It ensures that the scenario uses the correct integrations and permissions.

Design an automation scenario - BusinessProcess

The process defines the scenario steps, triggers, conditions and the necessary integrations. It takes into account the automation policy and the approval requirements.

Test the automation scenario - BusinessProcess

The process verifies the correctness of the scenario steps on test data or a controlled case. The result is confirmation of readiness or modifications to the scenario.

Receive an automation request - BusinessProcess

The process receives a request to create or modify an automated flow from SOC practice or from identified problems. It ensures basic recording of the purpose and the expected benefit.

Approve and version the scenario - BusinessProcess

The process ensures the formal approval of the scenario according to its impact and the subsequent creation of a version. It retains the approval trail and the conditions for deployment into operation.

Launch the automated flow - BusinessProcess

The process launches the execution of the scenario on the basis of a trigger or a manual instruction. It initialises the execution record and assigns the case context.

Evaluate the effectiveness of automation - BusinessProcess

The process monitors success metrics, time savings and the error rate of automated flows. The output consists of recommendations for modifications to scenarios and policies.

Evaluate the automation result - BusinessProcess

The process evaluates whether the automated flow achieved the expected result and whether further activity is needed. If further steps are needed, it creates a work task or prepares an escalation.

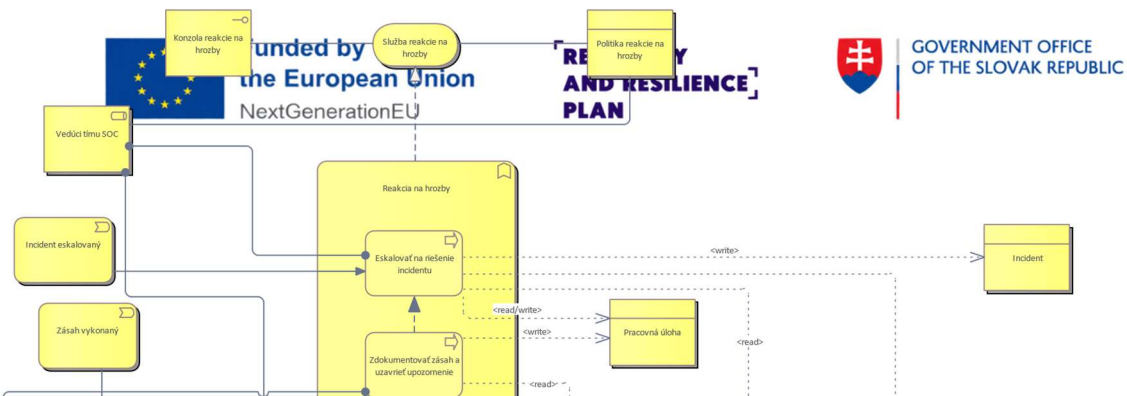
Execute automated actions - BusinessProcess

The process executes the individual actions of the scenario via integrated tools and services. At each step, it records the result and any errors.

Request manual approval - BusinessProcess

The process requires human confirmation for steps that have a higher impact or where the policy requires it. After approval, it continues the flow or terminates it.

4.1.3 Threat Response Management (EDR/XDR) – Business Realisation of the Service


Konzola reakcie na hrozby
Služba reakcie na hrozby
Politika reakcie na hrozby
Vedúci tímu SOC
Reakcia na hrozby
Incident eskalovaný
Eskalovať na riešenie incidentu
Incident
<write>
Zásah vykonaný
<read/write>
<write>
Pracovná úloha
<read>
Zdokumentovať zásah a uzavrieť upozornenie
<read>

Threat Response Management (EDR/XDR) - business realisation of the service

The diagram describes the provision and management of threat detection and response on endpoints and in the customer's extended telemetry sources. It captures the processes of deploying and managing agents and data sources, collecting and enriching telemetry, detecting suspicious behaviour, evaluating and triaging alerts, isolating or restricting compromised devices and handing the incident over for escalation. It includes the roles SOC Analyst, Operator and Manager, response rules and the recording of interventions and results.

Description of the elements used:

Incident escalated - BusinessEvent

The event occurs when a case is handed over to the security incident management process. It triggers further steps of investigation and coordination of resolution.

Endpoint registration request - BusinessEvent

The event occurs during customer onboarding or when a new endpoint is added to the monitoring scope. It expresses the need to enter the device into the register and prepare it for telemetry collection. It triggers the process of registering telemetry sources.

Threat response - BusinessFunction

The function ensures the coordination of telemetry collection, detection, evaluation and response to threats. It covers the processes of working with alerts and recording interventions.

Intervention approved - BusinessEvent

The event occurs after a response measure has been approved by the responsible role. It triggers the execution of the response and the subsequent verification of the result.

Threat response service - BusinessService

The service ensures detection and the execution of response measures in the event of suspicious behaviour on the customer's endpoints. It supports alert triage and the handover of confirmed cases for incident resolution.

Alert created - BusinessEvent

The event signals that the system has created a new threat alert with an assigned priority. It triggers triage and evaluation of the alert.

Suspicious activity captured - BusinessEvent

The event occurs when suspicious behaviour is identified in endpoint telemetry. It triggers the detection process and the subsequent creation of an alert.

Intervention performed - BusinessEvent

The event confirms that the response measure has been performed and is recorded. It triggers verification of the effectiveness of the intervention and closure of the alert or escalation.

Analyst - BusinessRole

The role performs alert triage and assesses the severity of findings. It decides on escalation and proposes appropriate response measures.

Detection rule - BusinessObject

The object defines the conditions according to which suspicious behaviour is evaluated. It makes it possible to record the version, priority and scope of application of the rule.

Detection result - BusinessObject

The object represents the result of evaluating telemetry according to a rule. It contains the indicators found, the severity score and a reference to the related data.

Incident - BusinessObject

The object represents a confirmed security incident that requires coordinated resolution. It holds links to alerts, interventions and work tasks.

Endpoint - BusinessObject

The object represents a customer device on which telemetry is collected and interventions are performed. It contains identifiers, the owner and the operational attributes needed for response.

Enriched telemetry - BusinessObject

The object represents telemetry supplemented with context and classification. It increases the usability of data for detection and reduces ambiguity during triage.

Operator - BusinessRole

The role performs approved response measures and oversees their implementation. It ensures communication with the customer in the case of interventions that affect operations.

Response permissions - BusinessObject

The object defines the access rights and the scope of permitted interventions for the individual roles. It ensures that responses are carried out in accordance with the policy.

Threat response policy - Contract

The policy defines the permitted response measures, intervention levels and role responsibilities. It determines the conditions for device isolation and the rules for restoring it to operation.

Work task - BusinessObject

The object represents an assigned task in the SOC team related to the response to a threat. It makes it possible to track status, responsibility and deadlines.

Response measure - BusinessObject

The object describes the proposed intervention and its parameters. It makes it possible to record the approval, the reasons and the expected impact on operations.

Device isolation status - BusinessObject

The object records the current status of the device restriction and the conditions for returning to operation. It helps to manage risk and coordinate recovery.

Endpoint telemetry - BusinessObject

The object contains technical records and events from the device that are used for threat detection. It includes timestamps, the source and basic event categories.

Threat alert - BusinessObject

The object represents a record of suspicious behaviour ready for triage. It carries the priority, processing status and the link to the detection result.

SOC team leader - BusinessRole

The role approves higher-impact interventions and manages the resolution procedure for recurring or large-scale threats. It is responsible for compliance with the threat response policy.

Intervention record - BusinessObject

The object captures the course of the steps performed and their results. It serves for audit, retrospective traceability and evaluation of the quality of responses.

Collection component - BusinessObject

The object represents a deployed component that collects and sends telemetry from the endpoint. It makes it possible to record the version, status and configuration of collection.

Threat response console - BusinessInterface

The interface provides a single place for an overview of telemetry, alerts and device statuses. It makes it possible to launch response steps and record the results of interventions.

Detect suspicious behaviour - BusinessProcess

The process evaluates enriched telemetry according to detection rules and identifies suspicious activities. When something is detected, it creates the basis for generating an alert.

Escalate to incident handling - BusinessProcess

The process hands a confirmed threat over to security incident management. It ensures the creation of an incident and the link to related records.

Deploy and maintain collection components - BusinessProcess

The process handles the deployment and ongoing maintenance of collection components on endpoints. It includes updates, status checks and the resolution of collection errors.

Propose a response measure - BusinessProcess

The process selects the appropriate type of response according to the policy and the threat context. It takes into account the impact on operations and the need for approval.

Enrich and classify telemetry - BusinessProcess

The process supplements telemetry with context, classification and related information. It prepares the data so that they are usable for detection and evaluation.

Verify the result of the intervention - BusinessProcess

The process verifies whether the intervention has eliminated or contained the threat and whether the device behaves as expected. In the event of failure, it triggers additional steps or escalation.

Approve the response measure - BusinessProcess

The process ensures the approval of the response according to the level of impact and the responsibilities of the roles. The result is permission for execution or a request to modify the proposal.

Evaluate and triage the alert - BusinessProcess

The process assesses the relevance of the alert, rules out false alarms and determines the next course of action. The result is confirmation of the threat, closure of the alert or escalation.

Execute the response measure - BusinessProcess

The process performs the selected intervention on the endpoint or in related systems. It includes isolating the device, restricting communication or stopping a malicious process.

Create a threat alert - BusinessProcess

The process creates an alert on the basis of the detection result and assigns it a priority. It ensures that the alert contains the data needed for triage.

Register telemetry sources - BusinessProcess

The process ensures the recording of endpoints and connected sources from which telemetry is collected. It sets the basic collection parameters and verifies data availability.

Collect telemetry from endpoints - BusinessProcess

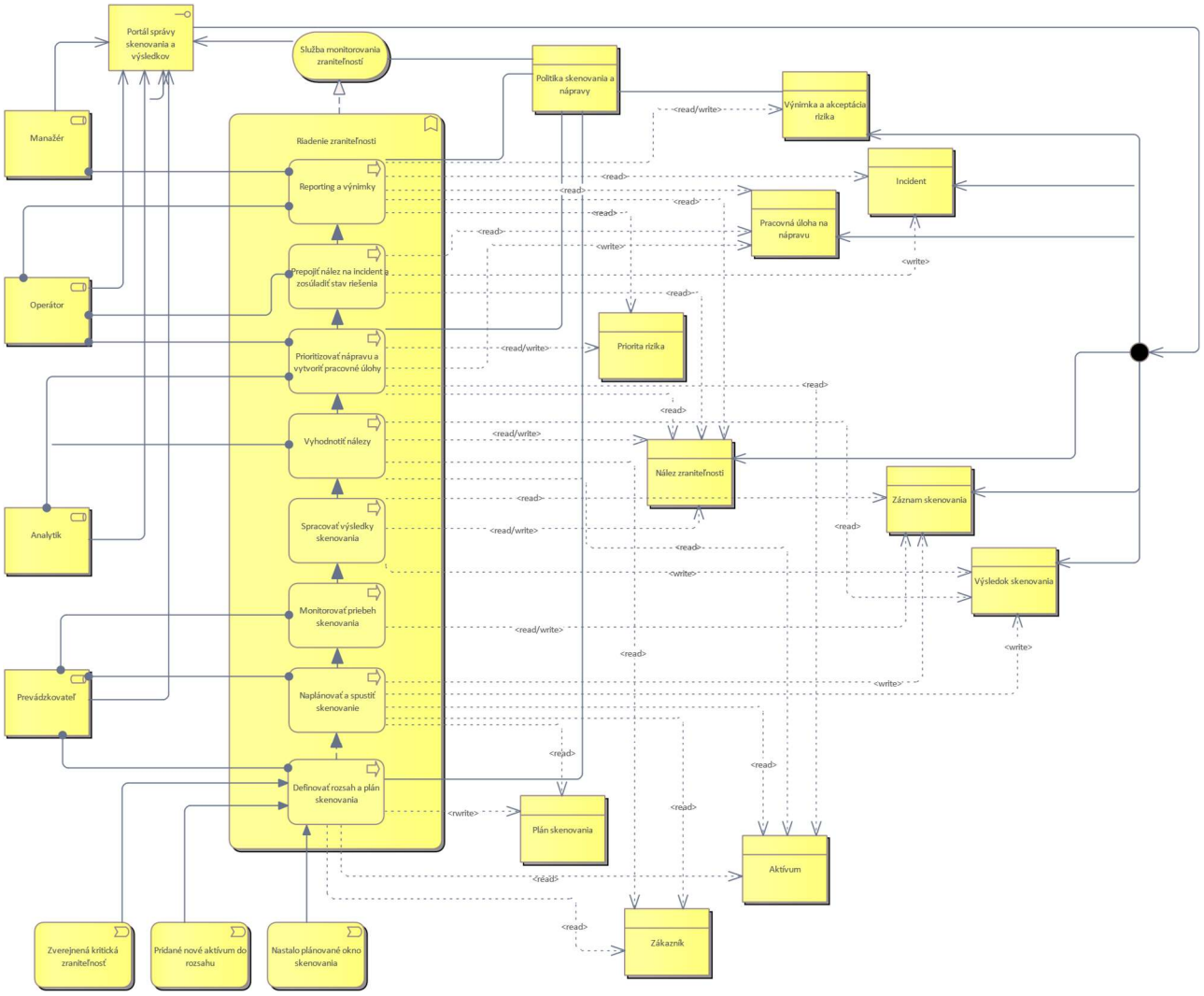
The process gathers events and technical records from endpoints according to the configured rules. It ensures their delivery to the console and a basic quality check.

Document the intervention and close the alert - BusinessProcess

The process creates a record of the steps performed, the results and recommendations. It closes the alert or links it to an incident if the investigation continues.

4.1.4 Vulnerability Monitoring – Business Realisation of the Service

Ensures the planning and execution of scans, validation of findings, risk assessment, remediation prioritisation, ticketing and reporting across tenants.



Vulnerability Monitoring – business realisation of the service

The diagram shows the vulnerability monitoring business service in the SOCaaS environment and its realisation through processes, roles and the information processed. The flow covers the definition of the scope and scanning plan, the launch and monitoring of progress, the processing and evaluation of results, remediation prioritisation, linking findings to incidents and closing the cycle through reporting and exceptions. The model highlights multiple customers and the separation of data by customer, traceability via the scan record and the link to work tasks and incidents. It also includes the portal as the business interface of the service and the scanning and remediation policy as the contractual framework, with events triggering key processes according to the schedule, changes in scope or the emergence of critical findings.

Description of the elements used:

Scheduled scanning window reached - BusinessEvent

Triggers the scheduled launch of scanning.

New asset added to the scope - BusinessEvent

Triggers an adjustment of the scope and often also an ad hoc launch of scanning.

Vulnerability management - BusinessFunction

Covers the SOCaaS capability to manage vulnerabilities from defining the scope, through monitoring scan progress, evaluating and prioritising findings, to managing remediation, exceptions and closing the cycle, including aligning the status of related incidents.

Vulnerability monitoring service - BusinessService

Provides the planning and execution of scans, processing of results, evaluation of findings, remediation prioritisation, management of remediation tasks, reporting and linking of findings to incidents across customers.

Critical vulnerability published - BusinessEvent

Triggers ad hoc prioritisation and often also ad hoc scanning.

Asset - BusinessObject

The scanning target to which the results and findings relate; it makes it possible to determine which services and processes would be affected if the vulnerability were exploited and to set the remediation priority accordingly.

Analyst - BusinessRole

Validates findings, adds context based on assets and relationships, evaluates the impact and decides on the course of action.

Incident - BusinessObject

A formal case record used when a vulnerability requires incident management, coordination and communication according to SOC rules. It includes the severity, status, timeline, responsibilities and related evidence and impacts. It is linked to findings and work tasks so that the course of resolution is uniform and auditable.

Manager - BusinessRole

Approves exceptions and risk acceptance, resolves priority conflicts, monitors trends and commitments to the customer.

Vulnerability finding - BusinessObject

A record of a specific vulnerability assigned to an asset, including the identifier, severity and processing status. It carries the context needed for a decision, for example exposure, the affected version, evidence of the finding and a reference to the scan result. It makes it possible to track the life cycle of a finding from detection to closure or exception.

Operator - BusinessRole

Ensures operational management of remediation, creates and tracks work tasks, oversees deadlines and processing status.

Scanning plan - BusinessObject

Scope, periodicity, scanning windows, rules for launching and processing.

Scanning and remediation policy - Contract

Defines the rules that the service must comply with, for example the scanning periodicity, permitted time windows and asset coverage requirements. It determines remediation deadlines according to severity and risk priority, escalation thresholds and the manner of monitoring the fulfilment of commitments. It also lays down the

procedure for exceptions and risk acceptance, including approval and validity, and the minimum requirements for reporting and auditability.

Remediation work task - BusinessObject

A work item that assigns responsibility for the fix, deadlines and the resolution status of a specific finding or group of findings. It maintains a history of steps, comments and the link to the asset, finding and risk priority. It enables operational monitoring of fulfilment and escalation in the event of delay.

Service operator - BusinessRole

Is responsible for setting the scanning scope, rules, scanning windows, the quality of input data and the operational readiness of the process.

Risk priority - BusinessObject

Expresses the order in which a finding is resolved based on a combination of the severity of the finding, the criticality of the asset and the expected impact on the customer's services. It turns a technical finding into a practical priority for remediation and escalations. It serves for setting deadlines, SLAs and capacity planning.

Exception and risk acceptance - BusinessObject

A documented decision to postpone or not carry out remediation, including the reason, scope, validity and approver. It defines compensating measures or conditions under which the risk is temporarily acceptable. It enables audit and control so that exceptions do not remain open without oversight.

Scan result - BusinessObject

A specific execution of a scan at a certain time with a unique identifier, status and scope. It contains information on who launched the scan, according to which plan it ran and whether it ended successfully or with an error. It serves for traceability, audit and explaining why some assets did not have up-to-date results. A summary of the scan outputs, including metadata such as the scope, time, profile used and asset coverage. It contains a list of findings, their parameters and the link to specific assets and the customer. It is the input for the validation of findings, prioritisation and further decision-making.

Customer - BusinessObject

Determines the ownership of data, rules and outputs and ensures separation between customers.

Scan record - BusinessObject

Contains the status of the scan progress and operational information about the run, for example the start, end, current status and any errors. It makes it possible to track which parts of the scope were actually processed and which failed or were skipped. It serves for traceability and troubleshooting of scanning and supports the auditability of the process.

Junction - Junction

Scanning and results management portal - BusinessInterface

Serves as a single place where the SOC and the customer can see the scanning status, asset coverage and current results. It allows findings to be filtered and searched, the progress of remediation tasks to be tracked and findings to be linked to incidents, so that the connection between the scan and the resolution is visible. It ensures a consistent view across customers with data separated by customer and supports exports and reports for audit.

Define the scanning scope and plan - BusinessProcess

Defines what is scanned, how often it is scanned, in which windows, what the evaluation rules are and the responsibilities by customer.

Monitor scan progress - BusinessProcess

Provides an operational overview of the status of scans, errors, and ongoing and completed runs.

Schedule and launch scanning - BusinessProcess

Carries out scheduled launches and enables an ad hoc launch in the event of a change in scope or an increased threat.

Link a finding to an incident and align the resolution status - BusinessProcess

Ensures the visibility of findings that are handled as incidents and maintains a uniform status of related incidents.

Prioritise remediation and create work tasks - BusinessProcess

Determines the resolution priority and creates work tasks for the fix, sets deadlines and responsibilities.

Reporting and exceptions - BusinessProcess

Creates reports, records exceptions and risk acceptance and closes the vulnerability management cycle.

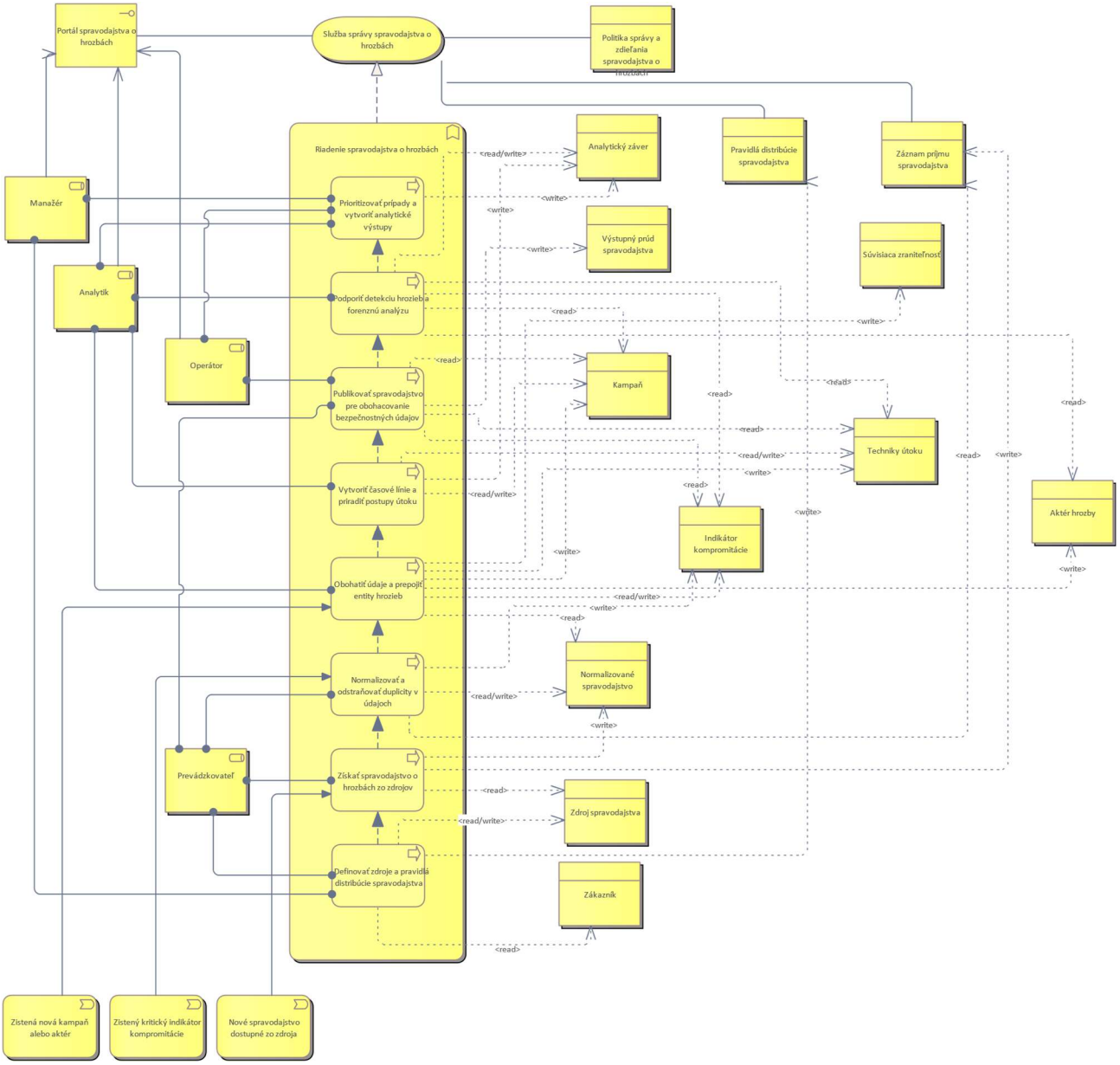
Process scan results - BusinessProcess

Collects and normalises results and ensures traceability and the assignment of results to the correct assets and customer.

Evaluate findings - BusinessProcess

Verifies the relevance of findings, filters out false findings and adds criticality and impact based on the assets and their relationships.

4.1.5 Threat Intelligence Management (CTI) – Business Realisation of the Service



Threat Intelligence Management (CTI) - business realisation of the service

The diagram describes how, in the SOCaaS environment, threat intelligence is received from sources, recorded via an intake record and then converted into normalised intelligence. Indicators of compromise are derived from it and context is added in the form of a campaign, threat actor, attack techniques and related vulnerabilities, so that the outputs are interpretable and defensible. Policy-driven distribution rules determine which intelligence output stream is provided to the customer, and the analytical conclusion summarises the main findings and recommendations for practice.

Description of the elements used:

New intelligence available from a source - BusinessEvent

The event signals the availability of new data and triggers their intake and processing. It may be regular according to a schedule or ad hoc in the event of a rapid change in the threat situation. It is important for keeping outputs up to date and for rapid enrichment of operational data.

Threat intelligence management - BusinessFunction

The function covers the entire life cycle of threat intelligence, from the selection of sources to the publication of usable outputs. It ensures quality, timeliness, removal of duplicates and the creation of relationships between threat entities for better interpretation. The function supports the enrichment of detections, threat hunting, forensic analysis and decision-making on severity and priority.

Threat intelligence management service - BusinessService

The service ensures the processing of external and internal threat information so that it is usable for security operations. The outputs are prepared for the enrichment of security data, investigation support and meaningful prioritisation of cases. The service is designed for a multi-customer environment and maintains the separation of data and rules by customer.

New campaign or actor identified - BusinessEvent

The event occurs when new connections are identified that require the linking of entities and the addition of context. It triggers steps to create a timeline and update the relationships between indicators, campaigns and attack procedures. It helps to capture a change in the nature of the threat and improve the interpretation of findings.

Critical indicator of compromise identified - BusinessEvent

The event signals that a high-importance indicator has appeared which requires rapid use in operations. It typically leads to the immediate publication of outputs and to support for threat hunting and case prioritisation. It ensures that critical information is not lost in the volume of data and is processed with greater urgency.

Threat actor - BusinessObject

Describes an attacker or group and provides context on motivation, typical targets and preferred procedures. It makes it possible to combine individual findings into a coherent picture and to distinguish random activity from a coordinated campaign. It supports decision-making on the priority and form of response, especially when recurring behaviour patterns appear.

Analytical conclusion - BusinessObject

Represents a summary and interpretation of intelligence for a specific purpose, for example recommended steps or a risk assessment. It combines technical data with context so that they can be used directly in decision-making and communication with the customer. It ensures that outputs are consistent, defensible and explainable, even during a retrospective review.

Analyst - BusinessRole

Assesses the relevance of intelligence for the customer's specific environment and adds the context needed for decision-making. Creates links between indicators, campaigns, actors and attack procedures and prepares analytical conclusions. Uses intelligence in threat hunting and investigations to speed up the understanding of the course and impact of events

Indicator of compromise - BusinessObject

Represents a specific threat identifier that can be searched for in security data and used to enrich detection. It contains the indicator type, confidence level, time validity and connections with other entities so that it can be interpreted correctly. For critical indicators, it enables a rapid operational response and a reduction in the time to threat detection. **Campaign** - BusinessObject

Groups related events and activities into a single unit so that the development of the threat can be tracked over time. It links indicators, actors and the attack procedures used, thereby creating context for investigation and threat hunting. It helps to identify recurring scenarios and to assess whether the threat concerns multiple customers or only a specific environment.

Manager - BusinessRole

Approves the sharing rules and processing priorities of intelligence according to customer needs and risk. Monitors the benefit of intelligence and reporting requirements and oversees that the processes support measurable improvement. Makes decisions in sensitive situations, for example in the case of significant campaigns that require clear communication of the impact.

Normalised intelligence - BusinessObject

Represents consolidated data after cleansing and unification of structure, ready for searching and linking. It contains the minimum required fields and uniform data types so that they can be reliably processed by other processes. It reduces noise and creates a stable basis for enrichment and the publication of outputs.

Operator - BusinessRole

Ensures the operational use of intelligence outputs in day-to-day operations and oversees the timely delivery of outputs. Monitors the publication status, resolves practical distribution issues and coordinates responses when critical indicators occur. Supports linking intelligence with case resolution so that outputs are available where they are worked with.

Threat intelligence management and sharing policy - Contract

The policy lays down the rules for source selection, minimum quality requirements, processing periodicity and data retention. It defines restrictions on sharing between customers, procedures for sensitive information and requirements for traceability and audit. It ensures that intelligence is managed consistently, predictably and in line with agreed expectations.

Intelligence distribution rules - BusinessObject

Defines which outputs are provided to which customers and what restrictions apply to sharing. It contains isolation rules, permitted forms of publication and the conditions for making sensitive information available. It is the basis for consistent and auditable distribution of intelligence in a multi-customer environment.

Service operator - BusinessRole

Is responsible for the reliable intake and processing of intelligence sources, setting schedules and checking availability. Resolves operational processing failures, monitors capacity and ensures that processing runs according to the defined rules. Also oversees the technical quality of data so that outputs are consistent and usable across customers.

Related vulnerability - BusinessObject

Represents a vulnerability that a threat may exploit or to which a specific indicator, actor or campaign relates. It makes it possible to assess the urgency of remediation according to the threat context, not only according to general severity. It supports coordination between threat intelligence and vulnerability management processes so that the riskiest points are fixed. **Attack techniques** - BusinessObject

Represent the described ways in which an attacker proceeds when intruding, moving within the environment and achieving the objective. They serve to map findings to known procedures so that they can be compared with previous cases and the connections can be better explained. They also make it possible to plan preventive measures and conduct more targeted searches for suspicious manifestations in the data.

Intelligence output stream - BusinessObject

Contains published outputs intended for the enrichment of security data and investigation support in operations. It is targeted by customer and distribution rules and contains only data that are permitted and relevant. It supports the automated use of intelligence and reduces the time needed for manual enrichment.

Customer - BusinessObject

Represents the organisation for which threat intelligence is processed and distributed. It carries requirements on the scope, sensitivity and manner of providing outputs so that they are usable for its environment. It also serves for the separation of data and for assigning the rules that determine what may be shared and with whom.

Intelligence intake record - BusinessObject

Captures the fact that intelligence was received, including the time, source, scope and processing result. It enables traceability during retrospective verification of why a certain item did or did not appear in the outputs and when it was last updated. It is also important for audit and troubleshooting in the event of source outages or inconsistent data.

Intelligence source - BusinessObject

Records the origin of intelligence, processing parameters, periodicity and the expected data quality. It contains the information needed to check trustworthiness and to control what is received and how it is processed. It enables oversight of source availability and of changes that may affect the quality of outputs.

Threat intelligence portal - BusinessInterface

The interface provides an overview of threat entities, their relationships and the outputs available to individual customers. It enables searching, filtering and quick display of the context of indicators, campaigns and attack procedures. It supports traceability by displaying the origin of data, the processing status and the change history of outputs.

Define intelligence sources and distribution rules - BusinessProcess

The process determines which intelligence sources are used, what their quality requirements are and how often they are to be processed. It sets distribution rules by customer and defines sharing restrictions so that data separation is maintained. The output is a clear framework according to which intelligence is received, processed and passed on.

Normalise and deduplicate data - BusinessProcess

The process converts received data into a uniform structure and removes duplicates and low-quality records. It ensures consistent naming, uniform data types and a minimum level of mandatory fields for further use. The output reduces noise, improves searching and prepares the data for linking and correlation.

Enrich data and link threat entities - BusinessProcess

The process adds context to indicators and creates links between indicators, actors, campaigns, attack procedures and related vulnerabilities. The aim is for intelligence to appear not as isolated items but as a connected picture of the threat with an explanation of the connections. This step significantly increases usability in investigations, threat hunting and decision-making on priority.

Support threat detection and forensic analysis - BusinessProcess

The process uses intelligence to create hypotheses, search for connections and explain findings within an investigation. It provides analysts with linked entities, timelines and context so that the scope and impact can be determined more quickly. The outputs of the process are also used as the basis for decisions on further steps and for communication with the customer.

Prioritise cases and create analytical outputs - BusinessProcess

The process uses threat context to determine the priority of cases and the recommended steps for resolution. It creates analytical conclusions and summaries that link technical findings with the impact on the customer's services and risk. The output consists of clear recommendations and the basis for reporting, decision-making and response management.

Publish intelligence for the enrichment of security data - BusinessProcessThe process prepares intelligence output streams and ensures their availability according to the distribution rules for individual customers. Publication is focused on rapid use in detection and investigation so that known threats can be identified in the data as early as possible. The process also checks that the published outputs meet the quality requirements and are auditable.

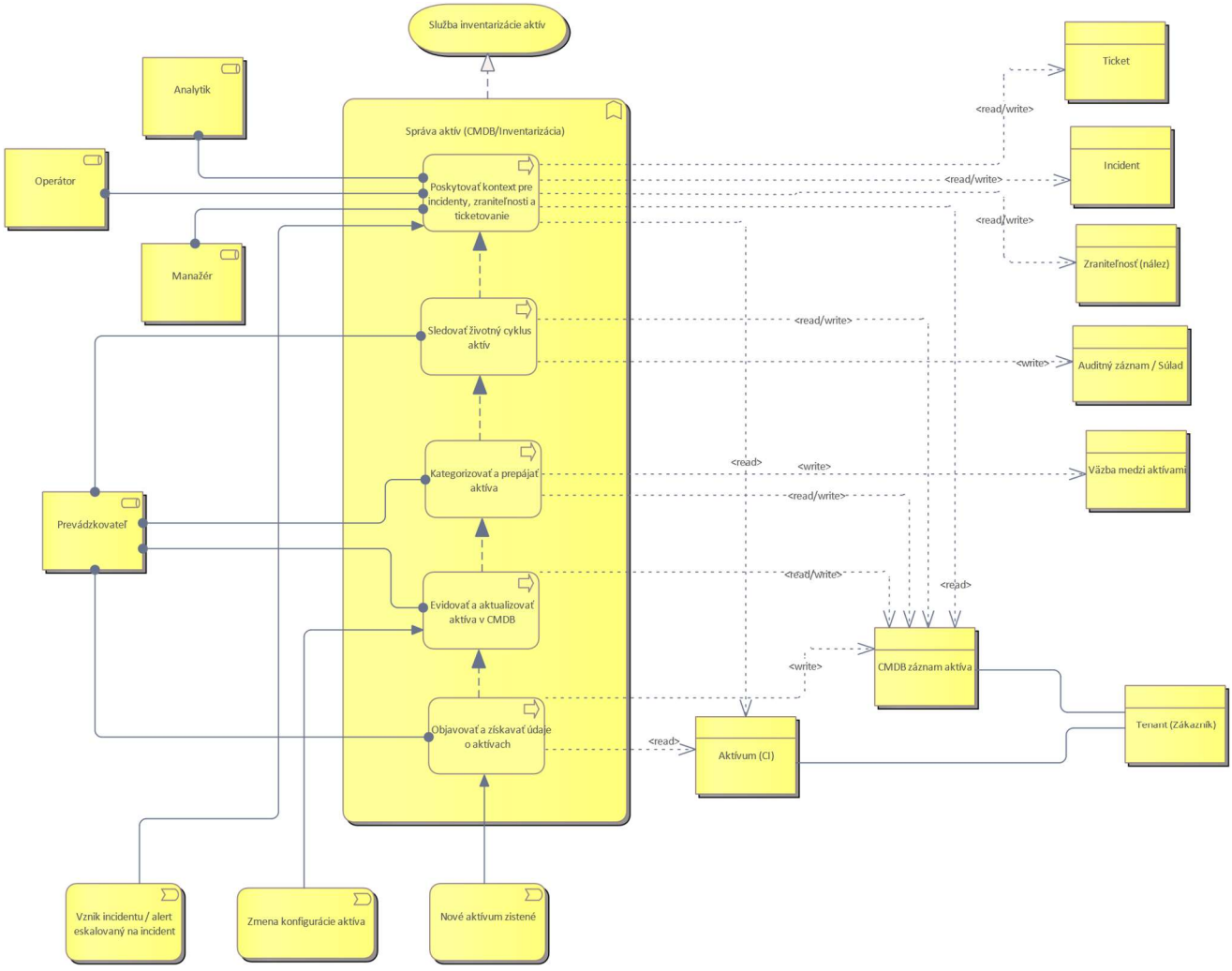
Create timelines and assign attack procedures - BusinessProcess

The process assembles information into a timeline and shows the development of campaigns and activities over time. It assigns typical attack procedures so that current findings can be compared with already known behaviour patterns. The output helps to understand the course of events and supports the formulation of hypotheses in threat hunting.

Obtain threat intelligence from sources - BusinessProcess

The process ensures the intake of intelligence from external and internal sources, including metadata on origin and time of receipt. It creates intake records so that the completeness and course of processing are traceable and outages or errors can be resolved. The aim is a consistent input for further processing regardless of the differing source formats.

4.1.6 Asset Inventory – Business View



Asset inventory – business view

Business view of the asset inventory service in SOCaaS and its realisation through processes. The diagram shows how asset data are obtained, recorded in the CMDB, categorised and linked, how the life cycle and auditability are tracked, and how this information is subsequently used to contextualise incidents, vulnerabilities and ticketing. It also captures the roles responsible for maintaining the quality of the records and using the context across tenants (customers).

Description of the elements used:

New asset discovered - BusinessEvent

Asset inventory service - BusinessService

Centralised recording and management of assets across tenants. It supports categorisation, relationships, tracking of security status and life cycle. It provides context for vulnerabilities, incidents, ticketing, audit and regulations.

Asset management (CMDB/Inventory) - BusinessFunction

Incident occurrence / alert escalated to an incident - BusinessEvent

Asset configuration change - BusinessEvent

Asset (CI) - BusinessObject

Device/system/application/virtual or cloud resource.

Analyst - BusinessRole

Uses the asset records to contextualise incidents and to analyse impact and priorities.

Audit record / Compliance - BusinessObject

Evidence of the status of and changes to assets for audit and regulations.

CMDB asset record - BusinessObject

Asset attributes, tenant, criticality, status, change history.

Incident - BusinessObject

Security incident contextualised by the asset records.

Manager - BusinessRole

Uses the asset records for audit, recovery planning, compliance and reporting.

Operator - BusinessRole

Works with tickets and SLAs; uses asset information for operational resolution and escalations.

Service operator - BusinessRole

Is responsible for the quality and currency of the asset records, categorisation, life cycle management and data collection.

Tenant (Customer) - BusinessObject

Ticket - BusinessObject

A work item for resolution; linked to an asset/incident/SLA.

Relationship between assets - BusinessObject

Dependencies and connections (topology, service-to-component, network relationships).

Vulnerability (finding) - BusinessObject

A finding linked to an asset; input for priority and impact.

Record and update assets in the CMDB - BusinessProcess

Maintenance of asset records (identity, type, owner/tenant, criticality, status).

Categorise and link assets - BusinessProcess

Tagging and creation of relationships between assets (dependencies, topology, relations).

Discover and obtain asset data - BusinessProcess

Obtaining data on devices, systems, applications and cloud resources; the aim is completeness and currency.

Provide context for incidents, vulnerabilities and ticketing - BusinessProcess

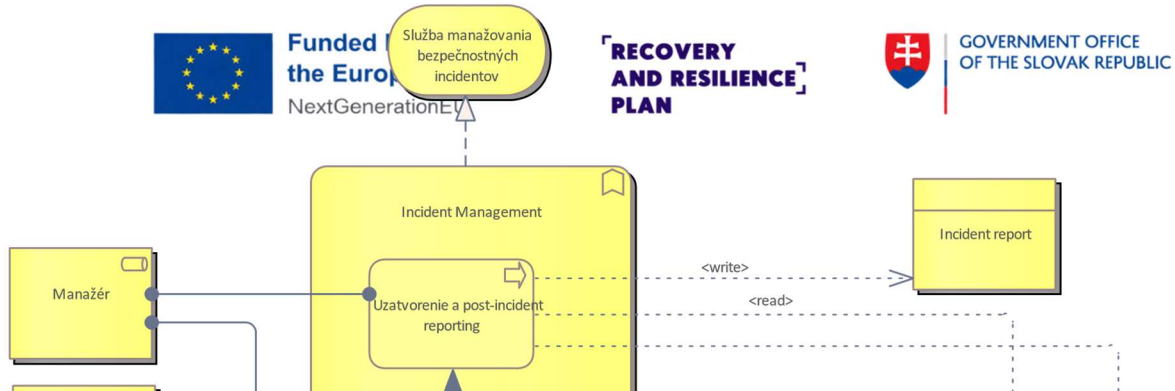
Contextualisation of incidents, analysis of impact and priorities, support for effective ticketing.

Track the asset life cycle - BusinessProcess

Status transitions, change history, recovery planning, auditability.

4.1.7 Cybersecurity Incident Management – Business Realisation of the Service

Business view of the security incident management service in SOCaaS and its end-to-end flow. The diagram shows the responsible roles and the main processes from receipt and triage, through investigation and escalation, to coordination of resolution and closure with post-incident reporting. It also includes work with the key business objects (incident/case, alert, ticket, evidence, communication record, report) and their link to the tenant, so that the incident management process is consistent and auditable.


Služba manažovania bezpečnostných incidentov
Incident Management
Manažér
Uzatvorenie a post-incident reporting
Incident report
<write>
<read>

Cybersecurity incident management – business realisation of the service **Description of the elements used:**

Alert created - BusinessEvent

Incident Management - BusinessFunction

The business capability of the SOC team to manage security incidents end-to-end, including recording, escalation, coordination and post-incident activities.

SLA threshold exceeded / escalation required - BusinessEvent

Security incident management service - BusinessService

Ensures triage, investigation, escalation, coordination of resolution, incident closure and reporting across tenants (customers).

Security incident detected - BusinessEvent

Asset (CI) - BusinessObject

Affected asset or service in the customer's environment.

Alert / Notification - BusinessObject

An input signal that may lead to an incident (triage decides).

Analyst - BusinessRole

Triage, investigation, enrichment of the incident with context, proposal of further steps.

Evidence / Artefacts - BusinessObject

Logs, indicators and analysis outputs needed for investigation and audit.

Incident (case) - BusinessObject

The main case record – status, severity, timeline, responsibilities.

Incident report - BusinessObject

Final report + recommendations/lessons learned.

Communication record - BusinessObject

Record of notifications and communication (who, when, what was communicated).

Manager - BusinessRole

Decisions on escalation, approvals, communication at management level, reporting.

Operator - BusinessRole

Operational coordination, communication, ticketing, SLA oversight, execution of routine actions.

Service operator - BusinessRole

Platform and data quality support; during an incident, helps to remove technical obstacles (e.g. missing logs, integrations).

Tenant (customer) - BusinessObject

Separation of data and responsibilities by customer.

Ticket - BusinessObject

A work item for carrying out remediation steps and monitoring SLAs.

Escalation and notifications - BusinessProcess

Escalate according to SLA and criticality; inform the appropriate contacts at the customer and internal roles.

Coordination of resolution and recovery - BusinessProcess

Management of remediation steps, status tracking, communication, ongoing updates.

Receive and record the incident - BusinessProcess

Creation of the case record and basic data, assignment of tenant, category and priority.

Triage and validation - BusinessProcess

Verify relevance, eliminate false positives, determine severity and the next course of action.

Closure and post-incident reporting - BusinessProcess

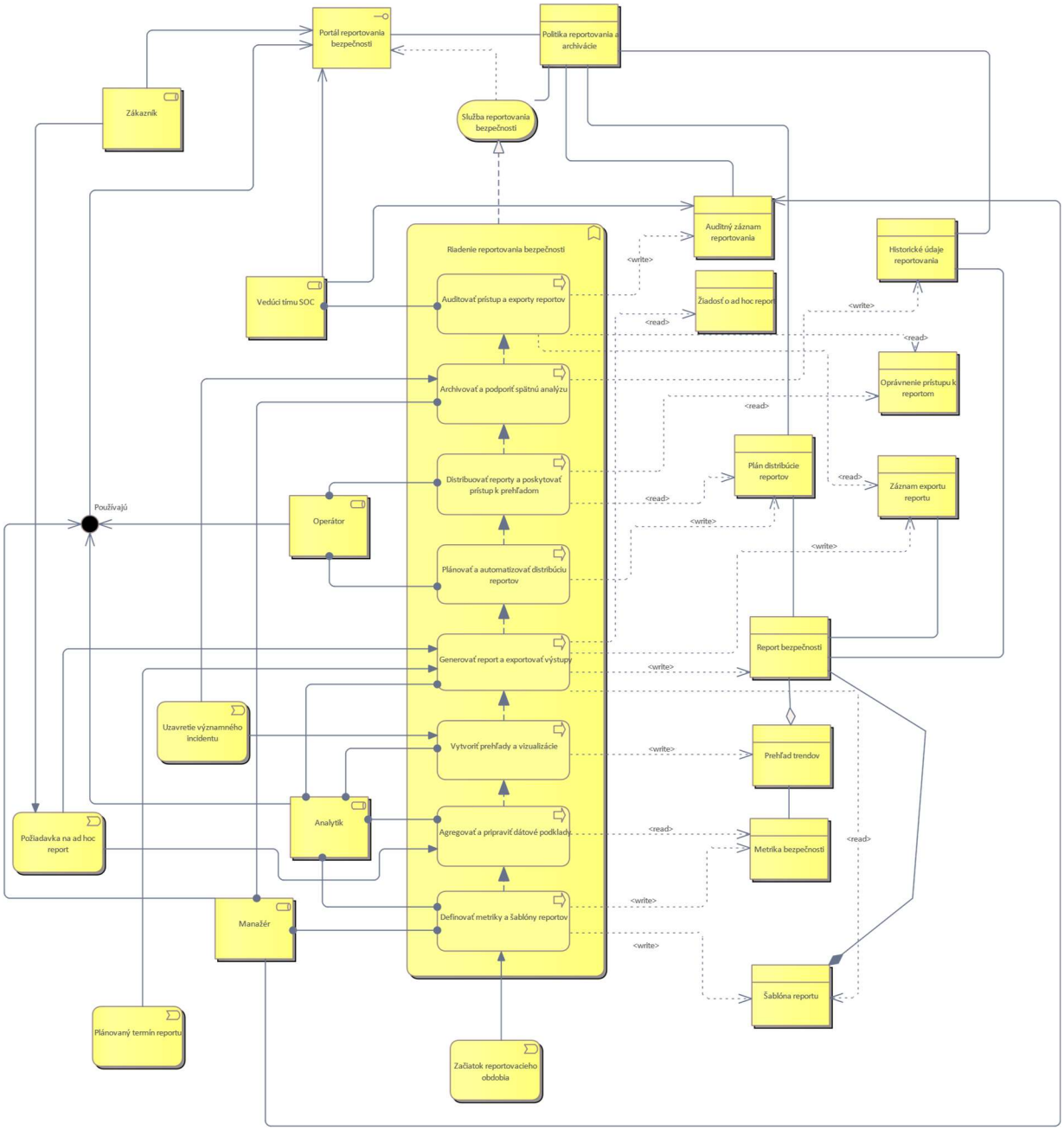
Incident conclusion, lessons learned, reports, recommendations, auditable trail.

Investigation and impact analysis - BusinessProcess

Determination of the cause, scope, affected assets and service; proposal of mitigation.

4.1.8 Security Reporting Management – Business Realisation of the Service

The diagram describes how security reporting is managed in the SOCaaS service, from triggering events to the creation of outputs, their distribution, archiving and audit. The business function groups together the processes that define metrics and templates, prepare the data basis, create visualisations, generate the report and ensure its delivery according to schedules and permissions. The roles Analyst, Operator, SOC Team Leader and Manager have divided responsibilities so that the outputs are consistent, traceable and usable both for the customer and for internal management.



Security reporting management - business realisation of the service

Description of the elements used:

Scheduled report date - BusinessEvent

The event expresses the regular reporting cycle according to the distribution plan and triggers the preparation and generation of outputs. It ensures that reports are produced consistently and on time for the agreed periods and recipients. It helps to maintain predictability and reduces dependence on manual initiation.

Ad hoc report request - BusinessEvent

The event arises when the customer or management needs an extraordinary output for a specific situation. It triggers the process of data preparation and report generation with a precisely defined scope and level of detail. It supports a rapid response during incidents, audits or when deciding on corrective steps. Relation to UC_04 Notify and escalate a security incident by supporting informed decision-making, traceability and the provision of outputs to responsible roles.

Security reporting management - BusinessFunction

The function manages the entire life cycle of reports, from defining metrics and templates through to publishing, distribution scheduling and archiving of outputs. It ensures that reports are consistent, comparable over time and adapted to the level of detail, from technical overviews to strategic summaries. The function supports traceability of exports and access control so that reporting is secure even in a multi-customer environment. Relation to UC_04 Notify and escalate a security incident by supporting informed decision-making, traceability and the provision of outputs to responsible roles.

Security reporting service - BusinessService

The service ensures the aggregation, visualisation and distribution of security information for various target groups in a multi-customer environment. It provides both regular and ad hoc outputs on incidents, alerts, the performance of controls and compliance with service level agreements in order to support decision-making and accountability. The service maintains history, enables retrospective analysis and supports audit and compliance with regulatory requirements. Relation to UC_04 Notify and escalate a security incident by supporting informed decision-making, traceability and the provision of outputs to responsible roles.

Closure of a significant incident - BusinessEvent

The event signals that the incident has been resolved and is ready for summarisation, lessons learned and assessment of impact in reports. It triggers an update of metrics and trends and, where appropriate, the preparation of an output for management or the customer. It improves accountability by ensuring that the outcome of the resolution is also captured in historical outputs. Relation to UC_04 Notify and escalate a security incident by supporting informed decision-making, traceability and the provision of outputs to responsible roles.

Start of the reporting period - BusinessEvent

The event marks the start of a new agreed reporting period, for example a week, a month or a quarter. It triggers verification of the validity of metrics and templates, followed by data preparation so that outputs are consistent and comparable over time. In a multi-customer environment, it also confirms for which customers and to what extent the given reporting cycle will be carried out.

Analyst - BusinessRole

Prepares the technical content of reports, interprets trends and explains the causes of changes in metrics and alerts. Uses reports to prioritise work, evaluate the effectiveness of detection and support incident investigation. Oversees the quality of the underlying data and adds comments on findings that are important for decision-making.

Reporting audit record - BusinessObject

A record of access to reports and of the creation of exports, containing who worked with the outputs, when, for what reason and to what extent. It enables retrospective verification of the legitimacy of access, investigation

of disputes and demonstration of the traceability of activities. In a multi-customer environment, the assignment to the customer and to the reporting period is also recorded.

Historical reporting data - BusinessObject

They represent archived outputs and underlying data that enable retrospective analysis of trends and verification of decisions. They maintain consistent time series and context so that the effectiveness of controls can be compared over a longer horizon. They also serve for audit purposes and to demonstrate compliance with record-retention requirements.

Manager - BusinessRole

Uses strategic summaries to decide on priorities, investments and responsibility for security controls. Requires outputs to be auditable and understandable so that they can be defended before internal audit or the regulator. Approves the rules for reporting, archiving and access to sensitive information in a multi-customer environment.

Security metric - BusinessObject

Represents a defined value used to measure the security posture, for example numbers of incidents and alerts or resolution time indicators. Contains the calculation methodology, the time window and the assignment to the customer so that results are comparable and interpretable. Serves as the basis for trend overviews, performance evaluation and compliance with service level agreements.

Operator - BusinessRole

Ensures the operational distribution of reports, checks the availability of outputs and handles practical requests for exports. Monitors scheduled report cycles, oversees delivery to the correct recipients and records outages or export errors. Helps maintain reporting dashboards so that they are always ready for analysts and management.

Report access authorisation - BusinessObject

Represents the rules that determine who can view specific dashboards, export reports or change templates. In a multi-customer environment, it ensures data separation and restricts access on the basis of role and customer. It supports secure distribution and minimises the risk of leakage of sensitive information.

Report distribution plan - BusinessObject

Contains schedules, recipients and rules for delivering reports to individual roles and customers. It takes into account the sensitivity of data and permissions so that information is disseminated only to the extent permitted. It supports automation and reduces the risk of delays or incorrect delivery.

Reporting and archiving policy - Contract

The policy defines the scope of reporting, the periodicity of outputs, minimum metrics and rules for the level of detail for different target groups. It governs the retention of historical data, requirements for the traceability of exports and rules for access to sensitive information. It ensures that reporting is consistent, auditable and compliant with internal and regulatory requirements. Relation to UC_04 Notify and escalate a security incident by supporting informed decision-making, traceability and the provision of outputs to responsible roles.

Trend overview - BusinessObject

Displays the development of selected metrics over time and makes it possible to identify deviations, deterioration or improvement of the situation. It supports comparison of periods and explanation of the causes of changes on the basis of events, interventions and configuration adjustments. It is used in planning controls, evaluating effectiveness and in communication with management and the customer.

Security report - BusinessObject

It is the resulting reporting output for a defined period or for a specific ad hoc need. It contains summaries of incidents, alerts, compliance with service level agreements and recommendations based on identified trends. It serves for decision-making, audit, communication with the customer and demonstration of accountability. Relation to UC_04 Notify and escalate a security incident by supporting informed decision-making, traceability and the provision of outputs to responsible roles.

Report template - BusinessObject

Defines the structure of the report, the scope of content, the level of detail and the method of presentation for a specific target group. It ensures consistency of outputs across periods and reduces the risk that important information is omitted. It also contains rules for labelling the period and version, and mandatory comments on interpretation.

SOC team leader - BusinessRole

Uses reports to manage team performance, check compliance with service level agreements and identify weaknesses in processes. Approves report templates and the level of detail for individual target groups so that outputs meet their needs. Coordinates improvement measures based on trends and recurring causes of incidents.

Customer - BusinessRole

Receives selected reports and dashboards according to the agreed scope and level of detail. Uses outputs to check the security posture and compliance with service level agreements, and to decide on corrective steps. May request ad hoc reports in the event of significant events or when compliance with audit requirements needs to be demonstrated.

Report export record - BusinessObject

Records the creation and handover of a report export, including the time, user, template and data scope. It enables retrospective verification of who created the output and supports the investigation of disputes or suspicious activities. It is important for audit and for traceability requirements in regulated environments.

Ad hoc report request - BusinessObject

Captures a request for an extraordinary output outside the regular cycle, for example in the event of a significant event or an audit. It contains the scope, target group, time period and expected level of detail so that the report is prepared correctly. It supports prioritisation of work and traceability of why the report was created and for whom it was intended.

Use - Junction

Security reporting portal - BusinessInterface

The interface provides access to dashboards, visualisations and report exports according to role and customer assignment. It enables filtering, comparison of periods and display of the context of metrics so that users understand the causes of changes. It supports traceability by also making available information on templates, versions and export history. Relation to UC_04 Notify and escalate a security incident by supporting informed decision-making, traceability and the provision of outputs to responsible roles.

Aggregate and prepare underlying data - BusinessProcess

The process collects data from security sources and prepares it in a form suitable for visualisation and metric calculation. It ensures consistency of time windows, normalisation of fields and quality control so that outputs are not distorted. In a multi-customer environment, it maintains data separation and the correct assignment of underlying data to the customer.

Archive and support retrospective analysis - BusinessProcess

The process retains historical reports and underlying data so that trends, decisions and compliance with requirements can be verified retrospectively. It makes it possible to compare periods, search for recurring causes of incidents and evaluate the effectiveness of security controls. Archiving supports audit and accountability by keeping outputs traceable and immutable.

Audit access to and exports of reports - BusinessProcess

The process records access to reports, export runs and significant changes to templates or permissions. It supports the investigation of unauthorised access and makes it possible to demonstrate who worked with sensitive information. It is important for multi-tier teams and for regulatory requirements that call for retrospective verification of activities.

Define report metrics and templates - BusinessProcess

The process determines which metrics are monitored, how they are calculated and which output templates are used for individual target groups. It takes into account both the technical view for analysts and strategic summaries for management and the customer so that the content is appropriate. The result is a unified framework that enables comparison over time and reduces the risk of ambiguous interpretation.

Distribute reports and provide access to dashboards - BusinessProcess

The process delivers reports and makes interactive dashboards available through the reporting portal in accordance with access rules. It supports ad hoc requests when a view of a specific event or period needs to be prepared quickly. It ensures that users see only data assigned to their role and customer.

Generate a report and export outputs - BusinessProcess

The process creates reports in formats suitable for distribution and archiving, for example tabular outputs and document reports. It ensures consistent labelling of the period, sources and template version so that the report remains traceable over time. On export, it creates a record that supports audit and the explanation of who created the output and when.

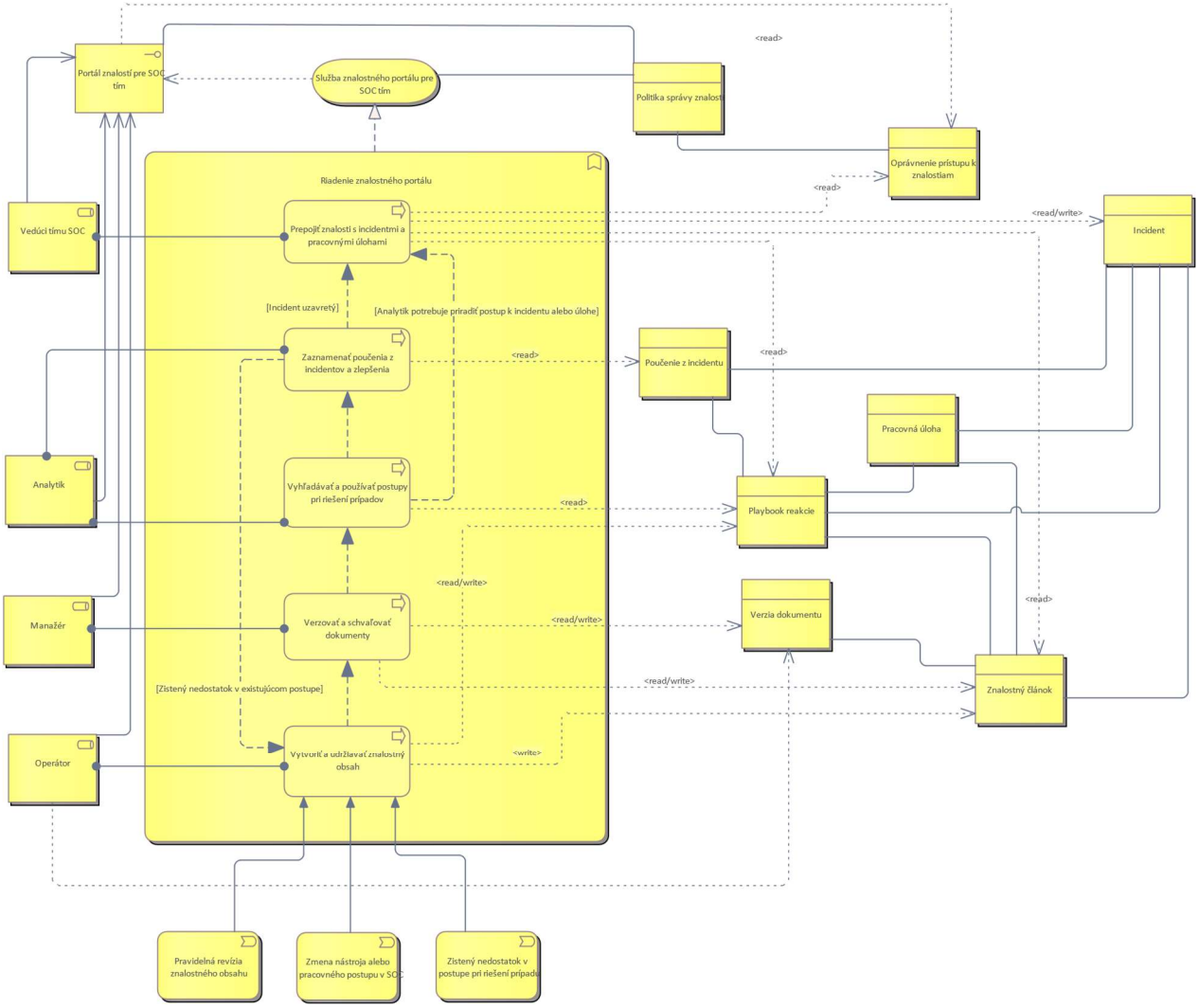
Plan and automate report distribution - BusinessProcess

The process sets up schedules for regular outputs and delivery rules for individual target groups and customers. It takes into account the sensitivity of data and permissions so that outputs reach only the correct recipients. It reduces the team's manual work and ensures that reporting takes place consistently in accordance with agreements.

Create dashboards and visualisations - BusinessProcess

The process creates overviews of trends over time, summaries of alerts and incidents, and views of compliance with service level agreements. Visualisations are designed to enable rapid filtering, comparison of periods and identification of deviations. Outputs are prepared for both technical and managerial use and support fact-based decision-making.

4.1.9 Knowledge portal management for the SOC team – business realisation of the service



Knowledge portal management for the SOC team - business realisation of the service

The diagram captures how SOCaaS builds and maintains the knowledge needed for a consistent response to incidents. It shows the responsibilities of roles in content creation, approval, searching for procedures and recording lessons learned from incidents, including access control and links to incidents and work tasks. The aim is for procedures to be traceable, versioned, controlled and immediately usable when handling cases.

Description of the elements used:

Knowledge portal management - BusinessFunction

The function manages the creation, approval, versioning and publishing of knowledge content so that it is up to date and usable in practice. It ensures the structure of content, findability, linking to incidents and traceability of changes for audit and retrospective learning. It supports integration with other SOC tools so that knowledge is

naturally used when handling cases. Relation to UC_04 Notify and escalate a security incident by supporting uniform procedures, traceability of decisions and collective learning from incidents.

Knowledge portal service for the SOC team - BusinessService

The service provides a central place for storing and sharing playbooks, procedures, recommendations and lessons learned from incidents. It increases the consistency of response, shortens the onboarding time of new members and supports the standardisation of methodologies and processes. In a multi-customer environment, it ensures permissions, change history, versioning and links to incidents and work tasks. Relation to UC_04 Notify and escalate a security incident by supporting uniform procedures, traceability of decisions and collective learning from incidents.

Manager - BusinessRole

The Manager approves content that has an impact on rules, responsibilities, escalations or requirements on the customer. Checks compliance with internal policies, audit requirements and commitments towards the customer. Is particularly suited to content with a higher level of formality.

Operator - BusinessRole

The Operator supplements the knowledge base with operational procedures, configurations and routine steps that ensure repeatability. Updates guides when tools and work procedures change and makes sure that the steps are feasible in practice. Is particularly suited to procedure- and guide-type content, not to analytical interpretations.

Knowledge access authorisation - BusinessObject

Defines who can read, edit or approve content and to what extent. Ensures that sensitive knowledge is available only to authorised roles and that no undesirable visibility occurs in a multi-customer environment. Supports access control and traceability of who has worked with the content.

Response playbook - BusinessObject

It is a practical step-by-step procedure for typical situations, for example malware, data leakage or suspicious communication. It ensures consistency of response, minimises improvisation and speeds up decision-making under time pressure. It contains checkpoints that help demonstrate the correct procedure during audit and retrospective review. Relation to UC_04 Notify and escalate a security incident by supporting uniform procedures, traceability of decisions and collective learning from incidents.

Knowledge management policy - Contract

The policy defines the structure of content, the rules for creating, approving and versioning documents and the minimum quality requirements. It governs permissions, access to sensitive knowledge, archiving and the way of linking to incidents and work tasks. It ensures that knowledge is consistent, auditable and maintainable in the long term in a multi-customer environment. Relation to UC_04 Notify and escalate a security incident by supporting uniform procedures, traceability of decisions and collective learning from incidents.

Lesson learned from an incident - BusinessObject

Records what worked, what failed and what improvements need to be made after an incident has been resolved. It is reflected in adjustments to playbooks, templates and settings so that the same mistakes are not repeated. It is important for building team knowledge and for increasing the effectiveness of response in the long term. Relation to UC_04 Notify and escalate a security incident by supporting uniform procedures, traceability of decisions and collective learning from incidents.

Document version - BusinessObject

Represents a specific state of a document at a point in time, including the author, date and list of changes. It makes it possible to return to previous content and to explain why a procedure or recommendation was modified. It supports audit and reduces the risk that outdated or unapproved content will be used.

Knowledge article - BusinessObject

Represents a standalone document with structured content that describes a topic, procedure or recommendation. It contains a description of the problem, resolution steps, conditions of use and references to related sources and incidents. It is versioned and governed by permissions so that quality and traceability of changes are maintained.

Knowledge portal for the SOC team - BusinessInterface

The interface provides access to documents, playbooks and lessons learned and enables searching by topics, incidents and attack techniques. It supports working with versions, comments and links to incidents and tasks so that knowledge can be used directly in operations. It ensures access control and displays the change history for traceability. Relation to UC_04 Notify and escalate a security incident by supporting uniform procedures, traceability of decisions and collective learning from incidents.

Link knowledge with incidents and work tasks - BusinessProcess

The process creates links between documents, incidents and tasks so that the correct procedure is used when handling cases. It makes it possible to understand quickly which procedure was used and what its outcome was, which supports audit and retrospective verification. It reduces duplicate documentation, as content is shared via references rather than repeated copying.

Version and approve documents - BusinessProcess

The process manages document versions, records changes and enables an approval procedure prior to publishing. It increases the credibility of content, as each change has an author and a reason and remains traceable over time. It supports audit and enables a return to previous versions in the event of disputes or when a comparison is needed.

Search for and use procedures when handling cases - BusinessProcess

The process makes it possible to quickly find a relevant playbook or guide according to the incident type, indicator or attack technique used. It supports a consistent procedure during response and reduces the risk of omitting important steps under time pressure. Knowledge is used directly when working with incidents, and the outcome can subsequently be added back into the content.

Create and maintain knowledge content - BusinessProcess

The process includes the creation of documents, playbooks and recommendations and their ongoing updating in line with the evolution of threats and tools. It ensures that knowledge is captured in a structured form and is available to the whole team without dependence on specific individuals. The content is also supplemented with practical notes and examples so that it is usable in real situations.

Record lessons learned from incidents and improvements - BusinessProcess

The process captures lessons learned from closed incidents and turns them into specific adjustments to playbooks, recommendations and procedures. It ensures that the team learns collectively and that knowledge is also available to new members. It helps to increase the quality of response over time and to reduce the repetition of the same mistakes.

Regular review of knowledge content - BusinessEvent

The event expresses a planned moment when the currency of articles and playbooks is checked according to the agreed cycle. It triggers a review of the content, identification of outdated procedures and preparation of changes for approval. It helps prevent the team from using outdated guides when handling cases.

Deficiency identified in the procedure when handling a case - BusinessEvent

The event arises when an analyst discovers during handling that an existing procedure is incomplete, unclear or does not cover a specific situation. It triggers the supplementing of content and its subsequent approval so that the finding is reflected in the standardised procedure. It supports continuous improvement without waiting for large review cycles.

Change of a tool or work procedure in the SOC - BusinessEvent

The event occurs when tools, workflows or responsibilities are modified in a way that changes how cases are handled. It triggers an update of the related articles and playbooks so that they correspond to actual operations. It reduces the risk of errors caused by a mismatch between documentation and practice.

Analyst - BusinessRole

Uses the knowledge portal when handling cases and searches for approved procedures. Is responsible for ensuring that current versions are used in practice and that deviations or new findings are fed back into the knowledge base. Participates in recording lessons learned after incidents are closed.

Incident - BusinessObject

A record of a security event that requires coordinated handling, recording of steps and closure with an outcome. It serves as an anchor for linking to the procedures used, relevant articles and lessons learned so that the course of handling can be defended retrospectively. It supports audit and accountability in a multi-customer environment.

Work task - BusinessObject

A record of a task that arises during remediation, supplementing of evidence or implementation of recommendations from knowledge content. It makes it possible to track status, deadlines, responsibility and outcome so that improvements are implemented and do not remain only in documentation. Linking to knowledge ensures that the task has a clear procedure and an expected output.

SOC team leader - BusinessRole

Ensures that knowledge is used consistently and that lessons learned from incidents are reflected in procedures. Manages the links between incidents, work tasks and relevant content so that the handling methodology is traceable. Oversees quality, completeness and adherence to the knowledge management policy.

4.2 BPMN – Business process model

Within this detailed solution design, the creation of BPMN diagrams or detailed modelling of business processes in BPMN notation is not envisaged. The subject of the document is primarily the architectural design of the solution, its functional areas, application and technology components, use cases and the links between the main parts of the proposed system.

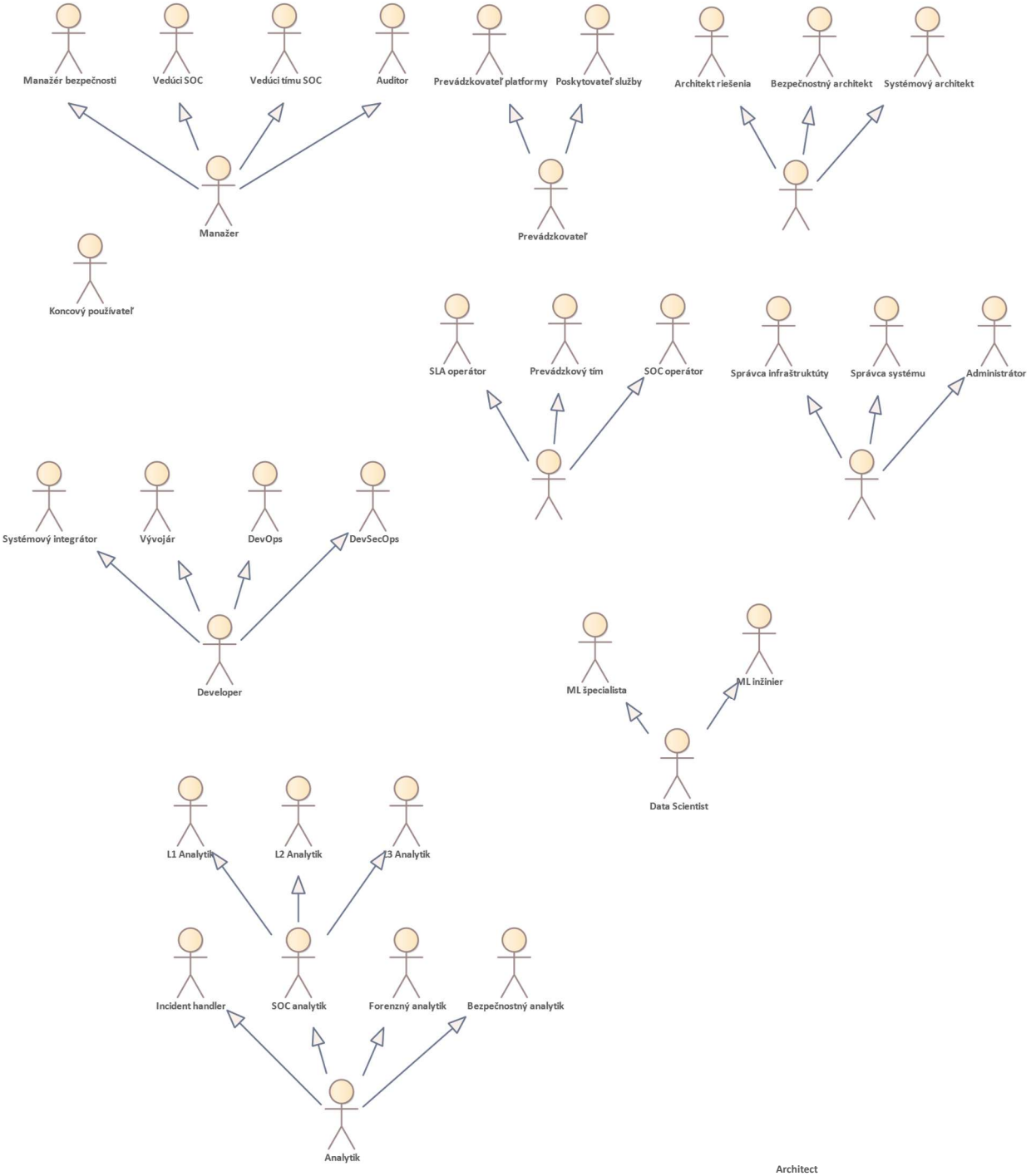
The business process view is covered in this document mainly through the description of the business architecture and the defined use cases, which sufficiently express the main activities, actor interactions and

supported process scenarios of the SOCaaS solution. For this reason, it was not necessary to produce separate BPMN diagrams. Should detailed process modelling be required in later phases of the project, BPMN notation may be used as a supplementary modelling tool, in particular for a more detailed design of selected operational, incident or support processes. Within this document, however, BPMN diagrams are not created and do not form part of the solution design.

4.3 Use cases

The following chapter presents the individual user scenarios that illustrate typical ways in which the proposed solution is used in practice. The scenarios describe user interactions with the system, their goals and expected outputs, while providing a comprehensive view of the functionality and practical usability of the solution in various situations. They also serve as a basis for a better understanding of processes and the identification of key system requirements.

4.3.1 Actors


Manažér bezpečnosti
Vedúci SOC
Vedúci tímu SOC
Auditor
Manažer
Koncový používateľ
Prevádzkovateľ platformy
Poskytovateľ služby
Prevádzkovateľ
Architekt riešenia
Bezpečnostný architekt
Systémový architekt
SLA operátor
Prevádzkový tím
SOC operátor
Správca infraštruktúty
Správca systému
Administrátor
Systémový integrátor
Vývojár
DevOps
DevSecOps
Developer
ML špecialista
ML inžinier
Data Scientist
L1 Analytik
L2 Analytik
L3 Analytik
Incident handler
SOC analytik
Forenzný analytik
Bezpečnostný analytik
Analytik
Architect

Actors

Actor/role name	Description
Administrator	Responsible for: SIEM, log processing, data visualisation, alerting, EDR/XDR, asset management, vulnerability scanner and security data sources.
Analyst	Responsible for areas such as: Log processing, SOAR, ML-based detection, Data visualisation, Alerting, EDR/XDR, Asset management, Vulnerability scanner, knowledge base, Incident recording and management, and Reporting.
Architect	Responsible for areas such as: ML-based detection and EDR/XDR.
Solution architect	Responsible for areas such as: SIEM and Log processing.
Auditor	Responsible for areas such as: SIEM, SOAR, Asset management, Incident recording and management, Reporting, Optional extensions, System administration, CTI and Alerting.
Security analyst	Responsible for areas such as: SIEM, Log processing, SOAR, ML-based detection, Data visualisation, Alerting, EDR/XDR, CTI, Asset management, Vulnerability scanner, Security data sources, Knowledge base, Ticketing, and Incident recording and management.

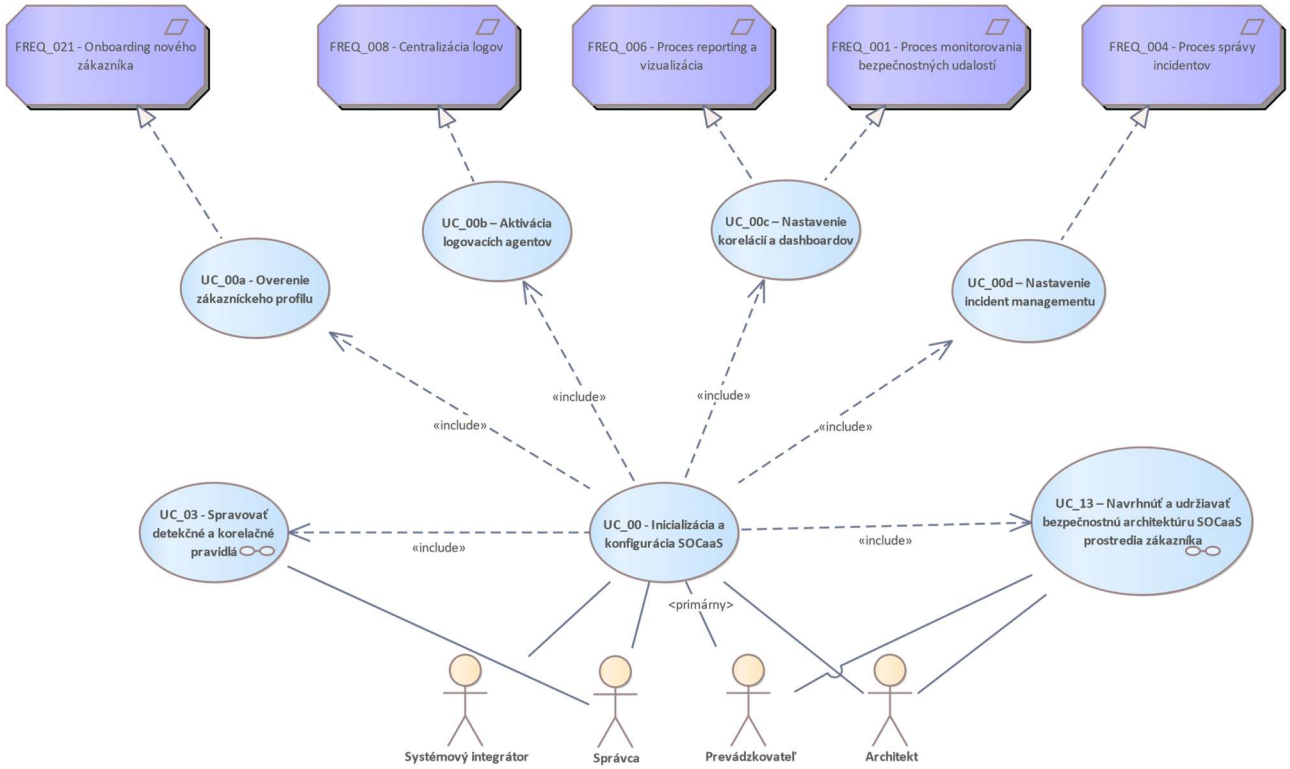
Security architect	Responsible for areas such as: SIEM, SOAR, ML-based detection, Data visualisation, EDR/XDR, Security data sources, Knowledge base, Incident recording and management, and System administration.
Data Scientist	Responsible for areas such as: ML-based detection.
Developer	Developer is a basic role focused on the design, development, modification and maintenance of software, scripts and integration solutions. It focuses on implementing logic, automating processes and providing technical support for development.
DevOps	Responsible for areas such as: SIEM, Log processing, ML-based detection, Vulnerability scanner, Security data sources, System administration and Optional extensions.
DevSecOps	Responsible for areas such as: SOAR, CTI, Vulnerability scanner and system administration.
Forensic analyst	Responsible for areas such as: optional extensions and EDR/XDR.
Incident handler	Responsible for areas such as: SOAR, and Incident recording and management.
Incident Handler	
End user	Responsible for areas such as: Data visualisation.
L1 Analyst	
L2 Analyst	
L3 Analyst	
Manager	Responsible for areas such as: SIEM, ML-based detection, data visualisation, alerting, Asset management, vulnerability scanner, knowledge base, Ticketing, Reporting and Optional extensions.
Security manager	Responsible for areas such as: SIEM, SOAR, ML-based detection, Data visualisation, Alerting, EDR/XDR, CTI, Asset management, Vulnerability scanner, Ticketing, Incident recording and management, Reporting, System administration and Optional extensions.
ML engineer	Responsible for areas such as: ML-based detection.
ML specialist	Responsible for areas such as: ML-based detection and EDR/XDR.
Operator	Responsible for areas such as: alerting, EDR/XDR, CTI, asset management, vulnerability scanner, security data sources, knowledge base, ticketing and system administration.
Service provider	Responsible for areas such as: SIEM.
Operating entity	Responsible for areas such as: data visualisation.
Platform operator	Responsible for areas such as: SIEM.
Operations team	Responsible for areas such as: alerting and optional extensions.
SLA operator	Responsible for areas such as: alerting.
SOC analyst	Responsible for areas such as: EDR/XDR, CTI, incident recording and management, reporting and optional extensions.
SOC operator	Responsible for areas such as: alerting, EDR/XDR, knowledge base and ticketing.
Custodian	The Custodian is a basic technical role responsible for the operation, maintenance and functioning of IT infrastructure and systems. It covers competences ranging from infrastructure, through systems management, to configuration administration.
Infrastructure administrator	Responsible for areas such as: Log processing, Asset management and System administration.
System administrator	Responsible for areas such as: SOAR, Data visualisation, alerting, EDR/XDR, Asset management, Vulnerability scanner, Security data sources, Knowledge base, Ticketing, Incident recording and management, Reporting and System administration data visualisation.

System architect	Responsible for areas such as: SIEM, Log processing, SOAR, ML-based detection, Vulnerability scanner and System administration.
System integrator	Responsible for areas such as: SIEM, Log processing, SOAR, ML-based detection,
Actor/role name	**Description**
	Data visualisation, Alerting, EDR/XDR, CTI, Asset management, Vulnerability scanner, Security data sources, Knowledge base, Ticketing, and Incident recording and management.
SOC lead	Responsible for areas such as: alerting.
SOC team lead	Responsible for areas such as: data visualisation.
Developer	Responsible for areas such as: SIEM, SOAR, ML-based detection and Security data sources.

4.3.2 General use cases

4.3.2.1 Use case diagrams

UC_00 – Initialisation and configuration of SOCaaS



UC_00 - Initialisation and configuration of SOCaaS

Use case	UC_00 - Initialisation and configuration of SOCaaS

Goal	The use case describes the initial initialisation and set-up of SOCaaS for a new customer. It includes the activation of logging sources and the set-up of integration points, dashboards and basic security processes.

Precondition	The customer has a tenant created.
Precondition	The Architect has configuration permissions.
Precondition	Basic network connectivity is available.

Postcondition	SOCaaS has been successfully initialised and is ready for monitoring and detection.
Postcondition	The basic modules are activated: logging, SIEM correlations, dashboards, incident management.
Postcondition	Logs from all mandatory sources contain the mandatory fields for detection.

Basic Path	Basic Path

Step	**Activity**
1	The Architect launches the initialisation process for a new customer.
2	The system loads the customer profile and prepares the configuration interface.
3	The Architect selects the modules to be activated (SIEM, SOAR, CTI, Ticketing).
4	The system performs the initialisation tasks (registration of sources, activation of agents, creation of dashboards).
5	The Architect confirms the configuration.
6	The system saves the configuration and announces successful completion.

ALT_00_01 – Incomplete customer profile	Alternate

Step	**Activity**
1	The system detects missing data.
2	It notifies the Architect to complete them.
3	The process continues once the information has been completed.

ALT_00_02 – Unavailable logging sources / infra	Alternate

Step	Activity
1	The system is unable to establish a connection with the logging agent or the customer's API.
2	It displays an error and offers the option to retry or skip the module.
3	The process continues with limited functionality.

Linked functional requirements	
UC_03 - Manage detection and correlation rules	This use case covers the activities of analysts and administrators in creating, managing and maintaining detection and correlation rules in the SOCaaS environment. These rules enable the detection of security threats on the basis of IOCs, behavioural anomalies, event chaining or threshold values. The use case focuses exclusively on the rule management process – their definition, tuning, classification and mapping to security frameworks (e.g. MITRE ATT&CK).
UC_00b – Activation of logging agents	The system activates the customer's logging agents to send logs to SOCaaS.
UC_13 – Design and maintain the security architecture of the customer's SOCaaS environment	UC_13 addresses the design and ongoing maintenance of the technical and security architecture for a specific customer: how the customer is connected to SOCaaS, which systems, zones and logging sources are integrated,
	through which channels data flows (agent, syslog, API, VPN, peering…), how SIEM, SOAR, TI, ticketing, EDR, vuln scanners and asset mgmt are interconnected. The output is an approved and maintained architectural design (topology, integration points, data flows) and a list of mandatory log sources and log types, which serves as the basis for onboarding (UC_00), monitoring (UC_01), asset management (UC_11), vulnerability management (UC_12) and automation (UC_07).
UC_00c – Set-up of correlations and dashboards	The system creates predefined correlation rules and dashboards for monitoring security events.
UC_00a - Verification of the customer profile	The system verifies whether the customer tenant contains all mandatory data (identification, technical, integration).
UC_00d – Set-up of incident management	The system activates the workflow for incident management and role assignment.

Use case	UC_00a - Verification of the customer profile
Goal	The system verifies whether the customer tenant contains all mandatory data (identification, technical, integration).

Precondition	The tenant exists in the system.

Postcondition	The customer profile is confirmed as complete.

Basic Path	Basic Path

Step	**Activity**
1	The Platform operator launches profile verification.
2	The system loads the customer data.
3	The system checks the mandatory fields.
4	Verification is successful.

ALT_00a_01 – Incomplete data	Alternate

Step	**Activity**
1	The system requests that the missing data be completed

ALT_00a_02 – Tenant lacks technical configuration	Alternate

Step	**Activity**
1	The system stops the process and notifies the administrator.

Linked functional requirements	
FREQ_021 - Onboarding of a new customer	The solution must ensure the onboarding of a new customer, during which their infrastructure is integrated into the multi-tenant SOCaaS environment. This includes: • setting up the logging agent • configuring correlation rules • creating separate dashboards • setting up incident management for the customer
UC_00a - Verification of the customer profile	The system verifies whether the customer tenant contains all mandatory data (identification, technical, integration).

Use case	UC_00b – Activation of logging agents
Goal	The system activates the customer's logging agents to send logs to SOCaaS.

Precondition	The customer systems are available.
Precondition	The communication channels are open.

Postcondition	The logging agents are active and sending data.

Basic Path	Basic Path

Step	**Activity**
1	The Platform operator launches activation.
2	The system contacts the logging agents.
3	The availability of the API and certificates is verified.
4	The agents are activated.

ALT_00b_01 – Agent does not respond	Alternate

Step	**Activity**
1	The system reports an error and continues later.

ALT_00b_02 – Invalid API keys	Alternate

Step	**Activity**
1	The system requests that the keys be uploaded again.

Linked functional requirements	
FREQ_008 - Log centralisation	The solution must ensure the collection and centralisation of logs from systems, networks, endpoints and applications.
UC_00b – Activation of logging agents	The system activates the customer's logging agents to send logs to SOCaaS.

Use case	UC_00c – Set-up of correlations and dashboards
Goal	The system creates predefined correlation rules and dashboards for monitoring security events.

Precondition	The SIEM module is active.
Precondition	Logs are arriving in the system.
Postcondition	Correlation rules and dashboards are configured.

Basic Path	Basic Path

Step	Activity
1	The Platform operator launches the set-up.
2	The system loads the correlation template.
3	The system creates dashboards for monitoring.
4	The configuration is saved.

ALT_00c_02 – Dashboard cannot be created	Alternate

Step	Activity
1	The system creates only a minimal version. (the last executable one)

ALT_00c_01 – Correlation rule is invalid	Alternate

Step	Activity
1	The system skips it and reports an error.

Linked functional requirements	
FREQ_006 - Reporting and visualisation process	Ensure the reporting and visualisation process – ensures regular output for management, auditors and SOC analysts.
FREQ_001 - Security event monitoring process	Ensure the security event monitoring process – ensures the collection and monitoring of logs, metric data and alerts from all integrated systems.
UC_00c – Set-up of correlations and dashboards	The system creates predefined correlation rules and dashboards for monitoring security events.

Use case	UC_00d – Set-up of incident management
Goal	The system activates the workflow for incident management and role assignment.

Precondition	The Ticketing or SOAR module is initialised.
Postcondition	The incident lifecycle (triage → analysis → closure) is ready.

Basic Path	Basic Path

Step	Activity
1	The Platform operator launches the configuration.

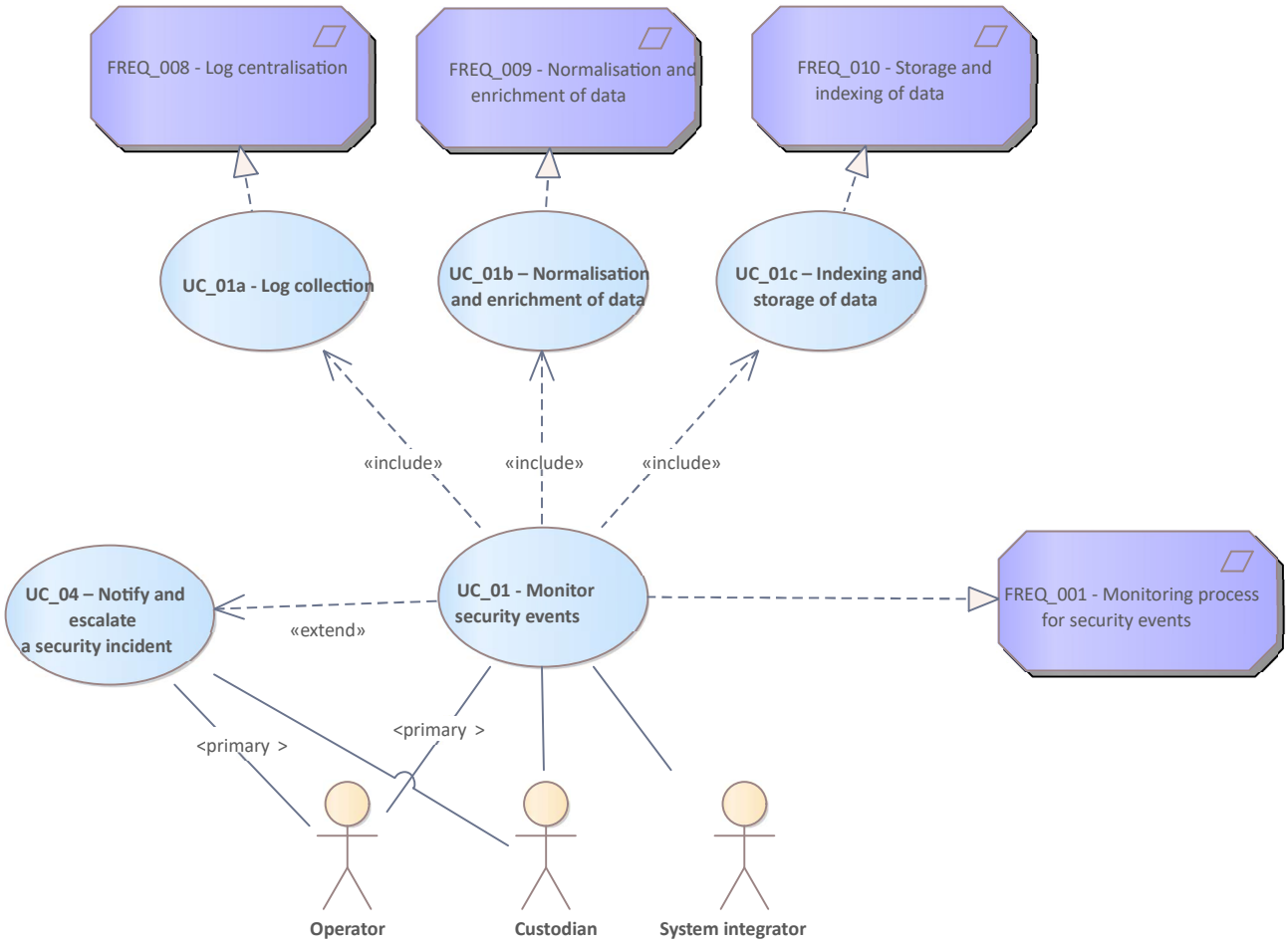
2	The system loads the workflow templates.
3	The system creates incident categories and priorities.
4	It assigns roles (Analyst, Operator, Manager).

ALT_00d_01 – No user roles exist	Alternate
Step	**Activity**
1	The system stops the workflow and asks the administrator to add roles.

ALT_00d_02 – Ticketing module is unavailable	Alternate
Step	**Activity**
1	The system activates only the basic SOAR fallback.

Linked functional requirements	
FREQ_004 - Incident management process	Ensure the incident management process – includes recording, classification, assignment of priorities, documentation and closure of incidents.
UC_00d – Set-up of incident management	The system activates the workflow for incident management and role assignment.

UC_01 – Monitor security events



UC_01 - Monitor security events

Use case	UC_01 - Monitor security events
Goal	The use case describes the continuous monitoring of security events within SOCaaS. It includes the collection of events, their normalisation and enrichment, and the preparation of alerts for the subsequent incident detection process.

Precondition	The logging agents are active.
Precondition	The SOCaaS SIEM module is functional.
Precondition	UC_00 (initialisation) has been successfully completed.
Postcondition	Events are processed and passed on to the incident detection process.

Postcondition	Alerts are displayed to the Operator.
Postcondition	Monitoring runs in real time.

Basic Path	Basic Path

Step	**Activity**
1	The Operator watches the control dashboard in SOCaaS.
2	UC_01a: The system receives events from logging sources.
3	UC_01b: The system normalises and enriches the data.
4	UC_01c: The system indexes the stored data for fast searching.
5	SOCaaS generates alerts on the basis of correlation rules.
6	The Operator sees the resulting security events.
7	The use case ends; monitoring continues without restriction. The Operator evaluates the alert.
8	If it is critical → activates UC_04 - Notify and escalate a security incident.
9	The Use Case ends – monitoring continues.

ALT_01_01 – Unavailable logs or agents	Exception

Step	**Activity**
1	SOCaaS is not receiving data from the source.
2	The system creates a “Missing logs” alert.
3	The administrator is notified and can resolve the problem manually.

ALT_01_02 – Overloaded SIEM or pipeline	Exception

Step	**Activity**
1	SOCaaS cannot keep up with processing the event stream.
2	The system activates protective mechanisms (rate-limiting, buffering).
3	The Operator is notified of the risk of event loss.

Linked functional requirements

UC_01c – Indexing and storage of data	The use case handles the storage of normalised security events in the repository and their indexing for fast searching and correlation mechanisms.
UC_04 – Notify and escalate a security incident	After the system detects and creates a security incident (UC_01, UC_03), the goal of UC_04 is to ensure that the right person (1st level) learns about it in time, that someone takes over the incident and that, in the event of inactivity or increased severity, the incident is automatically escalated to a higher level
UC_01b – Normalisation and enrichment of data	The use case ensures the normalisation of logs into a uniform format and their enrichment with information from CTI, the asset database or user data.
UC_01a - Log collection	It ensures the technical reception of log data from systems, applications, devices and sources into the SIEM platform. It is the foundation of the monitoring process.
FREQ_001 - Security event monitoring process	Ensure the security event monitoring process – ensures the collection and monitoring of logs, metric data and alerts from all integrated systems.

Use case	UC_01a - Log collection
Goal	It ensures the technical reception of log data from systems, applications, devices and sources into the SIEM platform. It is the foundation of the monitoring process.

Precondition	The communication channel between the system and the agent is functional.
Precondition	The logging agent is active.
Postcondition	The logs are ready for the next step – normalisation.
Postcondition	SOCaaS has received new logs.

Basic Path	Basic Path

Step	**Activity**
1	The logging agent sends security events.
2	SOCaaS receives the data.
3	The format and integrity of the logs are verified.
4	The logs are passed to the normalisation process.

ALT_01a_01 – Logs are not being delivered	Alternate

Step	**Activity**

1	SOCaaS does not record any data from the source.
2	The system creates a “Missing logs” notification.
3	The administrator receives a notification.

ALT_01a_02 – Invalid log format	Alternate

Step	**Activity**
1	The log has an incorrect format or missing fields.
2	SOCaaS rejects it and stores it in the quarantine buffer.
3	A notification is sent to the integrator.

Linked functional requirements	
FREQ_008 - Log centralisation	The solution must ensure the collection and centralisation of logs from systems, networks, endpoints and applications.
UC_01a - Log collection	It ensures the technical reception of log data from systems, applications, devices and sources into the SIEM platform. It is the foundation of the monitoring process.

Use case	UC_01b – Normalisation and enrichment of data
Goal	The use case ensures the normalisation of logs into a uniform format and their enrichment with information from CTI, the asset database or user data.

Precondition	A configuration of normalisation rules exists.
Precondition	UC_01a (log collection) was successful.
Postcondition	The log is normalised and enriched.
Postcondition	The data is ready for storage and indexing.

Basic Path	Basic Path

Step	**Activity**
1	SOCaaS loads the received logs.
2	A normalisation profile is applied according to the source type.
3	Data enrichment (IP reputation, hostname, user).

4	The resulting event is ready for indexing.

ALT_01a_01 – Unknown log type	Alternate

Step	**Activity**
1	The system uses a fallback normalisation profile.

ALT_01a_02 – CTI feed is missing	Alternate

Step	**Activity**
1	The system performs normalisation without enrichment and marks the event as “unenriched”.

Linked functional requirements	
FREQ_009 - Normalisation and enrichment of data	The solution must ensure the normalisation and enrichment of data.
UC_01b – Normalisation and enrichment of data	The use case ensures the normalisation of logs into a uniform format and their enrichment with information from CTI, the asset database or user data.

Use case	UC_01c – Indexing and storage of data
Goal	The use case handles the storage of normalised security events in the repository and their indexing for fast searching and correlation mechanisms.

Precondition	Data normalisation (UC_01b) is complete.
Precondition	The repository and cluster are available.
Postcondition	Events are stored in the SIEM database.
Postcondition	Indexes are ready for query systems.

Basic Path	Basic Path

Step	**Activity**
1	The system takes over the normalised data.
2	It writes it to the repository (DB or distrib. index).
3	It creates/updates the indexes.

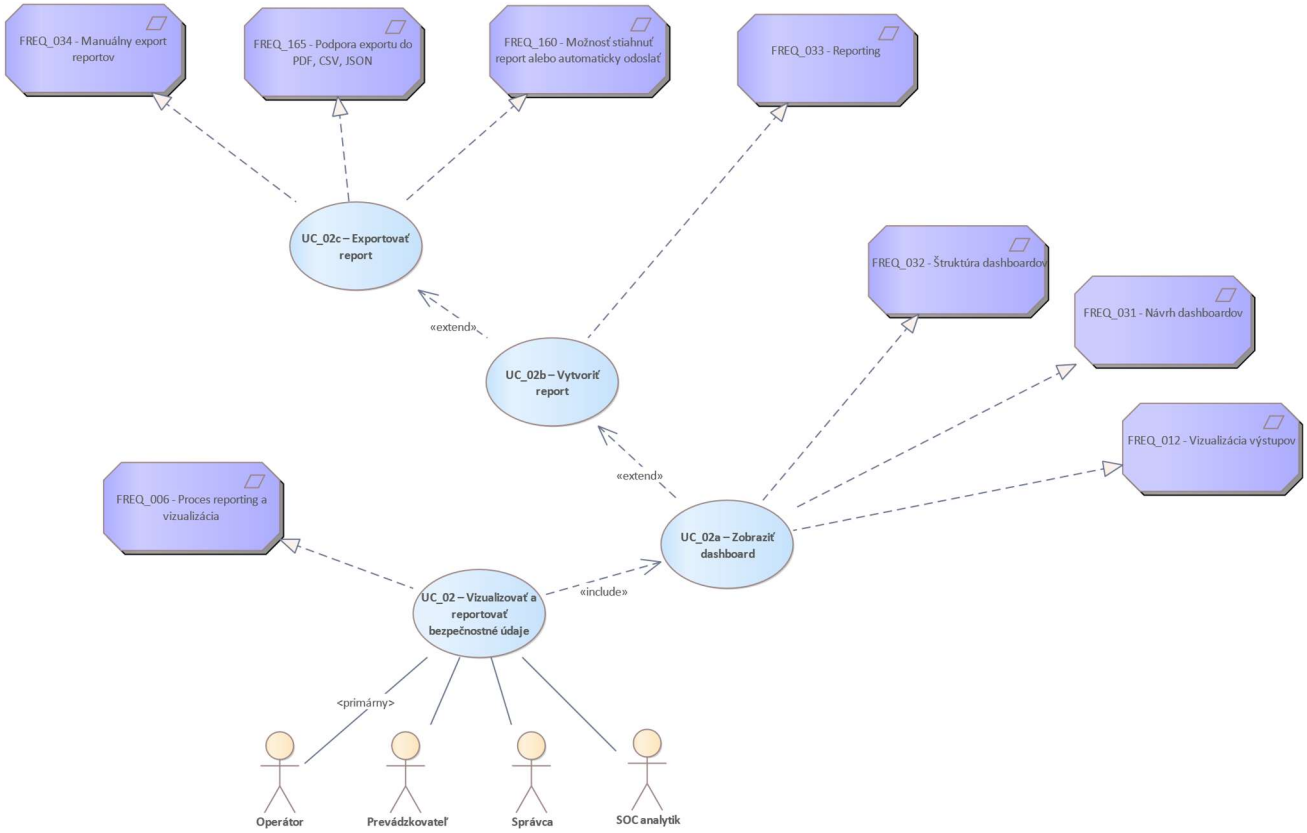
4	The event is available for correlation rules.

ALT_01c_01 – Repository unavailable	Alternate
Step	**Activity**
1	The system stores the data in a buffer and creates an alert.

ALT_01c_02 – Storage space exhausted	Alternate
Step	**Activity**
1	The system activates a cleanup or alerts operations.

Linked functional requirements	
FREQ_010 - Storage and indexing of data	The solution must ensure the storage, indexing and retention of data.
UC_01c – Indexing and storage of data	The use case handles the storage of normalised security events in the repository and their indexing for fast searching and correlation mechanisms.

UC_02 – Visualise and report security data



UC_02 - Visualise and report security data

Use case	UC_02 – Visualise and report security data
Goal	The system enables SOC operators and the administrator to display security events by means of dashboards, filters, charts and tabular outputs. It also enables the generation of reports and the export of data for management, audit or customers.

Precondition	Dashboards are defined in the configuration (UC_00c).
Precondition	UC_01c – Data is indexed and available for visualisation.
Precondition	UC_01 – Monitoring is running and providing data.
Postcondition	The activity is recorded in the audit log.
Postcondition	The report has been generated or exported (if initiated).

Postcondition	The user sees updated visualisations.

Basic Path	Basic Path

Step	**Activity**
1	The actor opens the “Visualisation and reporting” module.
2	The system loads the available dashboards and reports according to the actor's role.
3	The actor opens a dashboard → the system displays charts, metrics and tables.
4	The actor can apply filters (time, event type, source).
5	The actor can click on “Create report”.
6	The system compiles the report according to the selected format and template.
7	The actor can use the “Export report” function (PDF, CSV, JSON).

ALT_02_01 – Unavailable data	Alternate

Step	**Activity**
1	The dashboard returns an empty dataset.
2	The system displays a warning.
3	The Operator and the Operating entity can change the filtering criteria.

ALT_02_02 – Incorrect permissions	Alternate

Step	**Activity**
1	The actor (e.g. the Operating entity) opens a dashboard outside their tenant.
2	The system displays the error “Insufficient permissions”.

ALT_02_03 – Report generation failed	Alternate

Step	**Activity**
1	It displays an error message.
2	It records the error in the audit log.

3	The actor (e.g. the Operating entity) opens a dashboard outside their tenant.
Linked functional requirements	
UC_02a – Display a dashboard	The user opens a dashboard that visualises security events, metrics and system states in real time.
FREQ_006 - Reporting and visualisation process	Ensure the reporting and visualisation process – ensures regular output for management, auditors and SOC analysts.
UC_02 – Visualise and report security data	The system enables SOC operators and the administrator to display security events by means of dashboards, filters, charts and tabular outputs. It also enables the generation of reports and the export of data for management, audit or customers.
Use case	UC_02a – Display a dashboard
Goal	The user opens a dashboard that visualises security events, metrics and system states in real time.

Precondition	The system is initialised and configured (UC_00).
Precondition	All log sources are active and data is being received (UC_01).
Precondition	The user is authenticated and has a role that allows dashboards to be displayed.
Postcondition	The dashboard has been displayed with current data.
Postcondition	The user can save filters, views or sections (if they have permission).

Basic Path	Basic Path
Step	**Activity**
1	The actor opens the Dashboards module.
2	The system loads the available dashboards according to role.
3	The actor selects a dashboard to display.
4	The system loads and renders the visualisations (charts, heatmaps, tables).
5	The system updates the visualisations in real time or at defined intervals.
ALT_02a_02 – The user does not have permission for a specific dashboard	Alternate

Step	Activity
1	The system hides the dashboard or displays “Access Denied”.
2	The actor must request access from the administrator.

ALT_02a_01 – Data does not load	Alternate

Step	Activity
1	The system detects that sources are unavailable.
2	Instead of charts, it displays the message: “Data is not available.”
3	It offers to refresh or switch the source.

Linked functional requirements	
FREQ_012 - Visualisation of outputs	The solution must ensure the visualisation of outputs
FREQ_031 - Dashboard design	The solution must ensure the design of dashboards. To maintain clarity, dashboards within SOCaaS will be divided according to functional layers as follows: 1. Global overviews (SOC Overview) Intended for management and SLA reporting – they include daily/monthly incident statistics, threat types, top 10 IP addresses, countries of origin of attacks, ticket resolution time. 2. Alert monitoring dashboards Focused on operational monitoring – they display new alerts, breakdown by severity (Critical, High, Medium), time trends, and correlations between multiple sources. 3. Thematic dashboards They cover specific areas: • EDR activities: recorded files, processes, attempts to execute malicious code. • Vulnerabilities and scans: results from Greenbone or OpenSCAP. • Firewall and network logs: access, ports, detected scanning activities.
	4. Detailed analytical dashboards – They serve L2 analysts and threat hunting. They contain heatmaps, time slices, distributions by hour, country, ASN, occurrence of IOCs and the like. 5. Ad-hoc and incident-specific dashboards – They are created during the investigation of a specific case, where the analyst builds their own visualisation based on data from Elasticsearch.
UC_02b – Create a report	The user creates a one-off report or saves a reporting template based on data from dashboards or events.

FREQ_032 - Dashboard structure	The solution must ensure a dashboard structure in which dashboards are composed of individual panels that may contain charts, tables, heatmaps, distribution histograms, metrics or maps. Each dashboard within SOCaaS is designed as a composition of these visualisations focused on a specific aspect of monitoring or analysis. System security overview Correlation and incident dashboards Visualisation of network traffic and access Monitoring of authentications and user behaviour Integrated overview for the SOC analyst
UC_02a – Display a dashboard	The user opens a dashboard that visualises security events, metrics and system states in real time.
UC_02a – Display a dashboard	The user opens a dashboard that visualises security events, metrics and system states in real time.
UC_02a – Display a dashboard	The user opens a dashboard that visualises security events, metrics and system states in real time.
UC_02a – Display a dashboard	The user opens a dashboard that visualises security events, metrics and system states in real time.
Use case	UC_02b – Create a report
Goal	The user creates a one-off report or saves a reporting template based on data from dashboards or events.

Precondition	Dashboards are available and data is indexed (UC_01).
Precondition	The user has the “Create Report” permission.
Postcondition	The report is saved in the system.
Postcondition	It can be displayed, edited or exported (UC_02c).

Basic Path	Basic Path

Step	**Activity**
1	The actor opens the Reports module.
2	Selects “Create new report”.
3	Selects: report type, time range, template or custom design.
4	The system loads the necessary data.
5	The user confirms generation.
6	The system generates the report and saves it in the report database.

ALT_02b_02 – Data unavailable for the report	Alternate

Step	**Activity**
1	The system displays the message “Data for the selected period is not available”.
2	The Operator, Operating entity or Custodian selects another period.

ALT_02b_01 – Invalid report parameter	Alternate

Step	**Activity**
1	The system detects a missing parameter (time period, data source...).
2	It displays an error and requests completion.

Linked functional requirements	
FREQ_033 - Reporting	The solution must ensure reporting. Reporting in SOCaaS will fulfil several basic objectives: • Operational overview – display of the status of systems, number of incidents, agent activity. • Security reporting – outputs on detected threats, attack types, trends. • Compliance reporting – reports according to regulatory requirements (e.g. GDPR, NIS2). • Management reporting – aggregated statistics and indicators for the organisation's management. • Customer reporting – individual overviews for individual clients of the multi-tenant SOCaaS.
UC_02c – Export a report	The user exports an existing report to PDF, CSV, JSON or PNG and can send it by email or webhook.
UC_02b – Create a report	The user creates a one-off report or saves a reporting template based on data from dashboards or events.

Use case	UC_02c – Export a report
Goal	The user exports an existing report to PDF, CSV, JSON or PNG and can send it by email or webhook.

Precondition	A generated report exists (UC_02b).
Precondition	The user has “Export Report” rights.
Postcondition	The system has created a file in the selected format.
Postcondition	The file has been downloaded or sent to an external endpoint.

Basic Path	Basic Path

Step	**Activity**
1	The user opens the report detail.
2	Selects the Export option.
3	Selects the format (PDF/CSV/JSON/PNG).
4	The system generates the export.
5	The file is downloaded or sent according to the settings.

ALT_02c_02 – Insufficient permissions	Alternate

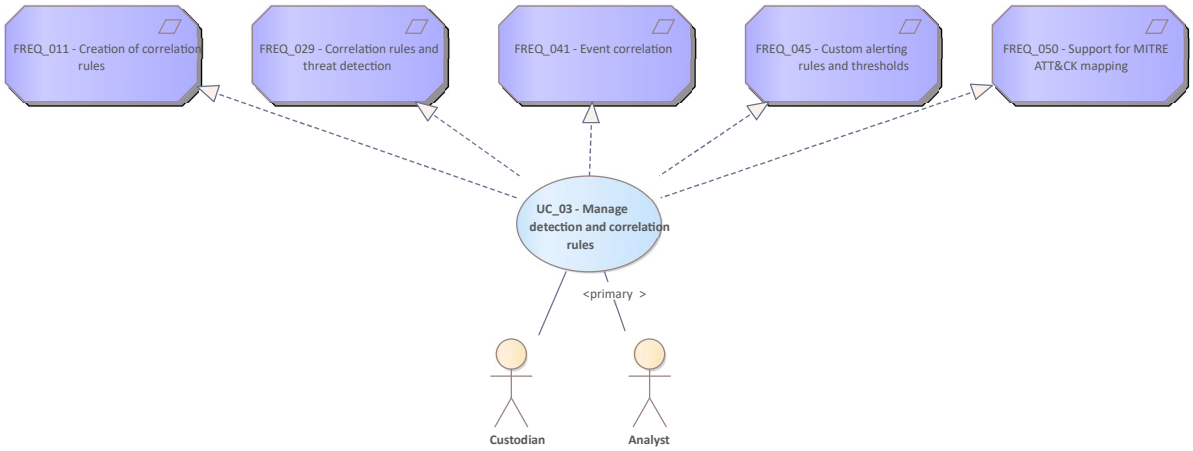
Step	**Activity**
1	The system displays “Access Denied”.
2	It offers the option to request access.

ALT_02c_01 – Export fails (timeout, large dataset)	Alternate

Step	**Activity**
1	The system reports an export error.
2	It offers an offline export or splitting of the data.

Linked functional requirements	
FREQ_160 - Option to download a report or send it automatically	Exporting reports manually or their scheduled automatic distribution. Option to download a report or send it automatically (e-mail, webhook)
FREQ_034 - Manual export of reports	The solution must ensure the manual export of reports. Reports will be exportable as: • PDF document – a generated snapshot of the current dashboard. • PNG image – static images of individual visualisations. • CSV files – raw data from tables and data fields. • JSON – for automated analysis and further processing.
FREQ_165 - Support for export to PDF, CSV, JSON	Option to export reports to various common formats. Support for export to PDF, CSV, JSON
UC_02c – Export a report	The user exports an existing report to PDF, CSV, JSON or PNG and can send it by email or webhook.

UC_03 – Manage detection and correlation rules



UC_03 - Manage detection and correlation rules

Use case	UC_03 - Manage detection and correlation rules
Goal	This use case covers the activities of analysts and administrators in creating, managing and maintaining detection and correlation rules in the SOCaaS environment. These rules enable the detection of security threats on the basis of IOCs, behavioural anomalies, event chaining or threshold values. The use case focuses exclusively on the rule management process – their definition, tuning, classification and mapping to security frameworks (e.g. MITRE ATT&CK).

Precondition	The actors have access to the administration interface for creating rules.
Precondition	The system supports the definition, management and versioning of detection rules.
Postcondition	A new rule has been created or an existing rule has been modified, deactivated or deleted.
Postcondition	The rule is versioned, validated and ready for deployment to the production environment.
Postcondition	The rule is available for evaluating security events (UC_04).

Basic Path	Basic Path

Step	**Activity**
1	The Analyst or administrator opens the interface for managing detection rules.
2	Creates a new rule or searches for an existing one.

3	Defines the conditions: event type, IOC/indicator, frequency of occurrence, source/destination entities,
4	Assigns the rule a severity, threat type and MITRE ATT&CK category.
5	The rule is saved and versioned.
6	It can be validated on historical data or marked as active.
7	The rule begins to be used in the evaluation of events (UC_04).

ALT_03_01 – The rule is tested on historical data (debugging)	Alternate

Step	**Activity**
1	The Analyst uses a tool to test rules against historical logs.
2	The results are displayed; the rule is then modified or approved.

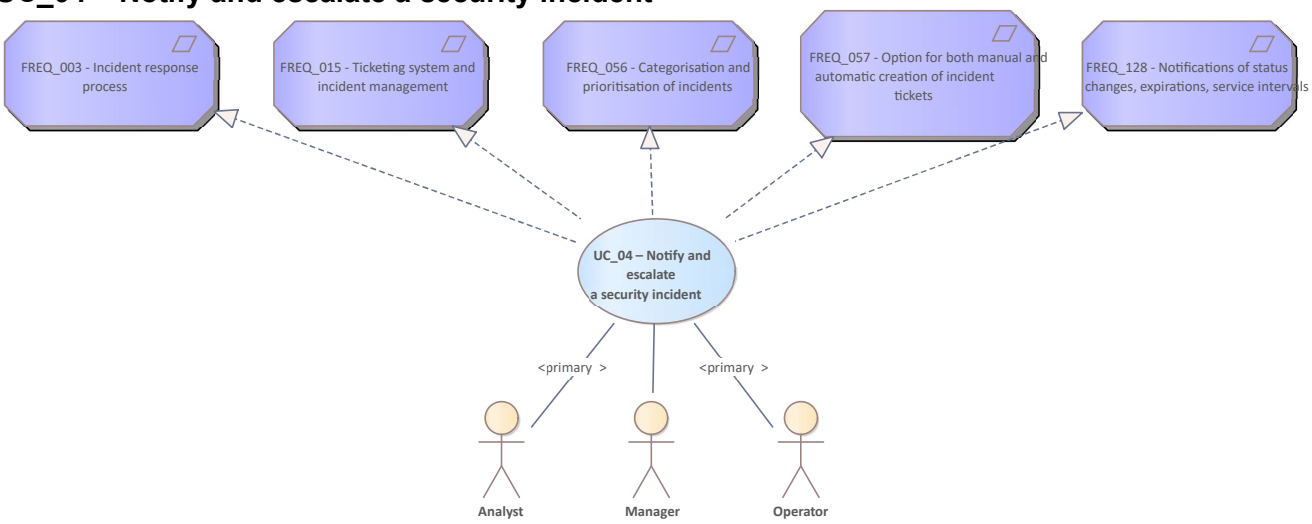
ALT_03_02 – The rule is created via a no-code interface	Alternate

Step	**Activity**
1	The Analyst uses a visual interface (drag & drop) to create conditions without writing syntax.
2	The platform automatically creates the underlying rule code.

Linked functional requirements	
FREQ_041 - Event correlation	Ability to link related events by means of correlation rules. Event correlation (predefined and custom rules)
FREQ_011 - Creation of correlation rules	The solution must ensure the creation of correlation rules and alerts
FREQ_050 - Support for MITRE ATT&CK mapping	Ability to map detections to tactics and techniques according to MITRE ATT&CK. Support for MITRE ATT&CK mapping

FREQ_029 - Correlation rules and threat detection	The solution must ensure correlation rules and threat detection. • Alerts on basic IOCs – comparison with a database of known indicators (IP, URL, hash). • Anomalous user behaviour – e.g. multiple failed logins within a short time, login from a new geographical region. • Non-standard network communications – e.g. DNS queries to dynamically generated domains (DGA), deviations in time and data volume. • Activities with administrative rights – e.g. adding a user to the "Administrators" group, ACL changes, running netsh, powershell. • Event chaining – correlation between multiple logs over time: e.g. login ? file transfer ? execution of a malicious process. These rules will be managed in a way that makes it possible to: • monitor detection alerts, • set threshold values (thresholds), • configure exceptions and ignored values (exceptions), • export and import rules as JSON objects.
FREQ_045 - Custom alerting rules and thresholds	Ability to define custom rules and conditions for generating alerts. Custom alerting rules and thresholds
UC_03 - Manage detection and correlation rules	This use case covers the activities of analysts and administrators in creating, managing and maintaining detection and correlation rules in the SOCaaS environment. These rules enable the detection of security threats on the basis of IOCs, behavioural anomalies, event chaining or threshold values. The use case focuses exclusively on the rule management process – their definition, tuning, classification and mapping to security frameworks (e.g. MITRE ATT&CK).
UC_03 - Manage detection and correlation rules	This use case covers the activities of analysts and administrators in creating, managing and maintaining detection and correlation rules in the SOCaaS environment. These rules enable the detection of security threats on the basis of IOCs, behavioural anomalies, event chaining or threshold values. The use case focuses exclusively on the rule management process – their definition, tuning, classification and mapping to security frameworks (e.g. MITRE ATT&CK).
UC_03 - Manage detection and correlation rules	This use case covers the activities of analysts and administrators in creating, managing and maintaining detection and correlation rules in the SOCaaS environment. These rules enable the detection of security threats on the basis of IOCs, behavioural anomalies, event chaining or threshold values. The use case focuses exclusively on the rule management process – their definition, tuning, classification and mapping to security frameworks (e.g. MITRE ATT&CK).

UC_04 – Notify and escalate a security incident



UC_04 – Notify and escalate a security incident

Use case	UC_04 – Notify and escalate a security incident
Goal	After the system detects and creates a security incident (UC_01, UC_03), the goal of UC_04 is to ensure that the right person (1st level) learns about it in time, that someone takes over the incident and that, in the event of inactivity or increased severity, the incident is automatically escalated to a higher level

Precondition	An incident exists – the incident has been created and has an assigned category and severity
Precondition	Notification and escalation policies are set up
Precondition	Contact channels and the on-call roster are known
Postcondition	The relevant recipient has been notified
Postcondition	The incident has a recorded history of notifications/escalations
Postcondition	In the event of inactivity, escalation has taken place in accordance with the SLA

Basic Path	Basic Path

Step	**Activity**
1	Creation or change of an incident
2	Evaluation of the notification policy

3	Selection of primary recipients
4	Generation of the notification message
5	Sending of the notification via primary channels
6	Waiting for confirmation (acknowledgement)
7	Acknowledgement of the incident by the Analyst/Operator
8	Stopping further escalation
9	Update of the incident history

ALT_01 – Failure of the primary notification channel	Alternate

Step	**Activity**
1	Sending the notification fails (e.g. the e-mail server is unavailable).
2	The system automatically tries an alternative channel (e.g. SMS, another mail gateway, chat).
3	Both the failed attempt and the subsequent fallback are recorded in the incident history.
4	If all channels fail, the incident is flagged as “notification failed” and the system automatically creates an internal ticket for the infrastructure administrator.

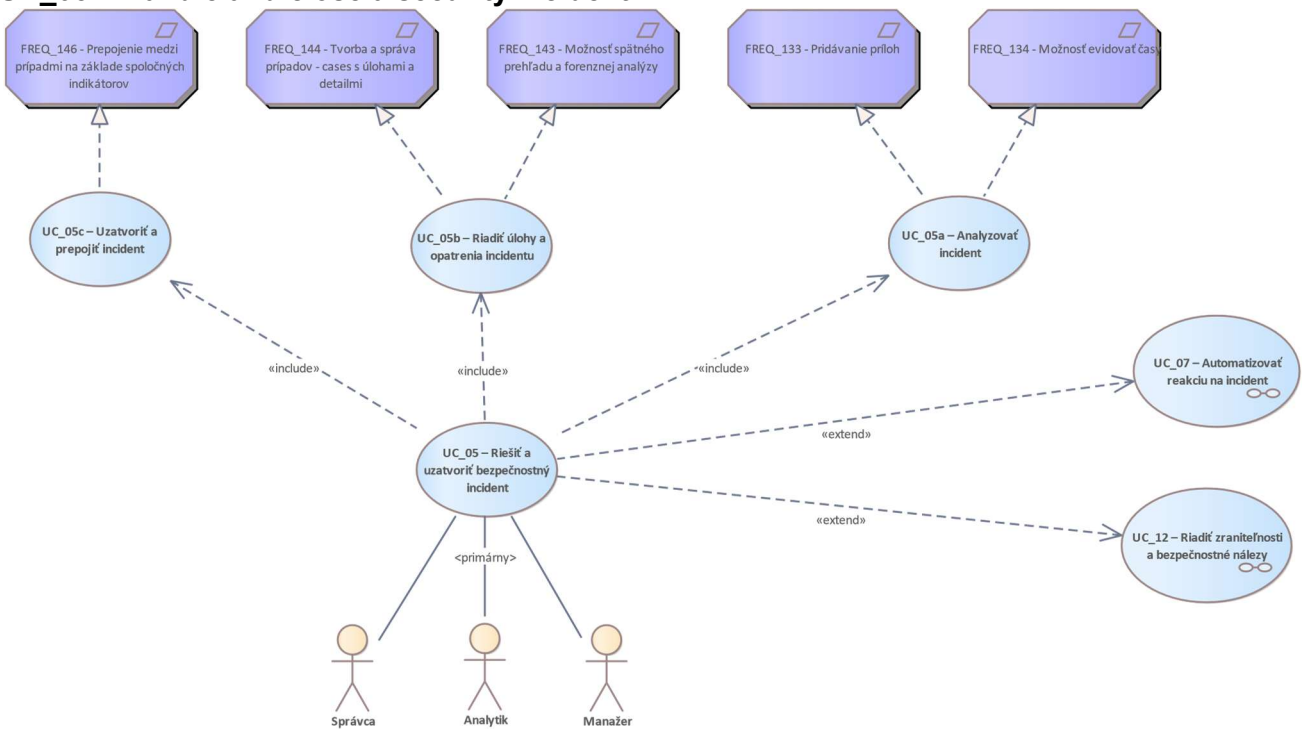
ALT_02 - Incident is not acknowledged within the SLA time - escalation	Alternate

Step	**Activity**
1	The incident has not been acknowledged after the SLA interval has elapsed.
2	The system automatically escalates the incident to a higher level (Manager, SOC lead): increasing the priority / severity of the incident, sending an escalation notification to the Manager via their preferred channels.
3	A record of the failure to take over within the SLA and of the escalation performed is added to the incident history.
4	Optionally, a further playbook is launched (e.g. partial automated measures, pre-prepared actions).

Linked functional requirements	
FREQ_003 - Incident response process	Ensure the incident response process – defines the way in which the team responds to identified incidents. It includes escalation, assignment of responsibility, implementation of measures, as well as communication with affected parties.

FREQ_015 - Ticketing system and incident management	The solution must ensure that the ticketing and incident management system provides a workflow for incident resolution, escalation, audit and tracking of resolution.
FREQ_128 - Notifications of status changes, expirations, service intervals	Automatic notifications upon a change of status or the need for service intervention. Notifications of status changes, expirations, service intervals
FREQ_057 - Option for both manual and automatic creation of incident tickets	Support for both manual and automatic creation of incident tickets from input data. Option for both manual and automatic creation of incident tickets
FREQ_056 - Categorisation and prioritisation of incidents	Ability to sort incidents by severity, type, category and importance. Categorisation and prioritisation of incidents
UC_04 – Notify and escalate a security incident	After the system detects and creates a security incident (UC_01, UC_03), the goal of UC_04 is to ensure that the right person (1st level) learns about it in time, that someone takes over the incident and that, in the event of inactivity or increased severity, the incident is automatically escalated to a higher level

UC_05 – Handle and close a security incident



UC_05 – Handle and close a security incident

Use case	UC_05 – Handle and close a security incident

Goal	After the detection of an incident and the sending of notifications (UC_01, UC_03, UC_04), this UC covers the entire incident handling process – from takeover, analysis and implementation of corrective measures through to closure of the incident and complete documentation of its course (timeline, evidence, linked cases). The result is an incident that has a clear owner, implemented steps, recorded times and prepared materials for any “lessons learned”

Precondition	The incident has been created and notified
Precondition	The incident has an assigned owner / team
Precondition	Relevant data and logs are available
Postcondition	The incident has a documented course and outcome
Postcondition	The agreed set of corrective measures has been carried out
Postcondition	The incident is closed or reclassified

Basic Path	Basic Path

Step	**Activity**
1	Takeover of the incident by the Analyst
2	Collection and supplementing of evidence
3	Analysis and identification of causes
4	Proposal of corrective measures
5	Implementation of measures by the Custodian
6	Tracking of times and the course of handling
7	Review and approval of the outcome
8	Closure of the incident
9	Preparation of materials for lessons learned

ALT_ 01 – The incident is evaluated as a false positive	Alternate

Step	**Activity**
1	After analysis, the Analyst finds that no incident occurred (incorrect correlation, test traffic).

2	The Analyst updates the incident to the “false positive” status and briefly documents the reason.
3	The incident is closed but remains in the history, including the evidence – it can later be used to adjust detection rules

ALT_02 – Corrective measures cannot be implemented immediately	Alternate

Step	**Activity**
1	The Custodian finds that the proposed measure has a high impact on operations (e.g. shutting down a critical service).
2	In the incident, they create scheduled tasks with deadlines and comments, or a request for approval to the Manager.
3	The incident remains in the “open – awaiting scheduled measures” status with recorded times and risk.

Linked functional requirements	
UC_05a – Analyse an incident	The Analyst examines the incident in detail – reviews the event timeline, logs and IOCs, supplements evidence and creates a comprehensive picture of the cause and scope of the incident.
UC_05b – Manage incident tasks and measures	The Analyst and the Custodian create, assign and track tasks that represent specific corrective or preventive measures related to the incident.
UC_12 – Manage vulnerabilities and security findings	UC_12 covers vulnerability management in SOCaaS: from receiving the outputs of vulnerability scanners, through their assignment to specific assets (UC_11) and risk-based prioritisation, to the planning and tracking of corrective measures. The aim is to have a comprehensive overview of where we have gaps, what is most important to fix and within what time.
UC_05c – Close and link an incident	After the corrective measures have been carried out, the Analyst (or the Manager) decides on the final status of the incident, documents the conclusion and links the incident to other related cases.
UC_07 – Automate incident response	UC_07 addresses the situation where an incident has already been detected and is being handled (UC_05), but instead of purely manual steps, automated or semi-automated playbooks are used. On the basis of the incident type and TTPs, the system proposes suitable playbooks, orchestrates technical interventions (blocking, isolation, account disabling, ticketing) and records everything for audit purposes.

Use case	UC_05a – Analyse an incident
Goal	The Analyst examines the incident in detail – reviews the event timeline, logs and IOCs, supplements evidence and creates a comprehensive picture of the cause and scope of the incident.

Precondition	The incident has been created and assigned to the Analyst.
Precondition	Basic logs and events from the monitoring system are available for the incident.

Postcondition	The incident contains supplementary evidence (logs, attachments, IOCs).
Postcondition	The preliminary cause of the incident and the scope of impact are described.

Basic Path	Basic Path

Step	**Activity**
1	The Analyst opens the incident detail.
2	Displays the timeline of related events and logs.
3	Identifies suspicious events and indicators of compromise.
4	Downloads or requests additional logs/attachments and attaches them to the incident.
5	On the basis of the collected data, describes the preliminary cause and scope of the incident.
6	Updates the incident priority if necessary.

ALT_01 – Unavailable or incomplete logs	Alternate

Step	**Activity**
1	The Analyst finds that logs are incomplete or missing; creates a task in the incident for the Custodian to remedy this (completing logging, export from the source system) and continues working with what is available.

ALT_02 – The incident turns out to be a false positive	Alternate

Step	**Activity**
1	After analysis, the Analyst determines that it is not a real incident; marks the incident as “false positive”, briefly documents the reason and UC_05 ends.

Linked functional requirements	
FREQ_134 - Ability to record times	Tracking and recording of time data for the entire ticket lifecycle. Ability to record times (creation, update, escalation, closure)
FREQ_133 - Adding attachments	Ability to add attachments such as logs, screenshots and other evidence. Adding attachments (logs, evidence, screenshots, tool outputs)
UC_05a – Analyse an incident	The Analyst examines the incident in detail – reviews the event timeline, logs and IOCs, supplements evidence and creates a comprehensive picture of the cause and scope of the incident.

Use case	UC_05b – Manage incident tasks and measures

Goal	The Analyst and the Custodian create, assign and track tasks that represent specific corrective or preventive measures related to the incident.

Precondition	A basic analysis exists for the incident (UC_05a).
Precondition	It is clear which systems or services the incident affects.
Postcondition	The incident has a list of tasks created with assigned responsible persons and deadlines.
Postcondition	The status of individual tasks reflects the actual progress of handling (planned, in progress, done).

Basic Path	Basic Path

Step	**Activity**
1	On the basis of the analysis, the Analyst proposes specific corrective measures.
2	In the incident, creates tasks and assigns them to specific Custodians or teams.
3	Sets a priority and deadline for each task.
4	The Custodians continuously update the status of tasks according to the implementation of measures.
5	The system records the times of creation, start and completion of tasks.
6	The Analyst monitors the status of tasks and, as needed, sends reminders about or escalates delayed tasks.
7	After the key tasks have been completed, the Analyst updates the incident status.

ALT_01 – The task has a high impact on operations	Alternate

Step	**Activity**
1	The Custodian finds that the proposed measure may cause an outage; marks the task as “requires approval” and the Manager approves or rejects it with a comment.

ALT_02 – The task cannot be implemented by the planned deadline	Alternate

Step	**Activity**
1	The Custodian updates the deadline and reason in the task; the Analyst adjusts the incident priority or informs the Manager as needed.

Linked functional requirements	
FREQ_143 - Ability to perform retrospective review and forensic analysis	Ability to retrospectively analyse the course and history of incidents. Ability to perform retrospective review and forensic analysis

FREQ_144 - Creation and management of cases with tasks and details	Ability to create incidents as cases with tasks, a description and attributes. Creation and management of cases with tasks and details
UC_05b – Manage incident tasks and measures	The Analyst and the Custodian create, assign and track tasks that represent specific corrective or preventive measures related to the incident.

Use case	UC_05c – Close and link an incident
Goal	After the corrective measures have been carried out, the Analyst (or the Manager) decides on the final status of the incident, documents the conclusion and links the incident to other related cases.

Precondition	The key tasks of the incident have been completed or a decision has been taken to accept the remaining risk.
Precondition	Monitoring shows no persisting indicators of attack, or their occurrence has been explained.
Postcondition	The incident is in one of the final statuses (closed, false positive, risk accepted, part of a campaign).
Postcondition	The incident is linked to other relevant incidents/cases or campaigns, if any exist.

Basic Path	Basic Path

Step	**Activity**
1	The Analyst checks that all important tasks have been completed and the systems are stable.
2	Verifies in monitoring that no new related events are appearing.
3	If the incident is part of a wider campaign, searches for and links related incidents.
4	Fills in the final summary: cause, impact, measures taken.
5	Sets the final status of the incident (e.g. “closed”).
6	For critical incidents, requests the Manager's approval of the closure.
7	The incident is moved to the archive and is ready for the “lessons learned” process.

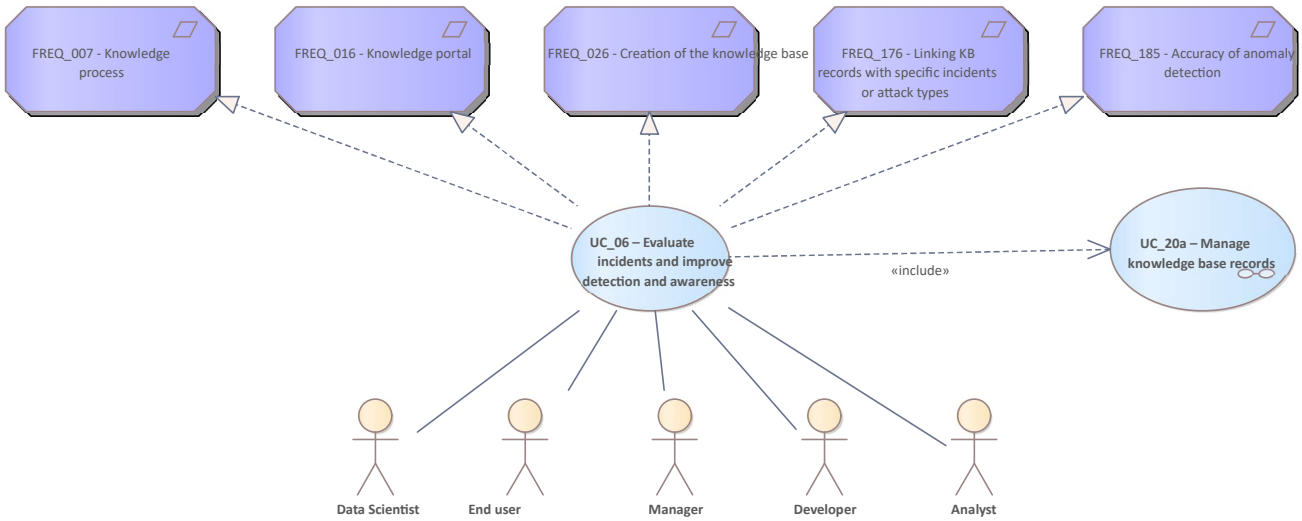
ALT_01 – New related events identified during closure	Alternate

Step	**Activity**
1	During the check, the Analyst identifies new suspicious events; leaves the incident open, creates new tasks and returns to UC_05a/05b.

ALT_02 – The incident is part of an extensive campaign	Alternate

Step	Activity
1	The Analyst identifies that the incident belongs to a larger campaign; sets it as subordinate to the parent case and adjusts the final summary.
Linked functional requirements	
FREQ_146 - Linking of cases based on shared indicators	Automatic or manual identification of related incidents. Linking of cases based on shared indicators
UC_05c – Close and link an incident	After the corrective measures have been carried out, the Analyst (or the Manager) decides on the final status of the incident, documents the conclusion and links the incident to other related cases.

UC_06 – Evaluate incidents and improve detection and awareness



UC_06 – Evaluate incidents and improve detection and awareness

Use case	UC_06 – Evaluate incidents and improve detection and awareness
Goal	After an incident has been resolved and closed (UC_05), this UC ensures that both the system and people learn from the incident: - the incident is evaluated (post-incident / post-mortem), - the findings are reflected in adjustments to detection rules and ML models, - training / guidance for employees is prepared or updated.

Precondition	The incident is closed
Precondition	Data on the incident are available

Precondition	At least a basic “lessons learned” process exists
Postcondition	The incident has a completed “lessons learned” record
Postcondition	Adjustments to detection/ML are proposed or implemented
Postcondition	Training / guidance for users is prepared or updated

Basic Path	Basic Path

Step	**Activity**
1	Selection of incidents for evaluation
2	Preparation of materials for evaluation
3	Post-incident meeting / evaluation
4	Recording of lessons learned and recommendations
5	Proposal of adjustments to detection and ML models
6	The team evaluates how the incident affects detection accuracy (e.g. the number of false positives/false negatives, the suitability of thresholds, the behaviour of the model in the case of anomalies) and incorporates these findings into the proposal of adjustments.
7	The knowledge base record is linked to the specific incident (or attack type) so that it can be found quickly in similar cases.
8	The outputs are stored in the knowledge portal / knowledge base and communicated to the relevant people.
9	Approval and planning of the deployment of changes
10	Preparation of training / awareness materials
11	Distribution of knowledge
12	Conclusion of the UC

ALT_01 – Insufficient data for a quality evaluation	Alternate

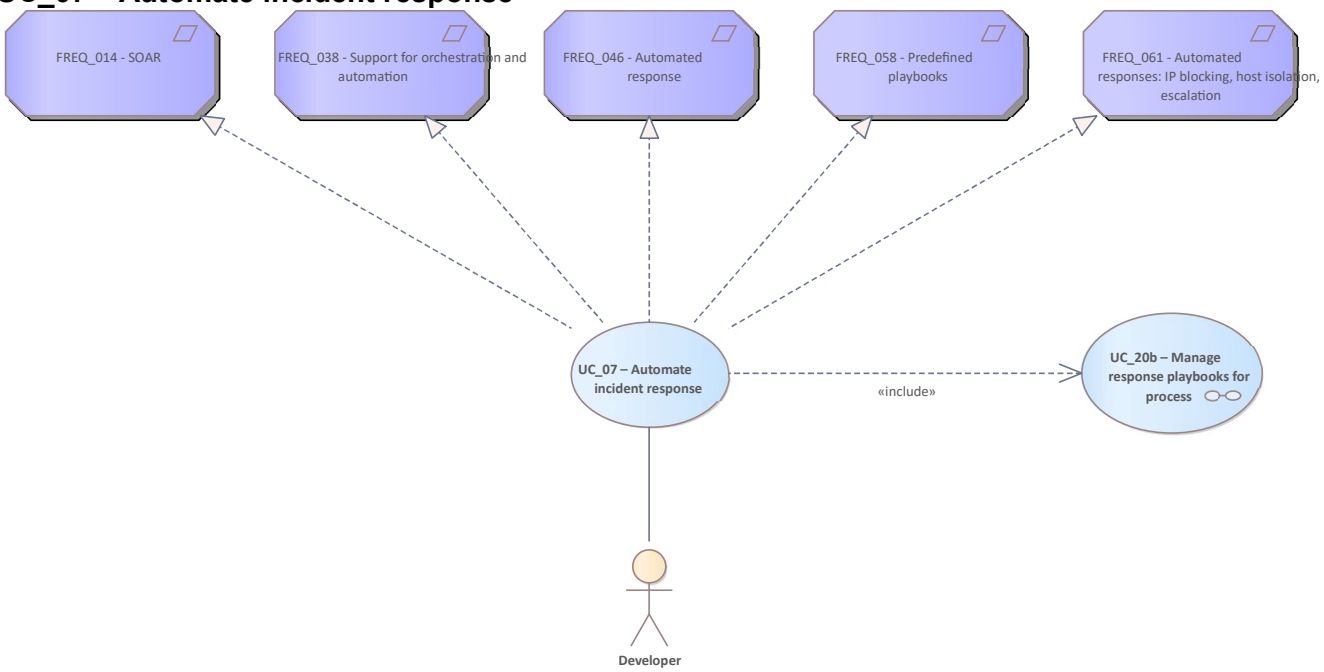
Step	**Activity**
1	The analysis reveals that not all the necessary logs or historical data are available; the team carries out a basic lessons learned exercise but at the same time creates tasks to improve logging and data collection in the future.

ALT_02 – The proposal of adjustments to detection/ML is rejected	Alternate

Step	**Activity**
1	The Data Scientist or Developer prepares a proposal of changes, but the Manager does not approve it on the basis of risks or costs; the recommendations are stored in the knowledge base and the changes may be reconsidered later.
ALT_03 – An identified need for targeted employee training	Alternate
Step	**Activity**
1	The evaluation shows that the main weakness was human error or lack of knowledge; the Manager decides on the preparation or modification of a specific training course, which is assigned to a selected group of users.
Linked functional requirements	
FREQ_026 - Creation of a knowledge base	The solution must ensure the creation of a knowledge base. Each incident or new type of threat will be documented in the form of: • incident summary + lessons learned • an entry in the knowledge portal • categorisation according to MITRE ATT&CK and dates
UC_20a – Manage knowledge base records	UC_20a addresses in detail the life cycle of KB articles – creation, editing, structure, tagging and linkage to incidents.
FREQ_007 - Knowledge management process	Ensure a knowledge management process – it ensures the maintenance of the knowledge base of playbooks, documentation and recommended responses.
FREQ_176 - Linking KB records to specific incidents or attack types	The ability to link knowledge articles to specific attack types and incidents in other tools. Linking KB records to specific incidents or attack types
UC_20 – Manage the knowledge base and playbook library	UC_20 ensures that the SOC has, centrally and in a controlled manner: - a knowledge base (KB) – articles, guides, lessons learned, procedures, - a playbook library – both manual and automated response procedures,
	- a process for the approval, versioning and publication of this content.
FREQ_185 - Accuracy of anomaly detection	The solution must ensure the evaluation of the accuracy of anomaly detection (Precision, Recall, F1). How accurately the algorithm detects anomalies regardless of class. Evaluation of overall accuracy using metrics such as Precision, Recall, F1-score or AUC-ROC.
FREQ_016 - Knowledge portal	The solution must provide a knowledge portal that will support response processes by sharing playbooks, internal policies and guides.

Funded by the European Union NextGenerationEU
RECOVERY AND RESILIENCE PLAN
GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC

UC_07 – Automate incident response



UC_07 – Automate incident response

Use case	UC_07 – Automate incident response
Goal	UC_07 addresses the situation where an incident has already been detected and is being handled (UC_05), but instead of purely manual steps, automated or semi-automated playbooks are used. Based on the incident type and TTPs, the system proposes suitable playbooks, orchestrates technical interventions (blocking, isolation, account disabling, ticketing) and records everything for audit purposes.

Precondition	The incident is in the “in progress” state
Precondition	Playbooks are available for the given incident type
Precondition	Integrations with the target systems are functional
Postcondition	The selected playbooks have been executed (auto / semi-auto)
Postcondition	All automated interventions are recorded in the incident for audit purposes
Postcondition	The incident is in a state that reflects the result of the automated response

Basic Path	Basic Path

Step	**Activity**
1	The Analyst opens an incident in the “in progress” state and the system displays a list of proposed playbooks according to the incident type and the techniques used.
2	The Analyst selects a suitable playbook and chooses the execution mode (e.g. simulation, semi-automatic mode, fully automatic for low risk).
3	The system displays an overview of the planned playbook steps (e.g. IP blocking, endpoint isolation, account disabling, ticket creation).
4	In semi-automatic mode, before executing each step the system requests the Analyst to confirm or skip the step.
5	The system gradually executes the approved steps through integrations with the target systems and continuously records the status (success/failure).
6	After the playbook finishes, the system summarises the result: which actions were completed successfully, what failed and where manual intervention is needed.
7	The results of the executed actions and the playbook timeline are stored in the incident as part of the resolution history.
8	Based on the result, the Analyst updates the incident status (e.g. containment completed, awaiting monitoring, awaiting further tasks according to UC_05).
9	The incident continues in UC_05 (detailed analysis, addition of tasks, closure).

ALT_01 – No suitable playbook was found	Alternate

Step	**Activity**
1	The system is unable to assign any relevant playbook to the given incident; the Analyst continues to handle the incident manually according to UC_05 and marks the incident as a candidate for the creation of a new playbook in the future.

ALT_02 – A critical playbook step fails	Alternate

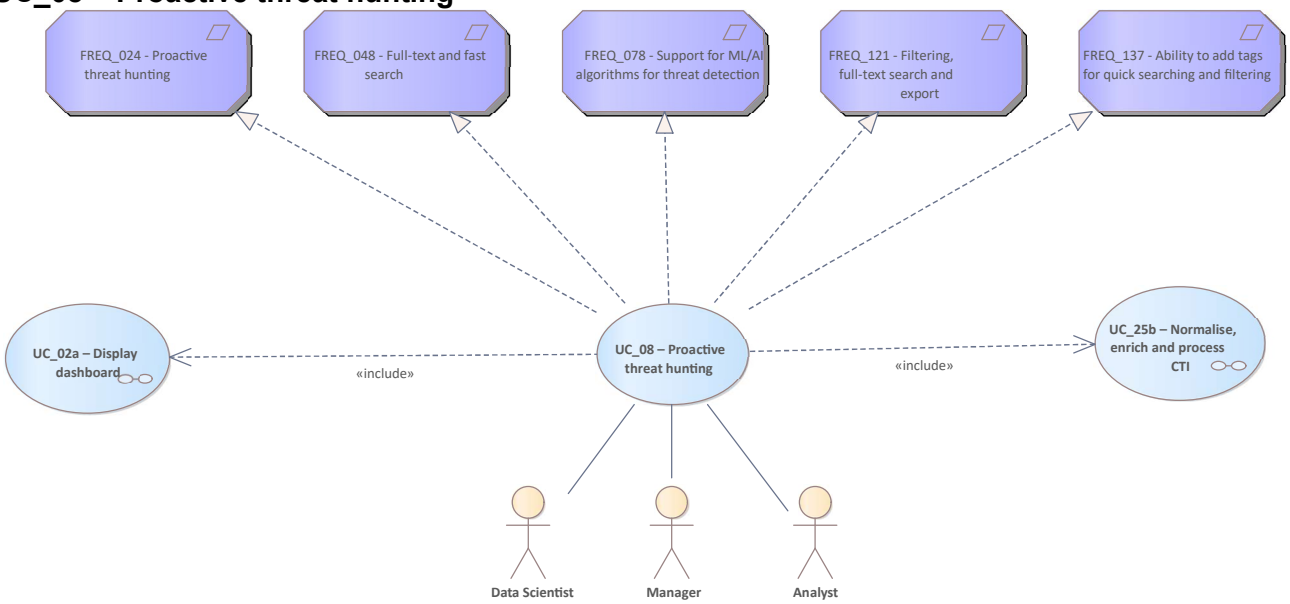
Step	**Activity**
1	An error occurs during the execution of an automated step (e.g. blocking on the firewall); the system marks the step as unsuccessful, records the error details and the Analyst decides whether to repeat the step, replace it with another measure or assign a manual task to the Administrator.

ALT_03 – The playbook is run only in simulation mode	Alternate

Step	**Activity**
1	Based on the Manager’s policy, selected playbooks are permitted only in simulation mode; the Analyst runs the simulation, the system shows which actions would be executed and what the impact would be, and only then does the Analyst decide which of them will be carried out manually or by means of partial automation.

Linked functional requirements	
FREQ_014 - SOAR	The solution must provide SOAR, which will serve to automate response scenarios, process alerts into cases and analyse them.
UC_20 – Manage the knowledge base and playbook library	UC_20 ensures that the SOC has, centrally and in a controlled manner: - a knowledge base (KB) – articles, guides, lessons learned, procedures, - a playbook library – both manual and automated response procedures, - a process for the approval, versioning and publication of this content.
UC_20b – Manage incident response playbooks	UC_20b addresses the design, modification and maintenance of the playbooks that the SOC uses in automated or semi-automated response (UC_07).
FREQ_046 - Automated responses	The ability to trigger a response (blocking, isolation, notification) automatically. Automated responses (e.g. IP blocking, SOAR integration)
FREQ_038 - Support for orchestration and automation	The solution should support tools such as Heat (OpenStack orchestration), Terraform (infrastructure as code), Ansible (automation) or Kubernetes Operator for the efficient management and deployment of complex infrastructure scenarios. This capability will enable rapid scaling of services, automated provisioning of environments, repeatable testing and recovery from outages. In a SOC environment, where requirements and network topology change frequently, orchestration is essential for
	ensuring flexibility, sustainability and operational efficiency.
FREQ_061 - Automated responses: IP blocking, host isolation, escalation	The ability to automatically perform responses such as blocking, isolation or notification. Automated responses: IP blocking, host isolation, escalation
FREQ_058 - Predefined playbooks	Availability of basic response scenarios and automations prepared in advance in the tool. Predefined playbooks
UC_07 – Automate incident response	UC_07 addresses the situation where an incident has already been detected and is being handled (UC_05), but instead of purely manual steps, automated or semi-automated playbooks are used. On the basis of the incident type and TTPs, the system proposes suitable playbooks, orchestrates technical interventions (blocking, isolation, account disabling, ticketing) and records everything for audit purposes.

UC_08 – Proactive threat hunting



UC_08 – Proactive threat hunting

Use case	UC_08 – Proactive threat hunting
Goal	UC_08 addresses proactive threat hunting in historical and current data beyond regular alerts. The Analyst purposefully searches for suspicious behaviour patterns (TTPs, IOCs) that have not yet triggered an incident or alert, saves hunting scenarios and can create new incidents directly from the findings.

Precondition	Historical and current logs/telemetry are available
Precondition	At least a basic set of hunting queries or scenarios exists
Precondition	The Analyst has access to dashboards and tools for ad hoc queries
Postcondition	At least one hunting run has taken place
Postcondition	The findings are documented and linked to incidents or the knowledge base

Postcondition	Hunting scenarios are saved for repeated use

Basic Path	Basic Path

Step	Activity
1	Selection of the hunting target
2	Opening of the hunting workspace
3	Creation or selection of a query / scenario
4	Execution of a hunting run over the data
5	Analysis of the findings
6	Conversion of findings into incidents or enrichment of existing incidents
7	Saving of the hunting scenario
8	Documentation of the results
9	Planning of repeated hunting

ALT_01 – Hunting reveals nothing significant	Alternate

Step	Activity
1	The hunting query returns no suspicious findings, or all the findings are legitimate after review; the Analyst saves the hunting record as “no significant findings” and may adjust the query for future use.

ALT_02 – Hunting reveals a large-scale campaign	Alternate

Step	Activity
1	Hunting identifies several related findings that indicate a more extensive attack; the Analyst creates a parent incident/campaign, assigns all the related findings and incidents to it and informs the Manager of the need for increased attention.

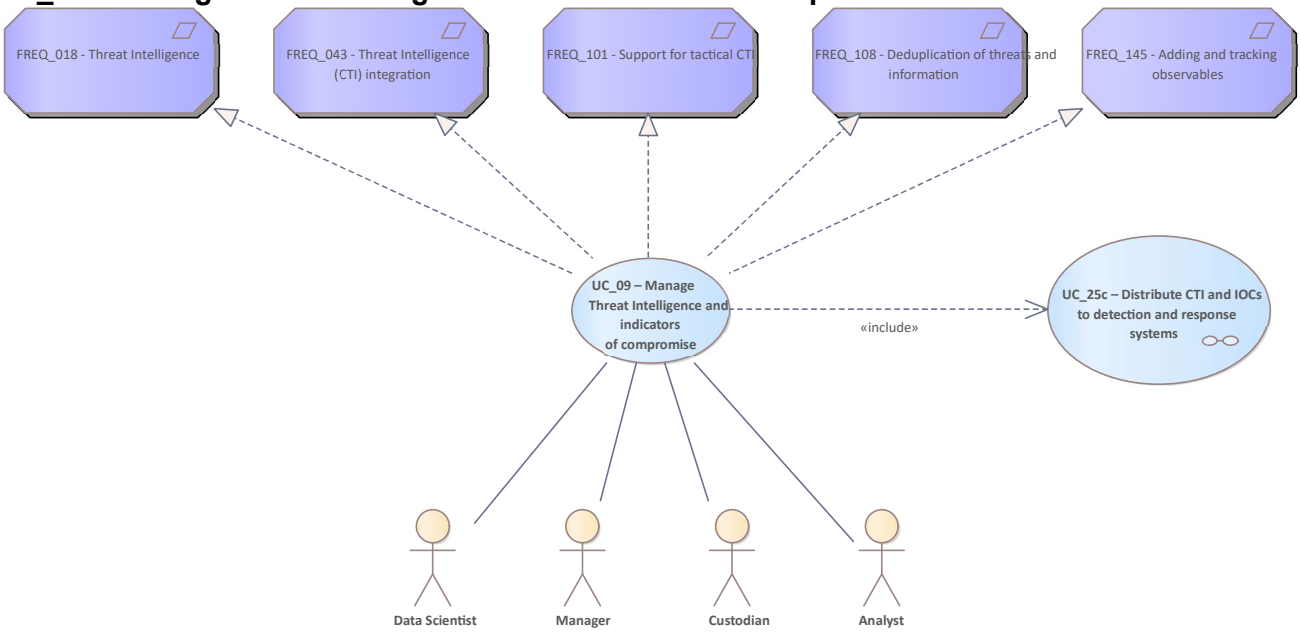
ALT_03 – The query is too demanding or slow	Alternate

Step	Activity
1	The hunting query that has been run processes a large volume of data and the systems evaluate it as too demanding; the system alerts the Analyst, stops or limits the query and offers the option to adjust the time range or granularity.

Linked functional requirements	
FREQ_121 - Filtering, full-text search and export capability	Efficient work with data – searching by various parameters and exporting data. Filtering, full-text search and export capability
UC_02a – Display a dashboard	The user opens a dashboard that visualises security events, metrics and system states in real time.

FREQ_024 - Proactive threat hunting	The solution must ensure proactive threat hunting (Threat Hunting). Advanced queries will be used, often supplemented with: • CTI data • manual analysis in the case of suspicious entities • the results are documented as playbooks
UC_25c – Distribute CTI and IOCs to detection and response systems	UC_25c addresses the push of CTI to the rest of the ecosystem: - forwarding of IOCs to SIEM/IDS/EDR/firewalls, - use of CTI in detection, playbooks and incidents,
FREQ_137 - Ability to add tags for quick searching and filtering	Use of tags to facilitate searching and categorisation. Ability to add tags for quick searching and filtering
FREQ_078 - Support for ML/AI algorithms for threat detection	Use of artificial intelligence algorithms to detect anomalies and threats. Support for ML/AI algorithms for threat detection
FREQ_048 - Full-text and fast search	The ability to search logs using operators, tags and filters. Full-text and fast search
UC_25b – Normalise, enrich and process CTI	UC_25b addresses the CTI processing pipeline: - normalisation of various formats into a unified model, - enrichment of IOCs and artefacts from external services, - a workflow for automatic/semi-automatic CTI processing.

UC_09 – Manage Threat Intelligence and indicators of compromise



UC_09 – Manage Threat Intelligence and indicators of compromise

Use case	UC_09 – Manage Threat Intelligence and indicators of compromise
Goal	UC_09 addresses the management of Threat Intelligence: receiving and integrating CTI feeds, working with indicators of compromise (IOCs), deduplicating them and tracking them over time.

	The aim is for the SOC to be able to manage threats centrally, enrich incidents and logs with TI information and keep observables in order.

Precondition	TI sources are available
Precondition	Integrations with TI sources are configured
Precondition	The system has a repository for TI and IOCs
Postcondition	TI data are loaded, normalised and stored
Postcondition	Redundant or conflicting IOCs are deduplicated or merged
Postcondition	IOCs/observables are linked to incidents, cases or assets and have a traceable history

Basic Path	Basic Path

Step	**Activity**
1	Loading of TI from external sources
2	Normalisation and storage of TI
3	Deduplication and merging of IOCs
4	Management of the IOC life cycleManagement of the IOC life cycle
5	Linking of IOCs to incidents and cases
6	Publication of TI for other parts of the system
7	Monitoring of TI quality

ALT_01 – Failure or unavailability of a TI source	Alternate

Step	**Activity**
1	When loading TI, the source is temporarily unavailable or returns errors; the system records an integration incident, uses the last valid set of IOCs and notifies the Administrator of the problem.

ALT_02 – Conflicting information about the same threat	Alternate

Step	**Activity**

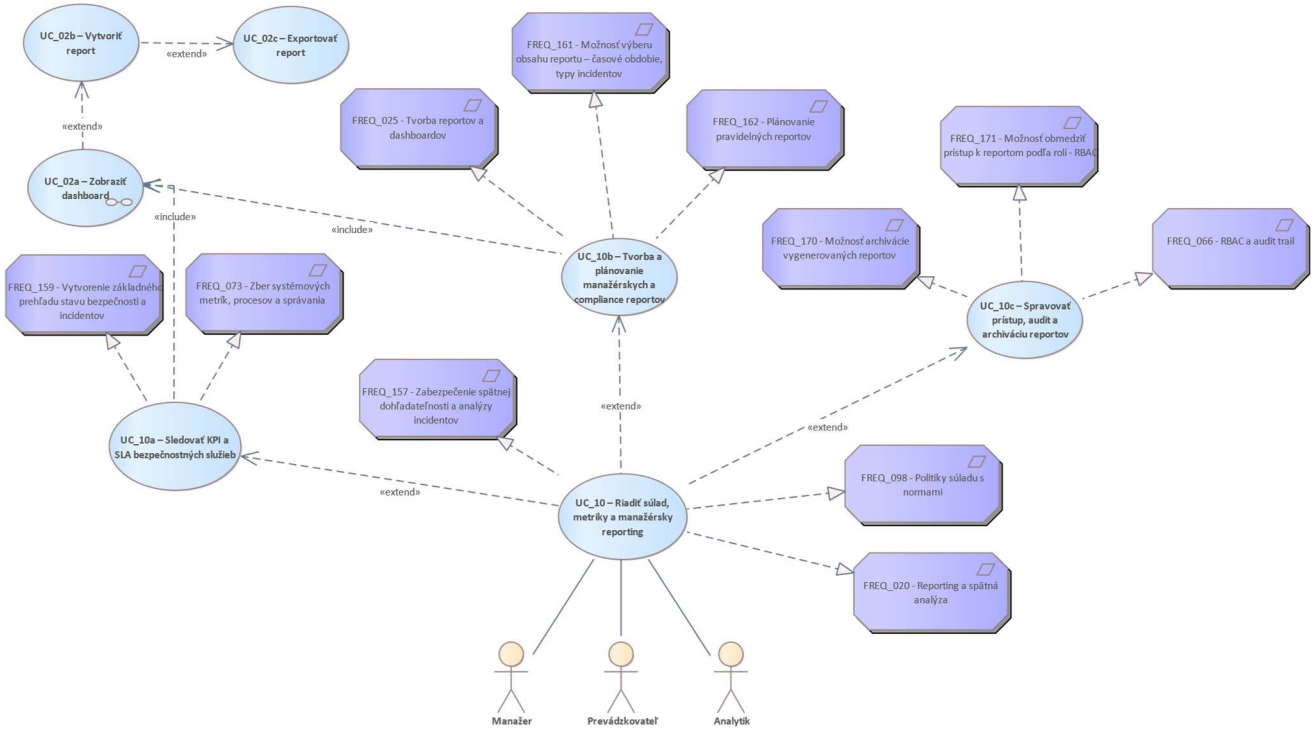
1	Several TI sources provide different or contradictory information about the same IOC; the system retains all variants together with their sources, marks the IOC as conflicting and leaves it to the Analyst to select the more trustworthy source.

ALT_03 – The IOC is assessed as outdated or unreliable	Alternate

Step	**Activity**
1	The Analyst finds that the given IOC is no longer relevant or has been marked as an error; they change its status to inactive, add a comment and the system ensures that it is no longer used in new detections.

Linked functional requirements	
FREQ_101 - Support for tactical CTI	Working with indicators of compromise (IOCs) and their recording, processing and distribution. Support for tactical CTI (IOCs, IPs, hashes, URLs, domains)
FREQ_108 - Deduplication of threats and information	Removal of redundant IOCs and indicators. Deduplication of threats and information
FREQ_145 - Adding and tracking observables	The ability to track IOCs as part of a case. Adding and tracking observables (IPs, domains, URLs, hash values)
FREQ_018 - Threat Intelligence	The solution must ensure Threat Intelligence. CTI (Cyber Threat Intelligence) processes enable the enrichment of incidents with known threats, TTPs and IOCs
UC_25c – Distribute CTI and IOCs to detection and response systems	UC_25c addresses the push of CTI to the rest of the ecosystem: - forwarding of IOCs to SIEM/IDS/EDR/firewalls, - use of CTI in detection, playbooks and incidents,
FREQ_043 - Threat Intelligence (CTI) integration	Integration with threat sources (IOCs, TTPs), automatic data enrichment. Threat Intelligence (CTI) integration

UC_10 – Manage compliance, metrics and management reporting



UC_10 – Manage compliance, metrics and management reporting

Use case	UC_10 – Manage compliance, metrics and management reporting
Goal	UC_10 addresses the “top floor” of SOCaaS – so that management and auditors have an overview of: - how well the SOC is performing (KPIs, SLAs, incident trends), - whether the system complies with standards/regulations, - whether all important matters can be traced retrospectively through reports, logs and audits.

Precondition	The system collects data on incidents, responses, playbooks and TI over the long term
Precondition	Defined metrics/KPIs and SLAs exist
Precondition	At least a basic compliance policy (standards, internal directives) is in place
Postcondition	The Manager and the Auditor have clear metrics and reports on the state of security and SOC services at their disposal
Postcondition	Problems identified in KPIs/SLAs or compliance are recorded and have recommendations/measures assigned

Postcondition	The reports created are archived and traceable for future audits

Basic Path	Basic Path

Step	**Activity**
1	The Manager opens the SOC management overview
2	Selection of the area of interest
3	Display of aggregated metrics and trends
4	Identification of deviations or problems
5	Detailed drill-down
6	Submission of a request for a report / analysis
7	Recording of findings and recommendations
8	Linking to measures

ALT_01 – Unavailable or incomplete metrics	Alternate

Step	**Activity**
1	When displaying the management overview, the system indicates that some metrics cannot be calculated reliably (missing data, incomplete logs); the Manager receives information on which parts are incomplete and can request remediation.

ALT_02 – An identified breach of an SLA or standard	Alternate

Step	**Activity**
1	The metrics and reports show that SLAs or the requirements of a standard have been breached; the Manager registers the finding as an official finding and initiates internal measures (e.g. a service quality incident, corrective measures).

Linked functional requirements	
UC_10a – Monitor KPIs and SLAs of security services	UC_10a extends UC_10 with detailed monitoring of KPIs, SLAs and the overall state of security. The Manager, Auditor or Operator views trend charts, overviews by team/customer and the overall “health status” of SOCaaS in the dashboards.
FREQ_098 - Policies for compliance with standards	The ability to assess vulnerabilities in relation to standards. Policies for compliance with standards (e.g. GDPR, ISO, CIS, BSI...)
FREQ_157 - Ensuring retrospective traceability and analysis of incidents	It makes it possible to obtain the complete timeline of an incident. Ensuring retrospective traceability and analysis of incidents

FREQ_020 - Reporting and retrospective analysis	The solution must ensure reporting and retrospective analysis.
UC_10b – Creation and scheduling of management and compliance reports	UC_10b addresses practical work with reports for management and audits: creation, content configuration, scheduling of regular reports and their one-off generation.
UC_10c – Manage access, audit and archiving of reports	UC_10c addresses the security and audit aspects of reporting: who sees what, who made what available, whether reports are archived and whether it is possible to reconstruct who had what data at their disposal.

Use case	UC_10a – Monitor KPIs and SLAs of security services
Goal	UC_10a extends UC_10 with detailed monitoring of KPIs, SLAs and the overall state of security. The Manager, Auditor or Operator views trend charts, overviews by team/customer and the overall “health status” of SOCaaS in the dashboards.

Precondition	From UC_10 – the management overview is available.
Precondition	KPIs and SLAs are defined (e.g. MTTR, MTTD, % of incidents within SLA).
Postcondition	Identified deviations in KPIs/SLAs are documented.
Postcondition	The Manager has an overview of which areas/teams/customers require attention.

Basic Path	Basic Path

Step	**Activity**
1	In UC_10, the Manager selects the “KPIs and SLAs” area.
2	The system displays a dashboard with KPIs and SLAs (include UC_02a – Display dashboard).
3	The Manager filters the metrics by period, team, incident type or customer.
4	The system displays trend charts, averages, SLA compliance percentages and comparisons with target values.
5	The Manager identifies problems (e.g. a specific team has long response times) and flags them for detailed analysis or corrective measures.

ALT_10a_01 – KPIs or SLAs have not yet been defined	Alternate

Step	**Activity**
1	When an attempt is made to display KPIs/SLAs, the system indicates that they are not configured; the Manager receives a recommendation to define the metrics and the associated data sources.

Linked functional requirements

FREQ_073 - Collection of system metrics, processes and behaviour	Monitoring of system behaviour and metrics such as CPU, memory, services and processes. Collection of system metrics, processes and behaviour
FREQ_159 - Creation of a basic overview of the state of security and incidents	The ability to create summaries of the state of security, incidents and system activity. Creation of a basic overview of the state of security and incidents
UC_02a – Display a dashboard	The user opens a dashboard that visualises security events, metrics and system states in real time.
UC_10a – Monitor KPIs and SLAs of security services	UC_10a extends UC_10 with detailed monitoring of KPIs, SLAs and the overall state of security. The Manager, Auditor or Operator views trend charts, overviews by team/customer and the overall "health status" of SOCaaS in the dashboards.

Use case	UC_10b – Creation and scheduling of management and compliance reports
Goal	UC_10b addresses practical work with reports for management and audits: creation, content configuration, scheduling of regular reports and their one-off generation.

Precondition	The system has aggregated data from the other UCs (incidents, TI, playbooks, KPIs) at its disposal.
Precondition	The user has sufficient permissions to create reports.
Postcondition	Report templates for management/the regulator are created.
Postcondition	Both regular and ad hoc reports are set up as needed.

Basic Path	Basic Path

Step	**Activity**
1	The Manager or Auditor opens the "Reports" section within UC_10.
2	They select the report type (management overview, incident overview, compliance report).
3	The system makes it possible to configure the report content – selection of metrics, time period and filters.
4	The user chooses whether it is a one-off report or a regular one (daily/weekly/monthly).
5	When generating the report, the system uses UC_02b (Create report) and UC_02c (Export report) – PDF/CSV/…
6	The generated report is saved and is available in the report archive.

ALT_01 – Incomplete report due to missing data	Alternate

Step	**Activity**

1	When generating the report, the system warns that some parts will be incomplete; the user may decide to generate the report anyway or to adjust the scope.

Linked functional requirements	
FREQ_025 - Creation of reports and dashboards	The solution must ensure the creation of reports and dashboards. The following will be generated regularly for management and customers: • incident overviews • visualisations from metrics and ticketing • exports for audit and SLA reporting
FREQ_162 - Scheduling of regular reports	The ability to set up report generation based on a schedule. Scheduling of regular (e.g. daily, weekly) reports
UC_02a – Display a dashboard	The user opens a dashboard that visualises security events, metrics and system states in real time.
FREQ_161 - Ability to select report content – time period, incident types	The ability to filter report content by time, type or source data. Ability to select report content – time period, incident types
UC_10b – Creation and scheduling of management and compliance reports	UC_10b addresses practical work with reports for management and audits: creation, content configuration, scheduling of regular reports and their one-off generation.

Use case	UC_10c – Manage access, audit and archiving of reports
Goal	UC_10c addresses the security and audit aspects of reporting: who sees what, who made what available, whether reports are archived and whether it is possible to reconstruct who had what data at their disposal.

Precondition	Generated reports exist in the system (UC_10b).
Precondition	Roles and access policies (RBAC) are defined.
Postcondition	Access to reports is restricted according to roles and responsibilities.
Postcondition	An audit trail exists of who worked with the reports and when.
Postcondition	Important reports are archived for the needs of future audits.

Basic Path	Basic Path

Step	**Activity**
1	The Manager or Auditor opens the management of reports and access.
2	For individual report types, they set which roles or specific users have access to them.
3	The system applies RBAC when reports are displayed and downloaded.
4	On each display, download or change of access settings, the system writes an event to the audit trail.

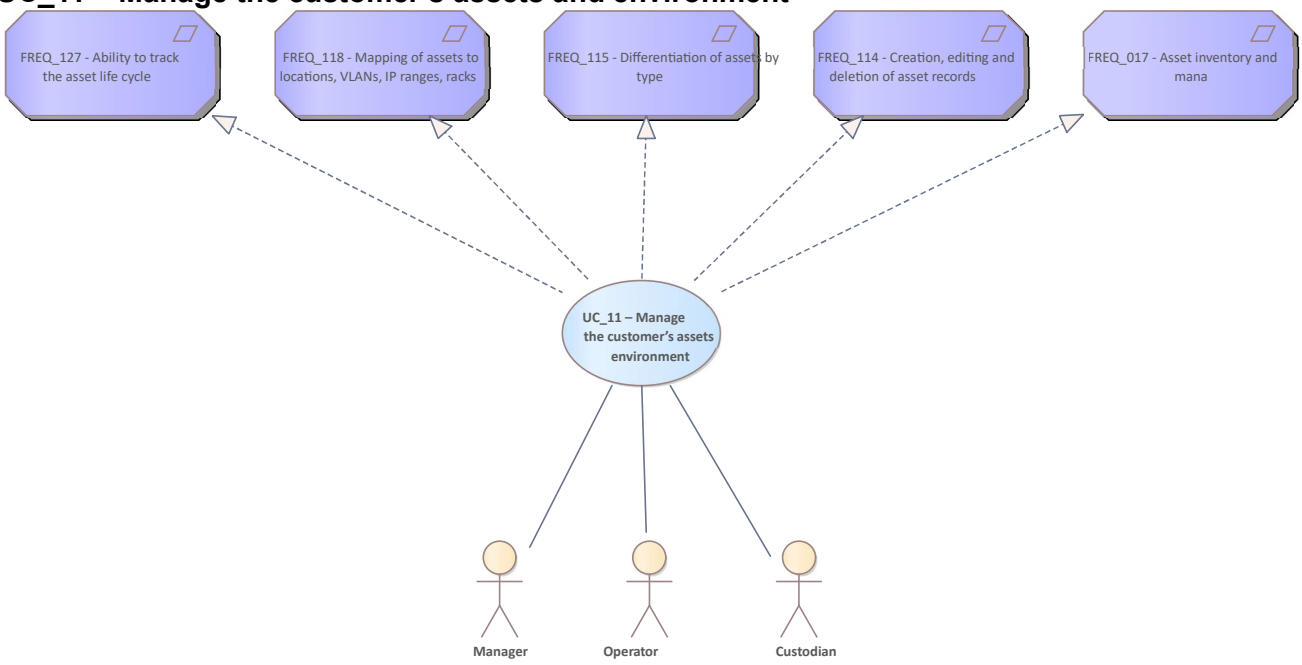
5	Older reports are archived automatically or manually so that they are available for audit but do not overload the regular interface.

ALT_01 – Attempt to access a report without permission	Alternate

Step	**Activity**
1	A user attempts to open a report for which they do not have the role; the system denies access, records the event in the audit trail and may notify the Manager or Auditor.

Linked functional requirements	
FREQ_066 - RBAC and audit trail	Role-based access control and logging of changes in the system. RBAC and audit trail
FREQ_170 - Ability to archive generated reports	The ability to retain reports for retrospective analysis and audit. Ability to archive generated reports
FREQ_171 - Ability to restrict access to reports according to roles - RBAC	Access to reports controlled by roles and permissions. Ability to restrict access to reports according to roles (RBAC)
UC_10c – Manage access, audit and archiving of reports	UC_10c addresses the security and audit aspects of reporting: who sees what, who made what available, whether reports are archived and whether it is possible to reconstruct who had what data at their disposal.

UC_11 – Manage the customer's assets and environment



UC_11 – Manage the customer's assets and environment

Use case	UC_11 – Manage the customer's assets and environment
Goal	The aim of UC_11 is for SOCaaS to know what it is actually protecting – all the customer's relevant assets (servers, applications, networks, cloud services...) and their relationships. It makes it possible to record assets, differentiate their types, map them to locations/VLANs/IPs and track their life cycle, so that incidents, threats and vulnerabilities can be linked to specific parts of the environment.

Precondition	The customer has been onboarded in the system (UC_00) and has a basic profile created.

Precondition	The system has a module / section for asset recording.
Precondition	At least a basic list of systems/services that need to be included in the inventory exists.
Postcondition	The customer's assets are recorded in the system with their type, location and basic properties.
Postcondition	Each asset has a unique identifier and can be used in incidents, TI, vuln mgmt and reports.
Postcondition	Changes to assets (addition, modification, decommissioning) are recorded and the life cycle is traceable.

Basic Path	Basic Path

Step	**Activity**
1	The Operator opens the "Assets" module for a specific customer.
2	They create a new asset record and enter its basic data (name, description, owner, criticality).
3	They select the asset type (e.g. server, application, cloud service, network element) according to predefined categories.
4	The Administrator adds technical data – IP addresses, hostname, environment (prod/test/dev).
5	The system makes it possible to map the asset to a specific location, VLAN, IP range or rack.
6	The Operator sets the asset's life-cycle status (e.g. "in operation", "being deployed", "decommissioned").
7	The system saves the asset record and makes it available to the other modules (incident mgmt, TI, vuln mgmt, reporting).
8	On a subsequent change (upgrade, migration, decommissioning), the Administrator updates the asset record and the system records the change in the history.

ALT_11_01 – Import of assets from an external CMDB	Alternate

Step	**Activity**

1	Instead of creating records manually, the Operator chooses to import from an external CMDB or inventory tool; the system loads the list of assets, displays a preview and allows them to be saved in bulk or matched with existing records.

ALT_11_02 – A duplicate asset is detected	Alternate

Step	**Activity**
1	When saving a new asset, the system finds that a record with the same key identifiers (e.g. IP/hostname) already exists; the user is given the option either to update the existing record or to create a separate asset with a justification.

ALT_11_03 – Change of asset criticality	Alternate

Step	**Activity**
1	The Manager decides that the asset has a higher/lower criticality; they modify the value in the asset record and the system flags the related incidents and vulnerabilities whose priority may change as a result of this decision.

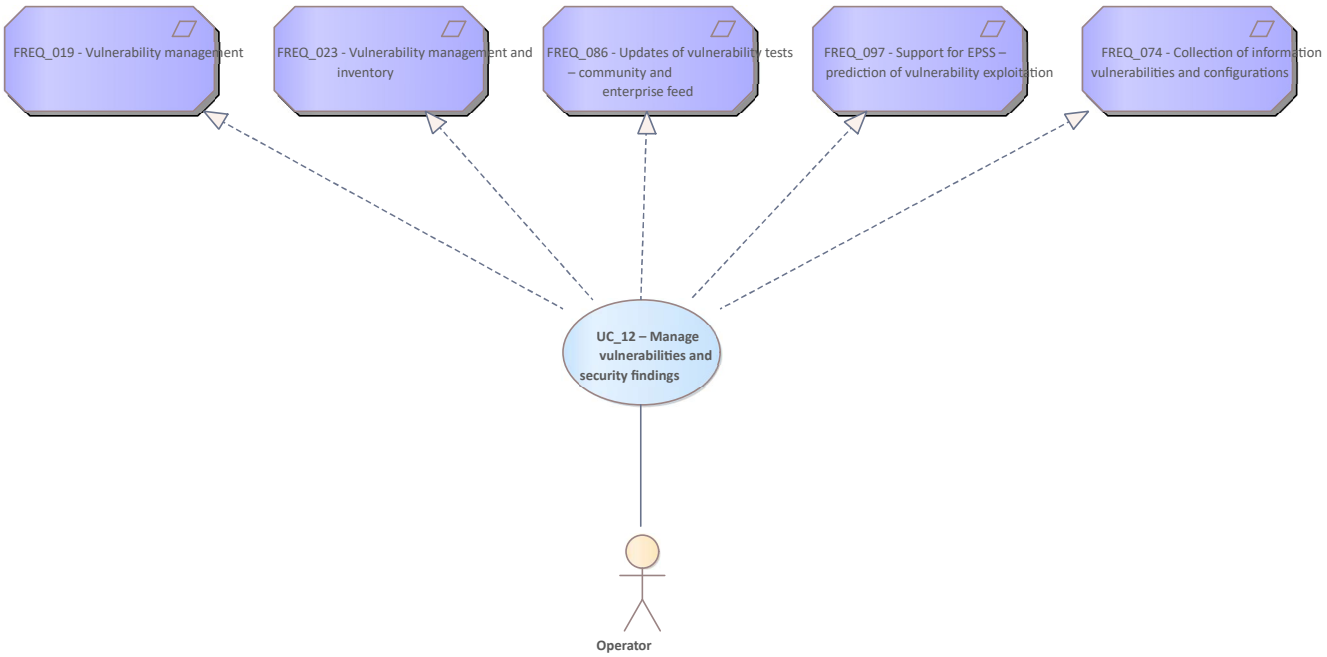
Linked functional requirements	
FREQ_127 - Ability to track the asset life cycle	Management of the individual phases of an asset's life – from inclusion to decommissioning. Ability to track the asset life cycle (deployment – decommissioning)
FREQ_115 - Differentiation of assets by type	The ability to differentiate asset types for better organisation and management. Differentiation of assets by type – hardware, software, virtual, cloud
FREQ_118 - Mapping of assets to locations, VLANs, IP ranges, racks	The ability to map the physical and logical location of assets in the network. Mapping of assets to locations, VLANs, IP ranges, racks
FREQ_017 - Asset inventory and management	The solution must ensure asset inventory and management. IT asset management is key to mapping the impacts of threats.
FREQ_114 - Creation, editing and deletion of asset records	The ability of the system to record, modify and delete information about individual assets. Creation, editing and deletion of asset records

UC_12 – Manage vulnerabilities and security findings

UC_12 covers vulnerability management in SOCaaS:

from receiving the outputs of vulnerability scanners, through their assignment to specific assets (UC_11) and risk-based prioritisation, to the planning and tracking of corrective measures.

The aim is to have a comprehensive overview of where we have gaps, what is most important to fix and within what time frame.



UC_12 – Manage vulnerabilities and security findings

Use case	UC_12 – Manage vulnerabilities and security findings
Goal	UC_12 covers vulnerability management in SOCaaS: from receiving the outputs of vulnerability scanners, through assigning them to specific assets (UC_11) and prioritising them according to risk, to planning and tracking remediation measures. The aim is to have a comprehensive overview of where we have gaps, what is most important to fix and within what time frame.

Precondition	Vulnerability scan results or other security findings exist
Precondition	Assets are recorded in the system
Precondition	A basic vulnerability prioritisation policy is defined
Postcondition	Vulnerabilities are recorded and assigned to specific assets.

Postcondition	Each vulnerability has a determined priority and a planned method of remediation.
Postcondition	The resolution status (not fixed / in progress / fixed / risk accepted) is tracked and traceable.

Basic Path	Basic Path
Step	**Activity**
1	Import or entry of vulnerabilities
2	Normalisation and grouping of findings
3	Assignment of vulnerabilities to assets
4	Calculation and setting of vulnerability priorityCalculation and setting of vulnerability priority
5	Creation of a remediation plan
6	Implementation of remediation measures
7	Re-scan or verification of status
8	Update of vulnerability status
9	Reporting and linking to incidents

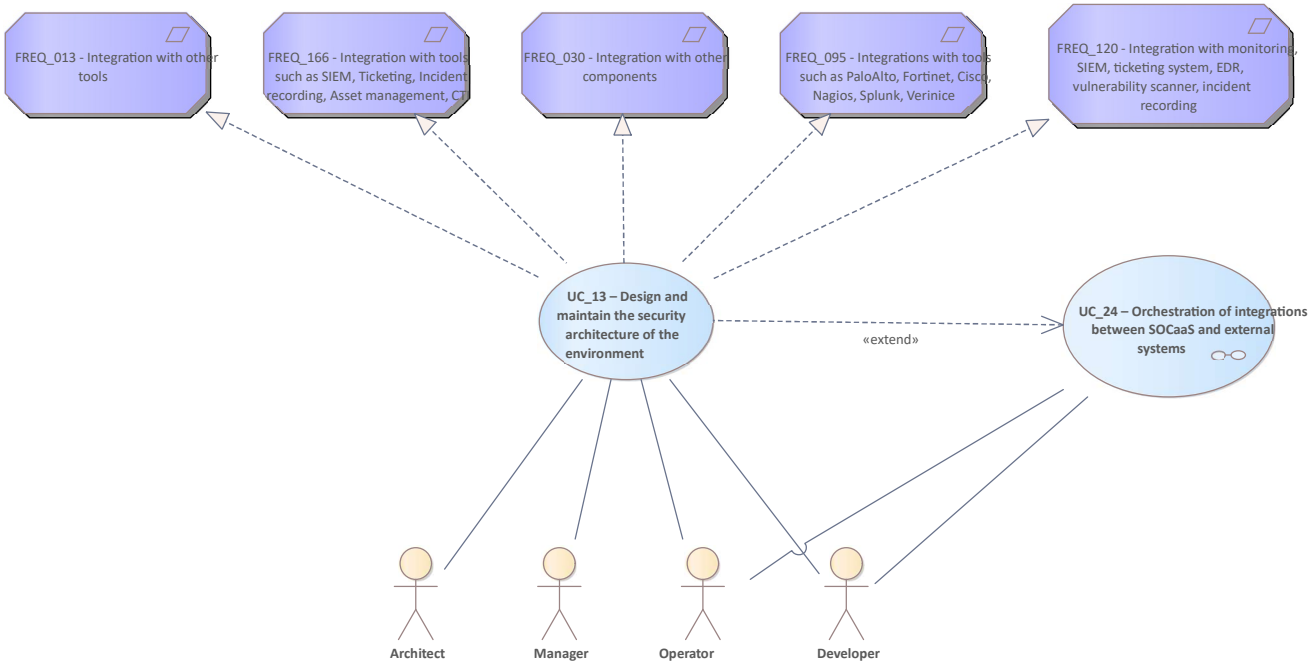
ALT_12_01 – The vulnerability cannot be fixed within the required time	Alternate
Step	**Activity**
1	The Administrator finds that the vulnerability cannot be fixed (technical limitations, no vendor patch exists, high impact on operations); after discussion, the Manager decides whether the risk is temporarily accepted or whether compensating measures are introduced (network segmentation, increased monitoring). The vulnerability status is set to “risk accepted” or “compensated”.

ALT_12_02 – A duplicate or obsolete finding is detected	Alternate
Step	**Activity**
1	When importing scan results, the system finds that the same vulnerability has already been recorded and in the meantime marked as fixed/irrelevant; the finding is marked as duplicate or obsolete, while a reference record is retained for audit.

ALT_12_03 – A vulnerability leads to the opening of an incident	Alternate
Step	**Activity**
1	The Analyst assesses a specific vulnerability (e.g. a critical, actively exploited one) as an acute threat; they create a security incident for it, which starts to be handled according to UC_05, while UC_12 is used to track long-term remediation.
Linked functional requirements	

FREQ_097 - Support for EPSS – prediction of vulnerability exploitation	Use of vulnerability exploitation probability models. Support for EPSS – prediction of vulnerability exploitation
FREQ_023 - Vulnerability management and inventory	The solution must ensure vulnerability management and inventory. Regular scans are to be linked to the CMDB. • Assets are updated on the basis of detection • incidents tied to vulnerabilities are linked into the management system
FREQ_086 - Updates of vulnerability tests – community and enterprise feed	Frequency and scope of updates of detection patterns. Updates of vulnerability tests – community and enterprise feed
FREQ_019 - Vulnerability management	The solution must ensure vulnerability management.
FREQ_074 - Collection of information on vulnerabilities and configurations	Obtaining information on vulnerabilities from configuration data and comparison with CVE databases. Collection of information on vulnerabilities and configurations
UC_12 – Manage vulnerabilities and security findings	UC_12 covers vulnerability management in SOCaaS: from receiving the outputs of vulnerability scanners, through their assignment to specific assets (UC_11) and risk-based prioritisation, to the planning and tracking of corrective measures. The aim is to have a comprehensive overview of where we have gaps, what is most important to fix and within what time.

UC_13 – Design and maintain the security architecture of the customer's SOCaaS environment



UC_13 – Design and maintain the security architecture of the customer's SOCaaS environment

Use case	UC_13 – Design and maintain the security architecture of the customer's SOCaaS environment
Goal	UC_13 addresses the design and ongoing maintenance of the technical and security architecture for a specific customer: - how the customer is connected to SOCaaS, - which systems, zones and logging sources are integrated, - through which channels data flows (agent, syslog, API, VPN, peering…), - how SIEM, SOAR, TI, ticketing, EDR, vuln scanners and asset mgmt are interconnected. The output is an approved and maintained architectural design (topology, integration points, data flows) and a list of mandatory log sources and log types, which serves as a basis for onboarding (UC_00), monitoring (UC_01), asset management (UC_11), vulnerability management (UC_12) and automation (UC_07).

Precondition	Cooperation has been agreed with the customer and the basic requirements are known (scope, regulations, criticality of services).
Precondition	The customer has provided the necessary overview of their environment (at least a high-level architecture, types of systems, networks).
Precondition	The SOCaaS platform has integration options available (connectors, APIs, agents, network paths) that can be used.

Postcondition	A documented and approved architectural design for connecting the customer's environment to SOCaaS exists (topology, integration points, data flow).
Postcondition	Integration channels are defined (which system → in what way → into which SOCaaS module).
Postcondition	Changes to the architecture are traceable (versions, who approved, when) and usable as input for onboarding and operation.
Postcondition	A list of mandatory log sources and log types for onboarding the customer to SOCaaS is defined and documented (including the minimum fields necessary for threat detection).

Basic Path	Basic Path

Step	**Activity**
1	Collection of requirements and inputs
2	Analysis of the customer's existing architecture
3	Design of data flows and integration patterns, including identification of mandatory log sources and log types
4	Design of integrations with the customer's tools
5	Assessment of security principles and risks
6	Approval of the architectural design
7	Submission of the design for implementation

8	Ongoing maintenance of the architecture

ALT_13_01 – Constraints on the customer side (network, security policies)	Alternate
Step	**Activity**
1	During the design it is found that the customer has constraints (e.g. new ports cannot be opened, an agent or VPN cannot be used); the Architect adjusts the design to use alternative integration patterns or a narrower monitoring scope and resubmits the changes for approval.

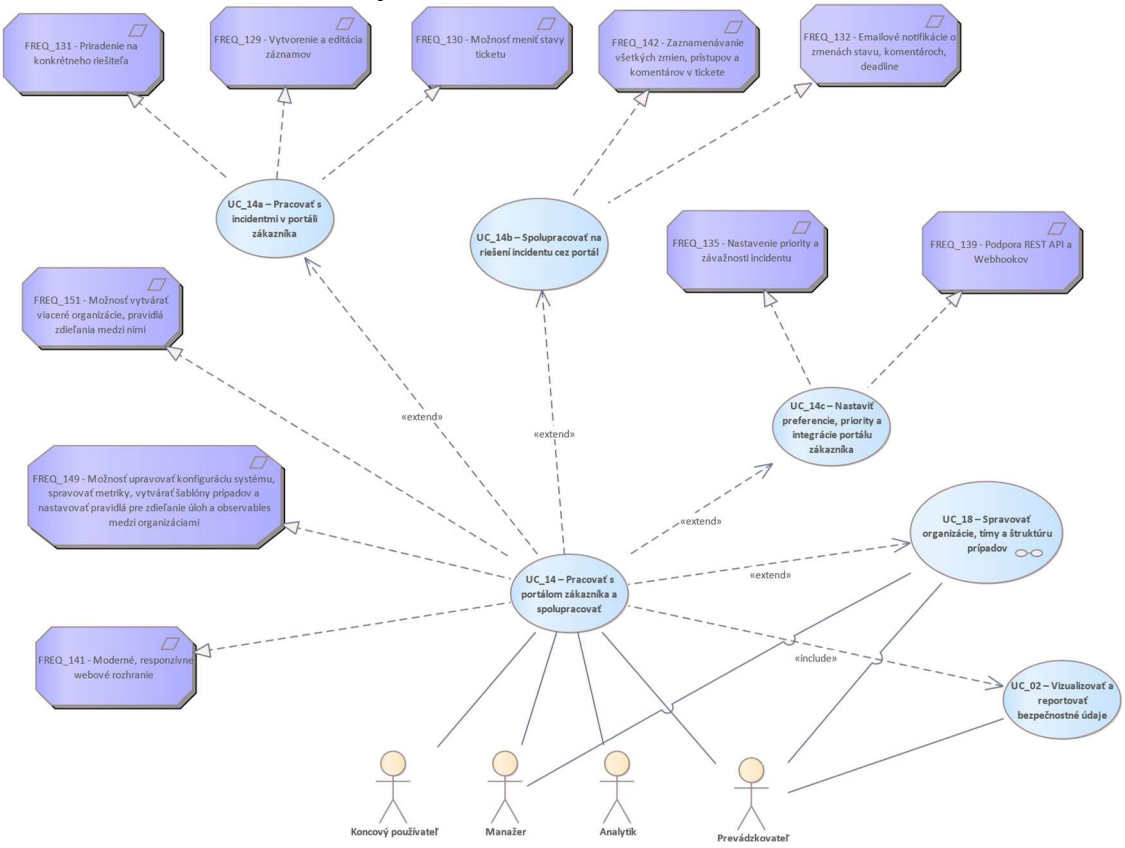
ALT_13_02 – Changes in the customer's environment after deployment	Alternate
Step	**Activity**
1	Over time the customer adds new systems, migrates to the cloud or changes the topology; the Architect creates a revision of the architectural design, adds new integration points and the Operator plans and carries out the necessary changes to the SOCaaS configuration.

ALT_13_03 – The design does not pass the security or performance assessment	Alternate
Step	**Activity**
1	During the assessment of the design it is found that the proposed solution is not satisfactory (excessive latencies, unacceptable risks, insufficient scaling); the design is returned to the Architect for rework and is resubmitted for approval only after the comments have been incorporated.

Linked functional requirements	
FREQ_120 - Integration with monitoring, SIEM, ticketing system, EDR, vulnerability scanner, incident recording	The ability to connect with other SOC tools for automation and data sharing. Integration with monitoring, SIEM, ticketing system, EDR, vulnerability scanner, incident recording
UC_24 – Orchestration of integrations between SOCaaS and external systems	UC_24 addresses the technical and procedural orchestration of integrations between SOCaaS and other systems: - SIEM, SOAR, CTI, ticketing, communication tools, firewall, EDR,
	- API integrations, webhooks, plug-ins and connectors, - unified control of data flows and actions across these systems. The aim is for incident response, automation (UC_07) and incident management to function as a single interconnected ecosystem, not as isolated tools.
FREQ_013 - Integration with other tools	The solution must ensure integration with other tools
FREQ_030 - Integration with other components	The solution must ensure integration with other components. Connection to data sources Integration with incident management and SOAR tools Knowledge database and playbooks
FREQ_166 - Integration with tools such as SIEM, Ticketing, Incident recording, Asset management, CTI	The ability to collect data from other systems and create linked reports. Integration with tools such as SIEM, Ticketing, Incident recording, Asset management, CTI, ...

FREQ_095 - Integrations with tools such as PaloAlto, Fortinet, Cisco, Nagios, Splunk, Verinice	The ability to connect with other security and IT systems. Integrations with tools such as PaloAlto, Fortinet, Cisco, Nagios, Splunk, Verinice
UC_13 – Design and maintain the security architecture of the customer's SOCaaS environment	UC_13 addresses the design and ongoing maintenance of the technical and security architecture for a specific customer: - how the customer is connected to SOCaaS, - which systems, zones and logging sources are integrated, - through which channels data flows (agent, syslog, API, VPN, peering…), - how SIEM, SOAR, TI, ticketing, EDR, vuln scanners and asset mgmt are interconnected. The output is an approved and maintained architectural design (topology, integration points, data flows) and a list of mandatory log sources and log types, which serves as the basis for onboarding (UC_00), monitoring (UC_01), asset management (UC_11), vulnerability management (UC_12) and automation (UC_07).

UC_14 – Work with the customer portal and collaborate



UC_14 – Work with the customer portal and collaborat

Use case	UC_14 – Work with the customer portal and collaborate

Goal	UC_14 addresses the customer's view of SOCaaS – via a web portal where they: - see their incidents, KPIs, reports and assets, - can collaborate with the SOC (comments, attachments, approval of measures), - can customise the portal for multiple organisations/units. The portal is a modern, responsive, multi-tenant web interface and also supports more advanced settings (templates, sharing, integration hooks in sub-UCs).

Precondition	The customer has been onboarded (UC_00) and has at least one organisation / tenant created.
Precondition	Accounts have been created for users and roles assigned (e.g. "Security Manager").
Precondition	The portal is accessible via a secure web interface (HTTPS, SSO/Federation if needed).
Postcondition	Through the portal, the user has an overview of their incidents, KPIs and assets.
Postcondition	It is possible to collaborate with the SOC directly in the context of incidents (comments, attachments, approval).
Postcondition	The portal respects the multi-tenant model – data are separated between organisations and the settings are adapted to the needs of the specific organisation.

Basic Path	Basic Path

Step	**Activity**
1	The end user logs in to the customer portal (SSO / username + password / MFA according to policy).
2	The system displays the home overview – a dashboard with the status of incidents, basic KPIs and notifications
3	The user selects the organisation/tenant on whose behalf they want to work (if they have more than one).
4	In the menu, they choose whether they want to handle incidents, view reports or manage settings and notifications.In the menu, they choose whether they want to handle incidents, view reports or manage settings and notifications.
5	When "Incidents" is selected, a list of incidents/tickets for the given organisation is displayed; the user can click through to the detail
6	When "Reports" is selected, they can view or download the management reports available for their organisation
7	When "Settings" is selected, the user can (depending on their role) change notifications, preferences and organisational settings of the portal
8	The system records all interactions (access to incidents, comments, changes to settings) for audit purposes and ensures that they are visible only within the given organisation.

ALT_14_01 – The user has no organisation assigned	Alternate

Step	**Activity**
1	After login, the system finds that the account is not linked to any organisation; it displays information about the missing assignment and recommends contacting the Operator or Manager.

ALT_14_02 – Restricted view according to role	Alternate

Step	**Activity**
1	The user only has a role with restricted rights (e.g. “read-only”); the portal hides certain action elements (editing, approval) and only allows incidents and reports to be viewed.

Linked functional requirements	
UC_14a – Work with incidents in the customer portal	UC_14a extends UC_14 with specific work with incidents/tickets from the customer’s perspective:
	- viewing the list of incidents, - opening the detail, - creating a new “ticket” or adding information, - changing statuses (New, In progress, Closed, Postponed), - routing the incident to the correct resolver (L1/L2/L3, a specific team).
FREQ_149 - Ability to modify the system configuration, manage metrics, create case templates and set rules for sharing tasks and observables between organisations	The ability to customise the platform according to the organisation’s needs. Ability to modify the system configuration, manage metrics, create case templates and set rules for sharing tasks and observables between organisations?
UC_14c – Set preferences, priorities and integrations of the customer portal	UC_14c extends UC_14 with a configuration part for the customer: - setting of incident priorities and severity from the customer’s perspective, - setting of integration channels (REST API / Webhooks) for connecting the portal to the customer’s internal tools (ticketing, chat, ITSM).
FREQ_151 - Ability to create multiple organisations and sharing rules between them	The ability to operate the system for multiple entities simultaneously. Ability to create multiple organisations and sharing rules between them
UC_02 – Visualise and report security data	The system enables SOC operators and the administrator to display security events by means of dashboards, filters, charts and tabular outputs. It also enables the generation of reports and the export of data for management, audit or customers.
UC_18 – Manage organisations, teams and case structure	UC_18 addresses the administration of organisations/tenants and the way in which incidents/cases are structured within them: - creation and modification of organisations / teams, - setting of case categories, types and owners, - definition of incident templates and the way they are shared between teams and external partners. It builds on UC_17 (users and roles) and UC_14 (customer portal) and supplements them with an “organisational and procedural” layer.
UC_14b – Collaborate on incident resolution via the portal	UC_14b extends UC_14 with a collaboration layer for incident resolution: - comments between the customer and the SOC, - e-mail notifications of status changes, comments and deadlines, - an audit trail of all changes and accesses in the ticket.
FREQ_141 - Modern, responsive web interface	Modern and responsive design for comfortable use. Modern, responsive web interface
Use case	UC_14a – Work with incidents in the customer portal

Goal	UC_14a extends UC_14 with specific work with incidents/tickets from the customer's perspective: - viewing the list of incidents, - opening the detail, - creating a new "ticket" or adding information, - changing statuses (New, In progress, Closed, Postponed), - routing the incident to the correct resolver (L1/L2/L3, a specific team).

Precondition	The precondition from UC_14 applies – the user is logged in and has an organisation.
Precondition	The user has a role that allows working with incidents (not just read-only).
Postcondition	Incidents/tickets are created or updated according to the customer's actions.
Postcondition	The incident has its current status set and a resolver assigned.

Basic Path	Basic Path

Step	**Activity**
1	In the portal, the user opens the "Incidents / Tickets" section.
2	The system displays a list of incidents with a filter by status, type, priority or period.
3	The user selects a specific incident and opens its detail.
4	In the incident detail, they can add information (description, impacts, contact person) or change the incident status (e.g. from "New" to "In progress").
5	The user can assign the incident to a specific resolver or team (e.g. "SOC L1", "SOC L2" , ...).
6	The system saves the changes, updates the view in the portal and makes them available to SOC analysts.

ALT_14a_01 – The user wants to create a new incident/ticket	Alternate

Step	**Activity**
1	The user clicks on "Create incident/ticket", fills in the form (description, criticality, affected systems) and submits it; the system creates a new record and assigns it to the SOC queue.

ALT_14a_02 – The status change is not permitted	Alternate

Step	**Activity**
1	The user attempts to change the status of an incident for which they do not have permission (e.g. to close an incident that only the SOC can close); the system rejects the change and displays information about insufficient rights.

Linked functional requirements	
FREQ_130 - Ability to change ticket statuses	The ability to switch between different ticket statuses during its life cycle. Ability to change ticket statuses (e.g. New, In progress, Closed, Postponed)
FREQ_131 - Assignment to a specific resolver	Assignment of tickets to specific resolvers, including escalation by level. Assignment to a specific resolver (L1 analyst, L2/L3 escalation)
FREQ_129 - Creation and editing of records	Basic functionality of the ticketing system – creating, editing and deleting tickets. Creation and editing of records (tickets)
UC_14a – Work with incidents in the customer portal	UC_14a extends UC_14 with specific work with incidents/tickets from the customer's perspective: - viewing the list of incidents, - opening the detail, - creating a new "ticket" or adding information, - changing statuses (New, In progress, Closed, Postponed), - routing the incident to the correct resolver (L1/L2/L3, a specific team).
Use case	UC_14b – Collaborate on incident resolution via the portal
Goal	UC_14b extends UC_14 with a collaboration layer for incident resolution: - comments between the customer and the SOC, - e-mail notifications of status changes, comments and deadlines, - an audit trail of all changes and accesses in the ticket.

Precondition	The incident exists and is available in the portal (UC_14a).
Precondition	The user has permission to view and comment on the given incident.
Postcondition	Comments and/or any status changes initiated by the customer have been added to the incident.
Postcondition	All changes and accesses are recorded for audit purposes; the relevant parties receive notifications.

Basic Path	Basic Path

Step	**Activity**
1	The user opens the incident detail in the portal.The user opens the incident detail in the portal.
2	The system saves the comment and records who added it and when.
3	The system saves the comment and records who added it and when.
4	The system sends e-mail notifications to the relevant parties (Analyst, other customer representatives) about the new comment or status change.
5	The Analyst replies via their tool; the reply is synchronised to the portal and displayed to the customer as another comment.
6	In the incident history, all comments, status changes and accesses can be viewed retrospectively.

ALT_14b_01 – Notifications are switched off for a specific user	Alternate
Linked functional requirements	
FREQ_132 - E-mail notifications of status changes, comments, deadlines	Automatic e-mail notifications of status changes, comments, deadlines. E-mail notifications of status changes, comments, deadlines
FREQ_142 - Recording of all changes, accesses and comments in the ticket	Recording of all actions, changes and comments in the ticket. Recording of all changes, accesses and comments in the ticket
UC_14b – Collaborate on incident resolution via the portal	UC_14b extends UC_14 with a collaboration layer for incident resolution: - comments between the customer and the SOC, - e-mail notifications of status changes, comments and deadlines, - an audit trail of all changes and accesses in the ticket.
Use case	UC_14c – Set preferences, priorities and integrations of the customer portal
Goal	UC_14c extends UC_14 with a configuration part for the customer: - setting of incident priorities and severity from the customer's perspective, - setting of integration channels (REST API / Webhooks) for connecting the portal to the customer's internal tools (ticketing, chat, ITSM).

Precondition	The user has a role with the right to change portal settings for their organisation.
Precondition	At least basic incident priorities and categories are defined for the organisation.
Postcondition	Preferred incident priorities/severities are defined for the given organisation.
Postcondition	Integration channels are configured (if needed) – e.g. Webhooks for notifications to the internal ticketing system.

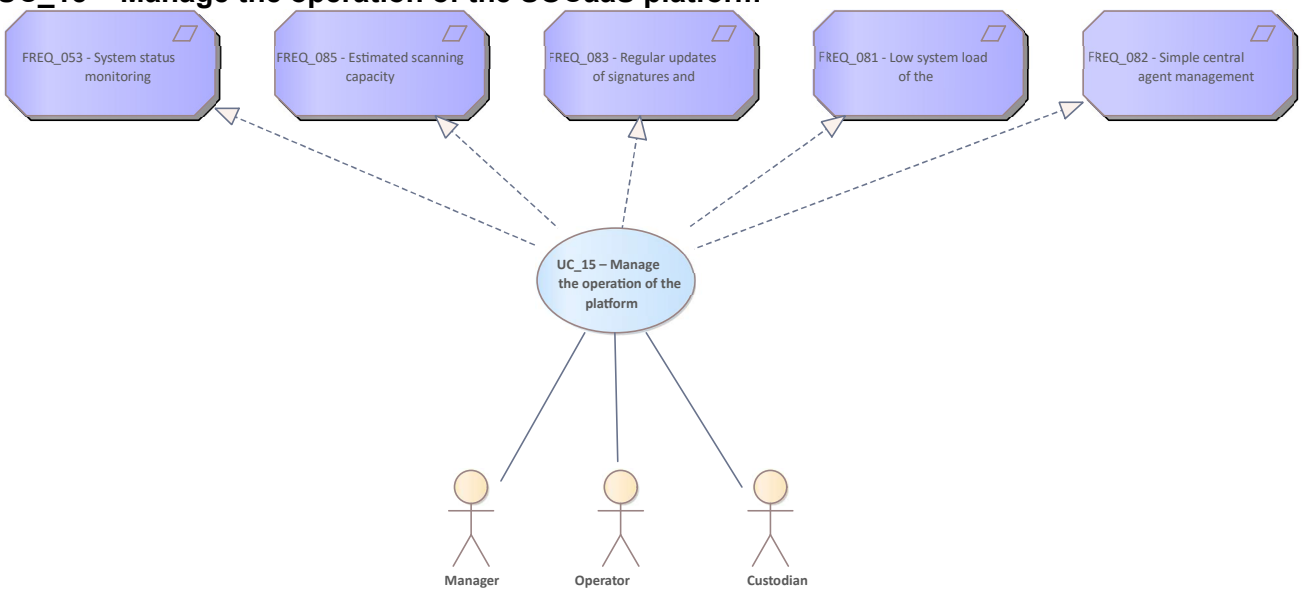
Basic Path	Basic Path

Step	**Activity**
1	In the portal, the Manager opens the "Organisation settings" section.
2	In the "Incidents" part, they set the mapping of priorities/severity (e.g. how SOC priorities are to be reflected in the customer's internal priorities).
3	In the "Integrations" part, they enter the endpoint URLs for Webhooks or API tokens for connecting to the internal ticketing/ITSM system.
4	The system verifies basic connectivity and saves the settings.
5	On subsequent incident status changes or for new incidents, the system sends notifications via REST API / Webhooks to the customer's systems according to the settings.

ALT_14c_01 – Invalid or unavailable integration interface	Alternate

Step	Activity
1	When the integration settings are saved, the test request fails (timeout, 4xx, 5xx); the system notifies the Manager and either saves the integration as inactive or requires the parameters to be corrected.
Linked functional requirements	
FREQ_135 - Setting of incident priority and severity	Setting of the ticket priority and severity level for better planning. Setting of incident priority and severity
FREQ_139 - Support for REST API and Webhooks	Support for REST API and Webhook interfaces for communication with other systems. Support for REST API and Webhooks
UC_14c – Set preferences, priorities and integrations of the customer portal	UC_14c extends UC_14 with a configuration part for the customer: - setting of incident priorities and severity from the customer's perspective, - setting of integration channels (REST API / Webhooks) for connecting the portal to the customer's internal tools (ticketing, chat, ITSM).

UC_15 – Manage the operation of the SOCaaS platform



UC_15 – Manage the operation of the SOCaaS platform

Use case	UC_15 – Manage the operation of the SOCaaS platform
Goal	UC_15 addresses the technical operation of the SOCaaS platform so that: - it is stable, performant and available, - agents and sensors do not place an unnecessary load on the customer's environment, - the platform is continuously updated (signatures, modules), - processing/scanning capacity is kept under control. The aim is for SOCaaS to run reliably, be able to handle the expected volume of data and not jeopardise the performance of the customer's systems.

Precondition	The SOCaaS platform is deployed in a production environment (on-prem/cloud/SaaS).
Precondition	Monitoring tools and mechanisms for tracking the status of the platform and agents exist.
Precondition	Basic operational parameters are defined – acceptable load, processing capacity, maintenance windows.
Postcondition	The status of the platform and the deployed agents is continuously monitored and known to the Operator.
Postcondition	Updates of signatures and modules are carried out in a controlled manner and without significant service outage.
Postcondition	Processing/scanning capacity is planned and the Operator is able to identify the need for expansion or optimisation.

Basic Path	Basic Path

Step	**Activity**
1	The Operator monitors the status of the SOCaaS platform in the operational overview (component availability, queues, latencies, error rates).
2	The system continuously monitors the status of the agents deployed at customers – availability, version, CPU/RAM load.
3	When the number of customers or the volume of data increases, the Operator evaluates the processing/scanning capacity and compares it with the expected growth.
4	Based on the update policy, the Operator schedules regular updates of signatures and modules (e.g. off-peak) to minimise the impact on performance.
5	The system performs the updates and verifies whether the agents and components are functional after the update and are reporting within the expected range.
6	If necessary, the Operator adjusts the configuration of the agents (scanning frequency estimates, level of detail, load limits) so that they do not overload the customers' systems.
7	All operational interventions (updates, configuration changes, capacity interventions) are recorded for audit and diagnostic purposes.

ALT_15_01 – Overload of agents or the platform is detected	Alternate

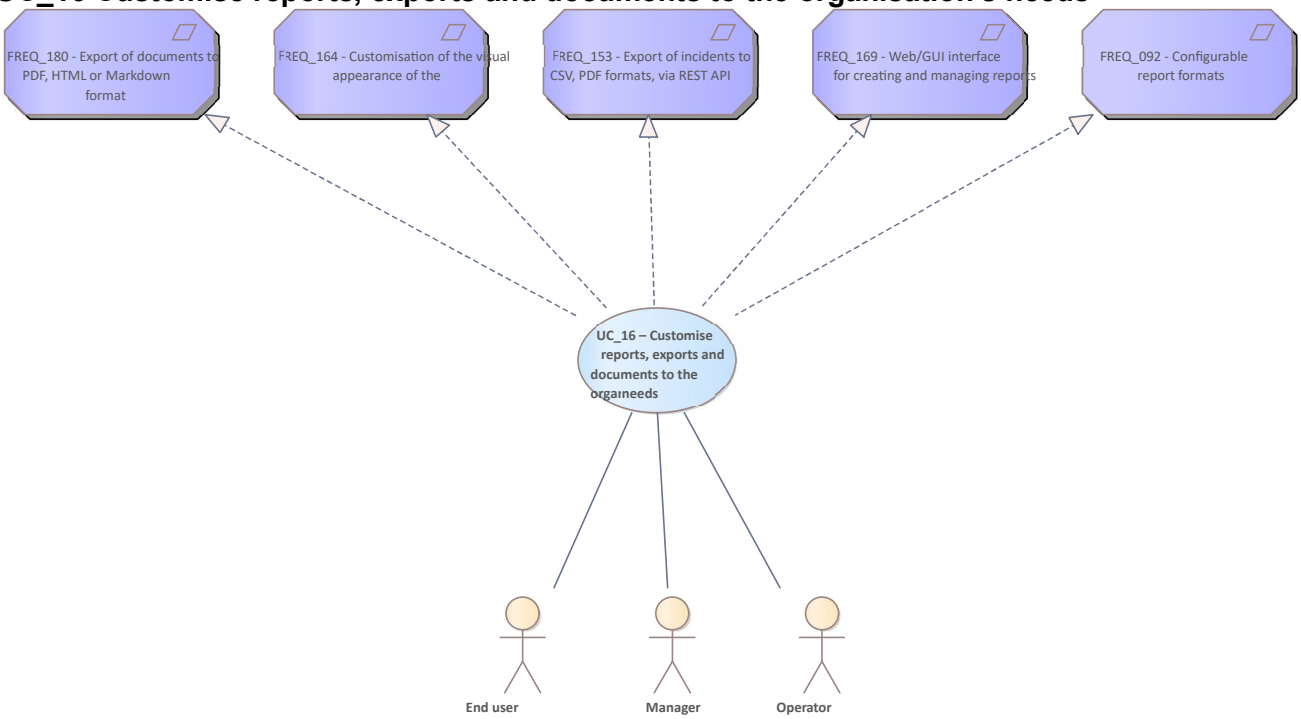
Step	**Activity**
1	Monitoring shows that the agents or the platform itself are placing an excessive load on the system (CPU, RAM, disk, network); the Operator adjusts the configuration (scanning frequency, data scope) or proposes a capacity expansion or deployment optimisation.

ALT_15_02 – Failure of a signature or module update	Alternate

Step	**Activity**
1	After the update is started, it is found that some of the agents or components did not complete the update or stopped reporting; the Operator identifies the affected systems, rolls them back to the last stable version or repeats the update with adjusted parameters.

Linked functional requirements	
FREQ_081 - Low system load of the agent	The agent has a low impact on system performance, which enables deployment at a large scale. Low system load of the agent
FREQ_083 - Regular updates of signatures and modules	Regular updates of threat databases, signatures and detection rules. Regular updates of signatures and modules
FREQ_082 - Simple central agent management	A central console for the management, configuration and supervision of all agents. Simple central agent management
FREQ_053 - System status monitoring	Monitoring of system performance, availability and functionality. System status monitoring (healthcheck)
FREQ_085 - Estimated scanning capacity	The ability of the tool to scan large networks efficiently. Estimated scanning capacity (for example, min. 100 IPs/24h)

UC_16 Customise reports, exports and documents to the organisation's needs



UC_16 – Customise reports, exports and documents to the organisation's needs

Use case	UC_16 – Customise reports, exports and documents to the organisation's needs
Goal	UC_16 addresses the configuration of the form of outputs from SOCaaS: - the appearance of reports according to corporate identity (logo, colours, header), - export formats (CSV, XML, PDF, HTML, Markdown), - export of incidents and documents for further processing or archiving. It follows on from UC_10b (which addresses the content and scheduling of reports) by enabling detailed configuration of presentation and export, so that the outputs fit into the customer's environment and tools.

Precondition	Defined reports and documents that can be generated exist (UC_10b, UC_02b).
Precondition	The user has a role that allows them to change the appearance and export settings of reports/documents.
Precondition	The SOCaaS platform supports multiple export formats and basic appearance templates.
Postcondition	Report appearance templates are defined for individual organisations (logo, colours, header).
Postcondition	Preferred export formats are set for specific types of reports/incidents.
Postcondition	Users are able to export reports, incidents and documents in the required formats for further processing or archiving.

Basic Path	Basic Path

Step	**Activity**
1	In the portal, the Manager or Auditor opens the “Report and export settings” section.
2	They select the type of output they want to customise (e.g. “management report”, “incident overview”, “knowledge document”).
3	In the Appearance part, they select or modify the template:
4	In the Export formats part, they select in which formats the report/document is to be available (e.g. CSV and PDF; for documents also HTML/Markdown).
5	The user displays a preview of the report with the new appearance and format and checks the layout and readability.
6	After confirmation, the system saves the appearance template and the preferred export formats for the given report/document type.
7	The next time a report is generated or an incident is displayed, the system makes it possible to select one of the configured exports directly (e.g. “Download as PDF”, “Export to CSV”, “Export document to HTML”).
8	The exported files can be downloaded by the user or sent to other systems (e.g. via REST API), according to the organisation’s settings.

ALT_16_01 – The required export format is not supported	Alternate

Step	**Activity**
1	The user attempts to set a format that the platform does not support; the system warns them and offers the available alternatives (e.g. CSV instead of XLSX, PDF instead of DOCX).

ALT_16_02 – Error when exporting a large volume of data	Alternate

Step	**Activity**
1	When exporting a report with a large number of records, a limit or timeout is exceeded; the system interrupts the export, informs the user and offers to narrow the scope (shorter period, filtered data) or an asynchronous export.

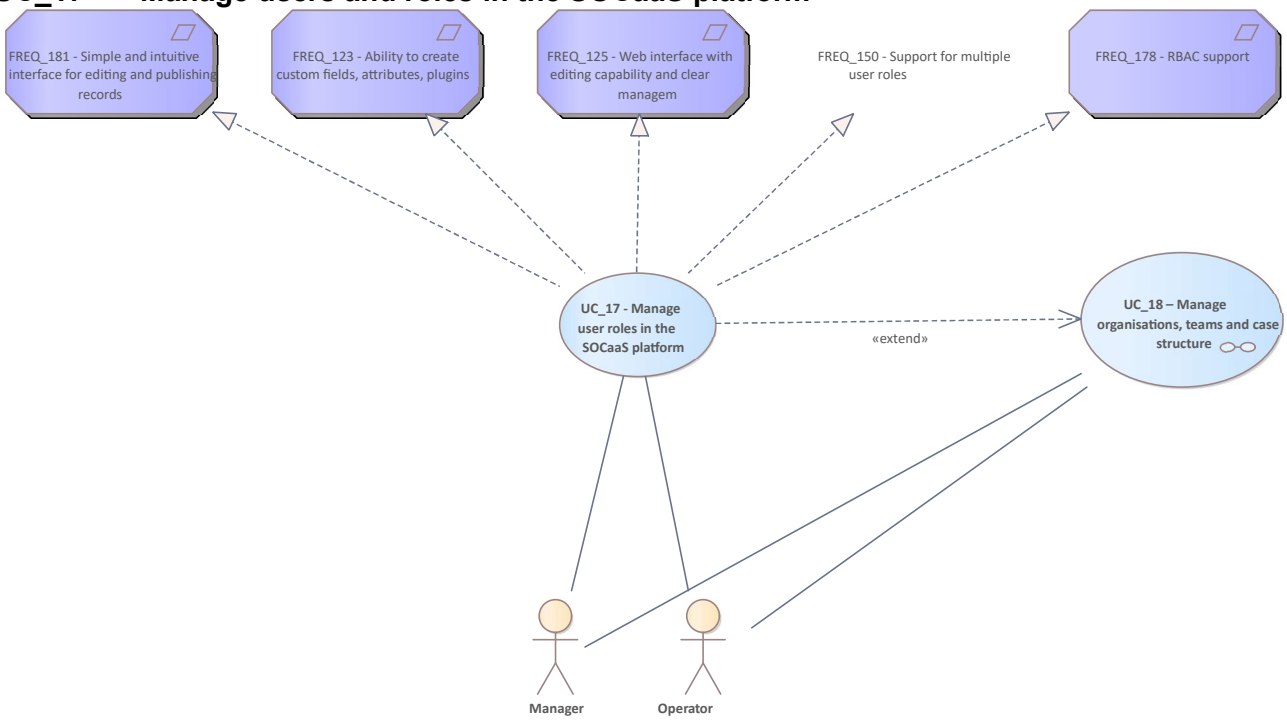
ALT_16_03 – Change of corporate identity	Alternate

Step	**Activity**
1	The organisation changes its logo or corporate colours; the Manager opens the appearance settings, modifies the template (logo, colours) and after saving, all newly generated reports will use the new identity, while older reports remain in their original form.

Linked functional requirements	
FREQ_180 - Export of documents to PDF, HTML or Markdown format	The ability to export documents for offline use or publication. Export of documents to PDF, HTML or Markdown format
FREQ_092 - Configurable report formats	Flexibility of outputs for further processing. Configurable report formats (e.g. CSV, XML, PDF)
FREQ_169 - Web/GUI interface for creating and managing reports	A user-friendly interface for designing, editing and managing reports. Web/GUI interface for creating and managing reports

FREQ_153 - Export of incidents to CSV, PDF formats, via REST API	Export of data for documentation or reporting. Export of incidents to CSV, PDF formats, via REST API
FREQ_164 - Customisation of the visual appearance of the report	The ability to modify the appearance of the report according to corporate identity. Customisation of the visual appearance of the report (logo, colours, header)

UC_17 Manage users and roles in the SOCaaS platform



UC_17 – Manage users and roles in the SOCaaS platform

Use case	UC_17 - Manage user roles in the SOCaaS platform
Goal	UC_17 addresses the administration of users and roles within SOCaaS: - creation and modification of accounts, - assignment of roles (SOC, customer, auditor…), - setting of access via RBAC, - addition of custom attributes where needed. It is a system/administrative UC that supports all the other UCs by determining who can see and do what in the platform.

Precondition	The SOCaaS platform is running and has an accessible administrative web interface.
Precondition	The currently logged-in user has administrative permissions to manage accounts and roles.
Postcondition	Created, modified or deactivated user accounts are stored in the system.
Postcondition	Users are assigned appropriate roles according to the RBAC model.
Postcondition	Any custom attributes/metadata are set and usable in other parts of the system.

Basic Path	Basic Path

Step	**Activity**
1	The Operator logs in to the administrative web interface.
2	They open the “Users and roles” section.
3	They choose whether to create a new user or modify an existing account.
4	When creating an account, they enter the basic data (name, contacts, account identifier) and select one or more user roles (e.g. SOC analyst, Manager, Customer, Auditor).
5	If necessary, they add custom fields/attributes (e.g. affiliation with an organisation, permission level, special tags).
6	The system saves the changes and displays the user in a clear list; the interface allows filtering and searching by name, role and other attributes.
7	The Operator can deactivate the account or change roles if the user’s responsibilities change.

ALT_17_01 – Attempt to make a change for which the user does not have permission	Alternate

Step	**Activity**
1	A user without administrative rights attempts to modify accounts or roles; the system rejects the change and displays information that they do not have sufficient permission.

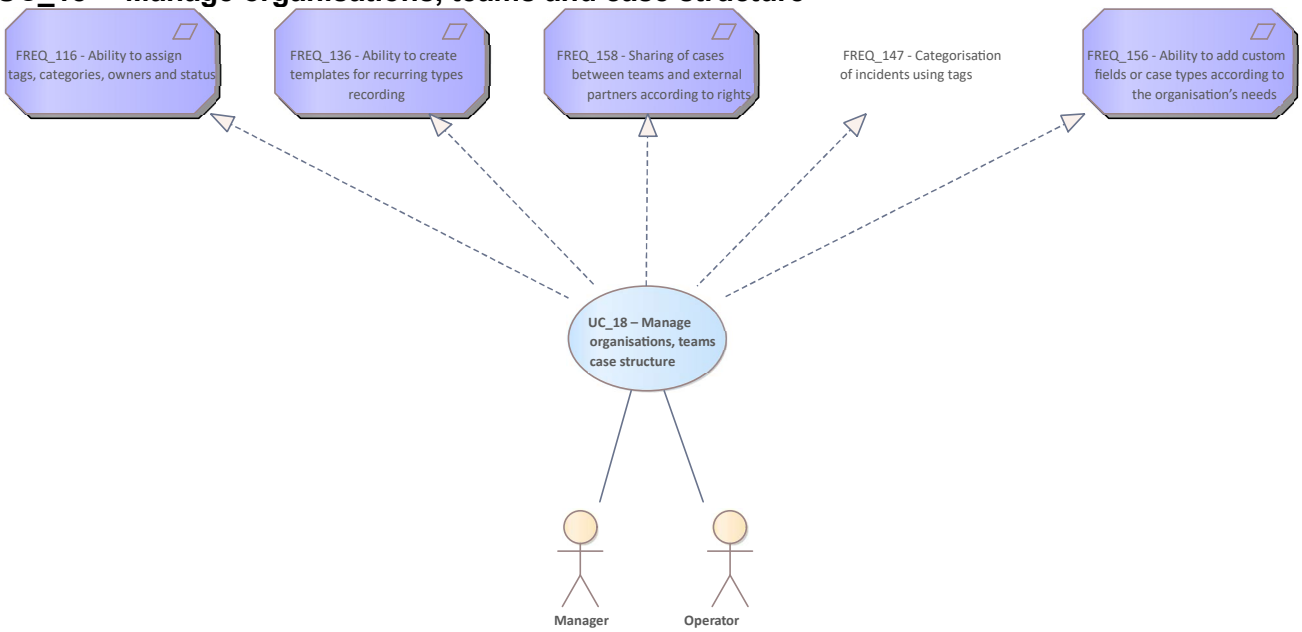
ALT_17_02 – Invalid combination of roles	Alternate

Step	**Activity**
1	When saving the account, the system finds that the selected combination of roles conflicts with the RBAC policy (e.g. conflict of interest); the administrator is informed and must modify the selected roles.

Linked functional requirements	
FREQ_092 - Configurable report formats	Flexibility of outputs for further processing. Configurable report formats (e.g. CSV, XML, PDF)
FREQ_125 - Web interface with editing capability and clear management	A user-friendly GUI for asset management. Web interface with editing capability and clear management

FREQ_150 - Support for multiple user roles	RBAC support for different roles. Support for multiple user roles – at least analyst and administrator
FREQ_153 - Export of incidents to CSV, PDF formats, via REST API	Export of data for documentation or reporting. Export of incidents to CSV, PDF formats, via REST API
FREQ_123 - Ability to create custom fields, attributes, plugins	User flexibility to adapt the system to one's own needs. Ability to create custom fields, attributes, plugins (e.g. NetBox/Django)
FREQ_169 - Web/GUI interface for creating and managing reports	A user-friendly interface for designing, editing and managing reports. Web/GUI interface for creating and managing reports
FREQ_178 - RBAC support	Access control according to user roles. RBAC support – different access levels for reading, editing, approval
FREQ_164 - Customisation of the visual appearance of the report	The ability to modify the appearance of the report according to corporate identity. Customisation of the visual appearance of the report (logo, colours, header)
FREQ_180 - Export of documents to PDF, HTML or Markdown format	The ability to export documents for offline use or publication. Export of documents to PDF, HTML or Markdown format
FREQ_181 - Simple and intuitive interface for editing and publishing records	A user-friendly GUI for knowledge management. Simple and intuitive interface for editing and publishing records
UC_18 – Manage organisations, teams and case structure	UC_18 addresses the administration of organisations/tenants and the way in which incidents/cases are structured within them: - creation and modification of organisations / teams, - setting of case categories, types and owners, - definition of incident templates and the way they are shared between teams and external partners. It builds on UC_17 (users and roles) and UC_14 (customer portal) and supplements them with an "organisational and procedural" layer.

UC_18 Manage organisations, teams and case structure



UC_18 – Manage organisations, teams and case structure

Use case	UC_18 – Manage organisations, teams and case structure
Goal	UC_18 addresses the administration of organisations/tenants and the way in which incidents/cases are structured within them: - creation and modification of organisations / teams, - setting of case categories, types and owners, - definition of incident templates and the way they are shared between teams and external partners. It builds on UC_17 (users and roles) and UC_14 (customer portal) and supplements them with an "organisational and procedural" layer.

Precondition	The user is logged in and has permissions to manage organisations/case structure.
Precondition	SOCaaS has at least a few users created (UC_17) and at least one organisation/tenant (UC_14).
Postcondition	Organisations/ teams have a case structure set up (types, categories, owners).
Postcondition	Templates for recurring incidents and the way they are shared between teams/partners are defined.
Postcondition	Incidents can be unambiguously assigned to an organisation, team, category and owner.

Basic Path		Basic Path
Step	**Activity**	

1	In the administration, the Manager or Operator opens the “Organisations and cases” section.
2	They select an existing organisation (tenant) or create a new one and assign teams to it (e.g. “SOC L1”, “SOC L2”, “Customer IT”).
3	For the organisation, they set the case/incident types (e.g. “Phishing”, “Malware”, “Policy”, “Vuln finding”) and basic categories.
4	For individual incident types, they create templates – pre-filled fields, recommended items, required information.
5	They assign owners, tags and categories (e.g. “critical”, “regulatory”, “internal”) to incident types and templates so that it is possible to filter and report on them.
6	They set which teams and, where applicable, external partners can view and edit the given cases (sharing between teams according to rights).They set which teams and, where applicable, external partners can view and edit the given cases (sharing between teams according to rights).
7	The system saves the organisation structure, templates and sharing rules; new incidents are created according to these settings.

ALT_18_01 – Creation of a custom case type for a specific organisation	Alternate

Step	**Activity**
1	The organisation has a specific case type (e.g. “Compliance finding”) that does not exist in the global schema; the Manager adds a new case type for this organisation only and creates a custom template for it.

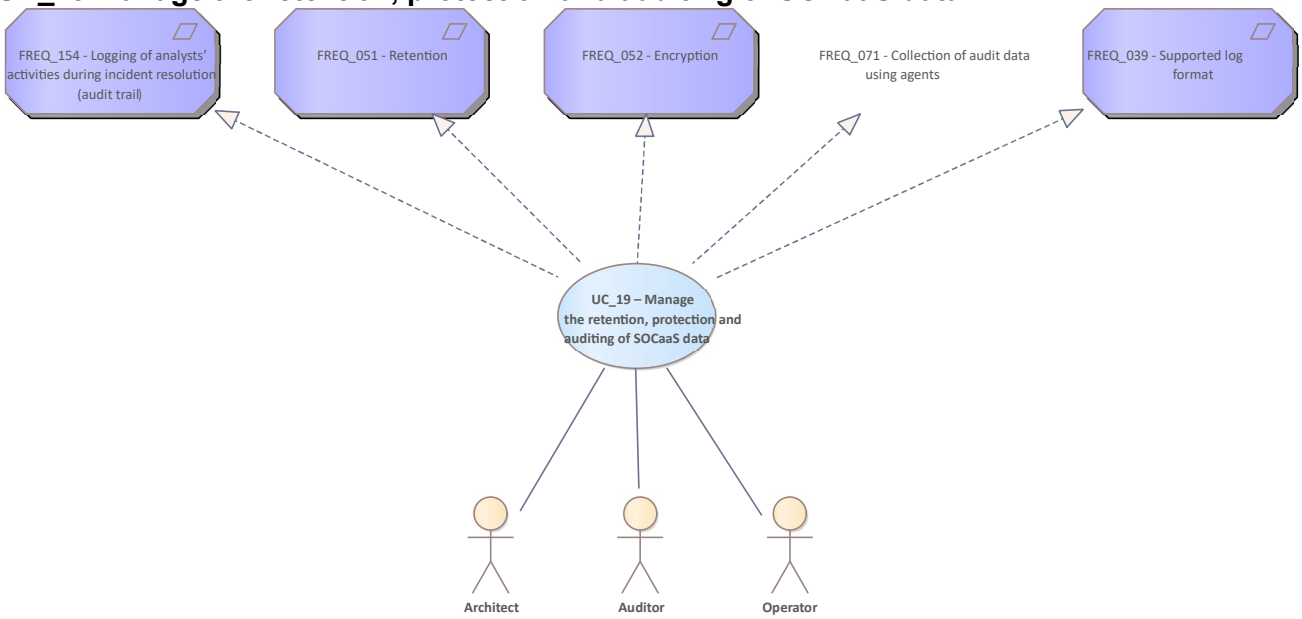
ALT_18_02 – Restriction of case sharing with an external partner	Alternate

Step	**Activity**
1	Originally, cases are shared with an external supplier; the Manager decides that certain categories (e.g. HR incidents) will not be shared and modifies the sharing rules so that these cases are accessible only to internal teams.

Linked functional requirements	
FREQ_156 - Ability to add custom fields or case types according to the organisation’s needs	The ability to extend the incident schema. Ability to add custom fields or case types according to the organisation’s needs
FREQ_116 - Ability to assign tags, categories, owners and status	Adding metadata for efficient sorting and management of assets. Ability to assign tags, categories, owners and status (e.g. active, under maintenance, decommissioned)
FREQ_147 - Categorisation of incidents using tags	The ability to categorise incidents using labels/tags. Categorisation of incidents using tags
FREQ_136 - Ability to create templates for recurring incident types	Creation of templates for quick entry of frequently recurring incidents. Ability to create templates for recurring incident types
FREQ_158 - Sharing of cases between teams and external partners according to rights	Support for controlled sharing of cases. Sharing of cases between teams and external partners according to rights

UC_18 – Manage organisations, teams and case structure	UC_18 addresses the administration of organisations/tenants and the way in which incidents/cases are structured within them: - creation and modification of organisations / teams, - setting of case categories, types and owners, - definition of incident templates and the way they are shared between teams and external partners. It builds on UC_17 (users and roles) and UC_14 (customer portal) and supplements them with an “organisational and procedural” layer.
UC_18 – Manage organisations, teams and case structure	UC_18 addresses the administration of organisations/tenants and the way in which incidents/cases are structured within them: - creation and modification of organisations / teams, - setting of case categories, types and owners, - definition of incident templates and the way they are shared between teams and external partners. It builds on UC_17 (users and roles) and UC_14 (customer portal) and supplements them with an “organisational and procedural” layer.

UC_19 Manage the retention, protection and auditing of SOCaaS data



UC_19 – Manage the retention, protection and auditing of SOCaaS data

Use case	UC_19 – Manage the retention, protection and auditing of SOCaaS data

Goal	UC_19 addresses data management in SOCaaS from the platform's perspective: - where and how logs are stored, indexed and retained, - which retention policies apply to individual log sources, - how logs are protected and encrypted, - how audit records of user and analyst activities are kept. The aim is for SOCaaS to meet the requirements for secure storage and auditability of data, while also supporting forensic analysis.

Precondition	The SOCaaS platform is in operation (UC_15) and receives logs from various sources (UC_01).
Precondition	A log / data repository (SIEM / data lake) exists that supports the configuration of retention and encryption.
Precondition	At least a basic log retention and security policy is defined (e.g. according to log type or regulation).
Postcondition	Retention policies are set for logs/data according to source and importance.
Postcondition	Logs are protected both in transit and at rest (encryption configured).
Postcondition	User and analyst activities are audited so that forensic analysis can be carried out and it can be traced who did what.

Basic Path	Basic Path

Step	**Activity**
1	Design of the storage and retention architecture
2	Setting of retention policies according to source and importance
3	Enabling and verification of log encryption
4	Configuration of the collection of audit records
5	Monitoring of the retention status and audit trail
6	Evaluation of auditors' requests
7	Adjustment of policies on the basis of findings

ALT_19_01 – Insufficient capacity for the configured retention	Alternate

Step	**Activity**
1	With longer retention periods, the Operator finds that the repository does not have sufficient capacity; together with the Architect, they either adjust the retention (e.g. differently according to log type) or propose an expansion of the repository.

ALT_19_02 – Auditor's request for longer retention	Alternate

Step	**Activity**

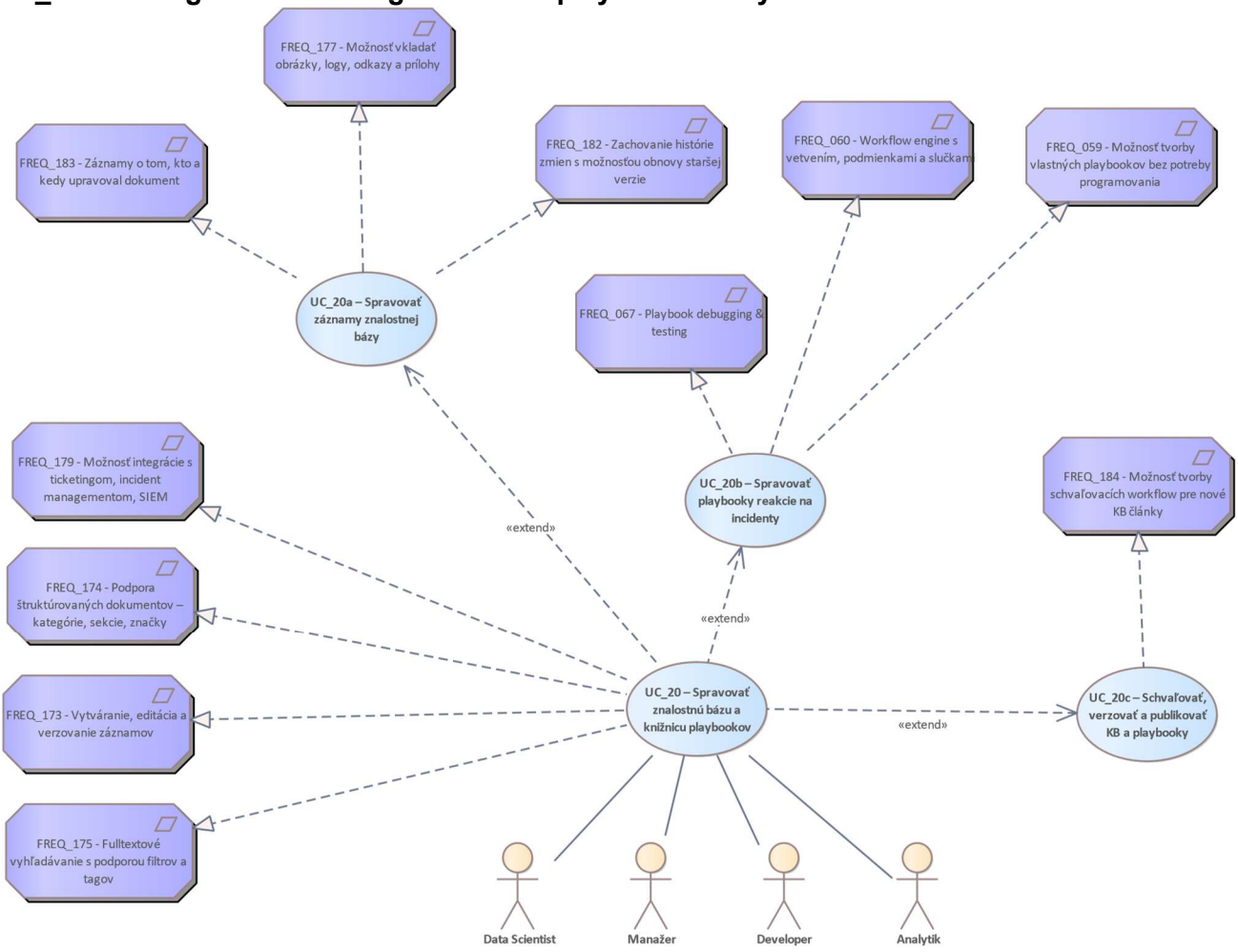
1	The Auditor or regulator requires longer retention for a specific type of logs (e.g. login logs); the Architect and the Operator adjust the relevant retention policy and record the change as part of the compliance documentation.

ALT_19_03 – Incident involving compromise of the log repository	Alternate

Step	**Activity**
1	An incident related to access to the log repository is detected; the audit trail makes it possible to determine who accessed the data and when. On the basis of the findings, the Architect and the Operator tighten the encryption, access rights or segmentation of the repository.

Linked functional requirements	
FREQ_154 - Logging of analysts' activities during incident resolution (audit trail)	Recording of activities for forensic analysis. Logging of analysts' activities during incident resolution (audit trail)
FREQ_039 - Supported log formats	The ability of the SIEM to receive and process logs in various standardised formats. Supported log formats (syslog, JSON, CSV, CEF, LEEF...)
FREQ_052 - Encryption	Protection of logs against unauthorised access in transit and at rest. Encryption (both in transit and at rest)
FREQ_071 - Collection of audit data using agents	Collection of audit records on user activities, processes and the system via an agent. Collection of audit data using agents
FREQ_051 - Retention	The ability to set log retention policies according to source and importance. Retention (flexible, according to log type)

UC_20 Manage the knowledge base and playbook library



UC_20 – Manage the knowledge base and playbook library

Use case	UC_20 – Manage the knowledge base and playbook library
Goal	UC_20 ensures that the SOC has, centrally and in a controlled manner: - a knowledge base (KB) – articles, guides, lessons learned, procedures, - a playbook library – both manual and automated response procedures, - a process for the approval, versioning and publication of this content.

Precondition	A module for the knowledge base and playbooks exists and is available to the Analyst and the Manager.
Precondition	Roles and rights for creating, editing and approving content are defined.
Precondition	At least basic KB articles or playbooks already exist in the system (e.g. an initial set).

Postcondition	The knowledge base contains up-to-date, versioned and categorised records.
Postcondition	The playbook library contains published and usable playbooks linked to incident types.
Postcondition	Each record has a traceable history and information on who created and approved it.

Basic Path	Basic Path
Step	**Activity**
1	The Analyst opens the “Knowledge base and playbooks” module and chooses whether to work with the KB or with playbooks.
2	The Analyst creates a new record or opens an existing one and adds content based on current knowledge from incidents, TI or hunting.
3	The Analyst assigns the record to incident types, categories and tags so that it can be easily found in other UCs.
4	The Analyst saves the record as a draft and submits it for approval to the Manager or the owner of the area.
5	The Manager checks the content, compares the changes with the previous version and decides whether to approve it or return it for revision.
6	After approval, the system marks the new version as valid and moves the old version to the history with a record of the date and author of the change.
7	The valid record is made available in the relevant parts of the system (e.g. in the incident detail, in the playbook engine, in the customer portal).
8	The system records all operations on the record (creation, edits, approval) for audit and forensic analysis.

ALT_20_01 – The record is returned for further work	Alternate
Step	**Activity**
1	The Manager does not approve the proposed content, adds a comment with remarks and the record is returned to the Analyst in the draft state.

ALT_20_02 – The record is marked as obsolete	Alternate
Step	**Activity**
1	The Manager or Analyst finds that the content is no longer valid; they change the state to “archived” and the system stops offering it as an active recommendation, but it remains available in the history.

Linked functional requirements	
UC_20a – Manage knowledge base records	UC_20a addresses in detail the life cycle of KB articles – creation, editing, structure, tagging and linkage to incidents.
FREQ_179 - Possibility of integration with ticketing, incident management, SIEM	Linking the KB with other systems for more efficient use of content. Possibility of integration with ticketing, incident management, SIEM
UC_20c – Approve, version and publish the KB and playbooks	UC_20c addresses the governance layer over the KB and playbooks – approval, versioning, change auditing and approval workflow.

FREQ_140 - Export to PDF, CSV, JSON formats for documentation and reporting	Possibility to export tickets for archiving or reporting purposes. Export to PDF, CSV, JSON formats for documentation and reporting
FREQ_175 - Full-text search with support for filters and tags	Searching the KB with the possibility of filtering by tags, categories and content. Full-text search with support for filters and tags
FREQ_173 - Creating, editing and versioning records	Possibility to create, edit and track versions of documents with change history. Creating, editing and versioning records (guides, playbooks, FAQ)
UC_20b – Manage incident response playbooks	UC_20b addresses the design, modification and maintenance of the playbooks that the SOC uses in automated or semi-automated response (UC_07).
FREQ_174 - Support for structured documents – categories, sections, tags	Organisation of KB documents according to structured rules (categories, tags, sections). Support for structured documents – categories, sections, labels (tags)
UC_20 – Manage the knowledge base and playbook library	UC_20 ensures that the SOC has, centrally and in a controlled manner: - a knowledge base (KB) – articles, guides, lessons learned, procedures, - a playbook library – both manual and automated response procedures, - a process for the approval, versioning and publication of this content.
UC_20 – Manage the knowledge base and playbook library	UC_20 ensures that the SOC has, centrally and in a controlled manner: - a knowledge base (KB) – articles, guides, lessons learned, procedures, - a playbook library – both manual and automated response procedures, - a process for the approval, versioning and publication of this content.
Use case	UC_20a – Manage knowledge base records
Goal	UC_20a addresses in detail the life cycle of KB articles – creation, editing, structure, tagging and linkage to incidents.

Precondition	The preconditions of UC_20 apply.
Precondition	The user has permission to edit KB records.
Postcondition	The KB contains up-to-date, structured articles with rich content, change history and an audit trail.

Basic Path	Basic Path

Step	**Activity**
1	The Analyst opens the list of KB articles in the “Knowledge Base” module.
2	The Analyst creates a new KB article or opens an existing record for editing.
3	The Analyst edits the article content and adds structure (sections, subsections), tags and categories.
4	The Analyst inserts relevant attachments, logs or screenshots that help in resolving similar incidents.
5	The Analyst saves the record; the system records the author and time of the change and, if necessary, sends it for approval.

ALT_20a_01 - Merging two related KB records		Alternate
Step	**Activity**	
1	The Analyst identifies two very similar articles in the knowledge base, creates a single consolidated record and marks the original articles as archived with a link to the new consolidated article.	
ALT_20a_02 - Attempt to edit without sufficient permissions		Alternate
Step	**Activity**	
1	A user with read-only access attempts to edit a KB article; the system does not allow the change, displays information about insufficient permissions and leaves the record unchanged.	
Linked functional requirements		
FREQ_183 - Records of who edited a document and when	Audit trail of document changes with information about authors and time. Records of who edited a document and when	
FREQ_177 - Possibility to insert images, logs, links and attachments	Ability to enrich documents with multimedia or data content. Possibility to insert images, logs, links and attachments	
FREQ_182 - Preservation of change history with the possibility of restoring an older version	Tracking and archiving of changes for each document. Preservation of change history with the possibility of restoring an older version	
UC_20a – Manage knowledge base records	UC_20a addresses in detail the life cycle of KB articles – creation, editing, structure, tagging and linkage to incidents.	
UC_20a – Manage knowledge base records	UC_20a addresses in detail the life cycle of KB articles – creation, editing, structure, tagging and linkage to incidents.	
Use case	UC_20b – Manage incident response playbooks	
Goal	UC_20b addresses the design, editing and maintenance of playbooks that the SOC uses in automated or semi-automated response (UC_07).	

Precondition	A workflow/playbook engine is available in the system.	
Precondition	The user has permission to create and modify playbooks.	
Postcondition	Playbooks have defined steps, branching and integration points and are tested before deployment to production.	
Basic Path		Basic Path
Step	**Activity**	
1	The Analyst or Developer opens the “Playbooks” section in the SOCaaS interface.	
2	The user creates a new playbook or opens an existing one for editing.	
3	In the graphical editor, they define the individual steps of the playbook, their order, branching and execution conditions.	
4	They assign integration actions to the steps (e.g. calling the API of a firewall, EDR, ticketing) and set the necessary parameters.	

5	The user runs a test of the playbook in debug/simulation mode and verifies that the steps proceed as expected.
6	After a successful test, they send the playbook to the Manager for approval and, once approved, it is marked as “production” for UC_07.

ALT_20b_01 - The playbook fails the test in simulation mode	Alternate

Step	**Activity**
1	After running the playbook test in simulation mode, the system detects an error in one of the steps or missing parameters; the test ends with an error and the playbook remains in the draft state until the Analyst or Developer modifies it.

ALT_20b_02 - Saving a playbook with an incomplete configuration	Alternate

Step	**Activity**
1	The user attempts to save a playbook in which mandatory fields are missing or not all integration steps are configured; the system refuses to save it and highlights the missing or invalid settings.

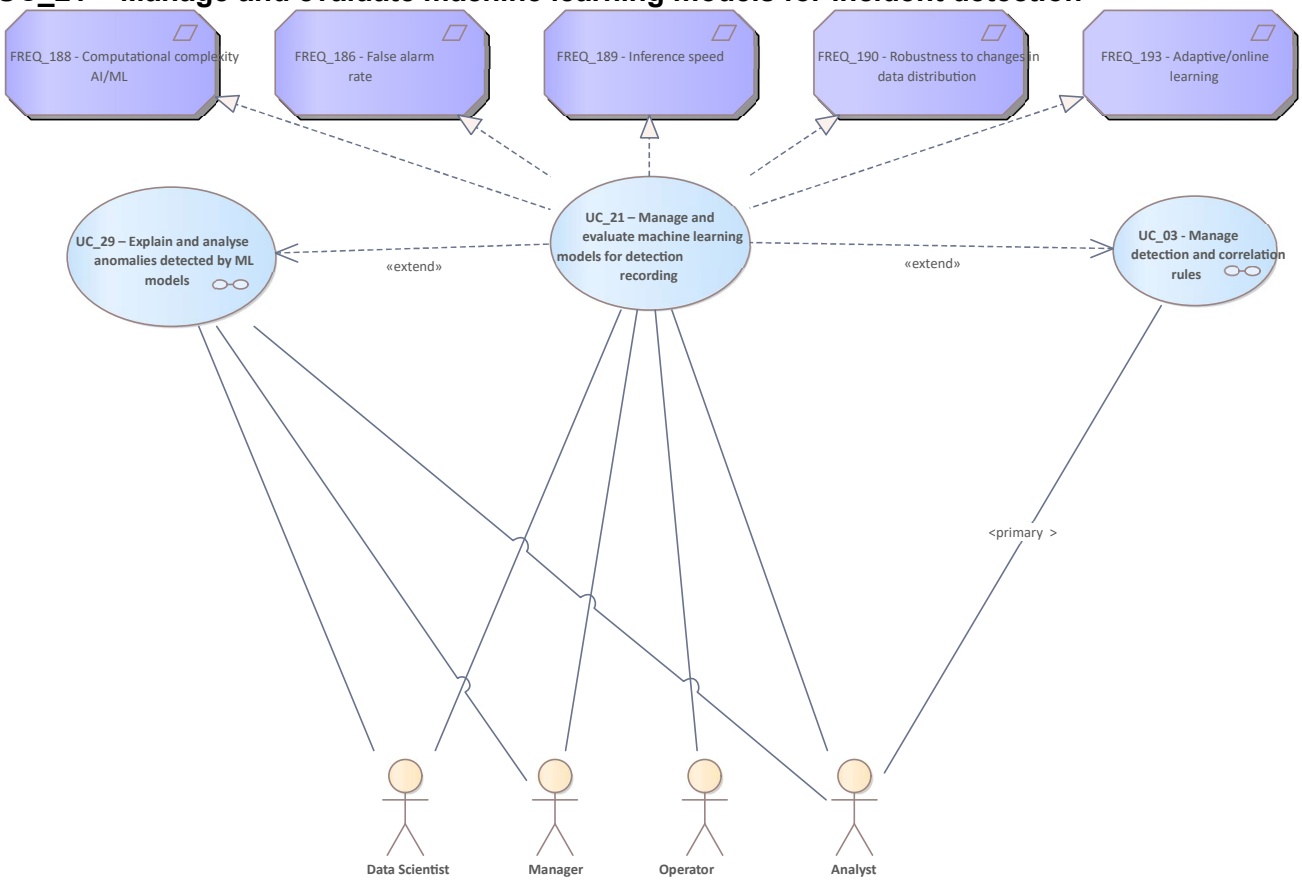
Linked functional requirements	
FREQ_059 - Possibility to create custom playbooks without the need for programming	Ability to create and manage playbooks using a visual interface without the need for coding. Possibility to create custom playbooks without the need for programming
FREQ_067 - Playbook debugging & testing	Possibility to test and debug playbooks before their live deployment. Playbook debugging & testing
FREQ_060 - Workflow engine with branching, conditions and loops	Possibility to define complex logic in playbooks using decision points. Workflow engine with branching, conditions and loops
UC_20b – Manage incident response playbooks	UC_20b addresses the design, modification and maintenance of the playbooks that the SOC uses in automated or semi-automated response (UC_07).
UC_20b – Manage incident response playbooks	UC_20b addresses the design, modification and maintenance of the playbooks that the SOC uses in automated or semi-automated response (UC_07).

Use case	UC_20c – Approve, version and publish the KB and playbooks
Goal	UC_20c addresses the governance layer over the KB and playbooks – approval, versioning, change auditing and approval workflow.

Precondition	Drafts or modified versions of KB articles and playbooks exist (UC_20a, UC_20b).
Precondition	The Manager has permission to approve and publish content.
Postcondition	Each record has a state (draft, under approval, published, archived) and a tracked history of changes and authors.

Basic Path		Basic Path
Step	**Activity**	
1	The Manager opens the list of KB articles and playbooks awaiting approval.	
2	The Manager selects a specific record and displays the current version together with an overview of changes compared with the previous version.	
3	The Manager decides whether to approve the record, return it for further work or mark it for archiving.	
4	Upon approval, the system sets the new version as valid, moves the older version to the history and records the date and author of the approval.	
5	According to the configured workflow, the system informs the relevant users that a new article or playbook has been published and is available for use.	
ALT_20c_01 - Error found in content after publication		Alternate
Step	**Activity**	
1	After a new version of a KB article or playbook is published, the Analyst notices a factual error, creates a corrected version, sends it for expedited approval and, once approved, the corrected version becomes valid.	
ALT_20c_02 - The primary approver is not available		Alternate
Step	**Activity**	
1	The record is awaiting approval, but the primary Manager is unavailable for a long period; after a defined time has elapsed, the system offers the approval to a substitute approver according to the configured approval workflow.	
Linked functional requirements		
FREQ_184 - Possibility to create approval workflows for new KB articles	Approval processes for the publication of new articles. Possibility to create approval workflows for new KB articles	
UC_20c – Approve, version and publish the KB and playbooks	UC_20c addresses the governance layer over the KB and playbooks – approval, versioning, change auditing and approval workflow.	

UC_21 – Manage and evaluate machine learning models for incident detection



UC_21 – Manage and evaluate machine learning models for incident detection

Use case	UC_21 – Manage and evaluate machine learning models for incident detection
Goal	UC_21 addresses the life cycle of the ML models that we use to detect incidents and anomalies in SOCaaS: - design and training of models, - quality measurement (FP/FN, accuracy, speed), - robustness to data changes, - adaptive/online learning while the system is running.

Precondition	SOCaaS has detection component(s) built on ML (anomalies, classifiers, scoring).
Precondition	Historical data/logs are available for training and testing models.
Precondition	Basic quality metrics that we want to achieve are set (e.g. a limit on FP/FN).
Postcondition	The ML models have their quality evaluated (FP/FN, speed, robustness) and are either deployed or rejected.

Postcondition	The system is able to respond to changes in data (drift) by means of retraining or online learning.
Postcondition	The model evaluation results are recorded for further decision-making and audit.

Basic Path	Basic Path

Step	**Activity**
1	The Data Scientist selects a dataset (historical logs, incidents) as the basis for training or testing the model.
2	The Data Scientist trains a new model or adjusts an existing model on the basis of current detection requirements.
3	The system evaluates the quality of the model on the validation set, including the metric of false alarms and missed incidents (FP/FN).
4	The system evaluates the computational complexity of the model during training (CPU, GPU, RAM, training time) and compares it with the available resources.
5	The system measures the inference speed of the model – the time needed to process new events under conditions close to real operation.
6	The Data Scientist evaluates the robustness of the model to changes in data distribution (e.g. new types of logs, changes in behaviour patterns) and the impact on detection quality
7	If the model meets the required metrics, the Manager approves its deployment to production, and the Operator ensures the deployment of the model in the detection engine.
8	After deployment, the system continuously collects feedback from real operation (actual FP/FN) and prepares input for further tuning or adaptive learning.

ALT_21_02 – The model is too slow for real time	Alternate

Step	**Activity**
1	The inference test shows that the model is unable to process incoming data within the required time; the Data Scientist selects a more efficient architecture, simplifies the model or proposes offloading to another environment (e.g. a dedicated GPU server).

ALT_21_03 – Significant data drift appears after deployment	Alternate

Step	**Activity**
1	After some time, it is found that new types of events or behaviour have reduced detection quality; the system warns of drift, the Data Scientist prepares an updated training set and launches adaptive/online learning so that the model reflects the new data.

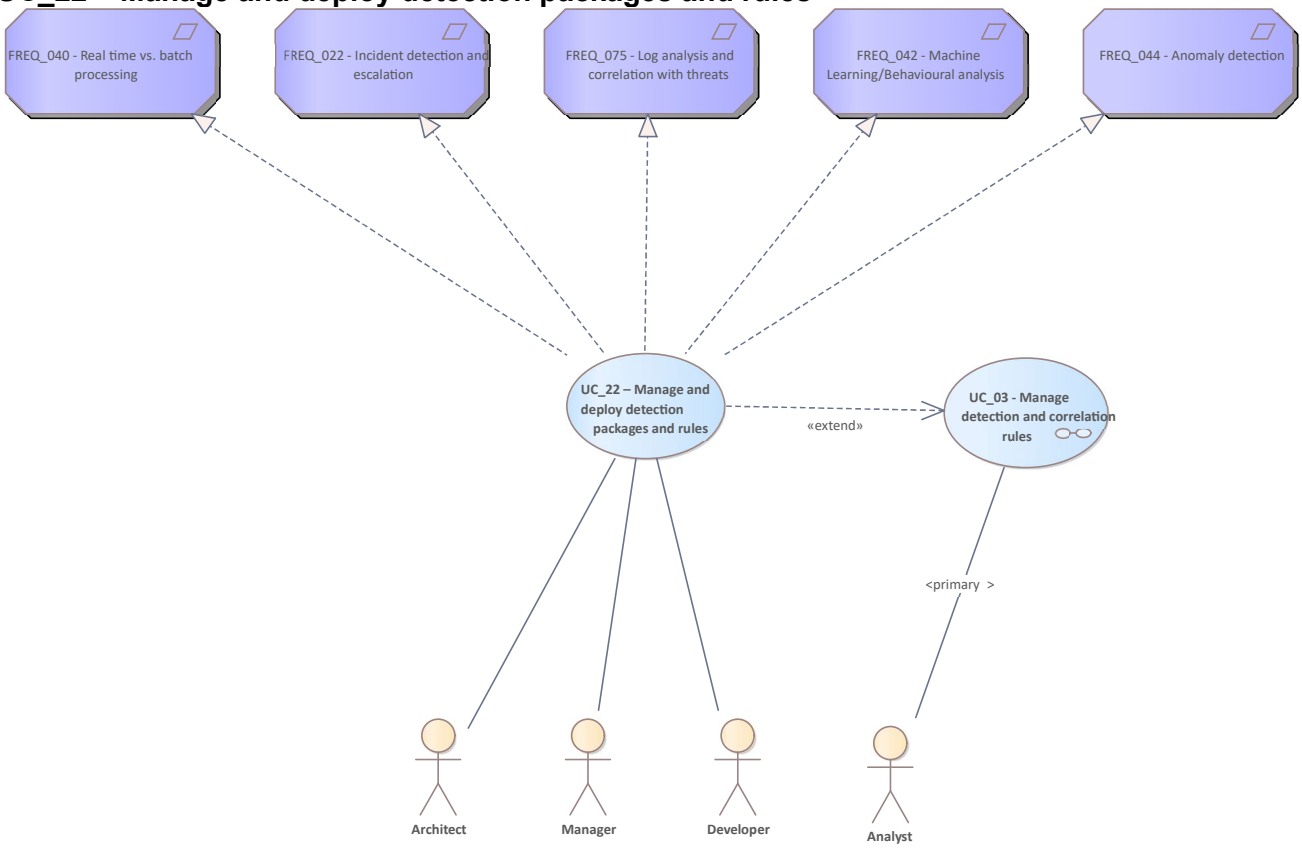
ALT_21_01 – The model has too high a number of false alarms (FP)	Alternate

Step	**Activity**
1	After evaluation, it turns out that the model generates an unacceptably high number of FPs; the Data Scientist adjusts the architecture, hyperparameters or thresholds and repeats the training and evaluation process until an acceptable level is achieved.

Linked functional requirements

FREQ_193 - Adaptive/online learning	The solution must provide the possibility of adaptive/online learning. Support for updating the model during operation without retraining. Ability to update the model without the need for complete retraining, e.g. by means of online learning.
FREQ_186 - False alarm rate	The solution must ensure evaluation of the false alarm rate (False Positives/Negatives). How often the algorithm generates false positives/negatives. Reducing the false alarm rate is crucial for usability in the SOC. Ideal models have a very low number of FP/FN.
UC_29 – Explain and analyse anomalies detected by ML models	UC_29 ensures that SOCaaS is able to explain comprehensibly the anomalies detected by machine learning models: - display the context of the data flow (logs, metrics, network flows), - show the time course of the signals that led to the anomaly, - work with unsupervised models (without labels), - and give the analyst an interpretation of “why the model flagged it”.
FREQ_188 - Computational complexity of AI/ML	The solution must be scalable with respect to computational complexity (CPU/GPU, memory). Resources required for training and inference of the model. Varying load on computational resources, including demands on GPU, RAM and training duration.
UC_03 - Manage detection and correlation rules	This use case covers the activities of analysts and administrators in creating, managing and maintaining detection and correlation rules in the SOCaaS environment. These rules enable the detection of security threats on the basis of IOCs, behavioural anomalies, event chaining or threshold values. The use case focuses exclusively on the rule management process – their definition, tuning, classification and mapping to security frameworks (e.g. MITRE ATT&CK).
FREQ_190 - Robustness to changes in data distribution	The solution must ensure robustness to changes in data distribution (drift, noise). Resilience to noise, changes in data or attacksThe model must be resilient to unexpected changes, fluctuations or noise, otherwise it will stop working correctly.
FREQ_189 - Inference speed	The solution must ensure evaluation of inference speed (detection time). Response speed in detection. How quickly the model is able to provide a result on new data. Critical for real-time deployment.

UC_22 – Manage and deploy detection packages and rules



UC_22 – Manage and deploy detection packages and rules

Use case	UC_22 – Manage and deploy detection packages and rules
Goal	UC_22 addresses the life cycle of detection content (rules, correlation rules, signatures, “content packs”) in SOCaaS: - design and modification of detection rules and packages, - testing on historical or synthetic data, - controlled deployment to production, including the possibility of rollback, - limited deployment for selected tenants/customers. UC_22 does not address the detection runtime itself (that is in UC_03), but the process of designing, testing and the controlled deployment of detection rules and packages into the engine.

Precondition	A detection engine exists (UC_03) that supports the configuration of rules/packages.
Precondition	The platform has a mechanism for testing rules (a separate test environment or test mode).
Precondition	Roles and competences are defined as to who may create/modify and who must approve detection rules.
Postcondition	New or modified detection rules are tested, approved and deployed to production, or rejected.

Postcondition	Rule changes are versioned and auditable and, in the event of problems, it is possible to revert to the previous version.

Basic Path	Basic Path

Step	**Activity**
1	The Analyst opens the “Detection rules and packages” module and displays the list of existing rules.
2	The Analyst creates a new rule or modifies an existing one, defining the conditions, thresholds and mapping to the incident type.
3	The Analyst saves the change as a draft and runs a test of the rule on historical or synthetic data in test mode.
4	The system evaluates the test results – the number of hits, potential false alarms and the impact on existing detection.
5	On the basis of the test results, the Analyst modifies the rule if necessary and repeats the test until acceptable behaviour is achieved.
6	The Analyst sends the rule for approval to the Manager or to the defined owner of the detection content.
7	The Manager assesses the impact of the rule (in particular on the number of alerts, threat types and affected customers/tenants) and decides whether to approve it or return it for modification.
8	After approval, the Developer / DevSecOps launches the deployment of the rule to production, or only for selected tenants or in a limited (“canary”) mode.
9	The system records all changes (rule version, author, approver, deployment time) and monitors the impacts on alerts and performance.

ALT_22_01 – The rule test reveals an extremely high number of hits	Alternate

Step	**Activity**
1	After step 4, the Analyst finds that the rule would generate a disproportionate number of alerts; they modify the rule (tightening the conditions, adding filters) and repeat the test until an acceptable number of hits is achieved.

ALT_22_02 – A deployed rule negatively affects production	Alternate

Step	**Activity**
1	After deployment in step 8, the system detects a significant increase in the number of alerts, latency or errors; the Operator or DevSecOps initiates a reversion to the previous version of the rules and the new rule is marked as “requires modification/test”.

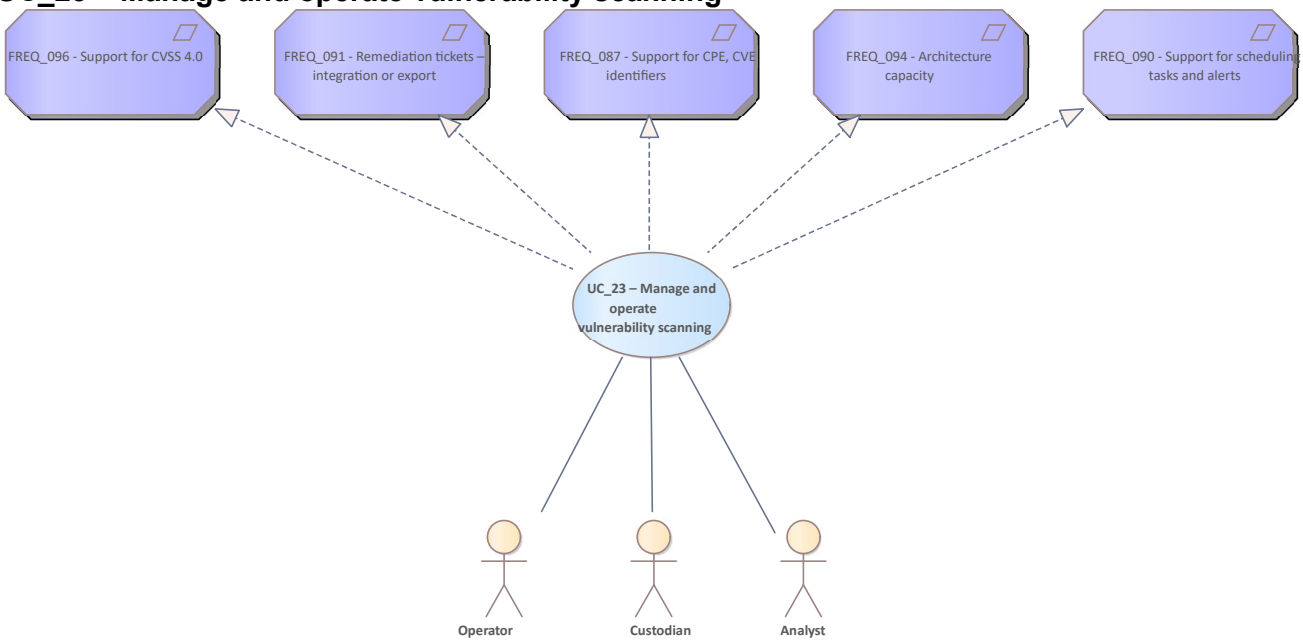
ALT_22_03 – The rule is to be deployed only for selected customers	Alternate

Step	**Activity**
1	During approval in step 7, the Manager decides that the rule is relevant only for a certain type of customer (e.g. a specific sector or tenant); during deployment, only a subset of tenants to which the rule is applied is selected

Linked functional requirements

FREQ_040 - Real time vs. batch processing	Ability to process logs in real time (streaming) or only in batches. Real time vs. batch processing
FREQ_044 - Anomaly detection	Detection of non-standard behaviour or deviations from expected behaviour. Anomaly detection
FREQ_075 - Log analysis and correlation with threats	Log analysis with the aim of detecting correlated events and potential threats. Log analysis and correlation with threats
FREQ_042 - Machine Learning/Behavioural analysis	Ability of the SIEM system to detect anomalies using machine learning. Machine Learning/Behavioural analysis
FREQ_022 - Incident detection and escalation	The solution must ensure incident detection and escalation. Rules for detecting anomalies and known threats must be configured in the environment. After detection: •an alert is generated • the alert is automatically transformed into an incident • Zammad provides a ticket for the L1 team and support for escalation, • the team evaluates the observables and uses enrichment
UC_03 - Manage detection and correlation rules	This use case covers the activities of analysts and administrators in creating, managing and maintaining detection and correlation rules in the SOCaaS environment. These rules enable the detection of security threats on the basis of IOCs, behavioural anomalies, event chaining or threshold values. The use case focuses exclusively on the rule management process – their definition, tuning, classification and mapping to security frameworks (e.g. MITRE ATT&CK).

UC_23 – Manage and operate vulnerability scanning



UC_23 – Manage and operate vulnerability scanning

Use case	UC_23 – Manage and operate vulnerability scanning

Goal	UC_23 addresses the operation and management of vulnerability scanning within SOCaaS: - scheduling and launching scans, - assigning scans to assets and networks, - the technical architecture of scanning (master/sensor), - integration of findings into tickets and into the vulnerability management process (UC_12).

Precondition	At least the basic assets and network ranges (UC_11) intended for scanning are known.
Precondition	A vulnerability scanning tool is deployed in the environment (on-prem, VM or SaaS).
Precondition	The Administrator and Operator have permissions to configure scanners and schedule scanning.
Postcondition	Vulnerability scanning for the selected ranges has been carried out according to schedule or on request.
Postcondition	Findings are enriched with CPE/CVE and CVSS scores and are available for UC_12.
Postcondition	Relevant findings are exported or linked to remediation tickets.

Basic Path	Basic Path

Step	**Activity**
1	The Operator or Administrator opens the “Vulnerability scanning” module and displays an overview of existing scanning profiles.
2	The Operator selects the ranges (IP, networks, asset groups) and chooses the scan type (full, quick, authenticated).
3	The Operator schedules regular execution of scans (e.g. weekly at night) or launches a one-off ad hoc scan.The Operator schedules regular execution of scans (e.g. weekly at night) or launches a one-off ad hoc scan.
4	The system assigns scans to the available scanning nodes according to the architecture (master/sensor) and optimises load distribution.The system assigns scans to the available scanning nodes according to the architecture (master/sensor) and optimises load distribution.
5	After the scan is completed, the system processes the results, assigns CPE/CVE identifiers to the vulnerabilities and calculates the CVSS score.After the scan is completed, the system processes the results, assigns CPE/CVE identifiers to the vulnerabilities and calculates the CVSS score.
6	The Analyst or Administrator displays the scan results, filters the findings by severity and scope and marks those that are to be addressed.The Analyst or Administrator displays the scan results, filters the findings by severity and scope and marks those that are to be addressed.
7	The system creates or updates remediation tickets in the integrated ticketing system or exports the findings in the selected format for further processing.

ALT_23_01 – Scanning cannot be completed on part of the infrastructure	Alternate

Step	**Activity**

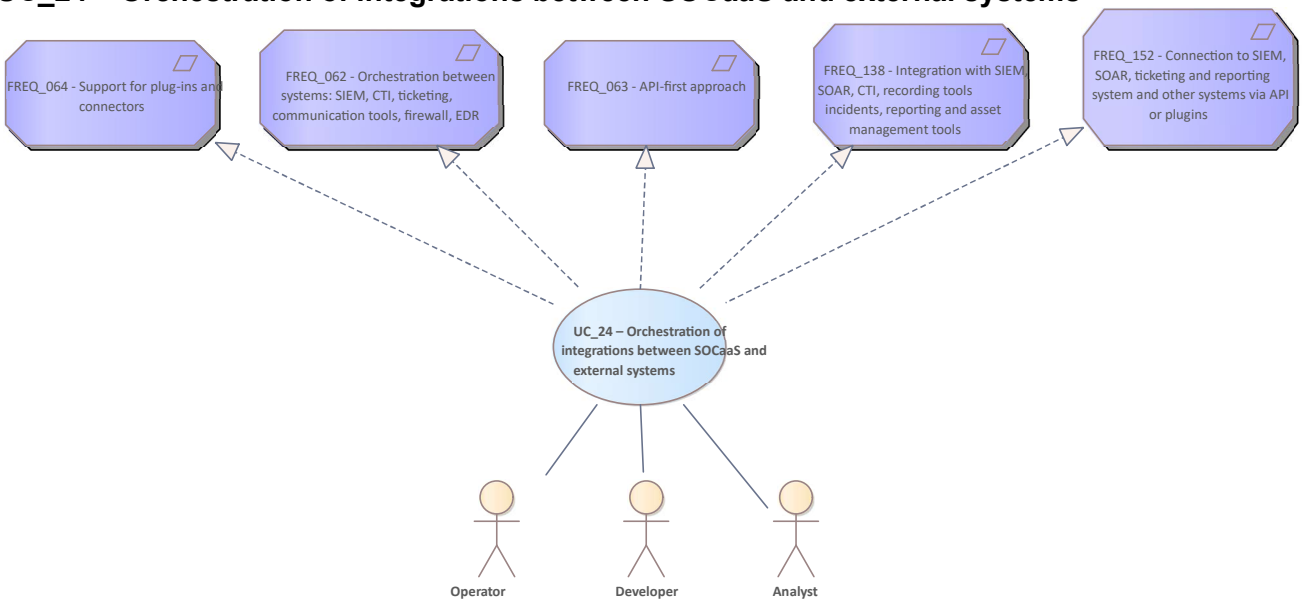
1	During the scan, some systems are unavailable or are blocked by a firewall; the system marks these parts as “not scanned”, the Operator receives information and can schedule a replacement scan with a modified configuration or in a different time window.

ALT_23_02 – Overload of the scanning infrastructure	Alternate

Step	**Activity**
1	After several scans are launched at once, the system detects excessive load on the scanning nodes; the scans are queued or spread over time so as not to exceed the permitted limits, and the Operator is notified that it would be advisable to strengthen the scanning architecture.

Linked functional requirements	
FREQ_087 - Support for CPE, CVE identifiers	Possibility to identify vulnerabilities according to standardised identifiers. Support for CPE, CVE identifiers
FREQ_096 - Support for CVSS 4.0	Support for the latest version of the CVSS scoring system. Support for CVSS 4.0
FREQ_090 - Support for scheduling tasks and alerts	Possibility to automate recurring tasks and the generation of notifications. Support for scheduling tasks and alerts (optional)
FREQ_091 - Remediation tickets – integration or export	Possibility to export findings to a ticketing system or to CSV, JSON, PDF. Remediation tickets – integration or export
FREQ_094 - Scanning architecture	Scalability and adaptation of scan distribution. Scanning architecture (monolithic or master/sensor)

UC_24 – Orchestration of integrations between SOCaaS and external systems



UC_24 – Orchestration of integrations between SOCaaS and external systems

Use case	UC_24 – Orchestration of integrations between SOCaaS and external systems

Goal	UC_24 addresses the technical and procedural orchestration of integrations between SOCaaS and other systems: - SIEM, SOAR, CTI, ticketing, communication tools, firewall, EDR, - API integrations, webhooks, plug-ins and connectors, - unified management of data flows and actions across these systems. The aim is for incident response, automation (UC_07) and incident management to function as a single interconnected ecosystem, not as isolated tools.

Precondition	The architecture of the customer's integrations with SOCaaS has been designed (UC_13) and at least a basic connection to SIEM/SOAR/ticketing exists.
Precondition	The SOCaaS platform supports a REST API, webhooks and a plug-in/connector mechanism.
Precondition	The Operator and Developer have permissions to configure integrations and orchestrations between systems.
Postcondition	The defined integration links and flows between SOCaaS, SIEM, SOAR, CTI, ticketing, firewalls, EDR and other systems are active and monitored.
Postcondition	•Incidents and events are consistently synchronised between systems (state, ID, references).
Postcondition	An audit and an overview are available of which integration channels are active, what actions they trigger and how they fail.

Basic Path	Basic Path

Step	**Activity**
1	The Operator opens the “Integrations and orchestration” module and displays an overview of existing integration links (SIEM, SOAR, CTI, ticketing, EDR, firewall, communication tools).
2	The Developer or Operator creates a new integration link or modifies an existing one, selects the target system (e.g. ticketing system, CTI, firewall) and the integration type (API, webhook, plug-in).
3	The user configures authentication, endpoints, data formats and field mapping between SOCaaS and the target system.
4	The user defines which events in SOCaaS are to trigger actions in external systems (e.g. incident creation → create ticket, incident state change → update ticket, IOC from TI → update block list in the firewall).
5	The system saves the configuration and allows a test of the integration link to be run on a small sample of data or a test incident.
6	After a successful test, the Operator activates the integration link and sets whether it is to be applied globally or only for selected customers/tenants.
7	In real operation, SOCaaS starts using the configured orchestration – it generates webhooks, calls APIs or runs plug-ins according to the defined rules and mappings.
8	The system continuously logs successful and unsuccessful integration calls, displays statistics and alerts the Operator to errors or outages of integration channels.

ALT_24_01 – Failure of an integration call	Alternate

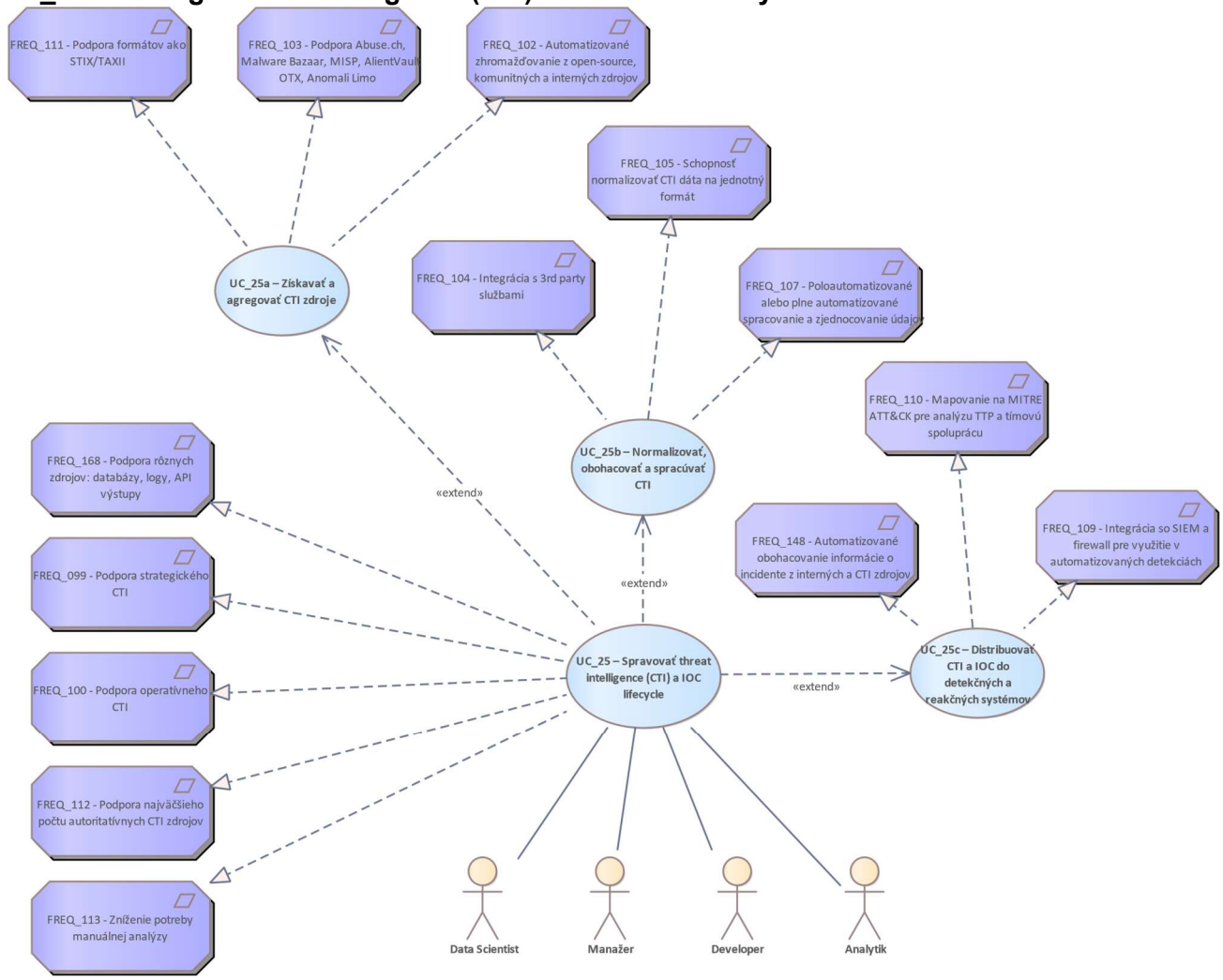
Step	**Activity**
1	During a test or production API/webhook call, the external system returns an error or is unavailable; SOCaaS marks the record as unsuccessful, stores the error details and, depending on the settings, either repeats the call with a retry policy or creates an internal incident/notification for the Operator.

ALT_24_02 – Conflict in data mapping between systems	Alternate

Step	**Activity**
1	When setting up field mapping between SOCaaS and an external system, it is found that some mandatory fields in the target system cannot be unambiguously populated from SOCaaS data; the Operator must add a transformation or default values, or restrict the type of events for which the integration is used.

Linked functional requirements	
FREQ_062 - Orchestration between systems: SIEM, CTI, ticketing, communication tools, firewall, EDR	Interconnection and data exchange with other systems within the incident response process. Orchestration between systems: SIEM, CTI, ticketing, communication tools, firewall, EDR
FREQ_063 - API-first approach	Accessibility of the tool's functions via API and support for webhooks for notifications and interactions. API-first approach (REST, Webhooks)
FREQ_138 - Integration with SIEM, SOAR, CTI, incident recording tools,reporting and asset management tools	Integration with other security tools for automation and workflow. Integration with SIEM, SOAR, CTI, incident recording tools,reporting and asset management tools
FREQ_152 - Connection to SIEM, SOAR, ticketing and reporting system and other systems via API or plugins	Possibility of bidirectional data exchange. Connection to SIEM, SOAR, ticketing and reporting system and other systems via API or plugins
FREQ_064 - Support for plug-ins and connectors	Ability to extend the tool's functionality via modules or connectors. Support for plug-ins and connectors
UC_24 – Orchestration of integrations between SOCaaS and external systems	UC_24 addresses the technical and procedural orchestration of integrations between SOCaaS and other systems: - SIEM, SOAR, CTI, ticketing, communication tools, firewall, EDR, - API integrations, webhooks, plug-ins and connectors, - unified control of data flows and actions across these systems. The aim is for incident response, automation (UC_07) and incident management to function as a single interconnected ecosystem, not as isolated tools.

UC_25 – Manage threat intelligence (CTI) and the IOC lifecycle



UC_25 – Manage threat intelligence (CTI) and the IOC lifecycle

Use case	UC_25 – Manage threat intelligence (CTI) and the IOC lifecycle
Goal	UC_25 addresses the complete CTI lifecycle in SOCaaS: - acquiring CTI from various sources (feeds, MISP, open-source, vendor), - normalisation, enrichment and prioritisation of CTI data, - distribution of IOCs/TTPs to detection (SIEM, EDR, firewall, SOAR) and to incidents.

Precondition	SOCaaS has a CTI module (or is integrated with an external CTI tool).
Precondition	CTI sources (community, commercial, internal) are defined, as well as a basic policy on which sources are “trusted”.

Precondition	Detection systems (SIEM, EDR, firewall, IDS, SOAR) support the receipt of IOCs/TTPs from CTI.
Postcondition	CTI data are systematically collected, normalised, enriched and aggregated.
Postcondition	Threats are prioritised and input for detection, hunting and incidents is created from the relevant IOCs/TTPs.

Postcondition	IOCs/TTPs are distributed to detection systems and incident management in a consistent manner (with an audit trail).

Basic Path	Basic Path

Step	**Activity**
1	The Analyst opens the “Threat Intelligence” module and displays an overview of active CTI sources and the latest updates.
2	Periodically or on request, the system downloads new CTI data from all configured sources (open-source, community, vendor, internal).Periodically or on request, the system downloads new CTI data from all configured sources (open-source, community, vendor, internal).
3	The CTI pipeline normalises, enriches and aggregates the data – it merges IOCs, adds context and links them with historical incidents and logs.
4	The Data Scientist defines or modifies the rules for scoring, prioritisation and categorisation of threats (criticality, relevance for specific customers).
5	On the basis of the scoring, the system automatically identifies the most relevant threats and IOCs that are to be used in detection, hunting and incidents.
6	For the selected IOCs/TTPs, the CTI module creates outputs for detection (e.g. block lists, SIEM rules, hunting queries) and prepares them for distribution to the connected systems.
7	The Analyst has enriched CTI available in incidents and hunting tools – they see campaigns, TTPs, groups and related IOCs.
8	The system continuously records which CTI sources and IOCs/TTPs were used in real incidents and hunting cases and uses this information for further fine-tuning of scoring and relevance.

ALT_25_01 – A CTI source ceases to be trustworthy or available	Alternate

Step	**Activity**
1	After some time, it turns out that a specific CTI source has poor-quality or duplicate data, or is unavailable for a long period; the Manager or CTI specialist deactivates the source or reduces its weight in the scoring.

ALT_25_02 – Excess of CTI and analyst overload	Alternate

Step	**Activity**
1	The system detects that the volume of new IOCs/threats significantly exceeds the analysts' capacity; the CTI specialist adjusts the scoring, filters and thresholds so that only the most relevant threats come to the fore (e.g. by sector, geolocation, asset exposure).

ALT_25_03 – CTI mismatch between customers/tenants	Alternate

Step	Activity
1	A particular IOC is relevant only for a specific customer (e.g. according to its technologies or geography); the system supports per-tenant filtering of CTI, so that IOCs/TTPs are distributed only to the customers they concern.
Linked functional requirements	
FREQ_112 - Support for the largest number of authoritative CTI sources	An extensive selection of community, public and paid sources. Support for the largest number of authoritative CTI sources
FREQ_168 - Support for various sources: databases, logs, API outputs	Ability to aggregate data from multiple technical and analytical sources. Support for various sources: databases, logs, API outputs
UC_25b – Normalise, enrich and process CTI	UC_25b addresses the CTI processing pipeline: - normalisation of various formats into a unified model, - enrichment of IOCs and artefacts from external services, - a workflow for automatic/semi-automatic CTI processing.
UC_25a – Acquire and aggregate CTI sources	UC_25a addresses the ingest and connection of CTI sources: - definition of sources (feeds, MISP, API, vendor portals), - automated downloading, - use of standardised formats.
FREQ_099 - Support for strategic CTI	Acquisition and analysis of information on long-term trends, impacts of attacks and strategic recommendations. Support for strategic CTI
	(reports, impact analyses, attack trends)
FREQ_113 - Reducing the need for manual analysis	Automation and prioritisation of threats so that analysts deal only with relevant ones. Reducing the need for manual analysis – saving analysts' time
FREQ_100 - Support for operational CTI	Ability to map techniques, tactics and procedures (TTPs) and identify attacker groups and their behaviour. Support for operational CTI (TTPs, group profiling)
UC_25c – Distribute CTI and IOCs to detection and response systems	UC_25c addresses the push of CTI to the rest of the ecosystem: - forwarding of IOCs to SIEM/IDS/EDR/firewalls, - use of CTI in detection, playbooks and incidents,
Use case	UC_25a – Acquire and aggregate CTI sources
Goal	UC_25a addresses the ingest and connection of CTI sources: - definition of sources (feeds, MISP, API, vendor portals), - automated downloading, - use of standardised formats.

Precondition	The CTI module is available in SOCaaS.
Precondition	The CTI sources to which we have access are known (API keys, URLs, authentication).
Postcondition	CTI sources are configured, activated and regularly download new data in supported formats.
Basic Path	Basic Path

Step	Activity
1	The Developer opens the “CTI sources” section and displays the list of existing sources.

2	The Developer adds a new CTI source (feed, MISP, vendor API) and enters the necessary access credentials.
3	They configure the download interval (periodic, on request) and the data type (IOCs, TTPs, campaigns, reports).
4	The system tests the connection, downloads a sample of data and verifies the format and consistency.
5	After a successful test, the source is marked as active and included in the regular CTI ingest.

ALT_25a_01 – Error when connecting to a CTI source	Alternate

Step	**Activity**
1	The test in step 4 fails (incorrect API key, expired token, unavailable endpoint); the system displays an error, deactivates the source and leaves it in the state “requires configuration correction”.

ALT_25a_02 – Limit or quotas on a CTI source	Alternate

Step	**Activity**
1	The CTI source has a limited number of requests per day; the system adjusts the download interval so that quotas are not exceeded and notifies the SOC analyst

Linked functional requirements	
FREQ_103 - Support for Abuse.ch, Malware Bazaar, MISP, AlientVault OTX, Anomali Limo	Ready-made connectors or integrations for well-known CTI sources. Support for Abuse.ch, Malware Bazaar, MISP, AlientVault OTX, Anomali Limo
FREQ_102 - Automated collection from open-source, community and internal sources	Automatic downloading of data from various sources (e.g. GitHub, feeds, MISP, API systems). Automated collection from open-source, community and internal sources
FREQ_111 - Support for formats such as STIX/TAXII	Use of standardised formats for the exchange of CTI data. Support for formats such as STIX/TAXII
UC_25a – Acquire and aggregate CTI sources	UC_25a addresses the ingest and connection of CTI sources: - definition of sources (feeds, MISP, API, vendor portals), - automated downloading, - use of standardised formats.

Use case	UC_25b – Normalise, enrich and process CTI
Goal	UC_25b addresses the CTI processing pipeline: - normalisation of various formats into a unified model, - enrichment of IOCs and artefacts from external services, - workflow for automatic/semi-automatic processing of CTI.

Precondition	CTI data have been downloaded from the sources (UC_25a).
Precondition	Integration services for enrichment are configured (VirusTotal, Shodan, …).
Postcondition	CTI data are in a unified format, enriched and verified, ready for scoring and distribution.

Basic Path	Basic Path

Step	**Activity**
1	The system receives new CTI data from the sources and stores them in a buffer for processing.
2	The CTI pipeline transforms the various formats into a unified structured model (IOCs, TTPs, campaigns, meta).
3	For each IOC or relevant artefact, the pipeline optionally performs enrichment via external services (VirusTotal, Shodan, DomainTools, …).
4	The workflow engine applies the defined processing steps – deduplication, filtering, first-seen/lastseen logic and technical validations.
5	The resulting CTI records are stored in a unified form and marked with a state (new, processed, unconfirmed, discarded).
ALT_25b_01 – External enrichment is temporarily unavailable	Alternate
Step	**Activity**
1	In step 3, an external service (e.g. VirusTotal) does not respond in time; the system marks the enrichment state as “pending” and schedules a repeated attempt later without blocking the entire pipeline.
ALT_25b_02 – Conflict during normalisation of CTI data	Alternate
Step	**Activity**
1	Different sources provide conflicting information for the same IOC; the pipeline marks the record for manual review by a CTI specialist, who selects or combines the correct values.
Linked functional requirements	
FREQ_104 - Integration with 3rd party services	Ability to enrich IOCs and other artefacts using external services. Integration with 3rd party services such as VirusTotal, Shodan, DomainTools, URLScan.io
FREQ_105 - Ability to normalise CTI data to a unified format	Consolidation of data from various formats into a structured unified output. Ability to normalise CTI data to a unified format
FREQ_107 - Semi-automated or fully automated processing and unification of data	Possibility of a workflow for data processing without the need for manual intervention. Semi-automated or fully automated processing and unification of data
UC_25b – Normalise, enrich and process CTI	UC_25b addresses the CTI processing pipeline: - normalisation of various formats into a unified model, - enrichment of IOCs and artefacts from external services, - a workflow for automatic/semi-automatic CTI processing.
UC_25b – Normalise, enrich and process CTI	UC_25b addresses the CTI processing pipeline: - normalisation of various formats into a unified model, - enrichment of IOCs and artefacts from external services, - a workflow for automatic/semi-automatic CTI processing.
Use case	UC_25c – Distribute CTI and IOCs to detection and response systems
Goal	UC_25c addresses the push of CTI to the rest of the ecosystem: - forwarding of IOCs to SIEM/IDS/EDR/firewalls, - use of CTI in detection, playbooks and incidents,

Precondition	CTI data are processed and prioritised (UC_25, UC_25b).
Precondition	Integration links to detection systems exist (UC_24).
Postcondition	Relevant IOCs/TTPs are deployed in detection systems.
Postcondition	Incidents and hunting cases are enriched with CTI metadata.

Basic Path	Basic Path

Step	**Activity**
1	The CTI module selects the IOCs/TTPs that are marked as “relevant for distribution” according to the scoring and policy.
2	The Developer selects to which detection systems the IOCs/TTPs are to be distributed (SIEM, IDS/IPS, EDR, firewall, SOAR).
3	The system generates output artefacts (block lists, SIEM rules, feeds) and sends them via integration channels (API, plug-in, CTI interface).
4	The detection systems start using the new IOCs/TTPs in detection; the CTI module tracks the success rate and manner of use in real incidents.
5	When an incident arises, SOCaaS automatically enriches the incident with information from CTI – it links IOCs, TTPs and MITRE ATT&CK techniques.

ALT_25c_01 – A detection system does not accept some IOCs	Alternate

Step	**Activity**
1	During distribution in step 3, the SIEM or firewall rejects some of the IOCs (e.g. because of format or a limit); the system marks these records as “not applied” and offers the CTI specialist the possibility of adjusting the format or scope.

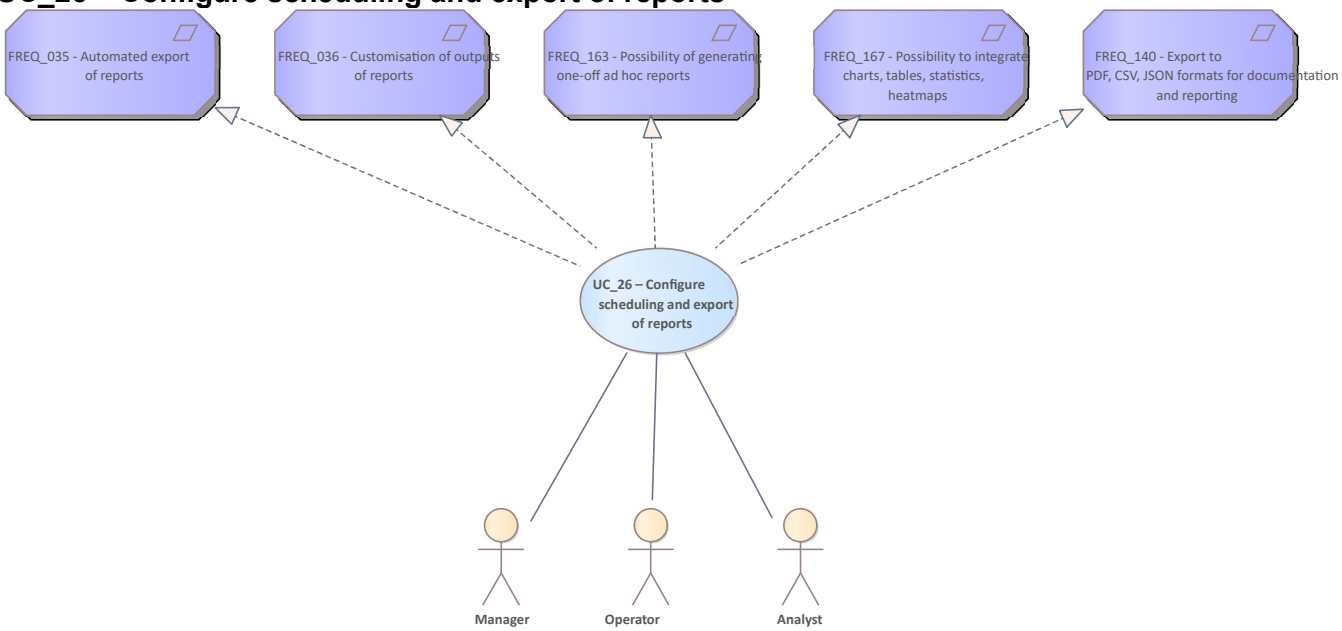
ALT_25c_02 – An IOC is later marked as a false positive or irrelevant	Alternate

Step	**Activity**
1	On the basis of feedback from incidents, it is found that some IOCs cause spam in detection; the CTI module withdraws these IOCs from distribution and adjusts the scoring/policy so that similar threats are assessed differently in the future.

Linked functional requirements	
FREQ_110 - Mapping to MITRE ATT&CK for TTP analysis and team collaboration	Linking incidents, IOCs and TTPs with the ATT&CK framework. Mapping to MITRE ATT&CK for TTP analysis and team collaboration
FREQ_109 - Integration with SIEM and firewall for use in automated detections	Forwarding IOCs and data directly to detection systems (SIEM, IDS, firewall). Integration with SIEM and firewall for use in automated detections
FREQ_148 - Automated enrichment of incident information from internal and CTI sources	Integration for enrichment of IOCs and metadata from CTI or other systems. Automated enrichment of incident information from internal and CTI sources

UC_25c – Distribute CTI and IOCs to detection and response systems	UC_25c addresses the push of CTI to the rest of the ecosystem: - forwarding of IOCs to SIEM/IDS/EDR/firewalls, - use of CTI in detection, playbooks and incidents,
UC_25c – Distribute CTI and IOCs to detection and response systems	UC_25c addresses the push of CTI to the rest of the ecosystem: - forwarding of IOCs to SIEM/IDS/EDR/firewalls, - use of CTI in detection, playbooks and incidents,
UC_25c – Distribute CTI and IOCs to detection and response systems	UC_25c addresses the push of CTI to the rest of the ecosystem: - forwarding of IOCs to SIEM/IDS/EDR/firewalls, - use of CTI in detection, playbooks and incidents,

UC_26 – Configure scheduling and export of reports



UC_26 – Configure scheduling and export of reports

Use case	UC_26 – Configure scheduling and export of reports
Goal	UC_26 addresses the practical management of reports in SOCaaS: - setting up automatic generation and distribution of reports, - customisation of report content and appearance, - generation of one-off ad hoc reports for managers or audit.

Precondition	A reporting module (UC_10) exists in the system with defined report types (incidents, SLA, KPIs…).
Precondition	Distribution channels are configured (at minimum e-mail, ideally also webhook/integration endpoints).
Postcondition	Generation schedules, recipients and export formats are set for the individual report types.
Postcondition	Users are able to generate one-off ad hoc reports outside the schedule.

Postcondition	Reports contain the selected metrics, charts and visualisations in the required design.

Basic Path	Basic Path

Step	**Activity**
1	The Manager or Operator opens the “Reports and export” module and displays the available report types.
2	The user selects a specific report type and chooses which metrics, time ranges and sections are to be included in the report.
3	The user configures the report generation schedule (daily, weekly, monthly) and the list of recipients or target endpoints.
4	The user adjusts the appearance of the report – the organisation's logo, header, notes and, where appropriate, the structure of sections.
5	The user selects the export format (PDF, CSV, JSON) and the method of delivery (e-mail, webhook, saving to storage).
6	At the defined times, the system automatically generates the report, adds the selected charts, tables and statistics and sends it to the selected channels.
7	The system retains a history of generated reports so that they can be downloaded retrospectively or used as input for an audit.

ALT_26_01 – Creation of a one-off ad hoc report	Alternate

Step	**Activity**
1	The Manager needs an extraordinary report outside the scheduled cycle; in the module, they select the report type, set the filters and time period and launch one-off generation, whereby the system generates the report only once without changing the existing schedules.

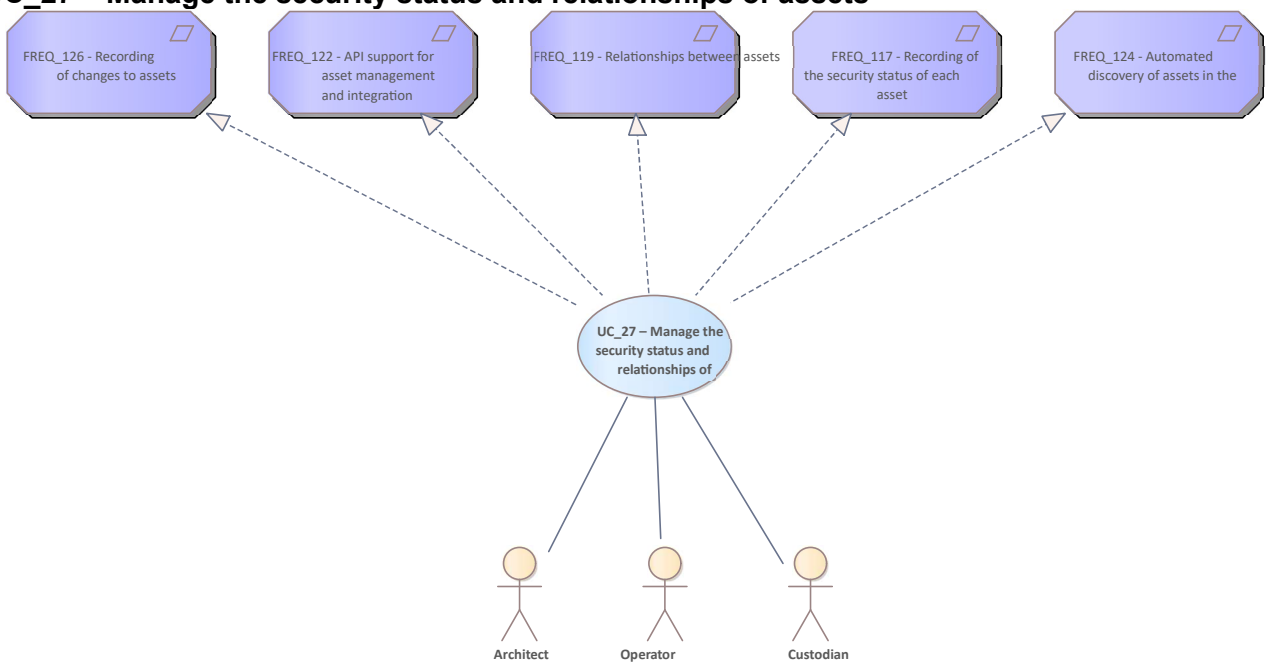
ALT_26_02 – Unsuccessful delivery of an automatically generated report	Alternate

Step	**Activity**
1	During a regular report export, an error occurs when sending the e-mail or webhook; the system saves the report, marks the distribution as unsuccessful and alerts the Operator, who can repeat the delivery or modify the target addresses.

Linked functional requirements	
FREQ_036 - Customisation of report outputs	The solution must ensure customisation of report outputs. It will include: • Possibility to select metrics, fields, time ranges • Personalisation of design – logos, header, colour scheme • Inclusion of explanatory notes, remarks and contextual commentary
FREQ_167 - Possibility to integrate charts, tables, statistics, heatmaps	Visualisation of data directly in reports in various forms. Possibility to integrate charts, tables, statistics, heatmaps
FREQ_163 - Possibility of generating one-off ad hoc reports	Creation of reports outside the regular schedule as needed. Possibility of generating one-off ad hoc reports

FREQ_035 - Automated export of reports	The solution must ensure automated export of reports. It will be possible to export reports in an automated manner: • Periodic reports – outputs generated and distributed daily, weekly or monthly. • Distribution via e-mail or webhook – reports can be automatically sent to specified addresses or endpoints. • Support for templates and custom filters – possibility to set conditions, selections of time periods and formats as needed.
FREQ_140 - Export to PDF, CSV, JSON formats for documentation and reporting	Possibility to export tickets for archiving or reporting purposes. Export to PDF, CSV, JSON formats for documentation and reporting

UC_27 – Manage the security status and relationships of assets



UC_27 – Manage the security status and relationships of assets

Use case	UC_27 – Manage the security status and relationships of assets
Goal	UC_27 extends asset management with a “smart CMDB”: - it records the security status of each asset (patch level, vulnerabilities, recent changes), - displays and manages relationships between assets (server–service–application), - uses automated discovery and API integrations, - maintains an audit of changes to assets.

Precondition	A basic asset inventory has been established (UC_11) – at least the key servers, networks and applications exist.
Precondition	Data sources for discovery are available (scans, CMDB, cloud API…).
Precondition	Users have defined roles for reading/editing assets.
Postcondition	For each asset, the current security status is recorded (versions, patch level, vulnerabilities).
Postcondition	Relationships between assets are defined (topology / dependencies).
Postcondition	Changes to assets are auditable (who, what, when) and available for forensic and operational analysis.

Basic Path	Basic Path

Step	**Activity**
1	The Administrator or Operator opens the “Assets” module and displays an overview of assets with their current security status.
2	The Operator configures automated discovery (e.g. network scanning, cloud API, integration with CMDB) for selected segments of the environment.
3	The system launches discovery and identifies new or changed assets, comparing them with the existing inventory.
4	The Administrator checks the newly found or changed assets, assigns them to the correct categories and adjusts the details as necessary (owner, criticality, location).
5	The Administrator or Architect adds or modifies relationships between assets (e.g. server → service → application → database) so that they reflect the actual dependencies.
6	The system automatically updates the security status of assets on the basis of inputs from scanning, patch management or other integration sources.
7	Each change to an asset (parameters, status, relationships) is saved with information on the author and time of the change so that the history can be analysed later.

ALT_27_01 – Conflict between discovery and the existing inventory	Alternate

Step	**Activity**
1	Discovery finds an asset that already exists in the inventory but with a different designation or parameters; the system flags the conflict, and the Administrator chooses whether to merge the record, overwrite it or keep it as a new asset.

ALT_27_02 – Failure of automated discovery	Alternate

Step	**Activity**
1	A discovery task fails (incorrect authentication, unavailable network, missing permissions); the system stops discovery, records the error details and informs the Operator so that they can adjust the configuration or run discovery again.

Linked functional requirements

FREQ_124 - Automated discovery of assets in the network	Automated collection and updating of asset data from the network and other systems. Automated discovery of assets in the network (e.g. SNMP, nmap, API from other systems)
FREQ_119 - Relationships between assets	Possibility to visualise and manage the interconnections between infrastructure components. Relationships between assets (e.g. server – service – application)
FREQ_126 - Recording of changes to assets	Auditability of all changes for forensic and operational transparency. Recording of changes to assets (who changed what and when)
FREQ_122 - API support for asset management and integration	Possibility to manage and update asset data using an API. API support (REST, GraphQL) for asset management and integration
FREQ_117 - Recording of the security status of each asset	Storing information on vulnerabilities, patch level and system updates. Recording of the security status of each asset (e.g. OS version, last update)

UC_28 – Deploy and maintain SOCaaS in various environments



UC_28 – Deploy and maintain SOCaaS in various environments

Use case	UC_28 – Deploy and maintain SOCaaS in various environments
Goal	UC_28 addresses the deployment and technical operation of the SOCaaS platform in various environments: - on-premise at the customer's site, in the cloud, hybrid, - as a dedicated IaaS environment for SOCaaS, - with clear rules for maintenance, updates and monitoring, - including the use of an open-source/free edition as the base platform. It is a technical-operational UC that extends the operation of the platform (e.g. the UC on SOCaaS operation) with a multi-deployment / IaaS perspective.

Precondition	The architecture of the SOCaaS solution has been designed (topology, modules, integration points).

Precondition	Infrastructure (on-prem, cloud or IaaS) is available in which SOCaaS can be deployed.
Precondition	The Operator has defined processes for monitoring, maintenance and updates of components.
Postcondition	SOCaaS is deployed in the selected environment (on-prem/cloud/hybrid/IaaS) and is monitored.
Postcondition	Rules are set for maintenance, updates and testing of changes in a separate environment.
Postcondition	The platform is ready to scale according to the needs of tenants / customers.

Basic Path	Basic Path

Step	**Activity**
1	The Architect, together with the Operator, decides whether SOCaaS will be deployed on-premise, in the cloud or in a hybrid model.
2	The Operator prepares the target environment – creates the necessary virtual machines, networks, storage and other resources in IaaS.
3	The Operator installs or deploys the SOCaaS components in the selected form (e.g. VM, Docker containers, cloud/SaaS instance).
4	The Operator sets up centralised monitoring of module functionality (metrics collection, health checks, status logging).
5	The Operator creates a test environment or staging instance where updates and configuration changes will be verified.
6	The Operator defines the rules for maintenance and updates – maintenance windows, the order of deployment of changes and the rollback procedure.
7	The Administrator or Operator continuously monitors the state of the deployment and, as necessary, adds further resources (VMs, storage, network elements) to ensure performance and availability.

ALT_28_01 – Need to change the deployment model (on-prem → cloud or hybrid)	Alternate

Step	**Activity**
1	On the basis of growth or customer requirements, it becomes apparent that the original on-prem deployment is no longer suitable; the Architect proposes a transition to the cloud or a hybrid model, and the Operator prepares the new environment and a gradual transfer of components so that the availability of the service is maintained.

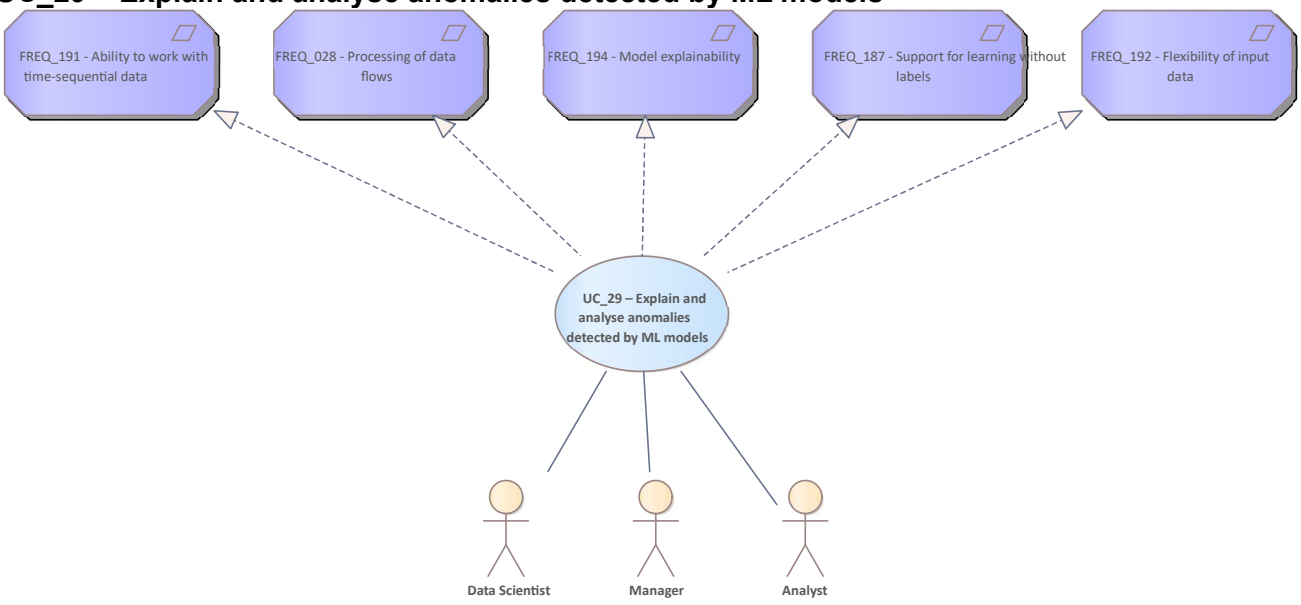
ALT_28_02 – An update causes problems in the production environment	Alternate

Step	**Activity**
1	After an update is deployed according to steps 5–6, it is found that a module is not working correctly; the Operator uses the prepared rollback procedure, reverts to the previous version and first fixes and tests the update in the staging environment before another attempt.

Linked functional requirements	
FREQ_084 - Free version, open-source solution with the possibility of extension	Availability of basic functionality without a licence and the possibility of extension. Free version, open-source solution with the possibility of extension

FREQ_069 - Deployment options: on-prem, cloud, hybrid	Possibility to deploy the solution in various environments, including a local, cloud or hybrid model. Deployment options: on-prem, cloud, hybrid
FREQ_037 - Support for IaaS functionality	The solution requires full IaaS functionality – management of virtual machines (VMs), disks, networks, images, snapshots and separate resources. The platform must support the creation and management of various types of instances, enable the allocation of resources according to the needs of tenants and at the same time offer a robust API for programmatic access. The ability to provide complete IaaS functionality is crucial for ensuring flexible and reliable operation of SOC as a service.
FREQ_093 - Deployment options: on-premise, cloud/SaaS, VM, Docker	Flexibility of installation in various environments. Deployment options: on-premise, cloud/SaaS, VM, Docker
FREQ_027 - System maintenance and updating	The solution must ensure system maintenance and updating. The operations team must have established: • rules for component updates • a test environment for verifying changes • centralised monitoring of the functionality of all modules

UC_29 – Explain and analyse anomalies detected by ML models



UC_29 – Explain and analyse anomalies detected by ML models

Use case	UC_29 – Explain and analyse anomalies detected by ML models
Goal	UC_29 ensures that SOCaaS is able to explain comprehensibly the anomalies detected by machine learning models: - display the context of the data flow (logs, metrics, network flows), - show the time course of the signals that led to the anomaly, - work with unsupervised models (without labels), - and give the analyst an interpretation of “why the model flagged it”.

Precondition	ML models for anomaly detection (UC_21) are deployed in SOCaaS and generate anomalous events/alerts.
Precondition	Data flows (logs, metrics, network flows) are processed through the pipeline – validated, normalised, enriched and stored.

Precondition	The user (Analyst, Data Scientist) has access to the “ML anomalies and explanations” module.
Postcondition	For the selected anomaly, an explanation of the detection exists – over time, over the relevant inputs and with a textual commentary.
Postcondition	The Analyst can label the model output (TP/FP) and add a comment, which is used as feedback for the models.

Basic Path	Basic Path

Step	**Activity**
1	The Analyst opens the “ML anomalies” module and selects a specific anomaly or incident that was detected by an ML model.
2	The system loads from the data pipeline all related events – logs, metrics and network flows – that have passed validation, normalisation and enrichment and have been assigned to the correct indices and timestamps.
3	The system identifies which ML model detected the anomaly (supervised/unsupervised) and loads its outputs (score, or cluster membership where applicable).
4	The Analyst launches the “explain detection” function and the system generates an explanation – the key signals, time segments and features that contributed most to the model's decision.
5	The system displays the time course of selected metrics and log events (time series), highlighting the interval in which the anomaly was detected.
6	The system displays a breakdown of inputs by type (logs, metrics, flow data) and shows which groups of inputs most influenced the detection, including the case where they originate from different sources.
7	The Analyst adds their own interpretation, labels the detection as “true positive”, “false positive” or “unclear” and saves the feedback; the Data Scientist can later use these labels when fine-tuning the models (including unsupervised ones).

ALT_29_01 – Insufficient data to explain the anomaly	Alternate

Step	**Activity**
1	When attempting an explanation, the system finds that some logs or metrics are missing in the selected time segment (the data were not stored or processed); the system warns the analyst of the limited explanation, displays only the available data and marks the explanation as “incomplete”.

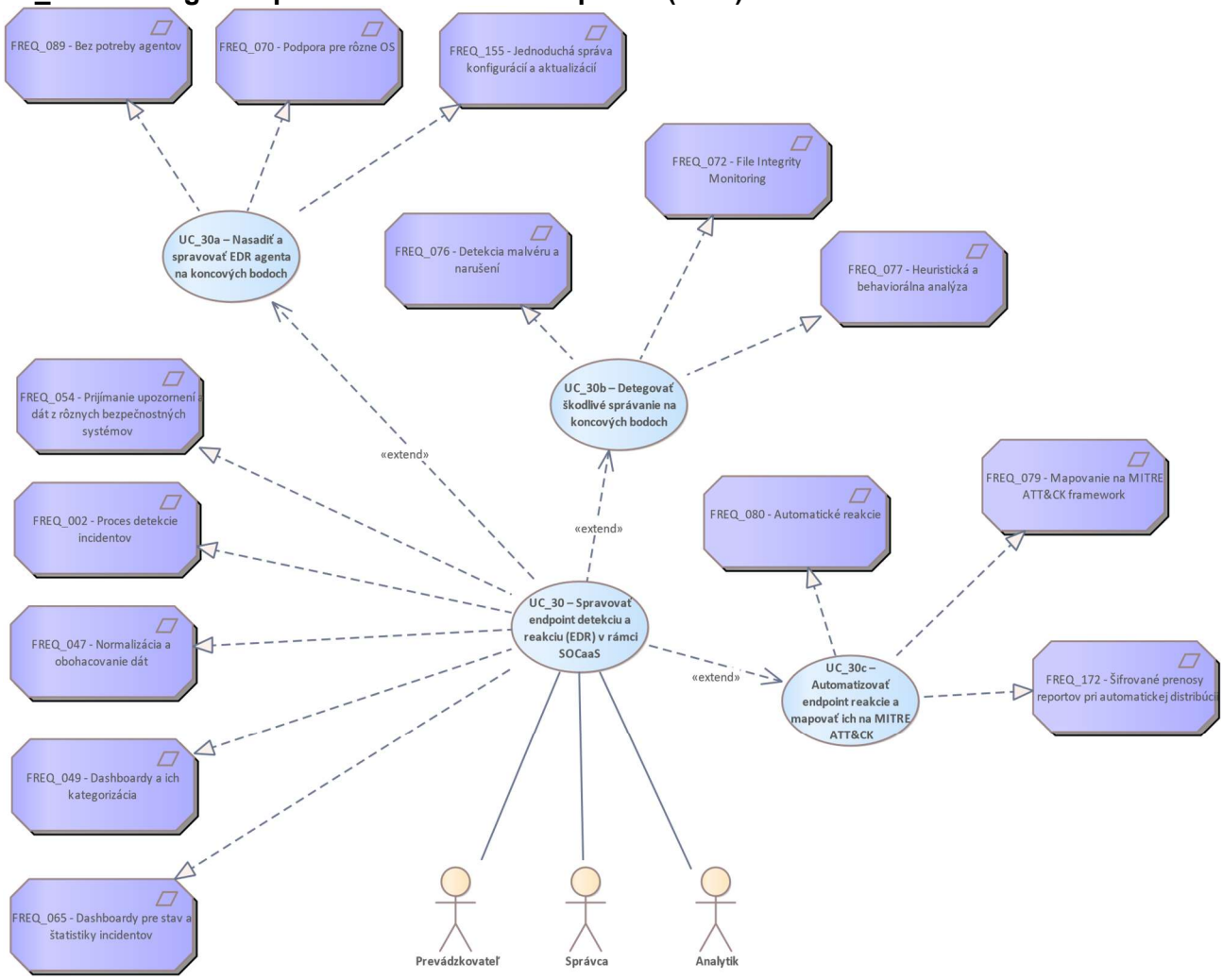
ALT_29_02 – The Analyst recalibrates trust in a specific model	Alternate

Step	**Activity**
1	The Analyst repeatedly labels anomalies detected by one model as “false positive”; the system aggregates this feedback and displays to the Data Scientist a recommendation to review the model (or to replace it with another type, including a change of threshold or architecture).

Linked functional requirements

FREQ_194 - Model explainability	The solution must have an explainable model (interpretability). Possibility to explain the detection output to the analyst. Possibility to provide the analyst with a comprehensible explanation of why the anomaly was detected.
FREQ_191 - Ability to work with time-sequential data	The solution must be able to work with time-sequential data. Suitability for data evolving over time (e.g. logs, network traffic). Ability to capture temporal patterns and anomalies in time series. Architectures such as LSTM and Transformer are an advantage.
FREQ_192 - Flexibility of input data	The solution must be able to ensure flexibility of input data (logs, metrics, flow data). How many types of data the model can handle without the need to modify the architecture. The model must be able to work with both structured and unstructured inputs – logs, metrics, network flows.
FREQ_028 - Processing of data flows	The solution must ensure the processing of data flows. The main tasks of data flow processing are: • Receipt of logs and events from multiple heterogeneous sources • Validation, normalisation and enrichment of data • Assignment of timestamps, identifiers and event categories • Redirection of data to appropriate indices for efficient analysis
	• Filtering out redundant or irrelevant data • Implementation of pipeline rules and processing modules
FREQ_187 - Support for learning without labels	The solution must ensure support for learning without labels (unsupervised). Ability to learn from input data without the need for annotations. Not all models require labelled data. Unsupervised models are advantageous particularly with large volumes of unlabelled data.
UC_29 – Explain and analyse anomalies detected by ML models	UC_29 ensures that SOCaaS is able to explain comprehensibly the anomalies detected by machine learning models: - display the context of the data flow (logs, metrics, network flows), - show the time course of the signals that led to the anomaly, - work with unsupervised models (without labels), - and give the analyst an interpretation of “why the model flagged it”.

UC_30 – Manage endpoint detection and response (EDR) within SOCaaS



UC_30 – Manage endpoint detection and response (EDR) within SOCaaS

Use case	UC_30 – Manage endpoint detection and response (EDR) within SOCaaS
Goal	UC_30 addresses how SOCaaS centrally manages the EDR/XDR layer: - it receives alerts and logs from EDR/antivirus/OS tools, - normalises and enriches them, - displays the status of incidents on dashboards, - connects them to the overall incident detection (UC_03) and response (UC_07) process.

Precondition	An EDR/AV/host-based sensor (or an integration agent) is deployed on the customer's endpoints.

Precondition	SOCaaS has integration connectors for collecting logs and alerts from EDR and other security tools.
Precondition	Basic correlation rules and heuristics for detecting endpoint incidents are defined.
Postcondition	Alerts and logs from endpoint tools are received, normalised and enriched and used in incident detection.
Postcondition	The status of endpoint incidents is clearly displayed on dashboards for analysts and management.
Postcondition	Endpoint incidents are linked to automated responses (UC_07), if this is permitted by policy.

Basic Path	Basic Path

Step	**Activity**
1	The Operator configures the integration of the EDR/AV tool with SOCaaS and defines which types of alerts and logs are to be sent.
2	The system receives alerts and data from various security tools (EDR, AV, OS logs) and places them into a unified processing pipeline.
3	The pipeline transforms the various logs from endpoints into a unified format, adds WHOIS/GEOIP and other context and stores them in appropriate indices for correlation.
4	The detection engine (UC_03) uses the normalised endpoint data in correlation rules and heuristics to identify potential incidents.
5	When an incident originating on an endpoint arises, the system displays it on the dashboards together with its context (host, user, threat type, timeline).
6	The Analyst works with the endpoint incident directly in SOCaaS – filters, sorts, drills down into the details of logs and enrichment, and where appropriate manually escalates or closes it.
7	If a playbook is defined for the given incident type, SOCaaS invokes UC_07 and offers automated or semi-automated responses (isolation, quarantine, blocking).

ALT_30_01 – Incomplete or faulty EDR integration	Alternate

Step	**Activity**
1	The EDR tool does not send all the expected types of alerts or logs; the system notifies the Operator that only partial data are arriving from the given source and recommends checking the configuration or permissions.

ALT_30_02 – Excess of endpoint alerts (alert fatigue)	Alternate

Step	**Activity**
1	After the EDR is connected, an extremely large number of low-severity alerts arrive; the Analyst and the Operator adjust the filters, correlation rules and thresholds so that only relevant incidents are displayed on the dashboards and the rest remain as low-priority events in the background.

Linked functional requirements	
FREQ_049 - Dashboards and their categorisation	Visualisation of data using charts, filters and widgets. Dashboards (including interactive ones) and their categorisation
UC_30c – Automate endpoint responses and map them to MITRE	UC_30c addresses automated/semi-automated responses to endpoint incidents and their mapping to MITRE ATT&CK:

ATT&CK	- device isolation, file quarantine, process blocking, - linkage with SOAR playbooks, - secure transfer of forensic data and reports.
FREQ_065 - Dashboards for incident status and statistics	Clear display of the system status and ongoing incidents. Dashboards for incident status and statistics
UC_30b – Detect malicious behaviour on endpoints	UC_30b addresses the actual detection of malware, intrusions and suspicious behaviour on endpoints: - FIM (File Integrity Monitoring), - malware detection, - heuristic/behavioural analysis.
UC_30a – Deploy and manage the EDR agent on endpoints	UC_30a addresses the deployment and life cycle of the EDR agent / sensor: - installation on various OSs, - updates and configuration, - or the use of agentless mode where an agent is not possible
FREQ_002 - Incident detection process	Ensure the incident detection process – it is responsible for identifying anomalies, attack patterns and other threats by means of correlation rules and heuristic mechanisms.
FREQ_047 - Data normalisation and enrichment	Transformation of various logs into a unified format and their extension (e.g. WHOIS, GEOIP). Data normalisation and enrichment
FREQ_054 - Receiving alerts and data from various security systems	Ability to receive alerts and data from various security sources (SIEM, EDR, CTI...). Receiving alerts and data from various security systems

Use case	UC_30a – Deploy and manage the EDR agent on endpoints
Goal	UC_30a addresses the deployment and life cycle of the EDR agent / sensor: - installation on various OSs, - updates and configuration, - or the use of agentless mode where an agent is not possible

Precondition	The EDR/endpoint tool that will be used in SOCaaS has been selected.
Precondition	Lists of endpoints and their OSs are available.
Postcondition	The EDR agent is deployed or configured (including in agentless mode) on the relevant endpoints.
Postcondition	Agent configurations and updates are managed centrally without code changes.

Basic Path	Basic Path

Step	**Activity**
1	The Administrator obtains an overview of the endpoints and their operating systems that are to be monitored.
2	The Operator prepares installation packages or policies for deploying the EDR agent on the supported OSs.
3	The Administrator deploys the agent (or configures an agentless connection) on the selected endpoints according to the customer's policy.
4	The Operator centrally sets the configuration of the agents – which logs to collect, which modules are active and how often to send data.

5	The system ensures the distribution of agent updates and configurations without the need to change code or intervene manually on each device.

ALT_30a_01 - The agent cannot be installed	Alternate

Step	**Activity**
1	On some devices (e.g. special servers, OT devices) it is not possible to install the agent; the Operator activates agentless mode (e.g. log collection via syslog, WMI, API) and marks these devices in the system as “agentless”.

Linked functional requirements	
FREQ_070 - Support for various OSs	Support for installing and running the agent on various operating systems. Support for various OSs (Windows, Linux, macOS)
FREQ_155 - Simple management of configurations and updates	Ability to manage the platform without the need for code changes. Simple management of configurations and updates
FREQ_089 - No need for agents	Possibility to perform scans without the need to install an agent on devices. No need for agents (agentless)
UC_30a – Deploy and manage the EDR agent on endpoints	UC_30a addresses the deployment and life cycle of the EDR agent / sensor: - installation on various OSs, - updates and configuration, - or the use of agentless mode where an agent is not possible

Use case	UC_30b – Detect malicious behaviour on endpoints
Goal	UC_30b addresses the actual detection of malware, intrusions and suspicious behaviour on endpoints: - FIM (File Integrity Monitoring), - malware detection, - heuristic/behavioural analysis.

Precondition	The EDR/agent is running on the endpoints (UC_30a).
Precondition	Policies are defined as to which types of behaviour are to be considered suspicious.
Postcondition	Suspicious behaviour (malware, intrusions, FIM events) is detected and mapped to incidents in SOCaaS.

Basic Path	Basic Path

Step	**Activity**
1	The EDR agent monitors changes to files, processes and system calls on the endpoint.
2	In the event of a suspicious file change (e.g. a system binary, a critical configuration), it creates an FIM event and sends it to SOCaaS.
3	The EDR engine analyses process and memory behaviour in order to identify signs of malware or a security breach.
4	Heuristic and behavioural rules (or models) evaluate the combination of events and determine whether it is suspicious activity.
5	When thresholds are exceeded or a pattern matches, the EDR creates an alert, which is subsequently processed in UC_30/UC_03 as an incident or a candidate incident.

ALT_30b_01 - Suspicious behaviour without a clear verdict	Alternate

Step	**Activity**
1	The heuristics evaluate the activity as suspicious, but not strong enough for automatic classification; the EDR sends a “low confidence” alert, which is displayed on the dashboard as a low-priority event for manual assessment by an analyst.

Linked functional requirements	
FREQ_077 - Heuristic and behavioural analysis	Basing detection on behaviour and heuristics instead of signatures alone. Heuristic and behavioural analysis
FREQ_072 - File Integrity Monitoring	Detection of file changes in real time for the purposes of forensic analysis and integrity. File Integrity Monitoring (FIM)
FREQ_076 - Malware and intrusion detection	Identification of malicious software and security breach incidents. Malware and intrusion detection
UC_30b – Detect malicious behaviour on endpoints	UC_30b addresses the actual detection of malware, intrusions and suspicious behaviour on endpoints: - FIM (File Integrity Monitoring), - malware detection, - heuristic/behavioural analysis.

Use case	UC_30c – Automate endpoint responses and map them to MITRE ATT&CK
Goal	UC_30c addresses automated/semi-automated responses to endpoint incidents and their mapping to MITRE ATT&CK: - device isolation, file quarantine, process blocking, - linkage with SOAR playbooks, - secure transfer of forensic data and reports.

Precondition	Response playbooks (UC_07) are defined for endpoint incident types.
Precondition	The EDR tool supports automatic actions on endpoints (isolation, quarantine, blocking).
Precondition	SOCaaS has a mapping of activities to MITRE ATT&CK.
Postcondition	Endpoint incidents are mapped to ATT&CK TTPs and used in reports and CTI.
Postcondition	Automated responses are performed securely and auditably, including the secure transfer of reports and forensic data.

Basic Path	Basic Path

Step	**Activity**
1	The Analyst or the system identifies an endpoint incident for which an automated response is appropriate (e.g. malware on a workstation).
2	SOCaaS assigns the relevant MITRE ATT&CK techniques to the incident and displays them in the incident detail.
3	The Analyst launches a playbook or the system automatically evaluates the conditions and performs the action (device isolation, file quarantine, process blocking).

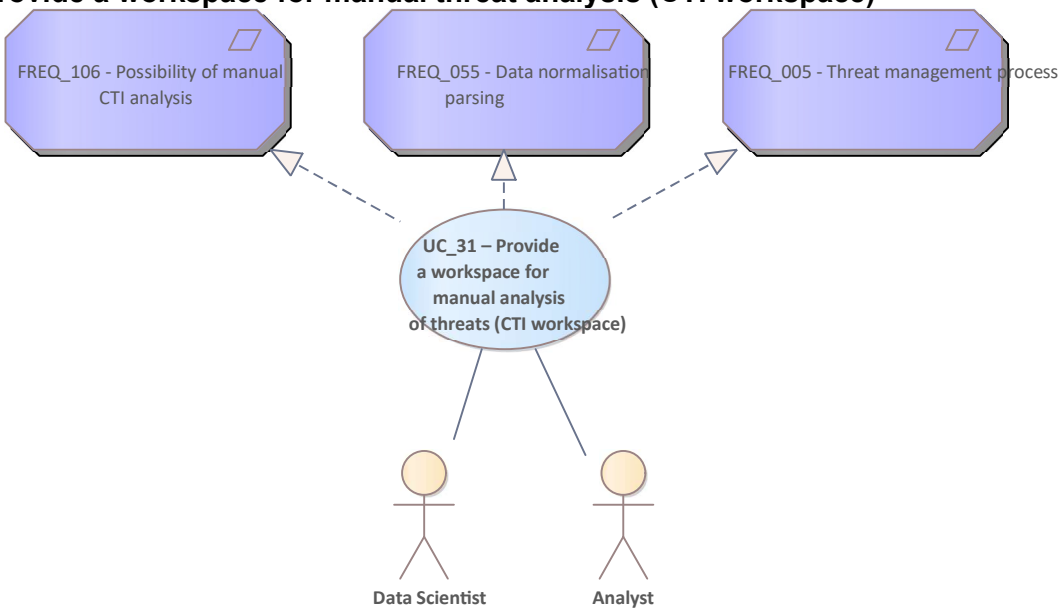
4	The EDR sends confirmation of the result of the action back to SOCaaS, where the incident status is updated and the steps performed are recorded.
5	The system generates a summary report on the course of the incident and the responses performed and sends it securely via an encrypted channel (e.g. to the manager or customer).

ALT_30c_01 - The automated response is blocked by policy	Alternate

Step	**Activity**
1	The Manager has set a policy that fully automatic isolation is not permitted on critical servers; in such an incident, SOCaaS offers only a “semi-automatic” action requiring explicit confirmation by an analyst.

Linked functional requirements	
FREQ_172 - Encrypted report transfers in automatic distribution	Secure transfer of sensitive data via encrypted channels. Encrypted report transfers in automatic distribution
FREQ_079 - Mapping to the MITRE ATT&CK framework	Classification of detected activities and techniques within the MITRE ATT&CK framework for the purposes of TTP analyses. Mapping to the MITRE ATT&CK framework
FREQ_080 - Automatic responses	Automatic response to incidents, such as endpoint isolation or file quarantine. Automatic responses (e.g. device isolation, file quarantine)
UC_30c – Automate endpoint responses and map them to MITRE ATT&CK	UC_30c addresses automated/semi-automated responses to endpoint incidents and their mapping to MITRE ATT&CK: - device isolation, file quarantine, process blocking, - linkage with SOAR playbooks, - secure transfer of forensic data and reports.

UC_31 – Provide a workspace for manual threat analysis (CTI workspace)



UC_31 – Provide a workspace for manual threat analysis (CTI workspace)

Use case	UC_31 – Provide a workspace for manual threat analysis (CTI workspace)
Goal	UC_31 ensures that SOCaaS provides a dedicated workspace for manual threat analysis (CTI): - the analyst can open a specific threat, IOC or campaign, - sees normalised data from multiple CTI sources, - has tools for manual investigation, filtering and adding comments.

Precondition	The CTI pipeline is functional (UC_25) – CTI data have been downloaded and technically processed.
Precondition	The user has a role that allows them to work with CTI data (reading, comments, labels).
Postcondition	For selected threats / IOCs, manually added analyses and comments from the analyst exist.
Postcondition	The results of the manual analysis are stored so that they can be used in incidents, hunting cases and further CTI processing.

Basic Path	Basic Path

Step	**Activity**
1	The Analyst opens the “CTI workspace” module and searches for the threat, IOC or campaign that they want to analyse.
2	The system loads from the CTI repositories all available information on the given threat – IOCs, descriptions, source feeds – and displays it in a unified, normalised format.
3	The Analyst uses the available tools (filtering, pivoting on IOCs, linking to incidents and logs) to manually examine the data and the context of the threat.
4	The Analyst adds their own comments, relevance assessment and recommendations and, where appropriate, flags the threat for specific customers/segments.
5	The system saves the result of the manual analysis as part of the CTI record and makes it available in related incidents, hunting cases and reports.

ALT_31_01 – CTI data for the threat are incomplete or inconsistent	Alternate

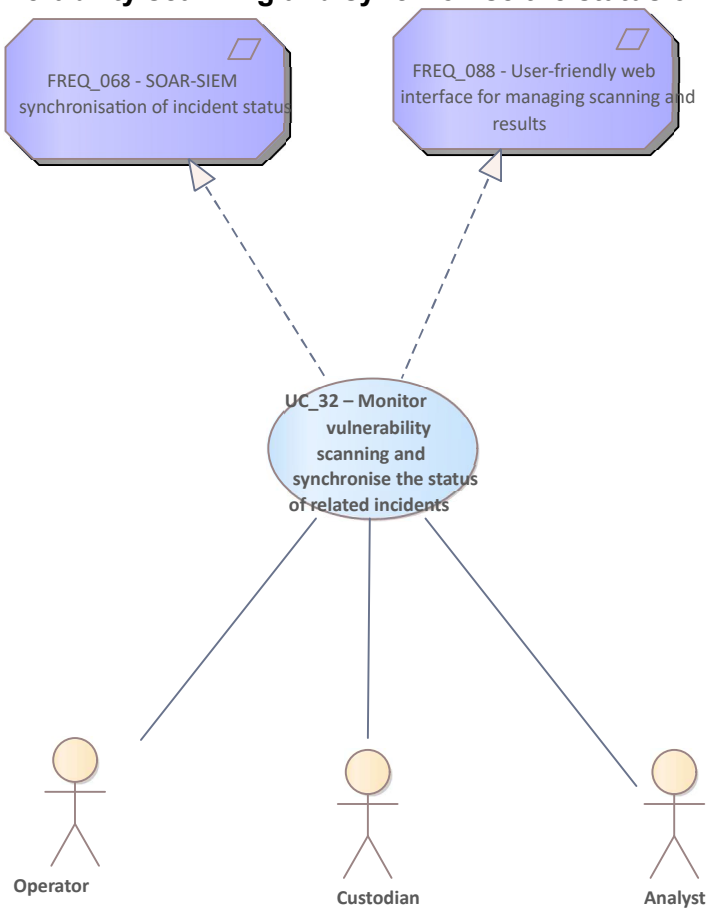
Step	**Activity**
1	In step 2, the Analyst finds that information on the threat from different sources is contradictory or incomplete; they mark the record as “requires completion” and add manual notes, and the CTI specialist later adjusts the sources or scoring.

ALT_31_02 – The Analyst opens the CTI workspace directly from an incident	Alternate

Step	**Activity**
1	While investigating an incident in UC_06, they click on the linked threat/IOC; the system opens the CTI workspace already pre-filtered to the relevant IOCs and campaign, so that the analyst does not have to search manually.

Linked functional requirements	
FREQ_005 - Threat management process	Ensure the threat management process (Threat Intelligence) – it enables systematic work with information on threats.
FREQ_055 - Data normalisation and parsing	Transformation of input data into a standardised format for further processing. Data normalisation and parsing
FREQ_106 - Possibility of manual CTI analysis	Interface and tools for security analysts for manual examination of data. Possibility of manual CTI analysis

UC_32 – Monitor vulnerability scanning and synchronise the status of related incidents



UC_32 – Monitor vulnerability scanning and synchronise the status of related incidents

Use case	UC_32 – Monitor vulnerability scanning and synchronise the status of related incidents
Goal	UC_32 addresses the operational view of vulnerability scanning and related incidents: - it provides a web interface for managing scans and results (on top of the functions of UC_23), - displays which findings have already been converted into incidents,

	- and ensures synchronisation of incident status between SIEM and SOAR so that analysts have a consistent status everywhere.

Precondition	Vulnerability scanning is technically set up and running in accordance with UC_23.
Precondition	An integration link between SIEM and SOAR exists (in accordance with UC_24).
Precondition	The user has permission to access the “Scanning and incidents” module.
Postcondition	Users have a clear web interface through which they can see the scans, their status and results.
Postcondition	Scan findings that are processed as incidents have their status synchronised between SIEM and SOAR (e.g. “open / in progress / resolved”).

Basic Path	Basic Path

Step	**Activity**
1	The Analyst or Administrator opens the “Scanning and incidents” module in the SOCaaS web interface and sees an overview of running and scheduled scans.
2	The user selects a specific scan and displays its detail – scope, start time, status, number of vulnerabilities found and basic statistics.
3	In the same interface, the user displays a list of scan findings, with each finding linked to its current status (new, in progress, closed) and to information on whether an incident already exists for it in SIEM/SOAR.
4	The system shows for which findings an incident has been created in SOAR and automatically verifies the status of that incident in SIEM (e.g. “open”, “assigned”, “resolved”).
5	When the status of an incident changes in SOAR (e.g. after a playbook has been executed), module UC_32 also updates the status of the incident in SIEM via the integration layer, so that analysts and management have consistent information everywhere.
6	The Analyst can filter findings in the GUI by incident status (e.g. “critical findings without an incident” or “incidents in progress”) and thereby prioritise further steps.

ALT_32_01 – An incident exists in SOAR, but SIEM does not know its status	Alternate

Step	**Activity**
1	In step 4 the system detects that an incident exists in SOAR for a specific finding, but there is no corresponding record in SIEM (e.g. an integration error or a historical incident); the system marks the finding as “inconsistent” and notifies the Operator to check the integration rules or align the incidents manually.

ALT_32_02 – The web interface cannot display some of the scans or results	Alternate

Step	**Activity**

1	When loading the overview in step 1 or 2, an API error occurs (e.g. the scanner is unavailable); the UI displays a notification stating which scans/results could not be loaded, keeps the rest of the overview functional and allows the view to be refreshed later.
Linked functional requirements	
FREQ_088 - Acceptable web interface for managing scanning and results	Usability of the GUI for managing scans and results. Acceptable web interface for managing scanning and results
FREQ_068 - SOAR-SIEM incident status synchronisation	Linking of incident status between SIEM and SOAR for up-to-date visibility. SOAR-SIEM incident status synchronisation

5 Data Architecture

The data architecture of the solution represents the design of the logical structure of data, its life cycle, mutual relationships and the principles of data management in the environment of the multi-tenant SOCaaS platform. Its aim is to ensure that the data required for monitoring, detection, correlation, incident management, asset management, vulnerability management, threat intelligence, ML/AI functionality, audit and reporting are stored consistently, securely, auditably and in a way that supports scaling of the service for multiple customers.

The proposed data architecture respects the nature of the solution as an open cloud service with multiple tenants. At the same time, it takes into account the need to separate customer data, manage access permissions, manage the life cycle of incidents and detection rules, retain logs and audit trails, as well as connect to other security domains, for example EDR/XDR, vulnerability scanning, CTI or ML models.

Objectives and principles of the data architecture

The data architecture of the solution is designed according to the following principles:

- Tenant separation and data security.

Each customer organisation represents a separate tenant. Tenant data must be logically separated at the level of database records, indexes, object storage, audit records and reporting outputs. All data objects therefore contain a link to the tenant, or to a team or organisational unit within the tenant. This principle is key for the management of organisations, teams, cases and user permissions.

- Unified reference model of security data.

Data coming from various sources, for example SIEM, EDR, vulnerability scanners, CTI feeds or ML components, must be normalised into a unified logical model after ingestion. This will enable correlation of events, linking of incidents to assets, linking of IOCs to incidents, as well as consistent reporting and audit.

- Auditability and traceability.

All significant operations on data must be retrospectively traceable. This includes not only the security data themselves, but also administrative changes, rule changes, changes to assets, model deployment, IOC distribution and handling of incidents. The data model therefore contains a separate area for audit records and change history.

- Support for the data life cycle.

The solution distinguishes several types of data with differing value and retention rules: configuration data, operational events, alerts, incidents, evidence, CTI objects, vulnerability findings, training datasets, model results and reports. Each group of data has its own life cycle and rules for retention, archiving and deletion.

- Support for performance and scaling.

The architecture must separate high-volume data flows from reference and configuration data. Operational logs, telemetry and alerts are data that are sensitive in terms of both volume and time, whereas tenants, roles, rules,

incidents or assets are reference domain objects. For this reason, a combination of several storage mechanisms according to the nature of the data is proposed.

Data domains of the solution

The proposed data architecture is divided into several main data domains.

Organisational and access domain

This domain covers the management of tenants, organisations, teams, users, roles and permissions. It ensures that it is possible to determine unambiguously which user belongs to which organisation, what permissions they have, which objects they have access to and what actions they can perform. This domain also includes custom user attributes, links to teams and an audit of changes in access rights. This domain is the foundation of multi-tenancy and the RBAC model.

Domain of customer assets and environment

This domain contains the customer's asset inventory and its technical and security context. Servers, workstations, applications, networks, cloud services, identities, endpoints and other protected objects are recorded here. In addition to the list of assets itself, their types, criticality, owners, technical identifiers, life cycle and relationships between assets are also recorded. An extension of this domain is the security status of assets, which includes the patch level, the link to vulnerabilities, recent changes and the dependency topology between assets.

Domain of security events, alerts and detection

This domain represents the core of operational monitoring. It contains received events, their normalised forms, enriched records, resulting alerts, correlation records and metadata on the source, time, severity and links to assets or tenants. The domain also includes the records of detection rules and correlation rules, their versions, statuses, test runs and deployments. The model must support rules based on IOCs, anomalies, threshold values and chaining of events over time.

Domain of incidents and cases

The incident domain covers the entire life cycle of a security incident from its creation to its closure. The incident is recorded here as the main case, together with its category, priority, status, owner, timeline, links to alerts, evidence, tasks, comments, decisions and the outcome of its resolution. The domain also contains the case structure defined at tenant level, incident templates, case types, categories and the rules for sharing them between teams or partners. The aim is to ensure uniform and auditable incident management.

Domain of data retention, protection and audit

This domain ensures the management of retention rules, encryption, archiving and audit records. Retention policies by data type and source, access history, configuration changes, administrative interventions and the audit trail of analysts and operational users are recorded here. The domain is essential for forensic analysis, compliance checks and support for both internal and external audits.

Domain of vulnerabilities and security findings

This domain covers the results of vulnerability scanning and their link to assets, networks and the remediation workflow. Scanning profiles, scan runs, findings, CPE/CVE identifiers, CVSS scores, remediation status and links to incidents or tickets are recorded here. The data model must make it possible to identify which assets are affected, what the severity of the finding is and whether the finding has been handed over for remediation.

Threat intelligence domain

The CTI domain covers external and internal sources of threat intelligence, their processing and distribution to detection and incident management. CTI feeds, indicators of compromise, TTPs, scoring, priority, source reliability, enrichment, mapping to incidents and the audit of distribution to other tools are recorded here. The

data model must support the aggregation of IOCs from multiple sources, deduplication and linking to historical incidents, logs and detection rules.

Domain of ML/AI and model explanation

This domain covers the life cycle of the machine learning models used in the detection of anomalies and incidents. Datasets, training and validation runs, model parameters, versions, quality metrics, robustness to drift, model deployments and inference results are recorded here. A separate part consists of explanations of model outputs, i.e. why a particular flow or event was marked as an anomaly, which input signals led to this and what the context was.

Endpoint telemetry domain

This domain contains alerts, logs and status data from endpoint tools, for example EDR, antivirus or host-based sensors. After ingestion, the data are normalised and linked to incidents, rules and assets. The aim is to enable a unified view of endpoint events together with other security data in SOCaaS.

Domain of reporting and exports

The reporting domain includes reports, dashboards, their templates, export jobs, the archive of outputs, distribution channels and the audit of access to reports. The parameters of time periods, filters used, metric types and links to tenants or roles are also recorded here. The purpose is to ensure repeatable, auditable and personalised reporting.

Design of the logical data model

From the perspective of the logical model, the most important relationships are the following:

- The tenant is the parent object for users, teams, assets, incidents, reports, local rules and retention policies.
- A user is assigned to one or more roles and can be a member of one or more teams.
- An asset belongs to a tenant, has a defined type, owner and criticality and may have links to other assets.
- A security event originates from a telemetry source and may be linked to an asset, identity or endpoint.
- An alert is generated on the basis of one or more rules, or an ML model, and may lead to the creation of an incident.
- An incident aggregates alerts, evidence, comments, tasks, timestamps, links to assets, IOCs and remediation steps.
- A detection rule may be deployed globally or for selected tenants and is versioned.
- A CTI object / IOC may be linked to rules, alerts, incidents and assets.
- A vulnerability finding is bound to an asset or a group of assets and may create or enrich an incident or ticket.
- An ML model uses datasets and produces inference results and explanations, which may be reflected in alerts or incidents.

Key entities of the data model

Entity	Purpose of the entity	Examples of key attributes
Tenant	Represents a customer organisation or a logically separated space of the service	tenant_id, name, status, organisation type, retention profile
Team	Represents an internal or customer team	team_id, tenant_id, name, team type, owner

User	Records of platform users	user_id, tenant_id, name, e-mail, account status, MFA status
Role	Definition of a role in the RBAC model	role_id, name, description, scope
UserRoleAssignment	Link of a user to a role	assignment_id, user_id, role_id, tenant_id, validity
Asset	Records of protected assets	asset_id, tenant_id, name, type, criticality, owner, life cycle status
AssetRelationship	Relationships between assets	relation_id, source_asset_id, target_asset_id, relationship type
Event	Normalised security event	event_id, tenant_id, source_type, timestamp, severity, asset_id, raw_ref
Alert	Result of detection over events	alert_id, tenant_id, rule_id/model_id, severity, status, asset_id
DetectionRule	Detection or correlation rule	rule_id, version, rule type, status, owner, mitre_mapping
DetectionPackage	Package of rules and detection content	package_id, name, version, scope, deployment status
Incident	Main case of a security incident	incident_id, tenant_id, type, category, priority, status, owner
Evidence	Evidence and artefacts in an incident	evidence_id, incident_id, evidence type, location, hash, timestamp
AuditLog	Audit trail of operations	audit_id, actor_id, tenant_id, object, action, timestamp, result
RetentionPolicy	Data retention policy	policy_id, data type, tenant_id/scope, retention period, archiving mode
VulnerabilityFinding	Finding from vulnerability scanning	finding_id, asset_id, cve, cvss, severity, remediation status
ScanJob	Vulnerability scan run	scan_id, profile, scope, start time, status, execution node
CTISource	Threat intelligence source	source_id, name, source type, reliability, periodicity
IOC	Indicator of compromise	ioc_id, type, value, severity, confidence, source_id, validity
TTPMapping	Link to attacker tactics and techniques	mapping_id, ioc_id/rule_id, framework, tactic, technique
MLModel	Records of a machine learning model	model_id, name, version, type, status, deployment date
Dataset	Dataset used for training or validation	dataset_id, source, period, label_info, tenant_scope
ModelEvaluation	Model evaluation results	evaluation_id, model_id, fp, fn, precision, recall, drift_score
PredictionExplanation	Explanation of a model output	explanation_id, model_id, alert_id/event_id, top_features, summary
ReportDefinition	Definition of a report or dashboard	report_id, tenant_id, name, type, template, access_scope

ReportExport	Generated report export	export_id, report_id, format, time, recipient, archived_location

Design of physical data storage

From the perspective of physical storage, it is proposed to divide the data into several layers according to their nature:

Relational storage will be used for reference and configuration data, i.e. tenants, users, roles, teams, assets, rules, incidents, policies, reports and audit records. This type of data requires transactional consistency, referential integrity and support for complex filters and relationships.

Indexing or analytical storage will be used for security events, alerts, endpoint telemetry and other high-volume time-series records. This layer must support fast search, aggregations, filtering by time and tenant scoping.

Object storage will be used for raw logs, archive data, evidence, report exports, model artefacts and larger datasets for ML training.

An optional graph model or graph representation may be used for relationships between assets, service dependencies, incident propagation, TTP mapping and enriched impact analysis.

Such a division will make it possible to separate high-volume operational data from master data and at the same time will support performance, scalability and simpler retention policies.

Key rules of data integrity and security

When implementing the data architecture, the following rules in particular must be observed:

- each data object must have a unique identifier,
- objects bound to a customer must contain a link to the tenant,
- all changes to reference and security-relevant objects must be audited,
- sensitive data must be protected both in transit and at rest,
- retention rules must be defined according to the data type and regulatory requirements,
- incidents, alerts, assets, IOCs and vulnerability findings must be mutually linkable,
- deployments of rules and models must be versioned and retrospectively traceable,
- reports and exports must respect the access model and the scope of tenants.

6 Application Architecture

The application architecture of the solution represents the logical design of the application services, components and their mutual relationships that provide the functionality of the proposed SOCaaS platform. Its purpose is to provide a comprehensive view of the application layer of the solution, in particular in terms of the division of responsibilities between individual components, their functional focus and the way in which they jointly support the main processes of security monitoring, detection, evaluation and resolution of security events and incidents. The architecture is based on the model prepared in the EA tool, with the descriptions of individual objects derived from the notes assigned to the modelled elements.

The proposed application architecture reflects the nature of the solution as a multi-tenant cybersecurity operations centre service. From the perspective of the application layer, it covers in particular the areas of collection and correlation of security events, response automation, incident management, threat intelligence,

asset management, vulnerability management, reporting and support for knowledge management of the SOC team. The application layer defined in this way also creates the precondition for supporting key use cases, such as monitoring of security events, management of detection rules, escalation and resolution of incidents, threat intelligence management, asset management, vulnerability management, report export and management of users, roles and organisations within the SOCaaS platform.

In the document, the application architecture is presented as a high-level list of the application components of the solution. The application viewpoints diagram shows the main components which together form the functional basis of the platform. The available model shows that the core of the solution is formed mainly by the components SIEM, SOAR, CTI, EDR/XDR, the analytical and automation engine, incident recording and management, the ticketing system, management of cybersecurity incidents, asset inventory, the vulnerability scanner, reporting and the knowledge portal for the SOC team. The diagram therefore does not represent a detailed design of internal integration flows, but above all an overview of the main application building blocks of the solution.

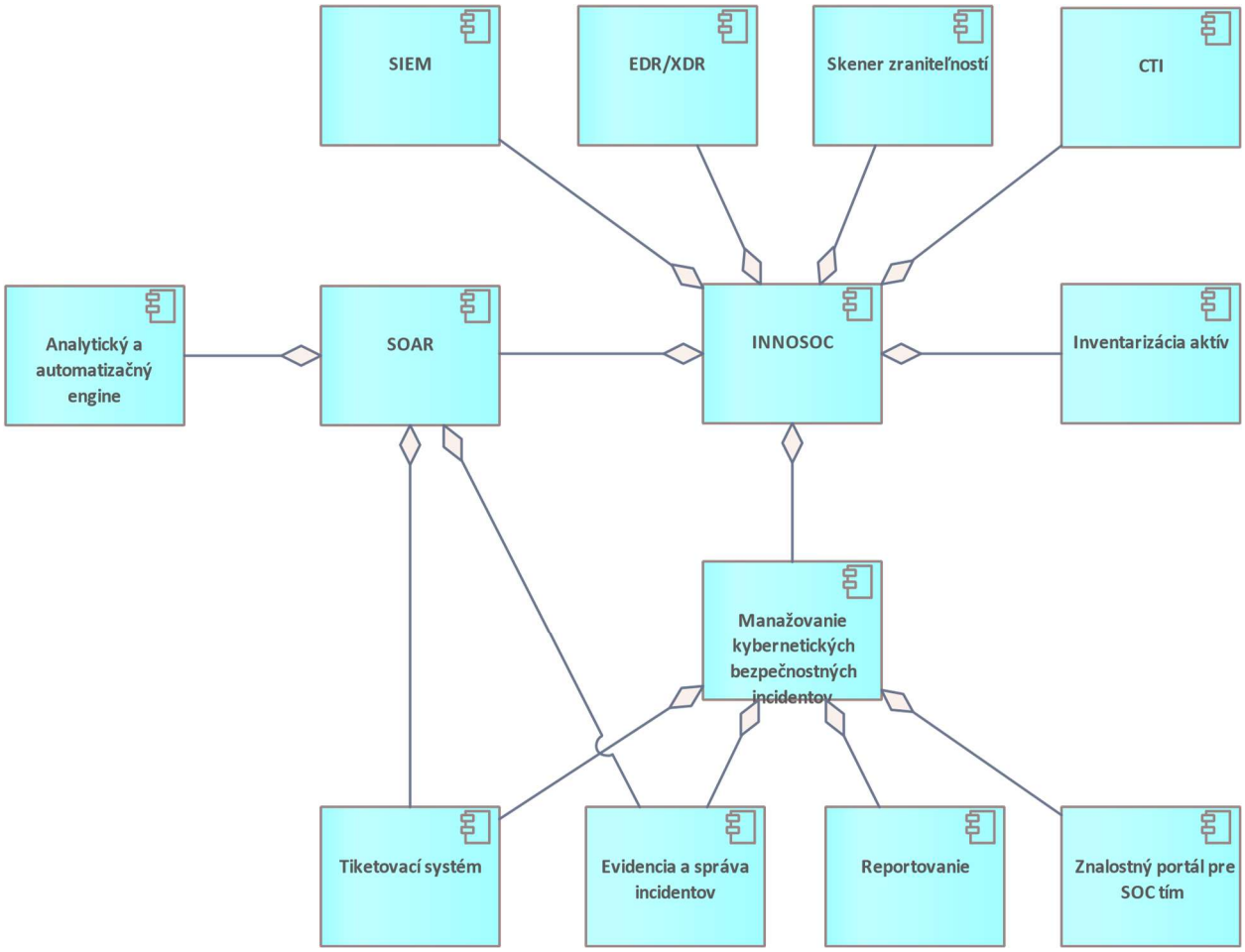
From a functional point of view, the individual application components can be divided into several logical groups. The first group consists of components oriented towards collection, analysis and detection, i.e. mainly SIEM, EDR/XDR, CTI and the analytical and automation engine. These components ensure the reception and processing of security data, their enrichment with threat context, correlation of events and identification of anomalies or suspicious states. The second group consists of components oriented towards response and management of the incident life cycle, above all SOAR, incident recording and management, the ticketing system and management of cybersecurity incidents. Their role is to enable systematic recording, coordination, automation and management of the response to incidents from their occurrence to their closure.

A further area of the application architecture consists of supporting and contextual components, in particular asset inventory, the vulnerability scanner, reporting and the knowledge portal for the SOC team. Asset inventory provides an overview of managed devices, systems and resources, thereby creating the basic context for monitoring and assessing the impacts of incidents. The vulnerability scanner supports a proactive approach to security and provides inputs for the prioritisation of remedial measures. Reporting ensures the aggregation, visualisation and distribution of security information to various target groups. The knowledge portal serves as a central place for storing playbooks, analytical procedures, recommendations and experience from incident resolution.

A special place in the application architecture is held by the analytical and automation engine which, according to the description in the model, combines data analysis and the automation of decision-making or operational processes. In the context of the solution, it is a component supporting the processing of input data, the application of analytical rules or models and the subsequent execution of automated responses via integrated tools and interfaces. This element is functionally linked to the objective of the solution to use elements of automation and machine learning methods in the detection and evaluation of security events.

From the perspective of the overall architecture of the solution, the application layer creates a bridge between the business layer and the technology layer. On the one hand, it builds on the defined services and use cases; on the other hand, it creates the basis for the subsequent technology design of specific platforms and tools. Its importance lies in the fact that it names the main application responsibilities of the solution and creates a structured framework for the implementation, integration and further development of the individual functional areas of the SOCaaS platform.

6.1 Application Viewpoints



Application Viewpoints Components

The diagram represents an HL list of the application components of the solution.

Description of the elements used:

Analytical and automation engine

The analytical and automation engine is a software component or service that combines data analysis and the automation of decision-making or operational processes with the aim of increasing the efficiency, accuracy and speed of the system's response. In the context of cybersecurity or IT operations, this engine processes input data (e.g. logs, events, metrics), applies analytical rules or models (e.g. anomaly detection, correlation) and subsequently performs automated responses or actions via integrated tools and API interfaces.

CTI

CTI (Cyber Threat Intelligence) is a key building block of the SOCaaS architecture, as it provides the ability to enrich detection rules, correlation mechanisms and playbooks with external information about threats – including

IOCs, TTPs, attacker profiling, campaigns and related vulnerabilities. In the environment of the multi-customer SOCaaS service, it plays a role in:

- Distribution and management of threat feeds across tenants,
- Automatic enrichment of SIEM and EDR data with known threats,
- Creation of attack timelines and mapping to MITRE ATT&CK techniques,
- Support for threat hunting and forensic analysis processes,
- Prioritisation of incidents according to threat context.

For the solution design, open-source platforms were selected that cover various aspects of CTI – from IOC collection through visualisation to STIX/TAXII compatibility.

EDR/XDR

EDR (Endpoint Detection and Response) and XDR (Extended Detection and Response) represent the fundamental technological pillars for threat detection at endpoint level. In the SOCaaS environment, they are used for the collection and correlation of, and response to, events that would otherwise escape traditional perimeter detection. XDR extends the EDR concept with the ability to integrate data from networks, clouds, e-mail systems and other layers.

Incident recording and management

Incident recording and management is a core element of every security operation (SOC). This subcomponent ensures the systematic documentation, tracking and management of security events and incidents throughout their life cycle – from detection through analysis to closure. A high-quality recording system must enable teams to collaborate in a coordinated way, document steps, record related artefacts (observables) and link incidents

with relevant threats, systems or historical cases. Ideally, it also supports advanced data enrichment, audit, multi-tenancy and the possibility of retrospective investigation of incidents.

INNOSOC

Represents a comprehensive aggregated component.

Asset inventory

Asset inventory is a fundamental pillar of every security solution. It enables the SOC team to obtain an overview of all devices, system entities, applications and virtual or cloud resources that make up the organisation's infrastructure. In the SOCaaS environment it plays an even more significant role, as it enables centralised management, categorisation, monitoring and tracking of the security status of the infrastructure across different customers.

Management of cybersecurity incidents

Effective resolution of cyber incidents is a key pillar of the SOCaaS architecture. It does not include only the recording of events itself, but a comprehensive cycle of their recording, management, response, reporting and knowledge retention. For the SOC team, it is essential that incidents are managed systematically, with an emphasis on traceability, analytical work, coordination and knowledge support. An effective and coordinated response to security incidents is an essential pillar of every SOCaaS solution. Within the INNOSOC architecture, four subcomponents have been identified which together form an integrated system for the management, recording, documentation and resolution of incidents:

1. Ticketing system
2. Incident recording and management
3. Reporting
4. Knowledge portal for the SOC team

Reporting

The reporting component within the SOCaaS architecture serves as a central means for the aggregation, visualisation and distribution of security information. Its role is to provide both regular and ad hoc outputs on the

state of cybersecurity for various target groups – from SOC analysts through team leaders to the management of the organisation or clients. Such a reporting system must cover:

- Generation of metrics on incidents, alerts, SLAs and KPIs,
- Overviews of trends over time and evaluation of the performance of security measures,
- Adaptation of content according to the level of detail (technical vs. strategic),
- Automation and scheduling of the distribution of outputs,
- Retrospective analysis and archiving of historical data.

SIEM

SIEM is the heart of the SOCaaS solution. It ensures the aggregation, processing, correlation and analysis of security events from various sources in real time. It enables analysts to identify anomalies, generate alerts and escalate incidents, while increasing visibility over the client's infrastructure.

Vulnerability scanner

The vulnerability scanner is a fundamental building block for ensuring a proactive approach in the SOCaaS architecture. It makes it possible to regularly identify weak points in the infrastructure, assess the state of updates, configurations and settings, and provide information for further prioritisation and planning of remedial measures. The scanner's outputs are often a source for SOAR systems, SLA metrics and reporting, as well as for assessing the client's security level.

SOAR

SOAR (Security Orchestration, Automation and Response) is the component responsible for automating responses to incidents, orchestrating security tools and supporting analysts' decision-making. In SOCaaS it represents a fundamental component enabling response capacities to be scaled without the need to increase the number of staff. It is essential particularly when processing a large number of alerts, where it reduces the MTTR and MTTC metrics.

Ticketing system

The ticketing system serves as the central communication and organisational channel for the management of security incidents throughout their entire life cycle. It enables the creation, assignment and tracking of tasks and communication between analysts, while supporting audit, SLA monitoring, escalation and notifications.

Knowledge portal for the SOC team

The knowledge portal for the SOC team is an important building block of every modern security operations centre. It serves as a central repository of know-how, playbooks, procedures, recommendations, analytical guides and often also historical lessons learned from incidents. The knowledge portal:

- Increases the efficiency and consistency of incident response,
- Enables rapid onboarding of new team members,
- Supports standardisation of processes and methodologies,
- Enables the team to retain and share proven procedures and its own analytical experience,
- Strengthens the ability to learn collectively from incidents and to build internal "threat intel" knowledge.

7 Technology Architecture

The technology architecture represents a structured design of technology components, their mutual relationships and the way they are integrated into the overall ecosystem of the organisation. It includes the definition of software layers, communication protocols, security mechanisms and operational processes. Its aim is to ensure the optimal functioning of the system, its scalability, interoperability and ability to respond to changing requirements of the business and regulatory environment.

The project responds to the needs mainly of small and medium-sized organisations, which often do not have sufficient expert capacity or budget to build their own security centre. The proposed solution provides centralised

management, automated detection of and response to incidents, monitoring and reporting, thereby reducing the risk of cyber attacks and increasing the level of compliance.

7.1 Design of the Technology Architecture

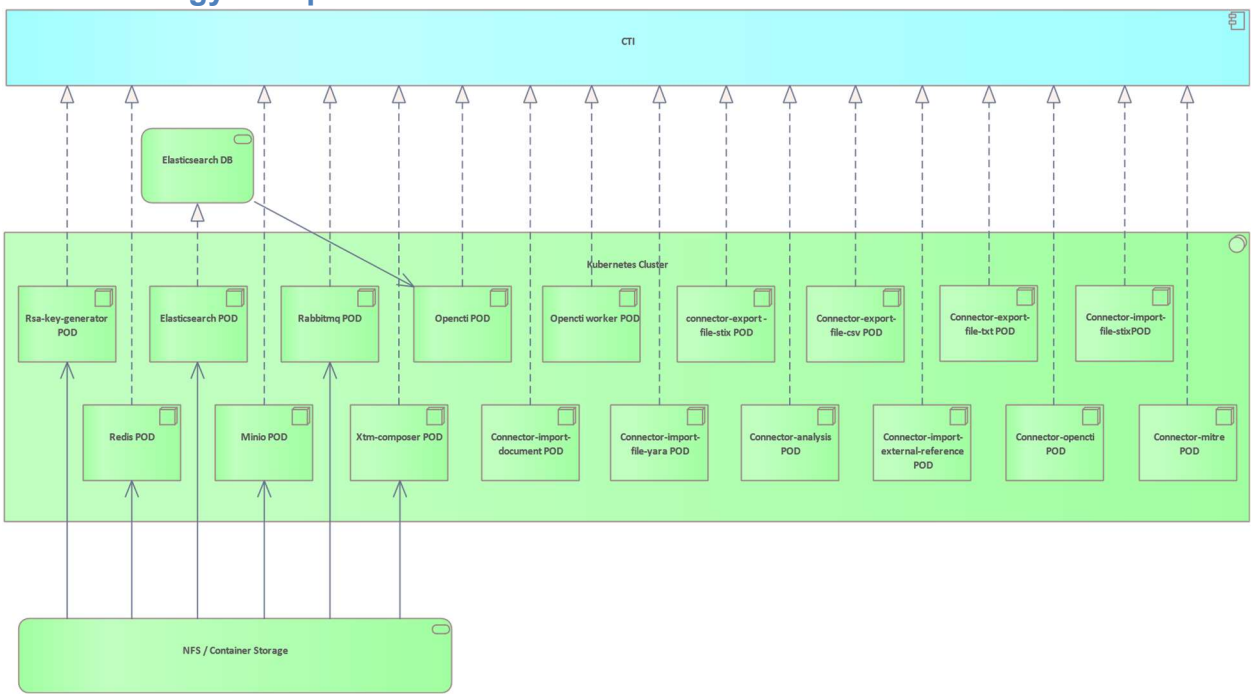
The technology architecture determines the method of deployment, operation and interconnection of the individual technical components, including containerised services, data storage, network layers and orchestration tools. It is designed with an emphasis on modularity, scalability, availability and security, while taking into account the requirements placed on a multi-tenant operational environment.

This section elaborates specific technology elements – such as Kubernetes clusters, database services, CI/CD tools, container registries, proxy layers and security mechanisms – and their links to application components. The subchapter also provides context for further technical specifications of the solution, including architecture views that map functional units such as CTI, EDR, SOAR and asset inventory, as well as data collection and analysis. The design of the technology architecture thus creates a reliable and well-manageable basis for the implementation of the entire solution in accordance with the defined business objectives.

The proposed operation is planned on two identical environments differing only in the quantity and type of resources:

- **PROD** environment – the production environment with real data and operation of the public portal.
- **TEST** environment – the testing, training and pre-production environment. The entire TEST environment is accessible via VPN for privileged users.

7.1.1 Technology Viewpoint CTI



Technology Usage Viewpoint CTI alt 1

The diagram shows a simplified assignment of the individual PODs to the corresponding application components. Alt 1 represents the use of the Elasticsearch database directly within the module setup. The design envisages a containerised version of the NoSQL databases for Elasticsearch.

Description of the elements used:

CTI

CTI (Cyber Threat Intelligence) is a key building block of the SOCaaS architecture, as it provides the ability to enrich detection rules, correlation mechanisms and playbooks with external information about threats – including IOCs, TTPs, attacker profiling, campaigns and related vulnerabilities. In the environment of the multi-customer SOCaaS service, it plays a role in:

- Distribution and management of threat feeds across tenants,
- Automatic enrichment of SIEM and EDR data with known threats,
- Creation of attack timelines and mapping to MITRE ATT&CK techniques,
- Support for threat hunting and forensic analysis processes,
- Prioritisation of incidents according to threat context.

For the solution design, open-source platforms were selected that cover various aspects of CTI – from IOC collection through visualisation to STIX/TAXII compatibility.

NFS / Container Storage

Represents an NFS service or other Container Storage. NFS (Network File System) is a network file protocol at the application layer of the OSI model that enables remote access to file systems over a network. It is one of the most common forms of shared storage in containerised environments, such as Kubernetes, where it is implemented as a PersistentVolume (PV) type. In a broader sense, container storage refers to mechanisms and services providing storage space for containers that need persistent or shared data between restarts or between cluster nodes.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

Elasticsearch DB

The NoSQL Elasticsearch database technology service is a powerful and distributed system designed for indexing, searching and analysing data in real time, based on a document-oriented model. It uses a JSON document schema, supports horizontal scaling and high availability, and is optimised for processing large volumes of unstructured or semi-structured data, such as logs, metrics or application records.

Elasticsearch POD

The Elasticsearch POD for the NoSQL database is a container instance deployed in a Kubernetes environment that hosts an Elasticsearch node as part of a distributed NoSQL database cluster. This POD ensures indexing, searching and storage of document-oriented data, while enabling horizontal scaling, high availability and efficient processing of large volumes of unstructured information in real time.

Connector-analysis POD

The connector-analysis POD is a container unit deployed in a Kubernetes cluster that hosts an analytical connector for OpenCTI, designed for automated processing, analysis and enrichment of threat intelligence. This POD receives input data from OpenCTI, performs analytical operations on them (e.g. reputation checks, IoC processing or entity correlation) and returns the results to the system for further intelligence activity and decision-making.

Connector-export- file-csv POD

The connector-export-file-csv POD is a container unit deployed in a Kubernetes cluster that hosts an export connector for OpenCTI, designed for generating and storing data in CSV (Comma-Separated Values) format.

This POD enables automated export of intelligence, such as threat indicators, attack techniques or actors, to CSV files, thereby supporting their easy integration with external systems, reporting or manual analysis.

Connector-export -file-stix POD

The connector-export-file-stix POD is a container unit deployed in a Kubernetes cluster that hosts a connector for OpenCTI designed for exporting data to STIX (Structured Threat Information Expression) format. This POD enables automated generation and storage of threat information in STIX 2.1 format, thereby supporting standardised sharing of cybersecurity data with other systems, partners or communities.

Connector-export- file-txt POD

The connector-export-file-txt POD is a container unit deployed in a Kubernetes cluster that hosts an export connector for OpenCTI, designed for generating text files (TXT) containing selected threat information. This POD enables automated export of indicators of compromise (IoC) and other CTI data in the form of simple, readable text output for further use in external systems, tools or manual processes.

Connector-import- document POD

The connector-import-document POD is a container unit deployed in a Kubernetes cluster that hosts a connector for OpenCTI, designed for automated import of documents containing intelligence or security information. This POD processes files in formats such as PDF, DOCX or TXT, extracts relevant data from them and subsequently imports them into the OpenCTI platform as structured objects for further analysis and correlation.

Connector-import- external-reference POD

The connector-import-external-reference POD is a container unit deployed in a Kubernetes cluster that hosts a connector for OpenCTI focused on importing external references from public or private sources of threat information. This POD regularly loads metadata and links to reports, articles or analytical documents, converts them into structured CTI (Cyber Threat Intelligence) objects and integrates them into OpenCTI for the purposes of correlation and analytical enrichment of the threat profile.

Connector-import- file-stixPOD

The connector-import-file-stix POD is a container unit deployed in a Kubernetes cluster that hosts an import connector for OpenCTI, focused on loading and processing files in STIX (Structured Threat Information Expression) format. This POD automatically imports threat information from STIX 2.0/2.1 files into OpenCTI, where it is subsequently transformed into structured CTI objects and integrated into existing data for the purposes of correlation, visualisation and intelligence analysis.

Connector-import- file-yara POD

The connector-import-file-yara POD is a container unit deployed in a Kubernetes cluster that hosts a connector for OpenCTI focused on importing YARA files – rules used for the detection of malware and threats. This POD automatically loads YARA files from a defined repository, analyses their content and converts them into structured threat objects, which it then imports into OpenCTI for the purposes of managing and correlating intelligence.

Connector-mitre POD

The connector-mitre POD is a container unit deployed in a Kubernetes cluster that hosts a connector for OpenCTI designed for automated import of data from the MITRE ATT&CK framework. This POD regularly downloads and synchronises the official tactics, techniques and procedures (TTPs), groups, software and campaigns from the MITRE database and subsequently transforms them into structured CTI objects in OpenCTI, thereby supporting accurate and up-to-date analysis of cyber threats.

Connector-opencti POD

The connector-opencti POD is a container unit deployed in a Kubernetes cluster that hosts the main OpenCTI connector, ensuring bidirectional communication between the OpenCTI platform and the individual connectors. This POD serves as middleware that registers connectors, coordinates their activity and ensures the exchange

of data via the OpenCTI API and a message broker (e.g. RabbitMQ), thereby enabling scalable, modular and efficient processing of cybersecurity information.

Minio POD

The MinIO POD is a container unit deployed in a Kubernetes cluster that hosts the MinIO service – a high-performance, scalable and S3-compatible object storage system. This POD provides storage for unstructured data, such as logs, backups, multimedia files or outputs of analytical tools, while supporting high availability, replication and integration with modern cloud-native applications.

Opencti POD

The OpenCTI POD is a container unit deployed in a Kubernetes cluster that hosts the OpenCTI (Open Cyber Threat Intelligence) platform – an open-source solution for the management, analysis and correlation of cyber threats and intelligence (Threat Intelligence). This POD provides a central hub for collecting, visualising and linking information about threats, including entities such as TTPs (tactics, techniques, procedures), indicators of compromise (IoC) or threat actors, and supports integration with other security tools via API.

Opencti worker POD

The OpenCTI Worker POD is a container unit deployed in a Kubernetes cluster that performs asynchronous processing of tasks and data operations within the OpenCTI platform. This POD ensures indexing, transformation, correlation and enrichment of cybersecurity information, while working closely with the main OpenCTI server and the connectors to ensure smooth data integration and distribution of computation.

Rabbitmq POD

The RabbitMQ POD is a container unit deployed in a Kubernetes cluster that hosts the RabbitMQ server – a reliable and widely used message broker implementing protocols such as AMQP, MQTT or STOMP. This POD ensures asynchronous communication and message exchange between services and components of the system, thereby supporting scalability, resilience and efficient processing of tasks in distributed applications.

Redis POD

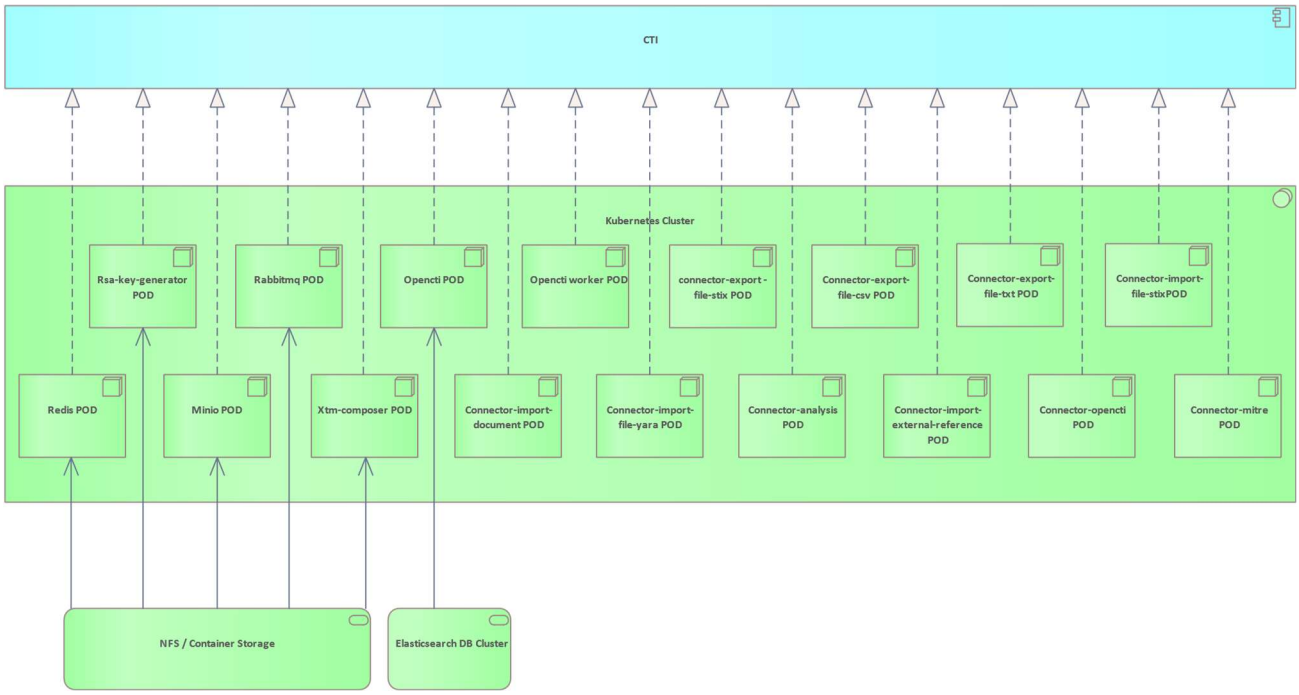
The redis POD is a container unit deployed in a Kubernetes cluster that hosts the Redis server – a high-performance in-memory key-value NoSQL database. This POD is used for fast storage, reading and manipulation of temporary or persistent data, often used for caching, message queuing, session storage and sharing of state between microservices.

Rsa-key-generator POD

The rsa-key-generator POD is a container unit deployed in a Kubernetes cluster that is used to generate RSA key pairs for the purposes of encryption, digital signing or secure communication. This POD ensures automated and secure creation of cryptographic keys, which can subsequently be used by other services within the infrastructure, for example for TLS certificates, GPG signatures or authentication.

Xtm-composer POD

The xtm-composer POD is a container unit deployed in a Kubernetes cluster that hosts the XTM Composer service – a component designed for the composition and generation of data or security outputs within a larger system, often related to cybersecurity management or automated reporting. This POD ensures the processing of input data, their transformation and output in a predefined format, and may cooperate with other components such as databases, analytical tools or API gateways.



Technology Usage Viewpoint CTI alt 2

The diagram shows a simplified assignment of the individual PODs to the corresponding application components. Alt 2 represents the use of the Elasticsearch database directly in the form of a centralised DB cluster.

Description of the elements used:

CTI

CTI (Cyber Threat Intelligence) is a key building block of the SOCaaS architecture, as it provides the ability to enrich detection rules, correlation mechanisms and playbooks with external information about threats – including IOCs, TTPs, attacker profiling, campaigns and related vulnerabilities. In the environment of the multi-customer SOCaaS service, it plays a role in:

- Distribution and management of threat feeds across tenants,
- Automatic enrichment of SIEM and EDR data with known threats,
- Creation of attack timelines and mapping to MITRE ATT&CK techniques,
- Support for threat hunting and forensic analysis processes,
- Prioritisation of incidents according to threat context.

For the solution design, open-source platforms were selected that cover various aspects of CTI – from IOC collection through visualisation to STIX/TAXII compatibility.

NFS / Container Storage

Represents an NFS service or other Container Storage. NFS (Network File System) is a network file protocol at the application layer of the OSI model that enables remote access to file systems over a network. It is one of the most common forms of shared storage in containerised environments, such as Kubernetes, where it is implemented as a PersistentVolume (PV) type. In a broader sense, container storage refers to mechanisms and services providing storage space for containers that need persistent or shared data between restarts or between cluster nodes.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

Elasticsearch DB Cluster

The NoSQL Elasticsearch database technology service in the form of a cluster represents a multi-node distributed solution in which individual nodes cooperate in storing, indexing and searching document-oriented data. Each node in the cluster performs a specific role (e.g. master, data or ingest node), which enables horizontal scaling, high availability and resilience to outages, while the system as a whole ensures the processing of a large volume of data in real time.

Connector-analysis POD

The connector-analysis POD is a container unit deployed in a Kubernetes cluster that hosts an analytical connector for OpenCTI, designed for automated processing, analysis and enrichment of threat intelligence. This POD receives input data from OpenCTI, performs analytical operations on them (e.g. reputation checks, IoC processing or entity correlation) and returns the results to the system for further intelligence activity and decision-making.

Connector-export- file-csv POD

The connector-export-file-csv POD is a container unit deployed in a Kubernetes cluster that hosts an export connector for OpenCTI, designed for generating and storing data in CSV (Comma-Separated Values) format. This POD enables automated export of intelligence, such as threat indicators, attack techniques or actors, to CSV files, thereby supporting their easy integration with external systems, reporting or manual analysis.

Connector-export -file-stix POD

The connector-export-file-stix POD is a container unit deployed in a Kubernetes cluster that hosts a connector for OpenCTI designed for exporting data to STIX (Structured Threat Information Expression) format. This POD enables automated generation and storage of threat information in STIX 2.1 format, thereby supporting standardised sharing of cybersecurity data with other systems, partners or communities.

Connector-export- file-txt POD

The connector-export-file-txt POD is a container unit deployed in a Kubernetes cluster that hosts an export connector for OpenCTI, designed for generating text files (TXT) containing selected threat information. This POD enables automated export of indicators of compromise (IoC) and other CTI data in the form of simple, readable text output for further use in external systems, tools or manual processes.

Connector-import- document POD

The connector-import-document POD is a container unit deployed in a Kubernetes cluster that hosts a connector for OpenCTI, designed for automated import of documents containing intelligence or security information. This POD processes files in formats such as PDF, DOCX or TXT, extracts relevant data from them and subsequently imports them into the OpenCTI platform as structured objects for further analysis and correlation.

Connector-import- external-reference POD

The connector-import-external-reference POD is a container unit deployed in a Kubernetes cluster that hosts a connector for OpenCTI focused on importing external references from public or private sources of threat information. This POD regularly loads metadata and links to reports, articles or analytical documents, converts them into structured CTI (Cyber Threat Intelligence) objects and integrates them into OpenCTI for the purposes of correlation and analytical enrichment of the threat profile.

Connector-import- file-stixPOD

The connector-import-file-stix POD is a container unit deployed in a Kubernetes cluster that hosts an import connector for OpenCTI, focused on loading and processing files in STIX (Structured Threat Information Expression) format. This POD automatically imports threat information from STIX 2.0/2.1 files into OpenCTI, where it is subsequently transformed into structured CTI objects and integrated into existing data for the purposes of correlation, visualisation and intelligence analysis.

Connector-import- file-yara POD

The connector-import-file-yara POD is a container unit deployed in a Kubernetes cluster that hosts a connector for OpenCTI focused on importing YARA files – rules used for the detection of malware and threats. This POD automatically loads YARA files from a defined repository, analyses their content and converts them into structured threat objects, which it then imports into OpenCTI for the purposes of managing and correlating intelligence.

Connector-mitre POD

The connector-mitre POD is a container unit deployed in a Kubernetes cluster that hosts a connector for OpenCTI designed for automated import of data from the MITRE ATT&CK framework. This POD regularly downloads and synchronises the official tactics, techniques and procedures (TTPs), groups, software and campaigns from the MITRE database and subsequently transforms them into structured CTI objects in OpenCTI, thereby supporting accurate and up-to-date analysis of cyber threats.

Connector-opencti POD

The connector-opencti POD is a container unit deployed in a Kubernetes cluster that hosts the main OpenCTI connector, ensuring bidirectional communication between the OpenCTI platform and the individual connectors. This POD serves as middleware that registers connectors, coordinates their activity and ensures the exchange of data via the OpenCTI API and a message broker (e.g. RabbitMQ), thereby enabling scalable, modular and efficient processing of cybersecurity information.

Minio POD

The MinIO POD is a container unit deployed in a Kubernetes cluster that hosts the MinIO service – a high-performance, scalable and S3-compatible object storage system. This POD provides storage for unstructured data, such as logs, backups, multimedia files or outputs of analytical tools, while supporting high availability, replication and integration with modern cloud-native applications.

Opencti POD

The OpenCTI POD is a container unit deployed in a Kubernetes cluster that hosts the OpenCTI (Open Cyber Threat Intelligence) platform – an open-source solution for the management, analysis and correlation of cyber threats and intelligence (Threat Intelligence). This POD provides a central hub for collecting, visualising and linking information about threats, including entities such as TTPs (tactics, techniques, procedures), indicators of compromise (IoC) or threat actors, and supports integration with other security tools via API.

Opencti worker POD

The OpenCTI Worker POD is a container unit deployed in a Kubernetes cluster that performs asynchronous processing of tasks and data operations within the OpenCTI platform. This POD ensures indexing, transformation, correlation and enrichment of cybersecurity information, while working closely with the main OpenCTI server and the connectors to ensure smooth data integration and distribution of computation.

Rabbitmq POD

The RabbitMQ POD is a container unit deployed in a Kubernetes cluster that hosts the RabbitMQ server – a reliable and widely used message broker implementing protocols such as AMQP, MQTT or STOMP. This POD ensures asynchronous communication and message exchange between services and components of the system, thereby supporting scalability, resilience and efficient processing of tasks in distributed applications.

Redis POD

The redis POD is a container unit deployed in a Kubernetes cluster that hosts the Redis server – a high-performance in-memory key-value NoSQL database. This POD is used for fast storage, reading and manipulation of temporary or persistent data, often used for caching, message queuing, session storage and sharing of state between microservices.

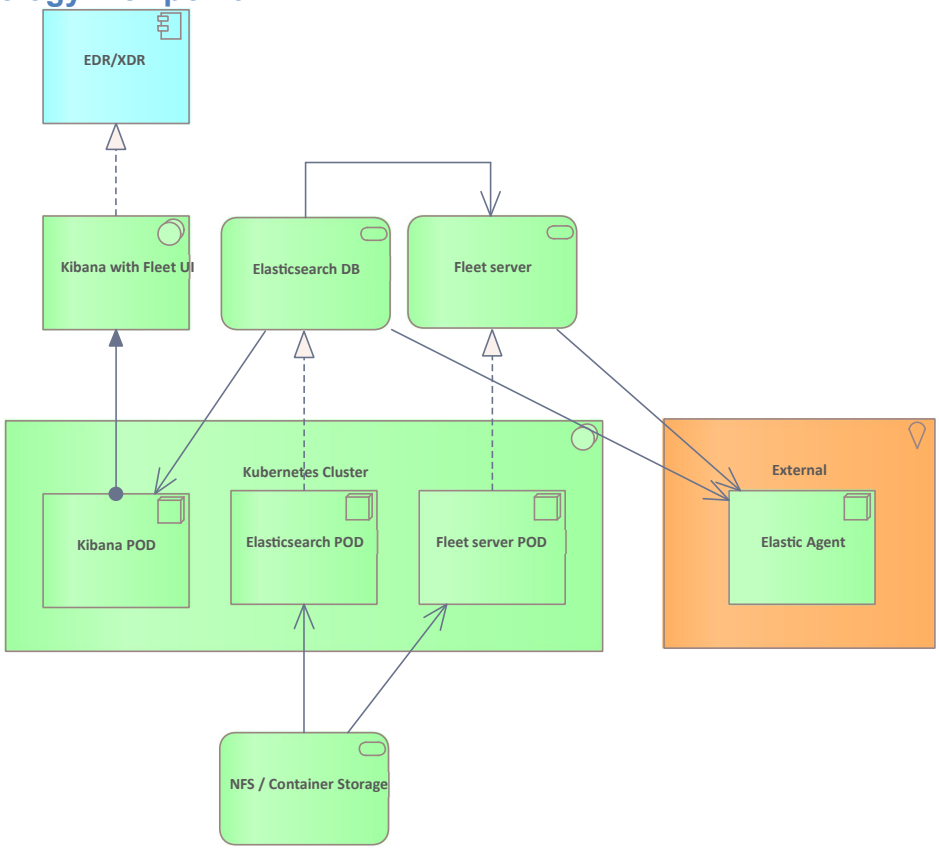
Rsa-key-generator POD

The rsa-key-generator POD is a container unit deployed in a Kubernetes cluster that is used to generate RSA key pairs for the purposes of encryption, digital signing or secure communication. This POD ensures automated and secure creation of cryptographic keys, which can subsequently be used by other services within the infrastructure, for example for TLS certificates, GPG signatures or authentication.

Xtm-composer POD

The xtm-composer POD is a container unit deployed in a Kubernetes cluster that hosts the XTM Composer service – a component designed for the composition and generation of data or security outputs within a larger system, often related to cybersecurity management or automated reporting. This POD ensures the processing of input data, their transformation and output in a predefined format, and may cooperate with other components such as databases, analytical tools or API gateways.

7.1.2 Technology Viewpoint EDR/XDR



Technology Usage Viewpoint EDR/XDR alt 1

The diagram shows a simplified assignment of the individual PODs to the corresponding application components. Alt 1 represents the use of the Elasticsearch database directly within the module setup. The design envisages a containerised version of the NoSQL databases for Elasticsearch.

Description of the elements used:

EDR/XDR

EDR (Endpoint Detection and Response) and XDR (Extended Detection and Response) represent the fundamental technological pillars for threat detection at endpoint level. In the SOCaaS environment, they are used for the collection and correlation of, and response to, events that would otherwise escape traditional perimeter detection. XDR extends the EDR concept with the ability to integrate data from networks, clouds, e-mail systems and other layers.

NFS / Container Storage

Represents an NFS service or other Container Storage. NFS (Network File System) is a network file protocol at the application layer of the OSI model that enables remote access to file systems over a network. It is one of the most common forms of shared storage in containerised environments, such as Kubernetes, where it is implemented as a PersistentVolume (PV) type. In a broader sense, container storage refers to mechanisms and services providing storage space for containers that need persistent or shared data between restarts or between cluster nodes.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

Elasticsearch DB

The NoSQL Elasticsearch database technology service is a powerful and distributed system designed for indexing, searching and analysing data in real time, based on a document-oriented model. It uses a JSON document schema, supports horizontal scaling and high availability, and is optimised for processing large volumes of unstructured or semi-structured data, such as logs, metrics or application records.

Elasticsearch POD

The Elasticsearch POD for the NoSQL database is a container instance deployed in a Kubernetes environment that hosts an Elasticsearch node as part of a distributed NoSQL database cluster. This POD ensures indexing, searching and storage of document-oriented data, while enabling horizontal scaling, high availability and efficient processing of large volumes of unstructured information in real time.

External

Represents the physical location at the client's premises, i.e. outside the INNOSOC environment.

Kibana POD

The Kibana POD for SIEM is a container instance deployed in a Kubernetes cluster that provides a graphical user interface for the visualisation and analysis of security data collected in Elasticsearch. It serves as the frontend component for the SIEM (Security Information and Event Management) solution, enabling interactive monitoring, event correlation, creation of dashboards and detection of security incidents in real time.

Fleet server

The Fleet Server technology service is a central component of the Elastic Stack used for the management, coordination and secure communication of Elastic Agents deployed on various systems. Fleet Server relays configurations, updates and commands from the Kibana Fleet UI to the individual agents, thereby enabling centralised management of the collection of logs, metrics and security data within large distributed environments.

Elastic Agent

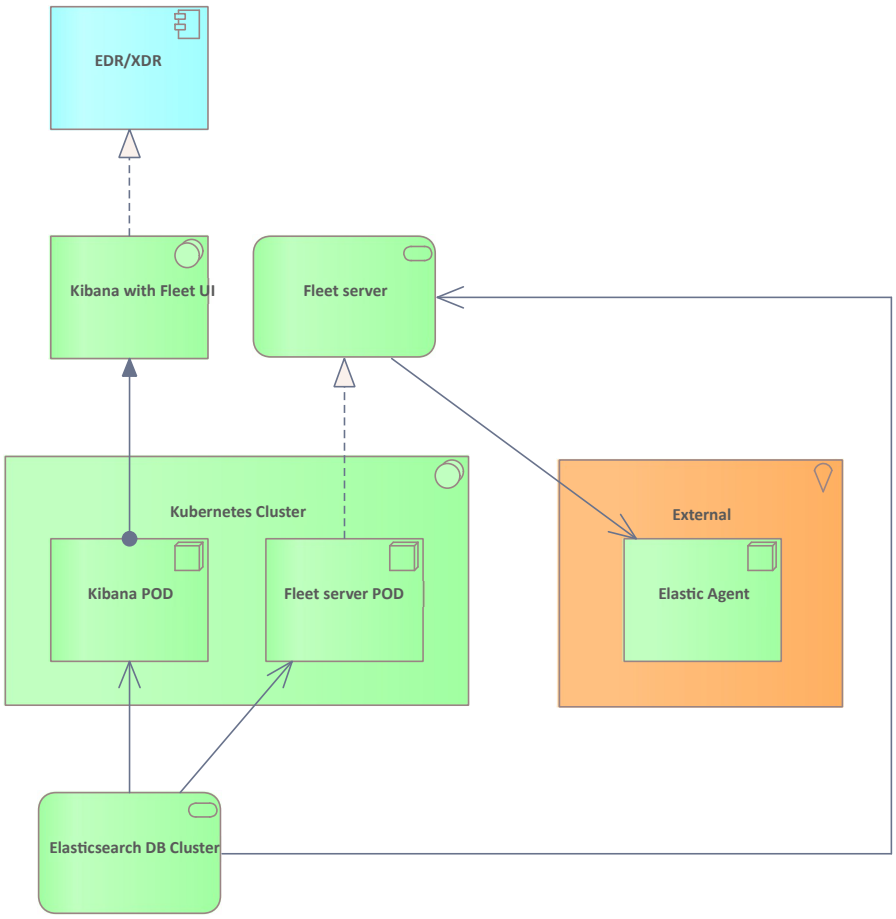
Represents an Elastic Agent instance for monitoring and processing logs, writing them to Elasticsearch and protecting the host against security threats.

Fleet server POD

The Elasticsearch POD for the NoSQL database is a container unit deployed in a Kubernetes cluster that hosts an Elasticsearch node as part of a distributed search and indexing system. This POD ensures the storage, searching and analysis of document-oriented data in real time, with support for horizontal scaling, high availability and integration with other tools in the data analytics or security ecosystem.

Kibana with Fleet UI

Kibana with Fleet UI is an extended web interface of the Kibana tool that provides centralised management of agents and data integrations within the Elastic Stack. Fleet UI enables simple deployment, configuration and monitoring of Elastic Agents, thereby simplifying the collection of logs, metrics and security data from various systems and their subsequent visualisation and analysis in Kibana.



Technology Usage Viewpoint EDR/XDR alt 2

The diagram shows a simplified assignment of the individual PODs to the corresponding application components. Alt 2 represents the use of the Elasticsearch database directly in the form of a centralised DB cluster.

Description of the elements used:

EDR/XDR

EDR (Endpoint Detection and Response) and XDR (Extended Detection and Response) represent the fundamental technological pillars for threat detection at endpoint level. In the SOCaaS environment, they are used for the collection and correlation of, and response to, events that would otherwise escape traditional perimeter detection. XDR extends the EDR concept with the ability to integrate data from networks, clouds, e-mail systems and other layers.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

Elasticsearch DB Cluster

The NoSQL Elasticsearch database technology service in the form of a cluster represents a multi-node distributed solution in which individual nodes cooperate in storing, indexing and searching document-oriented data. Each node in the cluster performs a specific role (e.g. master, data or ingest node), which enables horizontal scaling, high availability and resilience to outages, while the system as a whole ensures the processing of a large volume of data in real time.

External

Represents the physical location at the client's premises, i.e. outside the INNOSOC environment.

Kibana POD

The Kibana POD for SIEM is a container instance deployed in a Kubernetes cluster that provides a graphical user interface for the visualisation and analysis of security data collected in Elasticsearch. It serves as the frontend component for the SIEM (Security Information and Event Management) solution, enabling interactive monitoring, event correlation, creation of dashboards and detection of security incidents in real time.

Fleet server

The Fleet Server technology service is a central component of the Elastic Stack used for the management, coordination and secure communication of Elastic Agents deployed on various systems. Fleet Server relays configurations, updates and commands from the Kibana Fleet UI to the individual agents, thereby enabling centralised management of the collection of logs, metrics and security data within large distributed environments.

Elastic Agent

Represents an Elastic Agent instance for monitoring and processing logs, writing them to Elasticsearch and protecting the host against security threats.

Fleet server POD

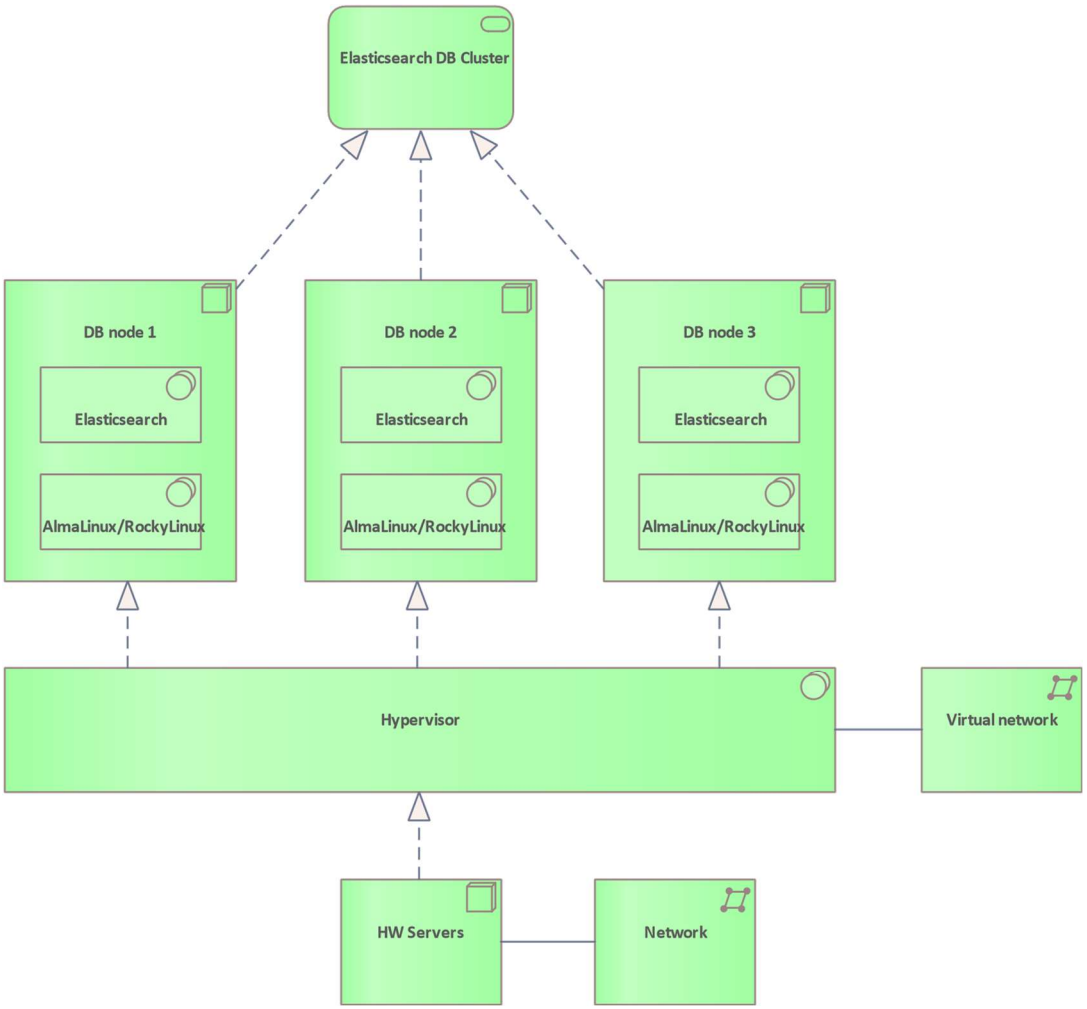
The Elasticsearch POD for the NoSQL database is a container unit deployed in a Kubernetes cluster that hosts an Elasticsearch node as part of a distributed search and indexing system. This POD ensures the storage,

searching and analysis of document-oriented data in real time, with support for horizontal scaling, high availability and integration with other tools in the data analytics or security ecosystem.

Kibana with Fleet UI

Kibana with Fleet UI is an extended web interface of the Kibana tool that provides centralised management of agents and data integrations within the Elastic Stack. Fleet UI enables simple deployment, configuration and monitoring of Elastic Agents, thereby simplifying the collection of logs, metrics and security data from various systems and their subsequent visualisation and analysis in Kibana.

7.1.3 Technology Viewpoint INNOSOC Common



Technology Viewpoint DB Cluster

The diagram shows the technology architecture of the PostgreSQL Cluster.

Description of the elements used:

DB node 1

Represents Node 1 of the Elasticsearch Cluster. An Elasticsearch Cluster node is an individual Elasticsearch instance that is part of a distributed cluster and performs one or more specific roles, such as data management, indexing or coordination of requests. Each node communicates with the other nodes in the cluster, and together they ensure scalability, high availability and parallel processing of queries and operations on large volumes of data.

DB node 2

Represents Node 2 of the Elasticsearch Cluster. An Elasticsearch Cluster node is an individual Elasticsearch instance that is part of a distributed cluster and performs one or more specific roles, such as data management, indexing or coordination of requests. Each node communicates with the other nodes in the cluster, and together they ensure scalability, high availability and parallel processing of queries and operations on large volumes of data.

DB node 3

Represents Node 3 of the Elasticsearch Cluster. An Elasticsearch Cluster node is an individual Elasticsearch instance that is part of a distributed cluster and performs one or more specific roles, such as data management, indexing or coordination of requests. Each node communicates with the other nodes in the cluster, and together they ensure scalability, high availability and parallel processing of queries and operations on large volumes of data.

HW Servers

Represents the HW server infrastructure. The HW server infrastructure represents the physical computing and network equipment that forms the basis for the operation of IT systems, including servers, storage, switches, routers and backup power supplies. This infrastructure is deployed either on-premise or as part of a cloud platform, and provides computing power, availability and connectivity for both virtualised and non-virtualised environments.

Hypervisor

Represents a Type 1 or Type 2 Hypervisor. A Type 1 hypervisor, also known as a "bare-metal" hypervisor, runs directly on physical hardware without an operating system, thereby providing high performance and efficiency of virtualisation; examples are VMware ESXi, Microsoft Hyper-V (core) or Xen. This type of hypervisor is intended mainly for data centres and enterprise environments, where emphasis is placed on performance, stability and security.

A Type 2 hypervisor, also called a "hosted" hypervisor, runs on top of an existing operating system as an application and thus shares system resources with the host; examples include VMware Workstation or Oracle VirtualBox. Although it is simpler to install and suitable for development or testing, its performance and isolation are lower compared with Type 1.

Network

Represents the network/subnet of the HW infrastructure. The network or subnet of the HW infrastructure is the physical and logical layer interconnecting servers, storage and network devices, which ensures data transmission, communication and access to services within the data centre.

Virtual network

Represents the virtual network/subnet of the Hypervisor. The virtual network or subnet of the Hypervisor is a logical network infrastructure that the hypervisor creates for the purpose of interconnecting virtual machines (VMs) with each other, with the host system or with the external network. It enables the definition of virtual switches and NAT, bridged or internal networking modes, thereby providing flexibility in the isolation, communication and routing of data between VMs and the physical network environment.

AlmaLinux/RockyLinux

Represents the AlmaLinux/RockyLinux Linux distribution. AlmaLinux and RockyLinux are open-source Linux distributions based on Red Hat Enterprise Linux (RHEL), which provide fully binary-compatible alternatives to

RHEL without licensing restrictions. They are designed for stable, secure and long-term supported server environments and are often used in enterprise, cloud and data centres as a replacement for CentOS following its discontinuation as a downstream project.

Elasticsearch

Represents the Elasticsearch NoSQL database. Elasticsearch is a high-performance, distributed NoSQL database designed for indexing, full-text search and analysis of large volumes of data in real time. It uses a JSON-based document-oriented model, supports horizontal scaling and high availability, and is widely used in areas such as log management, monitoring, security analytics and business intelligence.

AlmaLinux/RockyLinux

Represents the AlmaLinux/RockyLinux Linux distribution. AlmaLinux and RockyLinux are open-source Linux distributions based on Red Hat Enterprise Linux (RHEL), which provide fully binary-compatible alternatives to RHEL without licensing restrictions. They are designed for stable, secure and long-term supported server environments and are often used in enterprise, cloud and data centres as a replacement for CentOS following its discontinuation as a downstream project.

Elasticsearch

Represents the Elasticsearch NoSQL database. Elasticsearch is a high-performance, distributed NoSQL database designed for indexing, full-text search and analysis of large volumes of data in real time. It uses a JSON-based document-oriented model, supports horizontal scaling and high availability, and is widely used in areas such as log management, monitoring, security analytics and business intelligence.

AlmaLinux/RockyLinux

Represents the AlmaLinux/RockyLinux Linux distribution. AlmaLinux and RockyLinux are open-source Linux distributions based on Red Hat Enterprise Linux (RHEL), which provide fully binary-compatible alternatives to RHEL without licensing restrictions. They are designed for stable, secure and long-term supported server environments and are often used in enterprise, cloud and data centres as a replacement for CentOS following its discontinuation as a downstream project.

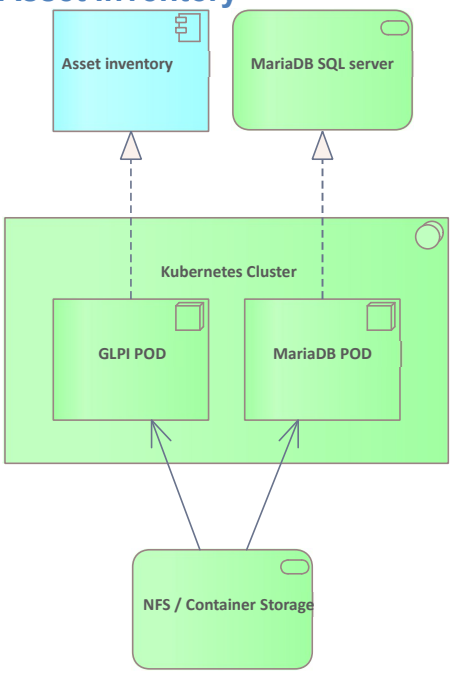
Elasticsearch

Represents the Elasticsearch NoSQL database. Elasticsearch is a high-performance, distributed NoSQL database designed for indexing, full-text search and analysis of large volumes of data in real time. It uses a JSON-based document-oriented model, supports horizontal scaling and high availability, and is widely used in areas such as log management, monitoring, security analytics and business intelligence.

Elasticsearch DB Cluster

The NoSQL Elasticsearch database technology service in the form of a cluster represents a multi-node distributed solution in which individual nodes cooperate in storing, indexing and searching document-oriented data. Each node in the cluster performs a specific role (e.g. master, data or ingest node), which enables horizontal scaling, high availability and resilience to outages, while the system as a whole ensures the processing of a large volume of data in real time.

7.1.4 Technology Viewpoint Asset Inventory



Technology Usage Viewpoint Asset Inventory alt 1

The diagram shows a simplified assignment of the individual PODs to the corresponding application components. Alt 1 represents the use of the database directly within the module setup. The design envisages a containerised version of the SQL server.

Description of the elements used:

Asset inventory

Asset inventory is a fundamental pillar of every security solution. It enables the SOC team to obtain an overview of all devices, system entities, applications and virtual or cloud resources that make up the organisation's infrastructure. In the SOCaaS environment it plays an even more significant role, as it enables centralised management, categorisation, monitoring and tracking of the security status of the infrastructure across different customers.

NFS / Container Storage

Represents an NFS service or other Container Storage. NFS (Network File System) is a network file protocol at the application layer of the OSI model that enables remote access to file systems over a network. It is one of the most common forms of shared storage in containerised environments, such as Kubernetes, where it is implemented as a PersistentVolume (PV) type. In a broader sense, container storage refers to mechanisms and services providing storage space for containers that need persistent or shared data between restarts or between cluster nodes.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

MariaDB SQL server

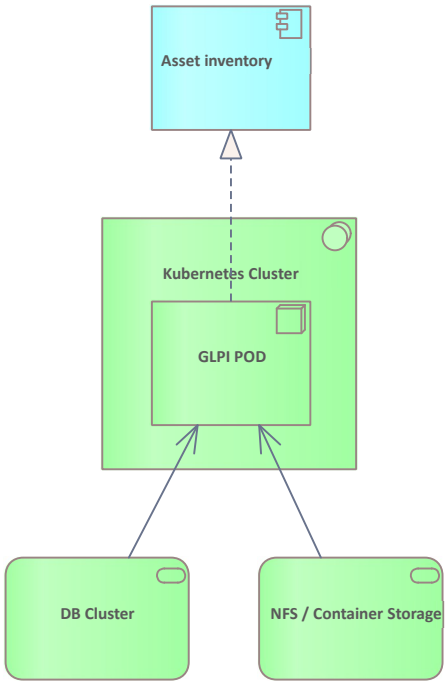
Represents the technology service of the relational MariaDB SQL database. A technology service in the field of database systems refers to a software component or platform that provides a consistent, managed and scalable environment for storing, manipulating and managing structured data. In the case of the MariaDB relational database, it is a platform-independent database service that makes the relational model accessible using SQL (Structured Query Language) through a consistent and transactionally safe application interface.

GLPI POD

Represents the GLPI POD for asset inventory. GLPI (Gestionnaire Libre de Parc Informatique) is an open-source IT service management (ITSM) system that offers functionalities such as IT asset management (hardware, software), incident management, service request management, management of stock and users, and a CMDB (Configuration Management Database). A POD (Point Of Deployment) in the context of GLPI represents a modular and distributed deployment of GLPI components at multiple places or locations, with each POD acting as a local node for the collection, updating and management of IT assets.

MariaDB POD

Represents the MariaDB POD for the relational SQL DB. MariaDB is an advanced open-source relational database management system (RDBMS) that originated as a fork of MySQL. It complies with the SQL (Structured Query Language) standard and provides powerful tools for managing and manipulating relational data. MariaDB supports the classic ACID architecture (Atomicity, Consistency, Isolation, Durability) and MVCC (Multiversion Concurrency Control) for secure and consistent database transactions.



Technology Usage Viewpoint Asset Inventory alt 2

The diagram shows a simplified assignment of the individual PODs to the corresponding application components. Alt 2 represents the use of the database directly in the form of a centralised DB cluster.

Description of the elements used:

Asset inventory

Asset inventory is a fundamental pillar of every security solution. It enables the SOC team to obtain an overview of all devices, system entities, applications and virtual or cloud resources that make up the organisation's infrastructure. In the SOCaaS environment it plays an even more significant role, as it enables centralised management, categorisation, monitoring and tracking of the security status of the infrastructure across different customers.

DB Cluster

Represents a clustered relational database service. A clustered relational database service is a distributed database system that provides relational database services through multiple interconnected database nodes (clusters), thereby ensuring high availability, scalability, redundancy and fault tolerance.

NFS / Container Storage

Represents an NFS service or other Container Storage. NFS (Network File System) is a network file protocol at the application layer of the OSI model that enables remote access to file systems over a network. It is one of the most common forms of shared storage in containerised environments, such as Kubernetes, where it is implemented as a PersistentVolume (PV) type. In a broader sense, container storage refers to mechanisms and services providing storage space for containers that need persistent or shared data between restarts or between cluster nodes.

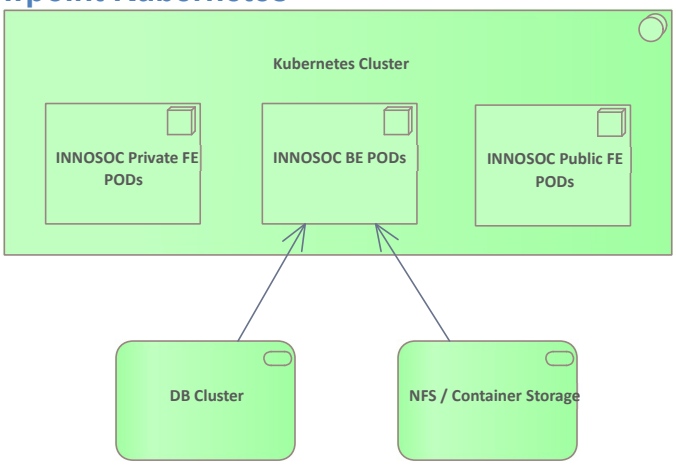
Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

GLPI POD

Represents the GLPI POD for asset inventory. GLPI (Gestionnaire Libre de Parc Informatique) is an open-source IT service management (ITSM) system that offers functionalities such as IT asset management (hardware, software), incident management, service request management, management of stock and users, and a CMDB (Configuration Management Database). A POD (Point Of Deployment) in the context of GLPI represents a modular and distributed deployment of GLPI components at multiple places or locations, with each POD acting as a local node for the collection, updating and management of IT assets.

7.1.5 Technology Viewpoint Kubernetes



Technology Usage Viewpoint and Containers maping

The diagram shows the assignment of the individual PODs to the corresponding application components.

Description of the elements used:

DB Cluster

Represents a clustered relational database service. A clustered relational database service is a distributed database system that provides relational database services through multiple interconnected database nodes (clusters), thereby ensuring high availability, scalability, redundancy and fault tolerance.

NFS / Container Storage

Represents an NFS service or other Container Storage. NFS (Network File System) is a network file protocol at the application layer of the OSI model that enables remote access to file systems over a network. It is one of the most common forms of shared storage in containerised environments, such as Kubernetes, where it is implemented as a PersistentVolume (PV) type. In a broader sense, container storage refers to mechanisms and services providing storage space for containers that need persistent or shared data between restarts or between cluster nodes.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

INNOSOC BE PODs

Represents the BE PODs. BE PODs refer to container instances of the system's backend services (BE – Backend) deployed in a distributed environment, typically in a Kubernetes cluster or other container orchestration. These PODs form the server (logical) layer of the architecture, which ensures data processing, business logic, transactions, API services and interaction with the database and network entities (core network functions).

INNOSOC Private FE PODs

FE Private PODs represent isolated and modular deployment units of the system's frontend components in the customer's private environment. This is an architectural element based on the principles of containerisation and microservice architecture, where each POD contains independently deployable elements of the presentation layer (FE – frontend), such as web applications, Single Page Applications (SPA) or UI modules.

INNOSOC Public FE PODs

FE Public PODs represent computing units for the deployment of the system's frontend components in a publicly accessible cloud or hybrid environment managed by the service provider or by the centralised infrastructure of the organisation. Within this architecture, the frontend components – typically Single Page Applications (SPA) or web interfaces – are operated as part of a multi-tenant system accessible via a public network (the internet), with communication and access governed by strict security policies. These PODs function as isolated deployments within a centralised cloud (e.g. Microsoft Azure, AWS, GCP) or in a managed Kubernetes cluster. Their aim is to provide a scalable, highly available and easily updatable interface for end users or customers who access the services without the need for local installation or private infrastructure.

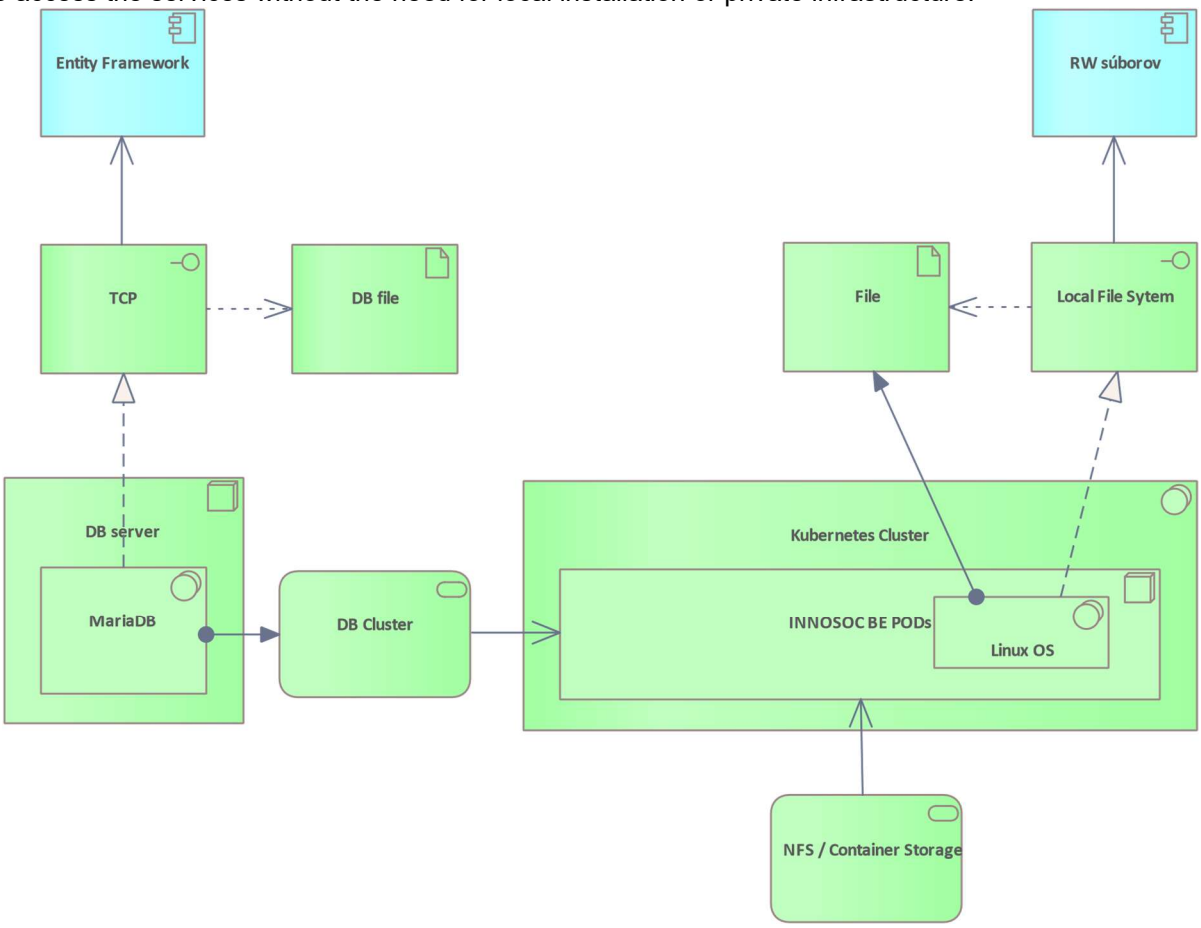


Figure 1 : Technology Usage Viewpoint BE, DB and File System

The diagram represents and describes in more detail the use of the persistence layer for the needs of BIT NC and the BE module by means of its Data Layer component, which handles work with PostgreSQL and the Local File System.

Description of the elements used:

DB Cluster

Represents a clustered relational database service. A clustered relational database service is a distributed database system that provides relational database services through multiple interconnected database nodes (clusters), thereby ensuring high availability, scalability, redundancy and fault tolerance.

NFS / Container Storage

Represents an NFS service or other Container Storage. NFS (Network File System) is a network file protocol at the application layer of the OSI model that enables remote access to file systems over a network. It is one of the most common forms of shared storage in containerised environments, such as Kubernetes, where it is implemented as a PersistentVolume (PV) type. In a broader sense, container storage refers to mechanisms and services providing storage space for containers that need persistent or shared data between restarts or between cluster nodes.

DB file

DB file – a file representing a MariaDB database file. A DB file (database file) in the context of the MariaDB system is a physical file on storage that represents the persistent data structure of the database. This file contains the data themselves, metadata, indexes, transaction logs or system structures used by the database engine (most commonly InnoDB or MyISAM) to manage relational data.

DB server

Represents the MariaDB server. MariaDB Server is an open-source relational database server system (RDBMS) designed for managing, storing and manipulating structured data. It is compatible with MySQL (of which it is originally a fork), uses the SQL language and is optimised for performance, security, transactionality and scalability.

File

File – a file is the basic unit of data storage in a file system, representing structured or unstructured data such as attachments, images, documents, multimedia and other digital objects. A file contains metadata (e.g. name, type, size, timestamps, access rights) and binary or text content, which is interpreted according to the MIME type or extension.

Entity Framework

The Entity Framework (EF) application module represents a logical software layer in the application that uses Entity Framework technology for object-relational mapping (ORM) between the database and programming language objects (most commonly C# in the .NET ecosystem). This module ensures that the application can work with database tables as objects and collections of classes, with all CRUD operations (Create, Read, Update, Delete) automatically translated into SQL statements.

File RW

The file RW module represents a software component or logical layer that ensures the reading (Read) and writing (Write) of data to files within the file system. This module serves as an abstraction between the application logic and the physical storage, using the standard system interfaces of the operating system for file manipulation (e.g. POSIX API or abstract I/O libraries).

Local File Sytem

A local file system providing work with files, implemented by the operating system under which the BE module runs. This file system enables low-latency access to files, typically via standard POSIX operations, and is bound to a specific physical or virtual node. In containerised environments (e.g. Kubernetes), this type of storage is usually represented by the hostPath or emptyDir type, which binds it to a specific node and causes it to lose persistence when the POD is restarted or moved. A local file system is suitable for temporary, high-performance operations (e.g. cache, scratch space), but is not intended for permanent or shared data storage, as it does not allow independent data recovery after an outage or scaling across nodes.

TCP

Represents client-server communication using the TCP protocol. Client-server communication via TCP (Transmission Control Protocol) is a method of data exchange between two entities in a computer network, in which the client initiates the connection and the server passively waits for the connection. The transfer is carried out via the TCP protocol – a layer 4 (transport layer) protocol of the OSI model, which provides reliable, controlled and connection-oriented data transfer over the network.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

INNOSOC BE PODs

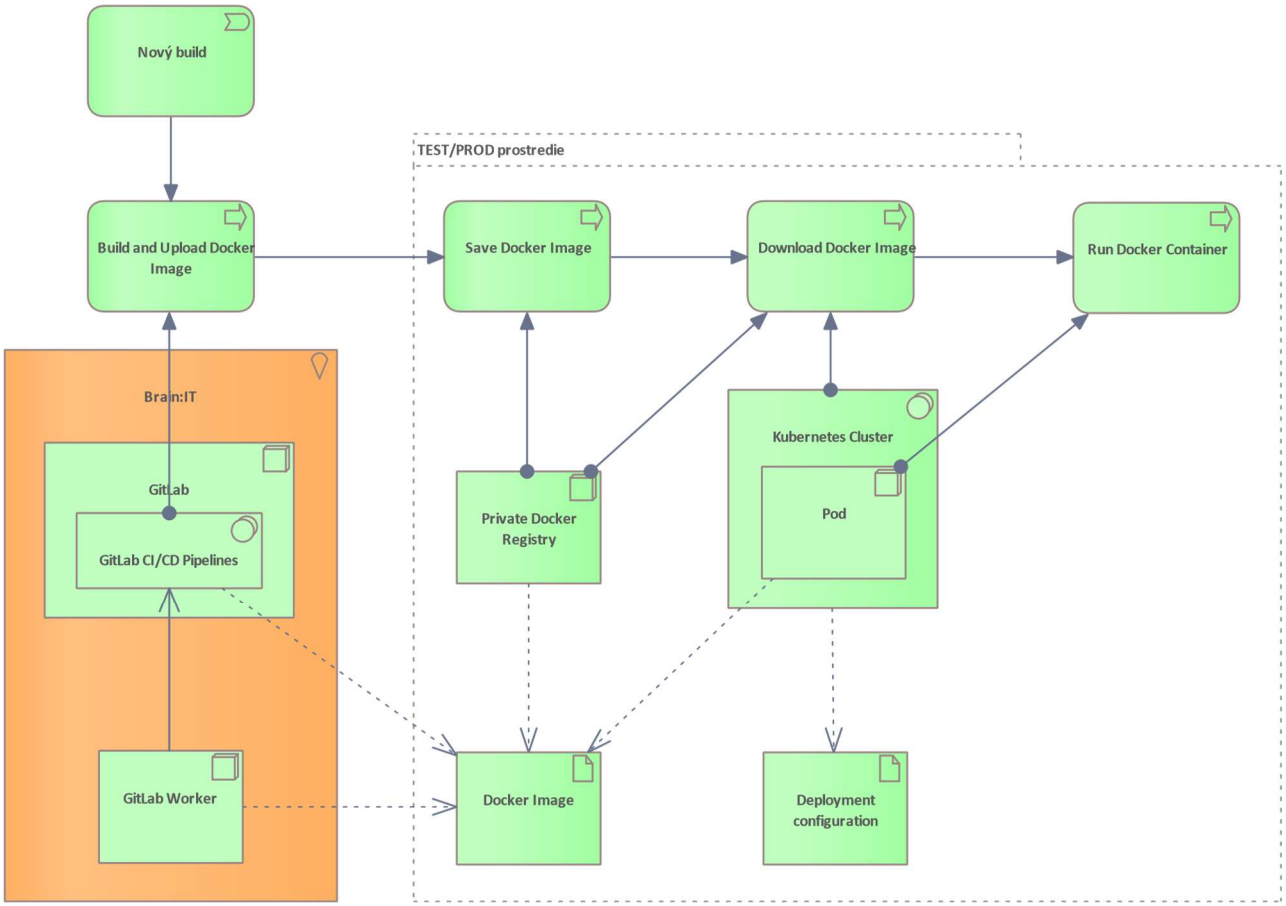
Represents the BE PODs. BE PODs refer to container instances of the system's backend services (BE – Backend) deployed in a distributed environment, typically in a Kubernetes cluster or other container orchestration. These PODs form the server (logical) layer of the architecture, which ensures data processing, business logic, transactions, API services and interaction with the database and network entities (core network functions).

Linux OS

Linux is a multitasking, multi-user, Unix-compatible operating system (OS) built on a monolithic kernel and developed as open-source software. It is characterised by a modular architecture and high customisability, and is used in servers, desktops, mobile devices, embedded systems and supercomputers. Technically, Linux is a software layer between the hardware and user applications that ensures the management of processes, memory, file systems, devices and network communication.

MariaDB

Represents the MariaDB relational database. MariaDB is a relational database (Relational Database Management System – RDBMS). In terms of a formal definition, MariaDB is a database system that: works with the relational model (data are organised into tables, columns and rows), uses the SQL (Structured Query Language) language for manipulating and querying data, ensures data integrity using primary and foreign keys and implements ACID properties through its storage engines.



Technology Viewpoint Development CI/CD TEST/PROD

The diagram shows the process of building and deploying a new version of a package in the form of a Docker image to the TEST/PROD environment. The diagram applies to custom packages; for other existing packages, the public DockerHub source is used.

Description of the elements used:

Build and Upload Docker Image

The build process followed by uploading the Docker Image to the Docker Registry is a standard step that ensures the automated assembly of the application container, its tagging and publication to a centralised repository, from where it can be deployed to development, testing or production environments.

Download Docker Image

Storing a Docker image in the Docker Registry begins with creating the image from a Dockerfile using the docker build command. The image is then tagged according to the name and version of the target registry. After authentication, the image is uploaded using the docker push command and becomes available for further deployment.

New build

Starting a new build process represents the initialisation and configuration of an automated chain of tasks whose aim is the assembly, testing and preparation of a software artefact for further processing or deployment.

Run Docker Container

Running a Docker container begins with downloading or selecting an existing Docker image, from which an instance of the application is created. The container is started using the docker run command, in which parameters such as ports, volume mounts, environment variables or network settings can be defined. Once started, the container runs as an isolated process, ready to communicate with services or the user.

Save Docker Image

The process of storing a Docker image in the Docker Registry includes creating the image using the docker build command and tagging it with the target registry address and version. This is followed by authentication to the registry and the actual upload via the docker push command. As a result, the image is stored in a central repository, from where it can be deployed to any environment.

TEST/PROD environment

Represents the internal DEV and TEST environment

Brain:IT

Brain:IT location

Private Docker Registry

Represents a private Docker Registry. A Private Docker Registry is a private repository in which an organisation securely stores and manages its own Docker images outside public registries such as Docker Hub. It enables full control over access, versioning, retention policies and security scanning of containers.

Deployment configuration

Representation of the Kubernetes Deployment artefact. A Kubernetes Deployment artefact is a configuration file that defines the desired state of an application deployment in a Kubernetes cluster – including the number of replicas, the Docker image, ports and update strategies. This artefact serves as input for kubectl or a CI/CD pipeline, which apply it to the cluster to ensure declarative and repeatable deployment.

Docker Image

Representation of the Docker Image artefact. A Docker Image as an artefact is the resulting application package assembled during the build process, which contains all the layers, dependencies and configuration necessary to run in a container environment. This artefact is stored in the Docker Registry, from where it can be reused for deployment in various environments.

GitLab

Represents a GitLab installation. A GitLab instance represents a separate deployment of the GitLab platform, which can run as a cloud service (GitLab.com) or as a self-managed version on one's own infrastructure. It contains all the components needed for repository management, CI/CD pipelines, issue tracking and DevOps workflows in a single integrated environment. Within the organisation, it enables full control over projects, users, security and access rights.

GitLab Worker

Represents a GitLab Worker instance. A GitLab Worker instance (also called GitLab Runner) is a separate service that executes jobs defined in the CI/CD pipeline, such as building, testing or deploying applications. It runs in a dedicated environment (Kubernetes) and communicates with the central GitLab server, from which it receives jobs for processing.

Kubernetes Cluster

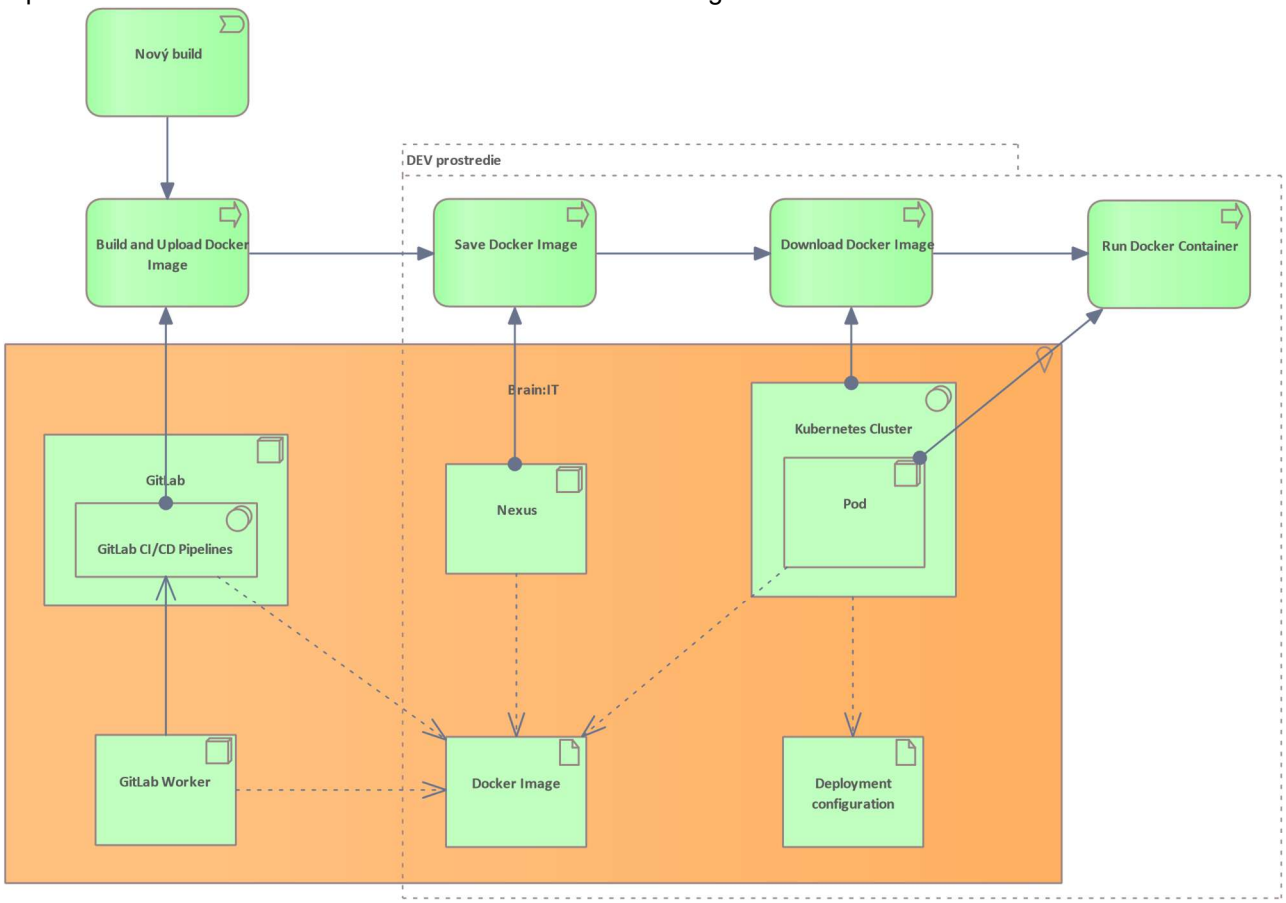
Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

GitLab CI/CD Pipelines

GitLab CI/CD Pipelines are a series of jobs ensuring compilation, tests, configuration, deployment and other tasks within GitLab.

Pod

Represents a Pod instance within which the new Docker Image is run in the form of a Docker Container



Technology Viewpoint Developmnent CI/CD DEV

The diagram shows the process of building and deploying a new version of a package in the form of a Docker image to the DEV environment. The diagram applies to custom packages; for other existing packages, the public DockerHub source is used.

Description of the elements used:

Build and Upload Docker Image

The build process followed by uploading the Docker Image to the Docker Registry is a standard step that ensures the automated assembly of the application container, its tagging and publication to a centralised repository, from where it can be deployed to development, testing or production environments.

Download Docker Image

Storing a Docker image in the Docker Registry begins with creating the image from a Dockerfile using the docker build command. The image is then tagged according to the name and version of the target registry. After authentication, the image is uploaded using the docker push command and becomes available for further deployment.

New build

Starting a new build process represents the initialisation and configuration of an automated chain of tasks whose aim is the assembly, testing and preparation of a software artefact for further processing or deployment.

Run Docker Container

Running a Docker container begins with downloading or selecting an existing Docker image, from which an instance of the application is created. The container is started using the docker run command, in which parameters such as ports, volume mounts, environment variables or network settings can be defined. Once started, the container runs as an isolated process, ready to communicate with services or the user.

Save Docker Image

The process of storing a Docker image in the Docker Registry includes creating the image using the docker build command and tagging it with the target registry address and version. This is followed by authentication to the registry and the actual upload via the docker push command. As a result, the image is stored in a central repository, from where it can be deployed to any environment.

DEV environment

Represents the internal DEV and TEST environment

Brain:IT

Brain:IT location

Deployment configuration

Representation of the Kubernetes Deployment artefact. A Kubernetes Deployment artefact is a configuration file that defines the desired state of an application deployment in a Kubernetes cluster – including the number of replicas, the Docker image, ports and update strategies. This artefact serves as input for kubectl or a CI/CD pipeline, which apply it to the cluster to ensure declarative and repeatable deployment.

Docker Image

Representation of the Docker Image artefact. A Docker Image as an artefact is the resulting application package assembled during the build process, which contains all the layers, dependencies and configuration necessary to run in a container environment. This artefact is stored in the Docker Registry, from where it can be reused for deployment in various environments.

GitLab

Represents a GitLab installation. A GitLab instance represents a separate deployment of the GitLab platform, which can run as a cloud service (GitLab.com) or as a self-managed version on one's own infrastructure. It contains all the components needed for repository management, CI/CD pipelines, issue tracking and DevOps workflows in a single integrated environment. Within the organisation, it enables full control over projects, users, security and access rights.

GitLab Worker

Represents a GitLab Worker instance. A GitLab Worker instance (also called GitLab Runner) is a separate service that executes jobs defined in the CI/CD pipeline, such as building, testing or deploying applications. It runs in a dedicated environment (Kubernetes) and communicates with the central GitLab server, from which it receives jobs for processing.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

Nexus

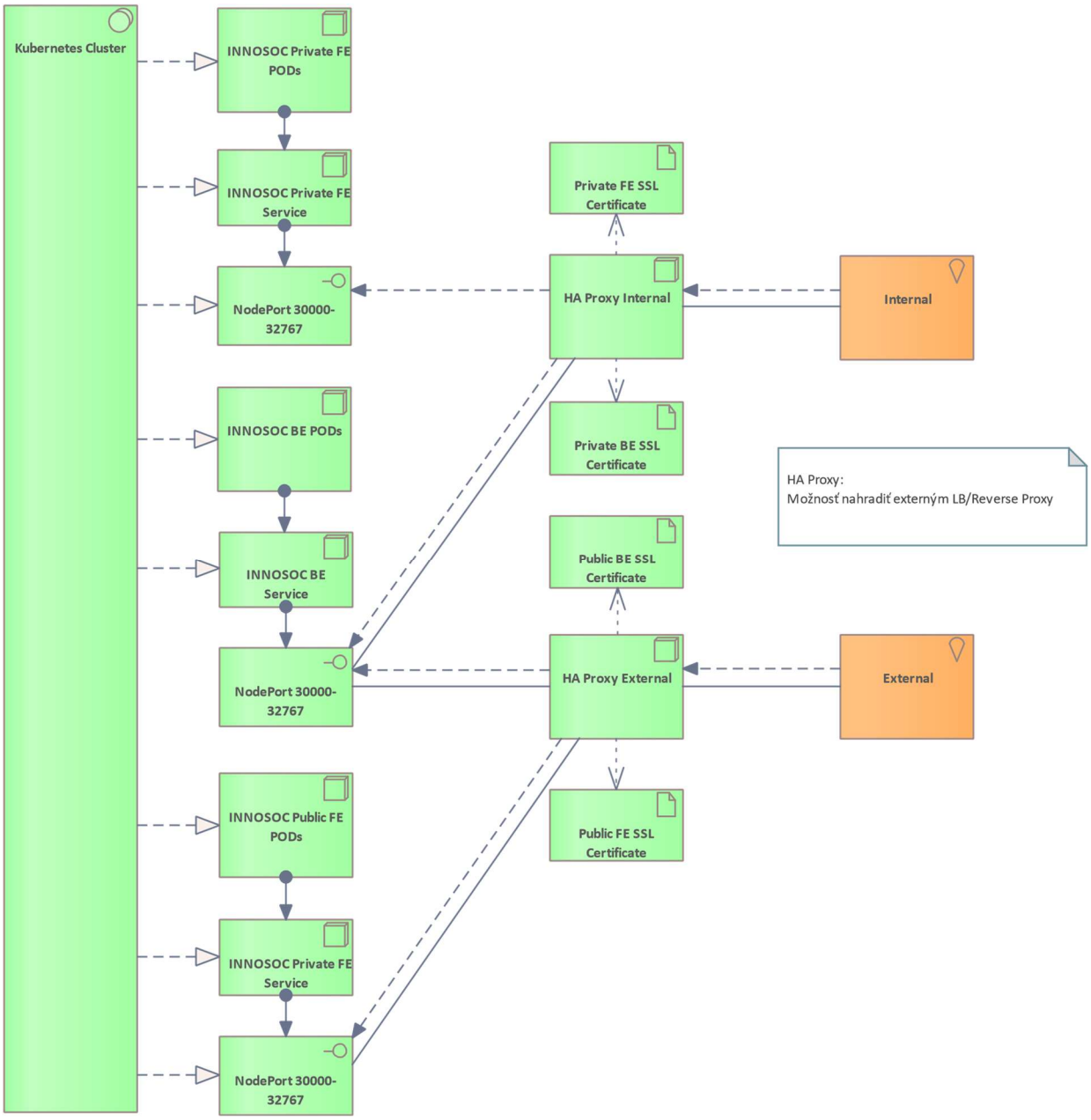
Represents a private Docker Registry. A Private Docker Registry is a private repository for Docker images that enables organisations to securely store and manage container images outside public services such as Docker Hub. It provides control over access, versions and distribution of images, and is often integrated with CI/CD pipelines for automated builds and deployments.

GitLab CI/CD Pipelines

GitLab CI/CD Pipelines are a series of jobs ensuring compilation, tests, configuration, deployment and other tasks within GitLab.

Pod

Represents a Pod instance within which the new Docker Image is run in the form of a Docker Container



Technology Viewpoint Interconnect

The diagram shows the deployment and interconnection of the exposed Kubernetes services and the LB/Reverse Proxy for two zones.

Description of the elements used:

External

Represents the external Internet environment

HA Proxy External

Represents the external reverse proxy built on HA Proxy. An external reverse proxy built on HAProxy is an intermediary layer located at the network perimeter that is used to receive requests from the internet and subsequently redirect them to internal backend services. In this role, HAProxy provides TLS termination, load balancing, protection via ACL rules, as well as the possibility of filtering or allowing traffic on the basis of defined security policies. This architecture increases the security, scalability and availability of publicly accessible services, while separating the external layer from the internal infrastructure.

HA Proxy Internal

Represents the internal reverse proxy built on HA Proxy. An internal reverse proxy built on HAProxy is an intermediary service that receives incoming requests within the internal network and redirects them to the relevant backend services according to configured rules. In this case, HAProxy provides load balancing, TLS termination, health checking and the possibility of centrally controlling the flow of data between microservices or internal applications.

INNOSOC BE Service

Represents the INNOSOC BE Service. It runs as part of a preconfigured POD, often in the form of a container or virtual machine, and is updated via a controlled CI/CD process.

INNOSOC Private FE Service

Represents the INNOSOC FE Private Service. It runs as part of a preconfigured POD, often in the form of a container or virtual machine, and is updated via a controlled CI/CD process.

INNOSOC Private FE Service

Represents the INNOSOC FE Public Service. It runs as part of a preconfigured POD, often in the form of a container or virtual machine, and is updated via a controlled CI/CD process.

Internal

Represents the internal BRAIN:IT environment

Private BE SSL Certificate

Represents a valid Private SSL certificate for HTTPS purposes. A valid BE SSL certificate for HTTPS purposes is a security certificate installed on the backend (BE) layer of the system that enables encrypted and authenticated communication between services or between the frontend and the backend. The certificate must be issued by a trusted certification authority (CA) or an internal PKI infrastructure and must ensure the integrity and confidentiality of data transmitted over HTTPS.

Private FE SSL Certificate

Represents a valid SSL certificate for HTTPS purposes. A valid FE SSL certificate for HTTPS purposes is a trusted digital certificate installed on the frontend (FE) layer of the application that ensures encrypted communication between the user's browser and the frontend server. This certificate must be issued by an accredited certification authority (CA), be valid for the specific domain and must not be expired or invalid.

Public BE SSL Certificate

Represents a valid Public SSL certificate for HTTPS purposes. A valid BE Public SSL certificate for HTTPS purposes is a certificate installed on a backend service that is accessible from the public network, and serves to secure encrypted and authenticated communication over HTTPS. It must be issued by a trusted public certification authority (CA), be valid for the specific public domain and comply with security standards.

Public FE SSL Certificate

Represents a valid Public SSL certificate for HTTPS purposes. A valid FE Public SSL certificate for HTTPS purposes is a trusted certificate installed on the frontend layer of an application that is publicly accessible via the internet, and ensures encrypted communication between the client browser and the frontend server. It must

be issued by a public certification authority (CA), be valid for the given domain and must not be expired or revoked, so that it is accepted by modern browsers.

NodePort 30000-32767

Represents NodePort-type routing from the range 30000-32767

NodePort 30000-32767

Represents NodePort-type routing from the range 30000-32767

NodePort 30000-32767

Represents NodePort-type routing from the range 30000-32767

Note

HA Proxy: Option to replace with an external LB/Reverse Proxy

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

INNOSOC BE PODs

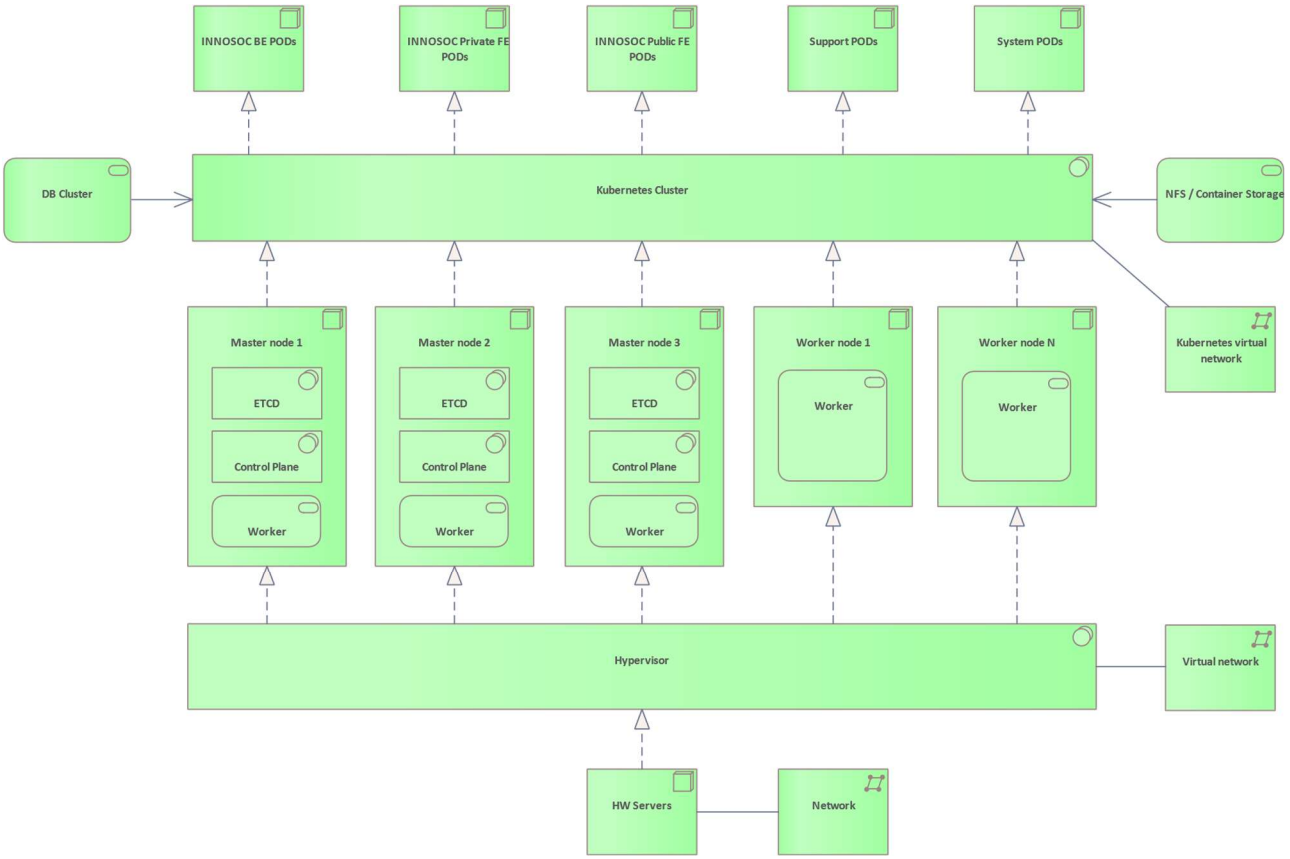
Represents the BE PODs. BE PODs refer to container instances of the system's backend services (BE – Backend) deployed in a distributed environment, typically in a Kubernetes cluster or other container orchestration. These PODs form the server (logical) layer of the architecture, which ensures data processing, business logic, transactions, API services and interaction with the database and network entities (core network functions).

INNOSOC Private FE PODs

FE Private PODs represent isolated and modular deployment units of the system's frontend components in the customer's private environment. This is an architectural element based on the principles of containerisation and microservice architecture, where each POD contains independently deployable elements of the presentation layer (FE – frontend), such as web applications, Single Page Applications (SPA) or UI modules.

INNOSOC Public FE PODs

FE Public PODs represent computing units for the deployment of the system's frontend components in a publicly accessible cloud or hybrid environment managed by the service provider or by the centralised infrastructure of the organisation. Within this architecture, the frontend components – typically Single Page Applications (SPA) or web interfaces – are operated as part of a multi-tenant system accessible via a public network (the internet), with communication and access governed by strict security policies. These PODs function as isolated deployments within a centralised cloud (e.g. Microsoft Azure, AWS, GCP) or in a managed Kubernetes cluster. Their aim is to provide a scalable, highly available and easily updatable interface for end users or customers who access the services without the need for local installation or private infrastructure.



Technology Viewpoint Kubernetes Cluster

Represents the role and service of the Worker node.

Description of the elements used:

DB Cluster

Represents a clustered relational database service. A clustered relational database service is a distributed database system that provides relational database services through multiple interconnected database nodes (clusters), thereby ensuring high availability, scalability, redundancy and fault tolerance.

NFS / Container Storage

Represents an NFS service or other Container Storage. NFS (Network File System) is a network file protocol at the application layer of the OSI model that enables remote access to file systems over a network. It is one of the most common forms of shared storage in containerised environments, such as Kubernetes, where it is implemented as a PersistentVolume (PV) type. In a broader sense, container storage refers to mechanisms and services providing storage space for containers that need persistent or shared data between restarts or between cluster nodes.

HW Servers

Represents the HW server infrastructure. The HW server infrastructure represents the physical computing and network equipment that forms the basis for the operation of IT systems, including servers, storage, switches, routers and backup power supplies. This infrastructure is deployed either on-premise or as part of a cloud

platform, and provides computing power, availability and connectivity for both virtualised and non-virtualised environments.

Hypervisor

Represents a Type 1 or Type 2 Hypervisor. A Type 1 hypervisor, also known as a "bare-metal" hypervisor, runs directly on physical hardware without an operating system, thereby providing high performance and efficiency of virtualisation; examples are VMware ESXi, Microsoft Hyper-V (core) or Xen. This type of hypervisor is intended mainly for data centres and enterprise environments, where emphasis is placed on performance, stability and security.

A Type 2 hypervisor, also called a "hosted" hypervisor, runs on top of an existing operating system as an application and thus shares system resources with the host; examples include VMware Workstation or Oracle VirtualBox. Although it is simpler to install and suitable for development or testing, its performance and isolation are lower compared with Type 1.

Kubernetes virtual network

Represents the virtual network/subnet of the Kubernetes Cluster. A virtual network or subnet in a Kubernetes cluster is a logical network architecture that enables communication between pods, services and other components in the cluster without the need for NAT. Each pod receives its own IP address within this network, which ensures predictable and fully routable connectivity in line with the principle "every pod can communicate with every other pod".

Master node 1

Represents the Master node of the Kubernetes Cluster. A Master node in a Kubernetes cluster is a control node that hosts the Control Plane components responsible for global decision-making, scheduling workloads and maintaining the state of the cluster. It contains key services such as kube-apiserver, the ETCD database, the scheduler and controller managers, which together ensure the orchestration and management of all Worker nodes and pods.

Master node 2

Represents the Master node of the Kubernetes Cluster. A Master node in a Kubernetes cluster is a control node that hosts the Control Plane components responsible for global decision-making, scheduling workloads and maintaining the state of the cluster. It contains key services such as kube-apiserver, the ETCD database, the scheduler and controller managers, which together ensure the orchestration and management of all Worker nodes and pods.

Master node 3

Represents the Master node of the Kubernetes Cluster. A Master node in a Kubernetes cluster is a control node that hosts the Control Plane components responsible for global decision-making, scheduling workloads and maintaining the state of the cluster. It contains key services such as kube-apiserver, the ETCD database, the scheduler and controller managers, which together ensure the orchestration and management of all Worker nodes and pods.

Network

Represents the network/subnet of the HW infrastructure. The network or subnet of the HW infrastructure is the physical and logical layer interconnecting servers, storage and network devices, which ensures data transmission, communication and access to services within the data centre.

Support PODs

Represents the PODs of support services. Support service PODs are container or virtual deployment units that provide non-application functions needed to run and manage the main system, such as monitoring, logging, authentication or configuration. These services typically operate as infrastructure components and are deployed in parallel with the application PODs within the same cluster or environment.

System PODs

Represents the PODs of system services. System service PODs are deployed components in a containerised environment that provide the basic functionality and operations necessary for the correct operation of the entire platform or cluster, such as networking, DNS, storage or orchestration. These services usually form part of the core infrastructure of Kubernetes or another orchestrator and operate independently of the application and support PODs, while ensuring the stability and operability of the system.

Virtual network

Represents the virtual network/subnet of the Hypervisor. The virtual network or subnet of the Hypervisor is a logical network infrastructure that the hypervisor creates for the purpose of interconnecting virtual machines (VMs) with each other, with the host system or with the external network. It enables the definition of virtual switches and NAT, bridged or internal networking modes, thereby providing flexibility in the isolation, communication and routing of data between VMs and the physical network environment.

Worker node 1

Represents a Worker node of the Kubernetes Cluster. A Worker node in a Kubernetes cluster is a compute node on which application Pods, i.e. the actual workloads of the system, are deployed and run. It contains basic components such as kubelet, kube-proxy and a container runtime (e.g. containerd), which ensure communication with the Control Plane and the correct execution of assigned tasks.

Worker node N

Represents a Worker node of the Kubernetes Cluster. A Worker node in a Kubernetes cluster is a compute node on which application Pods, i.e. the actual workloads of the system, are deployed and run. It contains basic components such as kubelet, kube-proxy and a container runtime (e.g. containerd), which ensure communication with the Control Plane and the correct execution of assigned tasks.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

INNOSOC BE PODs

Represents the BE PODs. BE PODs refer to container instances of the system's backend services (BE – Backend) deployed in a distributed environment, typically in a Kubernetes cluster or other container orchestration. These PODs form the server (logical) layer of the architecture, which ensures data processing, business logic, transactions, API services and interaction with the database and network entities (core network functions).

INNOSOC Private FE PODs

FE Private PODs represent isolated and modular deployment units of the system's frontend components in the customer's private environment. This is an architectural element based on the principles of containerisation and microservice architecture, where each POD contains independently deployable elements of the presentation layer (FE – frontend), such as web applications, Single Page Applications (SPA) or UI modules.

INNOSOC Public FE PODs

FE Public PODs represent computing units for the deployment of the system's frontend components in a publicly accessible cloud or hybrid environment managed by the service provider or by the centralised infrastructure of

the organisation. Within this architecture, the frontend components – typically Single Page Applications (SPA) or web interfaces – are operated as part of a multi-tenant system accessible via a public network (the internet), with communication and access governed by strict security policies. These PODs function as isolated deployments within a centralised cloud (e.g. Microsoft Azure, AWS, GCP) or in a managed Kubernetes cluster. Their aim is to provide a scalable, highly available and easily updatable interface for end users or customers who access the services without the need for local installation or private infrastructure.

Worker

Represents the role and service of the Worker node. The role and service of a Worker node in Kubernetes consists in hosting and running application pods, ensuring the execution of workloads based on instructions from the Control Plane.

Control Plane

Represents the Control Plane for managing the Kubernetes Cluster. The Control Plane for managing a Kubernetes cluster is the central control layer that ensures global orchestration, scheduling and management of all components in the cluster. It contains key services such as kube-apiserver, the ETCD database, kube-scheduler, kubecontroller-manager and possibly cloud-controller-manager. The Control Plane decides where and how applications will run, monitors their state and guarantees that the desired state (declared by the user) corresponds to the current state of the cluster.

ETCD

Represents the ETCD database. The ETCD database is a distributed, highly available key-value database designed for storing configuration data and metadata in real time. It is a core component of Kubernetes, where it serves as the central store of cluster state, including information about pods, configurations, secrets and other objects. Thanks to the Raft consensus mechanism, it ensures reliability, consistency and resilience to outages in distributed systems.

Worker

Represents the role and service of the Worker node. The role and service of a Worker node in Kubernetes consists in hosting and running application pods, ensuring the execution of workloads based on instructions from the Control Plane.

Control Plane

Represents the Control Plane for managing the Kubernetes Cluster. The Control Plane for managing a Kubernetes cluster is the central control layer that ensures global orchestration, scheduling and management of all components in the cluster. It contains key services such as kube-apiserver, the ETCD database, kube-scheduler, kubecontroller-manager and possibly cloud-controller-manager. The Control Plane decides where and how applications will run, monitors their state and guarantees that the desired state (declared by the user) corresponds to the current state of the cluster.

ETCD

Represents the ETCD database. The ETCD database is a distributed, highly available key-value database designed for storing configuration data and metadata in real time. It is a core component of Kubernetes, where it serves as the central store of cluster state, including information about pods, configurations, secrets and other objects. Thanks to the Raft consensus mechanism, it ensures reliability, consistency and resilience to outages in distributed systems.

Worker

Represents the role and service of the Worker node. The role and service of a Worker node in Kubernetes consists in hosting and running application pods, ensuring the execution of workloads based on instructions from the Control Plane.

Control Plane

Represents the Control Plane for managing the Kubernetes Cluster. The Control Plane for managing a Kubernetes cluster is the central control layer that ensures global orchestration, scheduling and management of all components in the cluster. It contains key services such as kube-apiserver, the ETCD database, kube-scheduler, kubecontroller-manager and possibly cloud-controller-manager. The Control Plane decides where and how applications will run, monitors their state and guarantees that the desired state (declared by the user) corresponds to the current state of the cluster.

ETCD

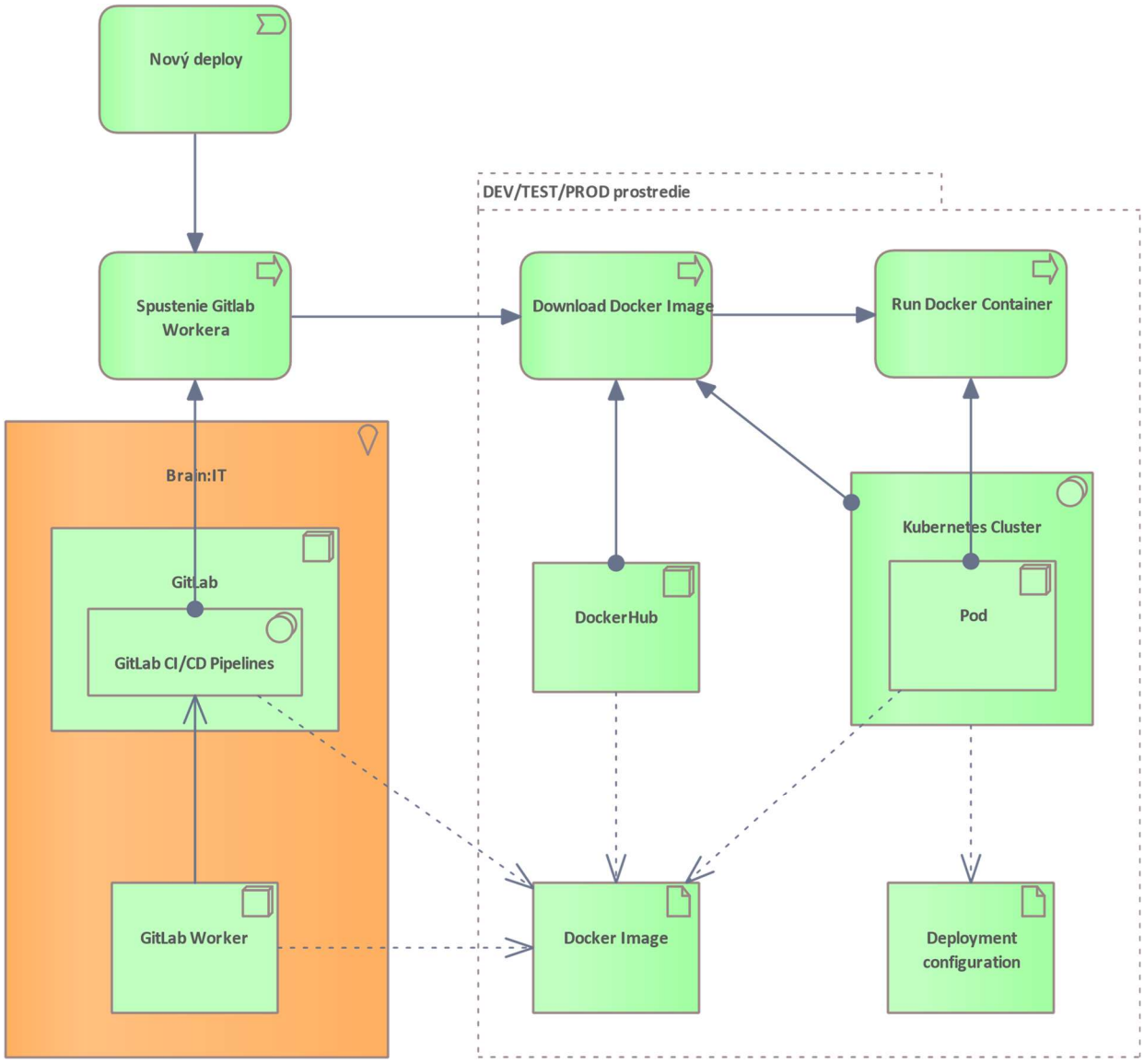
Represents the ETCD database. The ETCD database is a distributed, highly available key-value database designed for storing configuration data and metadata in real time. It is a core component of Kubernetes, where it serves as the central store of cluster state, including information about pods, configurations, secrets and other objects. Thanks to the Raft consensus mechanism, it ensures reliability, consistency and resilience to outages in distributed systems.

Worker

Represents the role and service of the Worker node. The role and service of a Worker node in Kubernetes consists in hosting and running application pods, ensuring the execution of workloads based on instructions from the Control Plane.

Worker

Represents the role and service of the Worker node. The role and service of a Worker node in Kubernetes consists in hosting and running application pods, ensuring the execution of workloads based on instructions from the Control Plane.



Technology Viewpoint standard CI/CD DEV/TEST/PROD

Description of the elements used:

Download Docker Image

Storing a Docker image in the Docker Registry begins with creating the image from a Dockerfile using the docker build command. The image is then tagged according to the name and version of the target registry. After authentication, the image is uploaded using the docker push command and becomes available for further deployment.

New deploy

Launching a new deploy process is an action in which a new version of an application or service is automatically deployed to the target environment on the basis of a defined CI/CD pipeline.

Run Docker Container

Running a Docker container begins with downloading or selecting an existing Docker image, from which an instance of the application is created. The container is started using the docker run command, in which parameters such as ports, volume mounts, environment variables or network settings can be defined. Once started, the container runs as an isolated process, ready to communicate with services or the user.

Launching the Gitlab Worker

The process of launching a GitLab Worker to deploy a new version of a package begins with receiving a job from the CI/CD pipeline defined in .gitlab-ci.yml, which contains the steps for deployment. The Worker then executes instructions such as downloading the artefact, connecting to the target environment and deploying the new version of the application, with the entire process being logged and traceable.

DEV/TEST/PROD environment

Represents the internal DEV and TEST environment

Brain:IT

Brain:IT location

DockerHub

Represents the public Docker Registry DockerHub. DockerHub is a public Docker Registry provided by Docker Inc. that enables developers to share, store and download container images via a cloud platform. It supports both public and private repositories, image versioning, automated builds and integration with CI/CD tools.

Deployment configuration

Representation of the Kubernetes Deployment artefact. A Kubernetes Deployment artefact is a configuration file that defines the desired state of an application deployment in a Kubernetes cluster – including the number of replicas, the Docker image, ports and update strategies. This artefact serves as input for kubectl or a CI/CD pipeline, which apply it to the cluster to ensure declarative and repeatable deployment.

Docker Image

Representation of the Docker Image artefact. A Docker Image as an artefact is the resulting application package assembled during the build process, which contains all the layers, dependencies and configuration necessary to run in a container environment. This artefact is stored in the Docker Registry, from where it can be reused for deployment in various environments.

GitLab

Represents a GitLab installation. A GitLab instance represents a separate deployment of the GitLab platform, which can run as a cloud service (GitLab.com) or as a self-managed version on one's own infrastructure. It contains all the components needed for repository management, CI/CD pipelines, issue tracking and DevOps workflows in a single integrated environment. Within the organisation, it enables full control over projects, users, security and access rights.

GitLab Worker

Represents a GitLab Worker instance. A GitLab Worker instance (also called GitLab Runner) is a separate service that executes jobs defined in the CI/CD pipeline, such as building, testing or deploying applications. It runs in a dedicated environment (Kubernetes) and communicates with the central GitLab server, from which it receives jobs for processing.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are

abstracted and managed as a unified system, which enables automated management of the application life cycle.

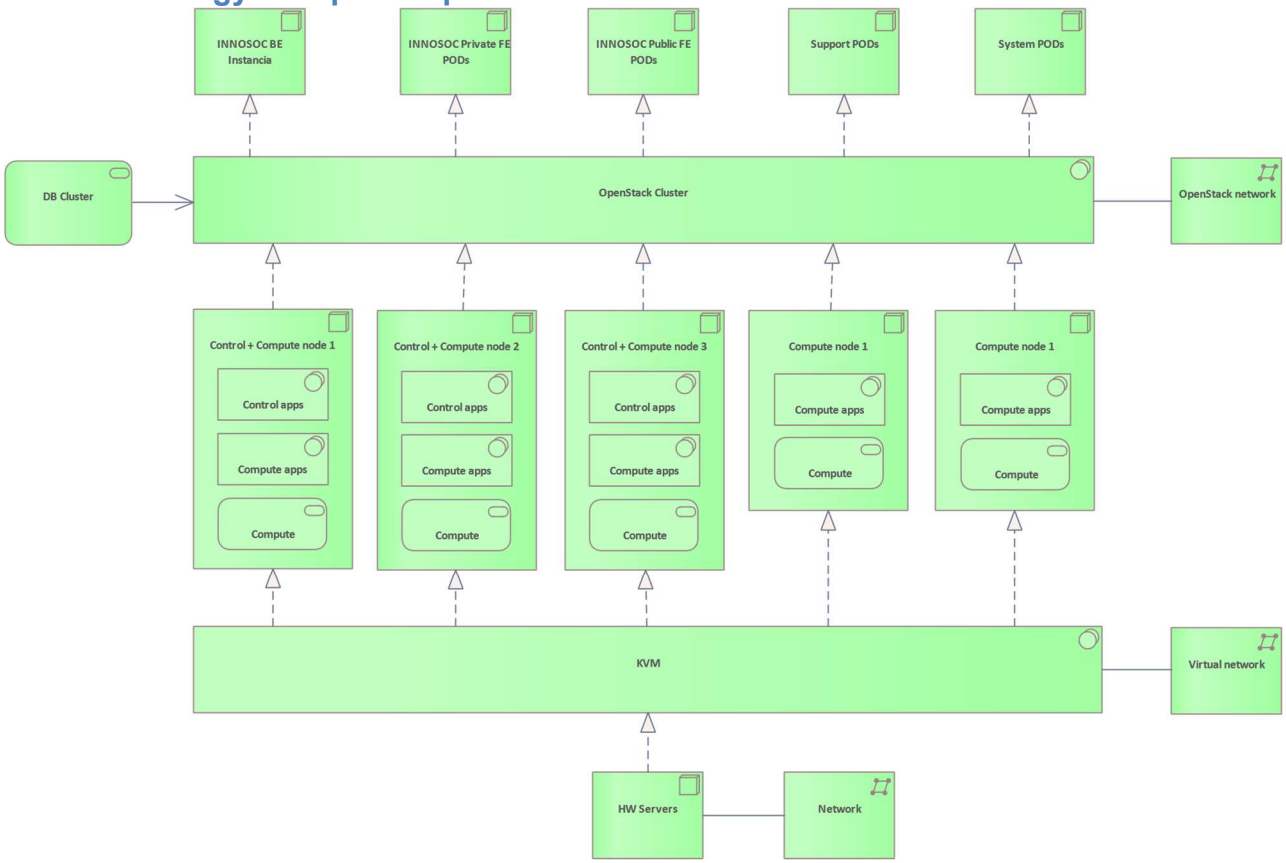
GitLab CI/CD Pipelines

GitLab CI/CD Pipelines are a series of jobs ensuring compilation, tests, configuration, deployment and other tasks within GitLab.

Pod

Represents a Pod instance within which the new Docker Image is run in the form of a Docker Container

7.1.6 Technology Viewpoint OpenStack



Technology Viewpoint OpenStack Cluster

Description of the elements used:

DB Cluster

Represents a clustered relational database service. A clustered relational database service is a multi-node database solution in which data are partitioned or replicated across several database nodes in order to ensure high availability, scalability and resilience to outages. Each node in the cluster may serve as an active or passive member, depending on the mode used. This type of database architecture is used in enterprise systems where data consistency, performance under load and continuous availability are critical.

Compute node 1

Represents a Compute node of the OpenStack Cluster. A Compute node in an OpenStack cluster is a compute node responsible for running and managing virtual machines (VMs) using the Nova Compute service. It receives instructions from the Control node and provides physical or virtual resources such as CPU, RAM and disk for individual instances, thereby forming the basis of the cluster's compute capacity.

Control + Compute node 1

Control and Compute nodes in an OpenStack cluster are the basic node types that together ensure the control and operation of the cloud environment. The Control node contains central services such as API servers, schedulers (Scheduler), the database, the message broker and authentication, thereby controlling orchestration, resource management and communication within the cluster. Compute node 1 provides compute capacity for running virtual machines (VMs) and ensures their local management using the Nova Compute component, according to instructions from the Control node.

Control + Compute node 2

Control and Compute nodes in an OpenStack cluster are the basic node types that together ensure the control and operation of the cloud environment. The Control node contains central services such as API servers, schedulers (Scheduler), the database, the message broker and authentication, thereby controlling orchestration, resource management and communication within the cluster. Compute node 2 provides compute capacity for running virtual machines (VMs) and ensures their local management using the Nova Compute component, according to instructions from the Control node.

Control + Compute node 3

Control and Compute nodes in an OpenStack cluster are the basic node types that together ensure the control and operation of the cloud environment. The Control node contains central services such as API servers, schedulers (Scheduler), the database, the message broker and authentication, thereby controlling orchestration, resource management and communication within the cluster. Compute node 3 provides compute capacity for running virtual machines (VMs) and ensures their local management using the Nova Compute component, according to instructions from the Control node.

HW Servers

HW server infrastructure is the physical foundation of the IT environment, comprising compute servers, storage, network elements, power supplies and supporting equipment, which together provide the performance, availability and connectivity needed to operate information systems. This infrastructure forms the platform for virtualisation, cloud services, databases and application workloads in both enterprise and data centres.

KVM

KVM (Kernel-based Virtual Machine) is a type 1 hypervisor that is integrated directly into the Linux kernel and allows multiple virtual machines to run with full isolation on a physical server. It uses hardware virtualisation and provides high performance, stability and efficiency, making it suitable for production and cloud environments such as OpenStack.

Network

Represents the network/subnet of the HW infrastructure. The network or subnet of the HW infrastructure represents the physical and logical interconnection between servers, storage, network devices and other components within the data centre. It serves to ensure data transmission, traffic segregation and the control of communication by means of routers, switches, VLANs and security rules.

OpenStack Cluster

Represents the OpenStack cluster, which is a distributed cloud environment consisting of multiple physical or virtual nodes jointly operating the individual OpenStack services such as compute (Nova), networking (Neutron), storage (Cinder, Swift) and control components. This cluster enables organisations to build a private or hybrid cloud with support for automated provisioning, resource management and a scalable multi-tenant architecture.

OpenStack network

The network and subnets of the OpenStack cluster serve for the logical and physical separation of different types of communication within the cloud infrastructure, with each fulfilling a specific role. The Corporate network ensures the interconnection of the OpenStack environment with the enterprise infrastructure, the OpenStack Tenant Data Network provides network connectivity for the virtual instances of individual tenants, and the OpenStack Internal Admin / API Subnet is reserved for internal communication between cluster components, including API calls, database replication and service management.

Support PODs

Represents the PODs of support services. Support service PODs are container units that provide non-functional, infrastructure or operational functions needed to run the application system, such as monitoring, logging, authentication or configuration. These services run independently of the application PODs and support the stability, security and management of the entire environment.

System PODs

Represents the PODs of system services. System service PODs are basic container units that provide key functionalities of the Kubernetes cluster and its infrastructure, such as network connectivity, DNS, workload scheduling or access control. They typically include components such as CoreDNS, kube-proxy, metrics-server or other system operators needed for the correct functioning of the environment. These PODs are critical for the stability and operability of the platform and run at the level of system namespaces outside the application layer.

Virtual network

Represents the virtual network/subnet of the Hypervisor. The virtual network or subnet of a hypervisor is a logical network layer that enables communication between virtual machines with each other, with the host or with an external network. Within this network, the hypervisor creates and manages virtual switches, bridges or NAT interfaces, thereby ensuring flexible routing, isolation and control of network traffic between the VMs and the physical infrastructure.

INNOSOC BE Instance

Represents an OpenStack instance, VM or container. An OpenStack instance in the form of a VM or container represents a virtual compute resource (compute instance) that is deployed within the OpenStack cloud platform by means of the Nova component. It may be a classic virtual machine (VM) or a lighter container instance, depending on the hypervisor used and the integration with containerisation technologies such as Kubernetes or Docker. **INNOSOC Private FE PODs**

Represents the FE Private PODs. FE Private PODs are isolated container deployment units of the frontend layer of the system that are operated in the customer's private environment, for example on-premise or in a private cloud. These PODs ensure secure and separate access to the user interface, while being integrated with the customer's local infrastructure, authentication and network policies.

INNOSOC Public FE PODs

Represents the FE Public PODs. FE Public PODs are container deployment units of the frontend layer of the system that are operated in a publicly accessible cloud environment and are accessible via the internet. They serve to provide the user interface to multiple clients (multitenancy), making use of centralised management, secured access and scalable cloud infrastructure.

Compute

The role and service of a Compute node in an OpenStack cluster consists in providing compute resources for running virtual instances by means of the Nova Compute component. This node is responsible for managing the VM life cycle and allocating CPU, memory and disk space, and communicates with the control services on the basis of requests from users or orchestration.

Compute apps

Represents a set of Compute applications. A set of Compute applications represents a group of software components that perform compute operations and manage compute resources within a cloud or virtualised infrastructure. It includes applications responsible for deploying, managing, scaling and monitoring virtual machines or containers, which work closely with the network, storage and orchestration layers of the system.

Control apps

The set of applications for the Control node represents the key software components that control and coordinate the operation of the entire cluster in an OpenStack or Kubernetes environment. This set includes services such as API servers, schedulers, state managers (controller managers), database services, authentication mechanisms and other system applications needed for orchestration, monitoring and management of the life cycle of workloads and infrastructure.

Compute

The role and service of a Compute node in an OpenStack cluster consists in providing compute resources for running virtual instances by means of the Nova Compute component. This node is responsible for managing the VM life cycle and allocating CPU, memory and disk space, and communicates with the control services on the basis of requests from users or orchestration.

Compute apps

Represents a set of Compute applications. A set of Compute applications represents a group of software components that perform compute operations and manage compute resources within a cloud or virtualised infrastructure. It includes applications responsible for deploying, managing, scaling and monitoring virtual machines or containers, which work closely with the network, storage and orchestration layers of the system.

Compute

The role and service of a Compute node in an OpenStack cluster consists in providing compute resources for running virtual instances by means of the Nova Compute component. This node is responsible for managing the VM life cycle and allocating CPU, memory and disk space, and communicates with the control services on the basis of requests from users or orchestration.

Compute apps

Represents a set of Compute applications. A set of Compute applications represents a group of software components that perform compute operations and manage compute resources within a cloud or virtualised infrastructure. It includes applications responsible for deploying, managing, scaling and monitoring virtual machines or containers, which work closely with the network, storage and orchestration layers of the system.

Control apps

The set of applications for the Control node represents the key software components that control and coordinate the operation of the entire cluster in an OpenStack or Kubernetes environment. This set includes services such as API servers, schedulers, state managers (controller managers), database services, authentication mechanisms and other system applications needed for orchestration, monitoring and management of the life cycle of workloads and infrastructure.

Compute

The role and service of a Compute node in an OpenStack cluster consists in providing compute resources for running virtual instances by means of the Nova Compute component. This node is responsible for managing the VM life cycle and allocating CPU, memory and disk space, and communicates with the control services on the basis of requests from users or orchestration.

Compute apps

Represents a set of Compute applications. A set of Compute applications represents a group of software components that perform compute operations and manage compute resources within a cloud or virtualised

infrastructure. It includes applications responsible for deploying, managing, scaling and monitoring virtual machines or containers, which work closely with the network, storage and orchestration layers of the system.

Control apps

The set of applications for the Control node represents the key software components that control and coordinate the operation of the entire cluster in an OpenStack or Kubernetes environment. This set includes services such as API servers, schedulers, state managers (controller managers), database services, authentication mechanisms and other system applications needed for orchestration, monitoring and management of the life cycle of workloads and infrastructure.

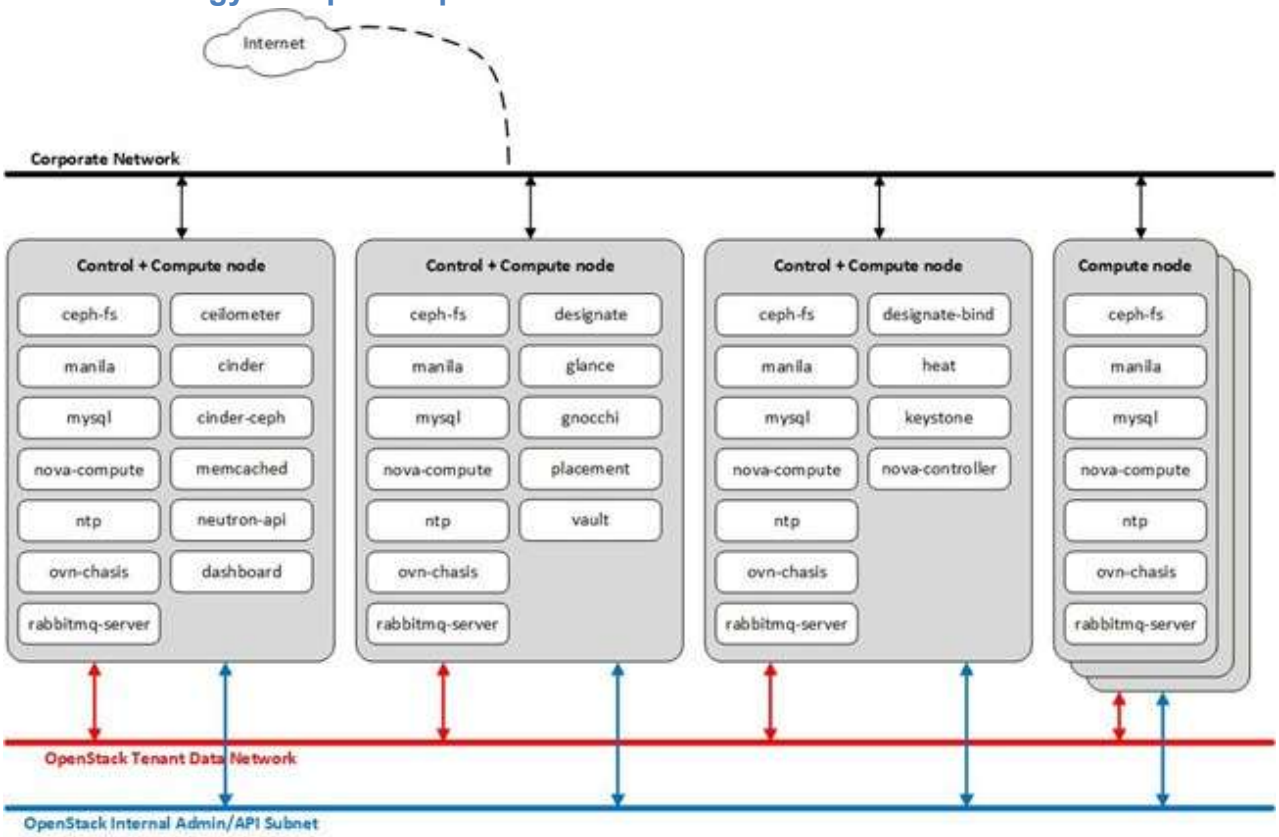
Compute

The role and service of a Compute node in an OpenStack cluster consists in providing compute resources for running virtual instances by means of the Nova Compute component. This node is responsible for managing the VM life cycle and allocating CPU, memory and disk space, and communicates with the control services on the basis of requests from users or orchestration.

Compute apps

Represents a set of Compute applications. A set of Compute applications represents a group of software components that perform compute operations and manage compute resources within a cloud or virtualised infrastructure. It includes applications responsible for deploying, managing, scaling and monitoring virtual machines or containers, which work closely with the network, storage and orchestration layers of the system.

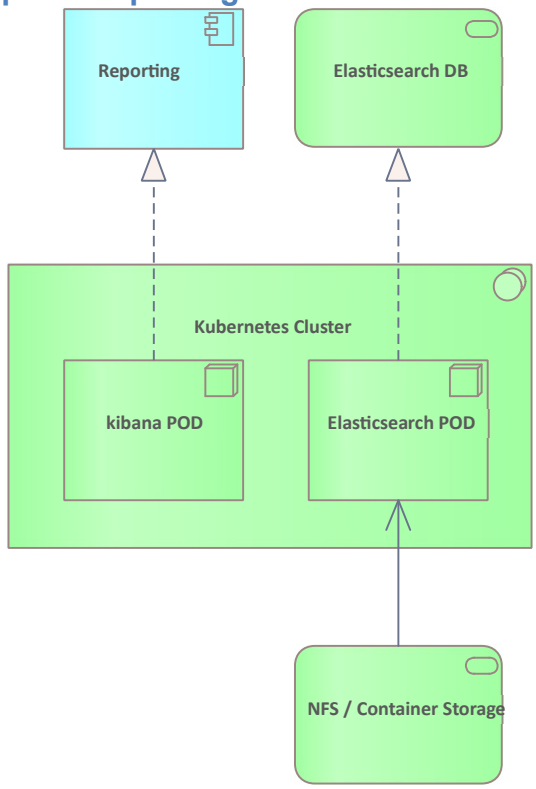
7.1.7 Technology Viewpoint OpenStack External



Obrázok1.png

Technology Viewpoint OpenStack External

7.1.8 Technology Viewpoint Reporting



Technology Usage Viewpoint Reporting alt 1

The diagram shows a simplified assignment of the individual PODs to the corresponding application components. Alt 1 represents the use of the Elasticsearch database directly within the module setup. The design envisages a containerised version of the NoSQL database for Elasticsearch.

Description of the elements used:

Reporting

The reporting component within the SOCaaS architecture serves as a central means for the aggregation, visualisation and distribution of security information. Its role is to provide both regular and ad hoc outputs on the state of cybersecurity for various target groups – from SOC analysts through team leaders to the management of the organisation or clients. Such a reporting system must cover:

- Generation of metrics on incidents, alerts, SLAs and KPIs,
- Overviews of trends over time and evaluation of the performance of security measures,
- Adaptation of content according to the level of detail (technical vs. strategic),
- Automation and scheduling of the distribution of outputs,

- Retrospective analysis and archiving of historical data.

NFS / Container Storage

Represents an NFS service or other Container Storage. NFS (Network File System) is a network file protocol at the application layer of the OSI model that enables remote access to file systems over a network. It is one of the most common forms of shared storage in containerised environments, such as Kubernetes, where it is implemented as a PersistentVolume (PV) type. In a broader sense, container storage refers to mechanisms and services providing storage space for containers that need persistent or shared data between restarts or between cluster nodes.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

Elasticsearch DB

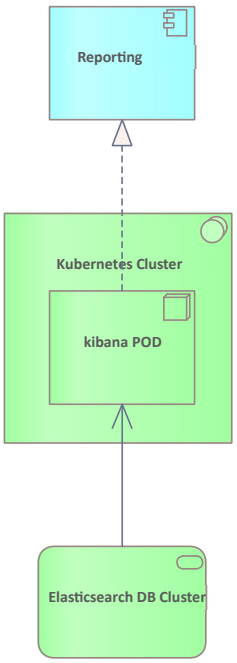
The NoSQL Elasticsearch database technology service is a distributed system for full-text search, indexing and data analysis that works on the basis of a document-oriented model. It enables efficient processing of large volumes of unstructured or semi-structured data in real time, while being optimised for horizontal scaling, high availability and powerful analytics over logs, metrics or application data.

Elasticsearch POD

Represents the Elasticsearch POD for the NoSQL database, which is a container unit deployed in a Kubernetes cluster that hosts an Elasticsearch node as part of a distributed, document-oriented NoSQL database. This POD ensures indexing, searching and storage of unstructured or semi-structured data, and is designed for horizontal scaling, high availability and fast response when working with large volumes of data.

Kibana POD

The Kibana POD for reporting is a container unit deployed in a Kubernetes cluster that hosts the Kibana web application intended for visualising and analysing data stored in Elasticsearch. This POD provides a graphical user interface for creating dashboards, reports and interactive overviews, thereby enabling users to efficiently monitor, analyse and interpret operational and logging data in real time.



Technology Usage Viewpoint Reporting alt 2

The diagram shows a simplified assignment of the individual PODs to the corresponding application components. Alt 2 represents the use of the Elasticsearch database directly in the form of a centralised DB cluster.

Description of the elements used:

Reporting

The reporting component within the SOCaaS architecture serves as a central means for the aggregation, visualisation and distribution of security information. Its role is to provide both regular and ad hoc outputs on the state of cybersecurity for various target groups – from SOC analysts through team leaders to the management of the organisation or clients. Such a reporting system must cover:

- Generation of metrics on incidents, alerts, SLAs and KPIs,
- Overviews of trends over time and evaluation of the performance of security measures,
- Adaptation of content according to the level of detail (technical vs. strategic),
- Automation and scheduling of the distribution of outputs,
- Retrospective analysis and archiving of historical data.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

Elasticsearch DB Cluster

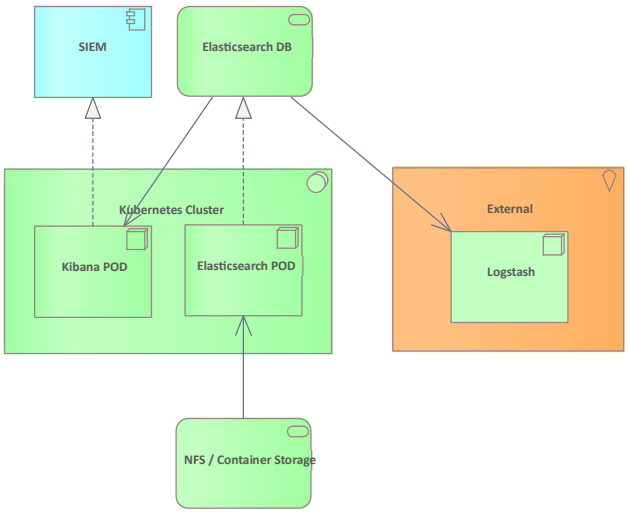
The NoSQL Elasticsearch database technology service in the form of a cluster represents a multi-node distributed solution in which individual nodes cooperate in storing, indexing and searching document-oriented

data. Each node in the cluster performs a specific role (e.g. master, data or ingest node), which enables horizontal scaling, high availability and resilience to outages, while the system as a whole ensures the processing of a large volume of data in real time.

kibana POD

The Kibana POD for reporting is a container unit deployed in a Kubernetes cluster that hosts the Kibana web application intended for visualising and analysing data stored in Elasticsearch. This POD provides a graphical user interface for creating dashboards, reports and interactive overviews, thereby enabling users to efficiently monitor, analyse and interpret operational and logging data in real time.

7.1.9 Technology Viewpoint SIEM



Technology Usage Viewpoint SIEM alt 1

The diagram shows a simplified assignment of the individual PODs to the corresponding application components. Alt 1 represents the use of the Elasticsearch database directly within the module setup. The design envisages a containerised version of the NoSQL databases for Elasticsearch.

Description of the elements used:

SIEM

SIEM is the heart of the SOCaaS solution. It ensures the aggregation, processing, correlation and analysis of security events from various sources in real time. It enables analysts to identify anomalies, generate alerts and escalate incidents, while increasing visibility over the client's infrastructure.

NFS / Container Storage

Represents an NFS service or other Container Storage. NFS (Network File System) is a network file protocol at the application layer of the OSI model that enables remote access to file systems over a network. It is one of the most common forms of shared storage in containerised environments, such as Kubernetes, where it is implemented as a PersistentVolume (PV) type. In a broader sense, container storage refers to mechanisms and services providing storage space for containers that need persistent or shared data between restarts or between cluster nodes.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical

or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

Elasticsearch DB

The NoSQL Elasticsearch database technology service is a powerful and distributed system designed for indexing, searching and analysing data in real time, based on a document-oriented model. It uses a JSON document schema, supports horizontal scaling and high availability, and is optimised for processing large volumes of unstructured or semi-structured data, such as logs, metrics or application records.

Elasticsearch POD

The Elasticsearch POD for the NoSQL database is a container instance deployed in a Kubernetes environment that hosts an Elasticsearch node as part of a distributed NoSQL database cluster. This POD ensures indexing, searching and storage of document-oriented data, while enabling horizontal scaling, high availability and efficient processing of large volumes of unstructured information in real time.

External

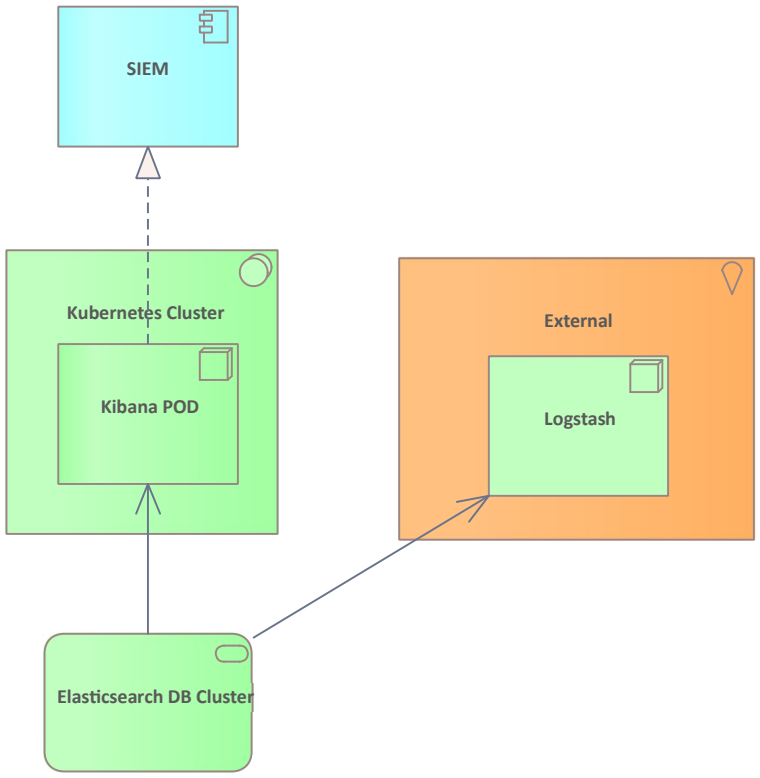
Represents the physical location at the client's premises, i.e. outside the INNOSOC environment.

Kibana POD

The Kibana POD for SIEM is a container instance deployed in a Kubernetes cluster that provides a graphical user interface for the visualisation and analysis of security data collected in Elasticsearch. It serves as the frontend component for the SIEM (Security Information and Event Management) solution, enabling interactive monitoring, event correlation, creation of dashboards and detection of security incidents in real time.

Logstash

The Logstash instance for log processing is a data processor deployed as a separate service or container that collects, transforms and routes logging data from various sources. Within the pipeline it performs parsing, enrichment and filtering of data, which it then writes to Elasticsearch, thereby enabling centralised, efficient and structured storage of logs for further analysis.



Technology Usage Viewpoint SIEM alt 2

The diagram shows a simplified assignment of the individual PODs to the corresponding application components. Alt 2 represents the use of the Elasticsearch database directly in the form of a centralised DB cluster.

Description of the elements used:

SIEM

SIEM is the heart of the SOCaaS solution. It ensures the aggregation, processing, correlation and analysis of security events from various sources in real time. It enables analysts to identify anomalies, generate alerts and escalate incidents, while increasing visibility over the client's infrastructure.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

Elasticsearch DB Cluster

The NoSQL Elasticsearch database technology service in the form of a cluster represents a multi-node distributed solution in which individual nodes cooperate in storing, indexing and searching document-oriented data. Each node in the cluster performs a specific role (e.g. master, data or ingest node), which enables

horizontal scaling, high availability and resilience to outages, while the system as a whole ensures the processing of a large volume of data in real time.

External

Represents the physical location at the client's premises, i.e. outside the INNOSOC environment.

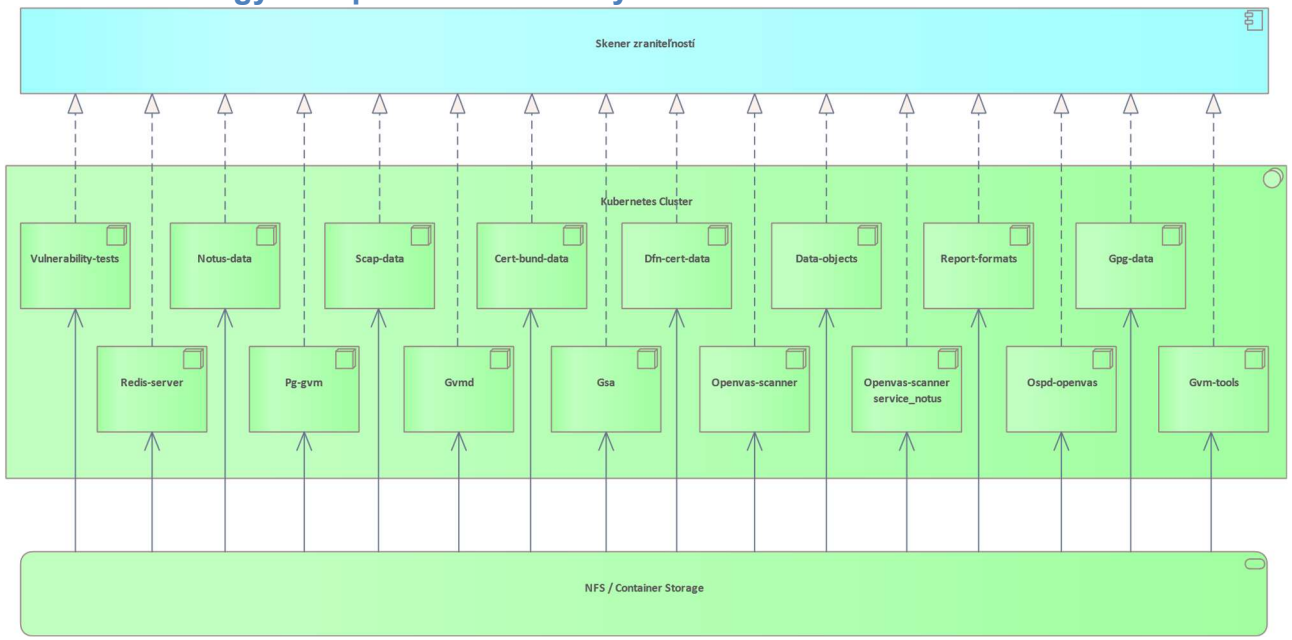
Kibana POD

The Kibana POD for SIEM is a container instance deployed in a Kubernetes cluster that provides a graphical user interface for the visualisation and analysis of security data collected in Elasticsearch. It serves as the frontend component for the SIEM (Security Information and Event Management) solution, enabling interactive monitoring, event correlation, creation of dashboards and detection of security incidents in real time.

Logstash

The Logstash instance for log processing is a data processor deployed as a separate service or container that collects, transforms and routes logging data from various sources. Within the pipeline it performs parsing, enrichment and filtering of data, which it then writes to Elasticsearch, thereby enabling centralised, efficient and structured storage of logs for further analysis.

7.1.10 Technology Viewpoint Vulnerability Scanner



Technology Usage Viewpoint Vulnerability Scanner

The diagram shows a simplified assignment of the individual PODs to the corresponding application components.

Description of the elements used:

Vulnerability scanner

The vulnerability scanner is a fundamental building block for ensuring a proactive approach in the SOCaaS architecture. It makes it possible to regularly identify weak points in the infrastructure, assess the state of updates, configurations and settings, and provide information for further prioritisation and planning of remedial

measures. The scanner's outputs are often a source for SOAR systems, SLA metrics and reporting, as well as for assessing the client's security level.

NFS / Container Storage

Represents an NFS service or other Container Storage. NFS (Network File System) is a network file protocol at the application layer of the OSI model that enables remote access to file systems over a network. It is one of the most common forms of shared storage in containerised environments, such as Kubernetes, where it is implemented as a PersistentVolume (PV) type. In a broader sense, container storage refers to mechanisms and services providing storage space for containers that need persistent or shared data between restarts or between cluster nodes.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

Cert-bund-data

The cert-bund-data POD is a container unit deployed in a Kubernetes cluster that provides up-to-date security information and warnings from official sources. This POD distributes data on threats, vulnerabilities and recommendations, which are used by security tools to improve detection of and response to current cyber risks.

Data-objects

The data-objects POD is a container unit deployed in a Kubernetes cluster that serves to store and provide structured data objects used within the security ecosystem, such as definitions of vulnerabilities, configurations or analytical rules. This POD ensures centralised access to reference data for the other components of the system, thereby supporting the consistency and currency of information in the processes of detecting and assessing security threats.

Dfn-cert-data

The dfn-cert-data POD is a container unit deployed in a Kubernetes cluster that serves to distribute security information and warnings. This POD provides up-to-date data on cyber threats, vulnerabilities and recommendations, which are used by security tools to improve detection capabilities and enable faster incident response.

Gpg-data

The gpg-data POD is a container unit deployed in a Kubernetes cluster that provides GPG (GNU Privacy Guard) keys and related data needed to verify the signatures of update packages and security data in the Greenbone Vulnerability Management (GVM) ecosystem. This POD ensures the integrity and trustworthiness of transmitted security information, such as vulnerability databases, report formats or configuration files.

Gsa

The gsa POD is a container unit deployed in a Kubernetes cluster that hosts the Greenbone Security Assistant (GSA) component – the web user interface for working with the Greenbone Vulnerability Management (GVM) system. This POD provides access to vulnerability scanning functions via the browser, including user management, scan scheduling, display of results and configuration of the entire security environment.

Gvmd

The gvmd POD is a container unit deployed in a Kubernetes cluster that hosts the Greenbone Vulnerability Management Daemon (gvmd) component – the central control service for the GVM/OpenVAS tool. This POD

handles user management, scheduling and launching of scans, processing of results and communication with the database and other services, thereby forming the core of the vulnerability scanning system.

Gvm-tools

The gvm-tools POD is a container unit deployed in a Kubernetes cluster that contains the GVM-Tools – client utilities for automated management of and interaction with the Greenbone Vulnerability Management (GVM) system. This POD enables the execution of commands, scripting of tasks, scheduling of scans and extraction of results via XML or GMP (Greenbone Management Protocol), thereby supporting the integration of GVM into larger security and orchestration workflows.

Notus-data

The notus-data POD is a container unit deployed in a Kubernetes cluster that provides data and updates for vulnerability detection mechanisms, particularly in the context of the Greenbone/OpenVAS tool. This POD distributes structured information on known vulnerabilities in JSON format, which is used by other components for precise and up-to-date testing of systems for security weaknesses.

Openvas-scanner

The openvas-scanner POD is a container unit deployed in a Kubernetes cluster that hosts the OpenVAS (Open Vulnerability Assessment System) component – a powerful scanning engine for vulnerability testing. This POD performs active scanning of target systems, analyses their security weaknesses on the basis of updated signatures and communicates with the central gvmd daemon to record and process scan results.

Openvas-scanner service_notus

Represents the openvas-scanner service_notus POD. The openvas-scanner service_notus POD is a container unit deployed in a Kubernetes cluster that extends the functionality of the OpenVAS scanning engine with integration with the Notus service. This service enables agentless and fast vulnerability testing based on data from the Notus database without the need for direct network access to all targets. The POD combines the performance of active scanning with dynamic detection of known vulnerabilities, thereby increasing the accuracy and efficiency of security analyses.notus.

Ospd-openvas

The ospd-openvas POD is a container unit deployed in a Kubernetes cluster that hosts the OSPd (Open Scanner Protocol daemon) component for integrating the OpenVAS scanner with the gvmd control system within Greenbone Vulnerability Management. This POD ensures standardised communication between gvmd and OpenVAS, processes scan requests, controls their execution and sends the results back for central processing, thereby enabling scalable and efficient orchestration of vulnerability scans.

Pg-gvm

The pg-gvm POD is a container unit deployed in a Kubernetes cluster that hosts a PostgreSQL database adapted for Greenbone Vulnerability Management (GVM). This POD serves as the central repository for all data on vulnerabilities, scans, results and configurations that are generated and managed by the GVM/OpenVAS tools within the security ecosystem.

Redis-server

The redis-server POD is a container unit deployed in a Kubernetes cluster that hosts a Redis instance – a high-performance in-memory NoSQL database. This POD provides temporarily or persistently stored key-value data for fast access, caching, message queuing or sharing state between applications within a distributed system.

Report-formats

Represents the report-formats POD. The report-formats POD is a container unit deployed in a Kubernetes cluster that provides and manages templates and formats of output reports within the Greenbone Vulnerability

Management (GVM) system. This POD enables the generation of scan results in various standardised formats, thereby supporting automated reporting, archiving and sharing of findings with various stakeholders.

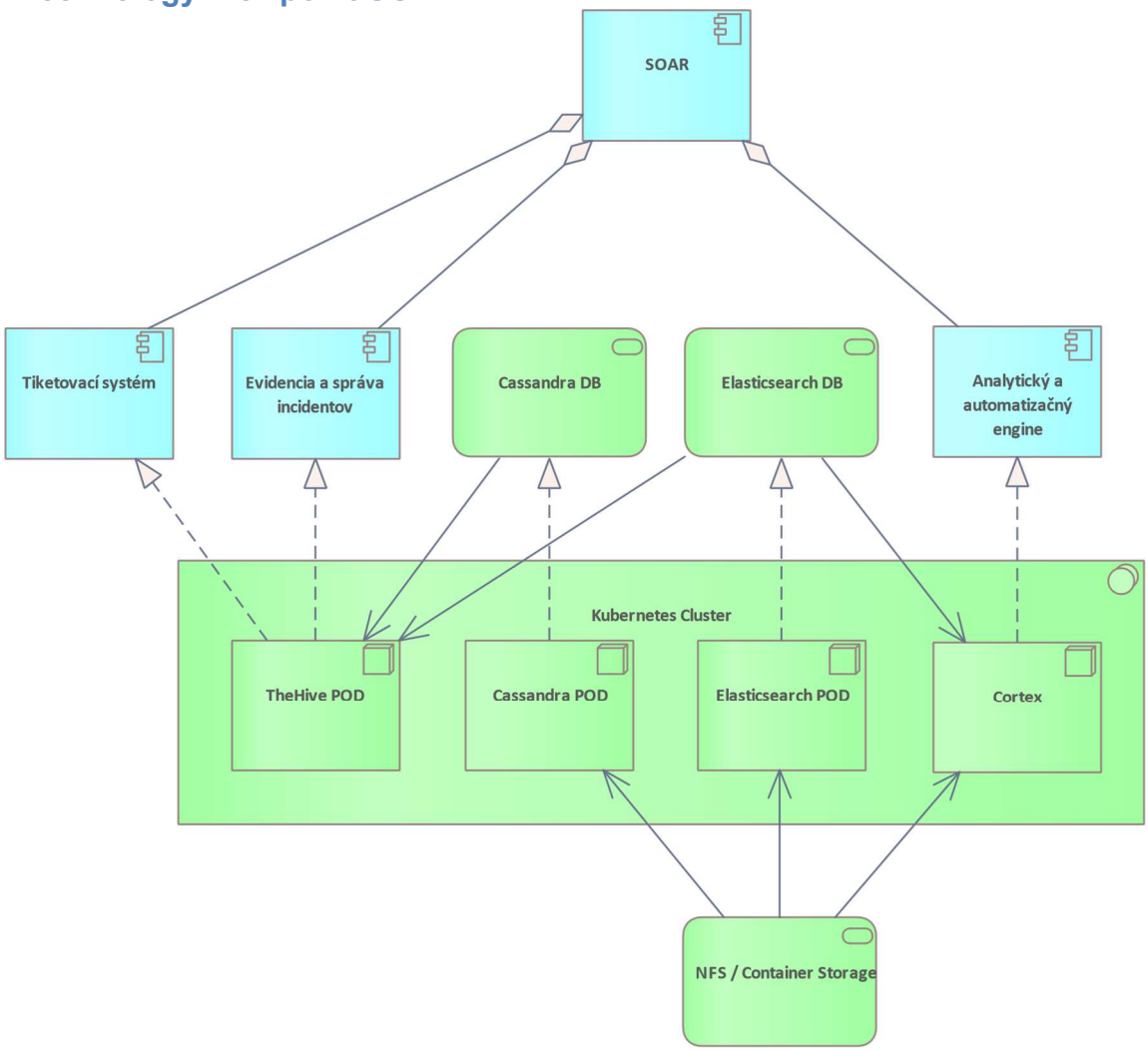
Scap-data

The scap-data POD is a container unit deployed in a Kubernetes cluster that provides structured security data packages in SCAP (Security Content Automation Protocol) format. These data contain standardised information on security configurations, vulnerabilities and policies, which is subsequently used by tools for compliance audits, conformity with security standards and automated assessment of systems.

Vulnerability-tests

The vulnerability-tests POD is a container unit deployed in a Kubernetes cluster that performs automated vulnerability tests against target systems, services or containers. This POD uses security scanners or analysis tools and serves to detect known weaknesses, assess risks and generate outputs for further processing within the security cycle.

7.1.11 Technology Viewpoint SOAR



Technology Usage Viewpoint SOAR alt 1

The diagram shows a simplified assignment of the individual PODs to the corresponding application components. Alt 1 represents the use of the Elasticsearch database directly within the module setup. The design envisages containerised versions of the NoSQL databases for Cassandra and Elasticsearch.

Description of the elements used:

Analytical and automation engine

The analytical and automation engine is a software component or service that combines data analysis and the automation of decision-making or operational processes with the aim of increasing the efficiency, accuracy and speed of the system's response. In the context of cybersecurity or IT operations, this engine processes input data (e.g. logs, events, metrics), applies analytical rules or models (e.g. anomaly detection, correlation) and subsequently performs automated responses or actions via integrated tools and API interfaces.

Incident recording and management

Incident recording and management is a core element of every security operation (SOC). This subcomponent ensures the systematic documentation, tracking and management of security events and incidents throughout their life cycle – from detection through analysis to closure. A high-quality recording system must enable teams to collaborate in a coordinated manner, document steps, record related artefacts (observables) and link incidents to relevant threats, systems or historical cases. Ideally, it also supports advanced data enrichment, auditing, multi-tenancy and the possibility of retrospective investigation of incidents.

SOAR

SOAR (Security Orchestration, Automation and Response) is the component responsible for automating responses to incidents, orchestrating security tools and supporting analysts' decision-making. In SOCaaS it represents a fundamental component enabling response capacities to be scaled without the need to increase the number of staff. It is essential particularly when processing a large number of alerts, where it reduces the MTTR and MTTC metrics.

Ticketing system

The ticketing system serves as the central communication and organisational channel for the management of security incidents throughout their entire life cycle. It enables the creation, assignment and tracking of tasks and communication between analysts, while supporting audit, SLA monitoring, escalation and notifications.

NFS / Container Storage

Represents an NFS service or other Container Storage. NFS (Network File System) is a network file protocol at the application layer of the OSI model that enables remote access to file systems over a network. It is one of the most common forms of shared storage in containerised environments, such as Kubernetes, where it is implemented as a PersistentVolume (PV) type. In a broader sense, container storage refers to mechanisms and services providing storage space for containers that need persistent or shared data between restarts or between cluster nodes.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

Cassandra DB

The NoSQL Cassandra database technology service is a distributed database system designed for storing large volumes of structured data with an emphasis on high availability, scalability and resilience to outages. It uses a masterless architecture with data replication across nodes, which enables fast writes and reads in a geographically distributed environment without a single point of failure.

Elasticsearch DB

The NoSQL Elasticsearch database technology service is a powerful, distributed system intended for indexing, searching and analysing data in real time, based on a document-oriented model.

It uses a JSON document schema, supports horizontal scaling and high availability, and is optimised for processing large volumes of unstructured or semi-structured data such as logs, metrics or application records.

Cassandra POD

The Apache Cassandra POD for the NoSQL database is a container instance deployed in a Kubernetes environment that hosts a Cassandra node as part of a distributed NoSQL cluster. This POD ensures scalable, highly available and resilient storage of structured data with an emphasis on fast writes and reads in an environment with large data volumes and high redundancy.

Elasticsearch POD

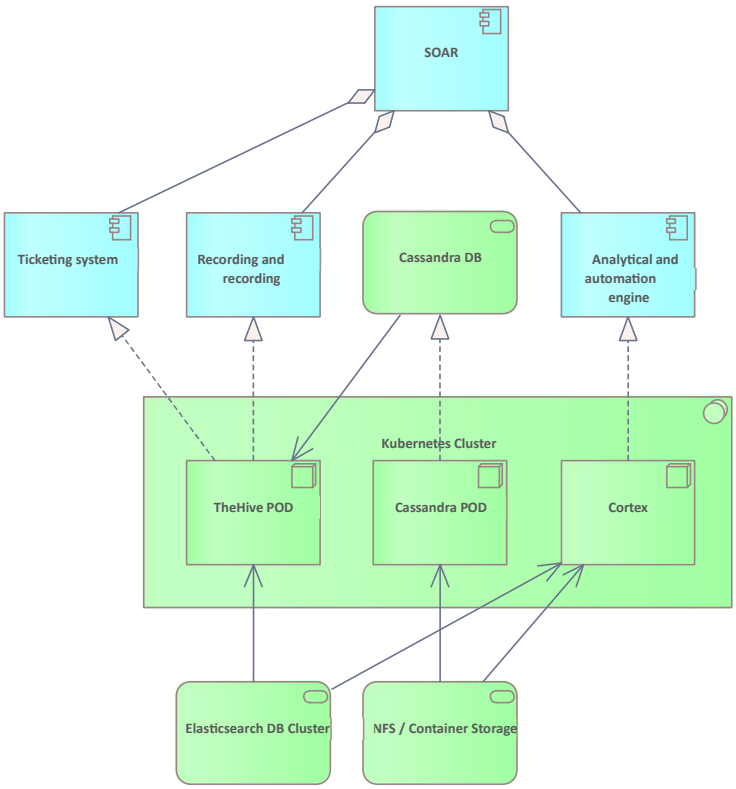
The Elasticsearch POD for the NoSQL database is a container instance deployed in a Kubernetes environment that hosts an Elasticsearch node as part of a distributed NoSQL database cluster. This POD ensures indexing, searching and storage of document-oriented data, while enabling horizontal scaling, high availability and efficient processing of large volumes of unstructured information in real time.

TheHive POD

The TheHive POD for asset inventory is a container instance deployed in a Kubernetes environment that hosts the TheHive application – an open-source platform for managing security incidents and digital assets. This POD enables centralised collection, categorisation and analysis of assets, their linking to incidents and support for collaborative investigation within cybersecurity teams.

Cortex

Represents the Cortex POD for SOAR. The Cortex POD for SOAR is a container unit deployed in a Kubernetes environment that hosts the Cortex service as part of the SOAR solution. This POD enables the automated launching of analytical modules (analyzers) and response actions (responders) based on security incidents, thereby making threat detection, incident response and integration with other security tools more efficient.



Technology Usage Viewpoint SOAR alt 2

The diagram shows a simplified assignment of the individual PODs to the corresponding application components. Alt 2 represents the use of the Elasticsearch Database directly in the form of a centralised DB Cluster. The design envisages a containerised version of the NoSQL database for Cassandra.

Description of the elements used:

Analytical and automation engine

The analytical and automation engine is a software component or service that combines data analysis and the automation of decision-making or operational processes with the aim of increasing the efficiency, accuracy and speed of the system's response. In the context of cybersecurity or IT operations, this engine processes input data (e.g. logs, events, metrics), applies analytical rules or models (e.g. anomaly detection, correlation) and subsequently performs automated responses or actions via integrated tools and API interfaces.

Incident recording and management

Incident recording and management is a core element of every security operation (SOC). This subcomponent ensures the systematic documentation, tracking and management of security events and incidents throughout their life cycle – from detection through analysis to closure. A high-quality recording system must enable teams to collaborate in a coordinated manner, document steps, record related artefacts (observables) and link incidents to relevant threats, systems or historical cases. Ideally, it also supports advanced data enrichment, auditing, multi-tenancy and the possibility of retrospective investigation of incidents.

SOAR

SOAR (Security Orchestration, Automation and Response) is the component responsible for automating responses to incidents, orchestrating security tools and supporting analysts' decision-making. In SOCaaS it represents a fundamental component enabling response capacities to be scaled without the need to increase the number of staff. It is essential particularly when processing a large number of alerts, where it reduces the MTTR and MTTC metrics.

Ticketing system

The ticketing system serves as the central communication and organisational channel for the management of security incidents throughout their entire life cycle. It enables the creation, assignment and tracking of tasks and communication between analysts, while supporting audit, SLA monitoring, escalation and notifications.

NFS / Container Storage

Represents an NFS service or other Container Storage. NFS (Network File System) is a network file protocol at the application layer of the OSI model that enables remote access to file systems over a network. It is one of the most common forms of shared storage in containerised environments, such as Kubernetes, where it is implemented as a PersistentVolume (PV) type. In a broader sense, container storage refers to mechanisms and services providing storage space for containers that need persistent or shared data between restarts or between cluster nodes.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

Cassandra DB

The NoSQL Cassandra database technology service is a distributed database system designed for storing large volumes of structured data with an emphasis on high availability, scalability and resilience to outages. It uses a masterless architecture with data replication across nodes, which enables fast writes and reads in a geographically distributed environment without a single point of failure.

Elasticsearch DB Cluster

The NoSQL Elasticsearch database technology service in the form of a cluster represents a multi-node distributed solution in which individual nodes cooperate in storing, indexing and searching document-oriented data. Each node in the cluster performs a specific role (e.g. master, data or ingest node), which enables horizontal scaling, high availability and resilience to outages, while the system as a whole ensures the processing of a large volume of data in real time.

Cassandra POD

The Apache Cassandra POD for the NoSQL database is a container instance deployed in a Kubernetes environment that hosts a Cassandra node as part of a distributed NoSQL cluster. This POD ensures scalable, highly available and resilient storage of structured data with an emphasis on fast writes and reads in an environment with large data volumes and high redundancy.

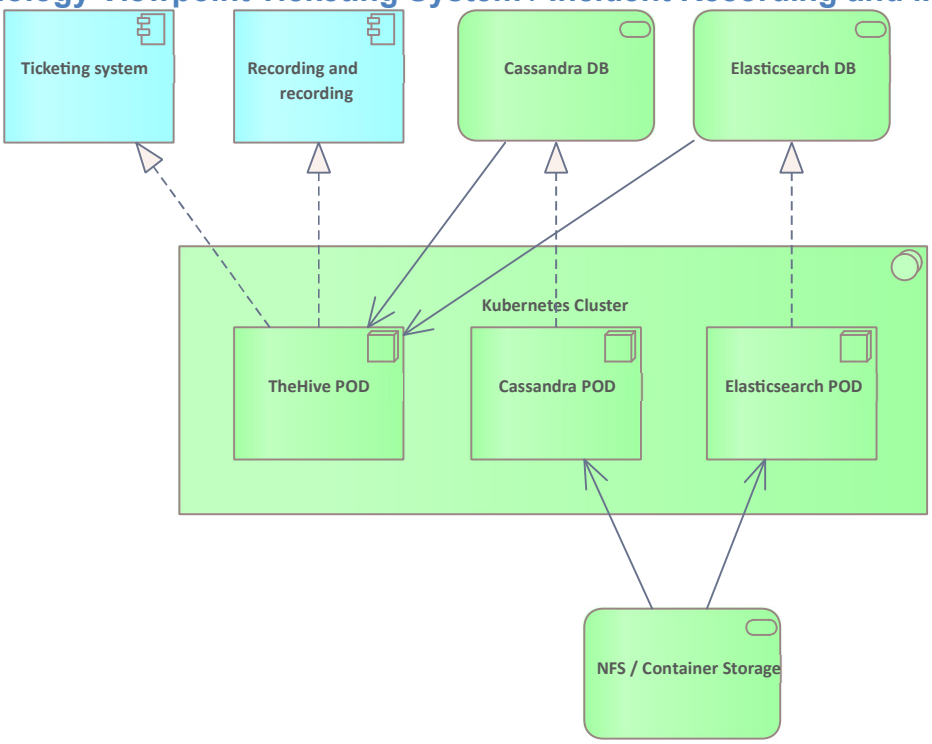
TheHive POD

The TheHive POD for asset inventory is a container instance deployed in a Kubernetes environment that hosts the TheHive application – an open-source platform for managing security incidents and digital assets. This POD enables centralised collection, categorisation and analysis of assets, their linking to incidents and support for collaborative investigation within cybersecurity teams.

Cortex

Represents the Cortex POD for SOAR. The Cortex POD for SOAR is a container unit deployed in a Kubernetes environment that hosts the Cortex service as part of the SOAR solution. This POD enables the automated launching of analytical modules (analyzers) and response actions (responders) based on security incidents, thereby making threat detection, incident response and integration with other security tools more efficient.

7.1.12 Technology Viewpoint Ticketing System / Incident Recording and Management



Technology Usage Viewpoint Ticketing System / Incident Recording and Management alt 1

The diagram shows a simplified assignment of the individual PODs to the corresponding application components. Alt 1 represents the use of the Elasticsearch database directly within the module setup. The design envisages containerised versions of the NoSQL databases for Cassandra and Elasticsearch.

Description of the elements used:

Incident recording and management

Incident recording and management is a core element of every security operation (SOC). This subcomponent ensures the systematic documentation, tracking and management of security events and incidents throughout their life cycle – from detection through analysis to closure. A high-quality recording system must enable teams to collaborate in a coordinated manner, document steps, record related artefacts (observables) and link incidents to relevant threats, systems or historical cases. Ideally, it also supports advanced data enrichment, auditing, multi-tenancy and the possibility of retrospective investigation of incidents.

Ticketing system

The ticketing system serves as the central communication and organisational channel for the management of security incidents throughout their entire life cycle. It enables the creation, assignment and tracking of tasks and communication between analysts, while supporting audit, SLA monitoring, escalation and notifications.

NFS / Container Storage

Represents an NFS service or other Container Storage. NFS (Network File System) is a network file protocol at the application layer of the OSI model that enables remote access to file systems over a network. It is one of the most common forms of shared storage in containerised environments, such as Kubernetes, where it is implemented as a PersistentVolume (PV) type. In a broader sense, container storage refers to mechanisms and services providing storage space for containers that need persistent or shared data between restarts or between cluster nodes.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

Cassandra DB

The NoSQL Cassandra database technology service is a distributed database system designed for storing large volumes of structured data with an emphasis on high availability, scalability and resilience to outages. It uses a masterless architecture with data replication across nodes, which enables fast writes and reads in a geographically distributed environment without a single point of failure.

Elasticsearch DB

The NoSQL Elasticsearch database technology service is a powerful and distributed system designed for indexing, searching and analysing data in real time, based on a document-oriented model. It uses a JSON document schema, supports horizontal scaling and high availability, and is optimised for processing large volumes of unstructured or semi-structured data, such as logs, metrics or application records.

Cassandra POD

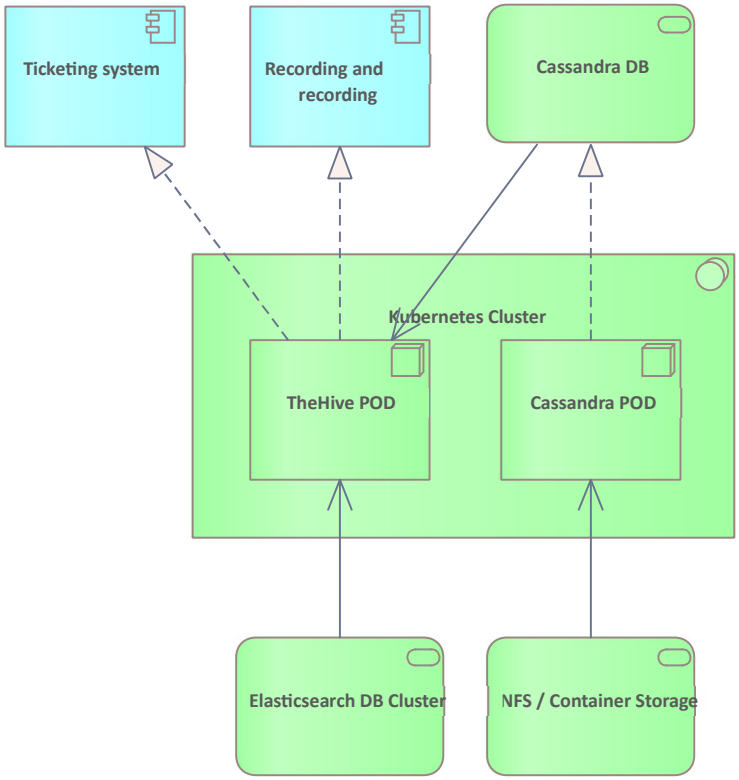
The Apache Cassandra POD for the NoSQL database is a container instance deployed in a Kubernetes environment that hosts a Cassandra node as part of a distributed NoSQL cluster. This POD ensures scalable, highly available and resilient storage of structured data with an emphasis on fast writes and reads in an environment with large data volumes and high redundancy.

Elasticsearch POD

The Elasticsearch POD for the NoSQL database is a container instance deployed in a Kubernetes environment that hosts an Elasticsearch node as part of a distributed NoSQL database cluster. This POD ensures indexing, searching and storage of document-oriented data, while enabling horizontal scaling, high availability and efficient processing of large volumes of unstructured information in real time.

TheHive POD

The TheHive POD for asset inventory is a container instance deployed in a Kubernetes environment that hosts the TheHive application – an open-source platform for managing security incidents and digital assets. This POD enables centralised collection, categorisation and analysis of assets, their linking to incidents and support for collaborative investigation within cybersecurity teams.



Technology Usage Viewpoint Ticketing System / Incident Recording and Management alt 2

The diagram shows a simplified assignment of the individual PODs to the corresponding application components. Alt 2 represents the use of the Elasticsearch Database directly in the form of a centralised DB Cluster. The design envisages a containerised version of the NoSQL database for Cassandra.

Description of the elements used:

Incident recording and management

Incident recording and management is a core element of every security operation (SOC). This subcomponent ensures the systematic documentation, tracking and management of security events and incidents throughout their life cycle – from detection through analysis to closure. A high-quality recording system must enable teams to collaborate in a coordinated manner, document steps, record related artefacts (observables) and link incidents to relevant threats, systems or historical cases.

Ideally, it also supports advanced data enrichment, auditing, multi-tenancy and the possibility of retrospective investigation of incidents.

Ticketing system

The ticketing system serves as the central communication and organisational channel for the management of security incidents throughout their entire life cycle. It enables the creation, assignment and tracking of tasks and communication between analysts, while supporting audit, SLA monitoring, escalation and notifications.

NFS / Container Storage

Represents an NFS service or other Container Storage. NFS (Network File System) is a network file protocol at the application layer of the OSI model that enables remote access to file systems over a network. It is one of the most common forms of shared storage in containerised environments, such as Kubernetes, where it is

implemented as a PersistentVolume (PV) type. In a broader sense, container storage refers to mechanisms and services providing storage space for containers that need persistent or shared data between restarts or between cluster nodes.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

Cassandra DB

The NoSQL Cassandra database technology service is a distributed database system designed for storing large volumes of structured data with an emphasis on high availability, scalability and resilience to outages. It uses a masterless architecture with data replication across nodes, which enables fast writes and reads in a geographically distributed environment without a single point of failure.

Elasticsearch DB Cluster

The NoSQL Elasticsearch database technology service in the form of a cluster represents a multi-node distributed solution in which individual nodes cooperate in storing, indexing and searching document-oriented data. Each node in the cluster performs a specific role (e.g. master, data or ingest node), which enables horizontal scaling, high availability and resilience to outages, while the system as a whole ensures the processing of a large volume of data in real time.

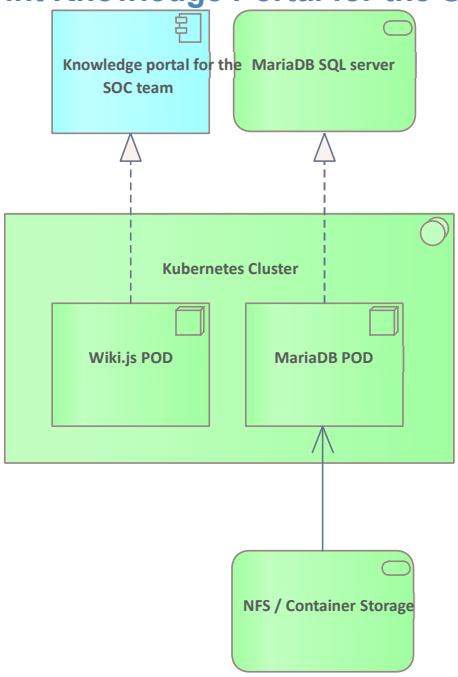
Cassandra POD

The Apache Cassandra POD for the NoSQL database is a container instance deployed in a Kubernetes environment that hosts a Cassandra node as part of a distributed NoSQL cluster. This POD ensures scalable, highly available and resilient storage of structured data with an emphasis on fast writes and reads in an environment with large data volumes and high redundancy.

TheHive POD

The TheHive POD for asset inventory is a container instance deployed in a Kubernetes environment that hosts the TheHive application – an open-source platform for managing security incidents and digital assets. This POD enables centralised collection, categorisation and analysis of assets, their linking to incidents and support for collaborative investigation within cybersecurity teams.

7.1.13 Technology Viewpoint Knowledge Portal for the SOC Team



Technology Usage Viewpoint Knowledge Portal for the SOC Team alt 1

The diagram shows a simplified assignment of the individual PODs to the corresponding application components. Alt 1 represents the use of the database directly within the module setup. The design envisages a containerised version of the SQL server.

Description of the elements used:

Knowledge portal for the SOC team

The knowledge portal for the SOC team is an important building block of every modern security operations centre. It serves as a central repository of know-how, playbooks, procedures, recommendations, analytical guides and often also historical lessons learned from incidents. The knowledge portal:

- Increases the efficiency and consistency of incident response,
- Enables rapid onboarding of new team members,
- Supports standardisation of processes and methodologies,
- Enables the team to retain and share proven procedures and its own analytical experience,
- Strengthens the ability to learn collectively from incidents and to build internal "threat intel" knowledge.

NFS / Container Storage

Represents an NFS service or other Container Storage. NFS (Network File System) is a network file protocol at the application layer of the OSI model that enables remote access to file systems over a network. It is one of the most common forms of shared storage in containerised environments, such as Kubernetes, where it is implemented as a PersistentVolume (PV) type. In a broader sense, container storage refers to mechanisms and services providing storage space for containers that need persistent or shared data between restarts or between cluster nodes.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

MariaDB SQL server

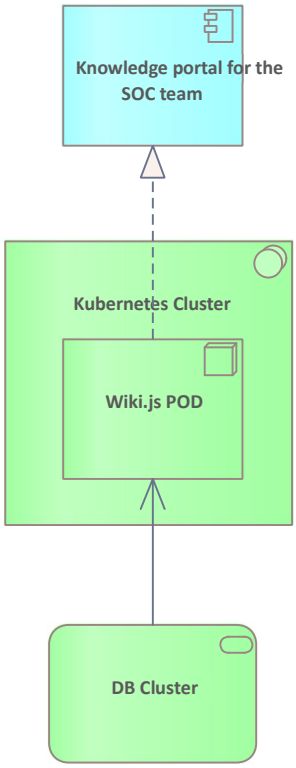
The MariaDB relational database technology service is a powerful and scalable relational database system built on an SQL architecture that serves for the persistent storage, management and querying of structured data using relational tables. MariaDB supports transactions, referential integrity, replication and high availability, making it suitable for deployment in critical enterprise applications where performance, reliability and security of data processing are important.

MariaDB POD

The MariaDB POD for the relational SQL database is a container unit deployed in a Kubernetes cluster that hosts the MariaDB database server – a powerful open-source relational database system compatible with MySQL. This POD provides persistently stored, structured data for applications or services, supports transactions and data integrity, and enables horizontal or vertical scaling within the cloud infrastructure.

Wiki.js POD

The Wiki.js POD for the Knowledge Portal for the SOC team is a container unit deployed in a Kubernetes cluster that hosts the Wiki.js application – a modern and powerful system for managing documentation and knowledge. This POD serves as the central knowledge portal for the cybersecurity (SOC) team, where information on procedures, incidents, analyses and internal processes is stored, managed and shared with an emphasis on security, auditability and access rights.



Technology Usage Viewpoint Knowledge Portal for the SOC Team alt 2

The diagram shows a simplified assignment of the individual PODs to the corresponding application components. Alt 2 represents the use of the database directly in the form of a centralised DB cluster.

Description of the elements used:

Knowledge portal for the SOC team The knowledge portal for the SOC team is an important building block of every modern security operations centre. It serves as a central repository of know-how, playbooks, procedures, recommendations, analytical guides and often also historical lessons learned from incidents. The knowledge portal:

- Increases the efficiency and consistency of incident response,
- Enables rapid onboarding of new team members,
- Supports standardisation of processes and methodologies,
- Enables the team to retain and share proven procedures and its own analytical experience,
- Strengthens the ability to learn collectively from incidents and to build internal "threat intel" knowledge.

DB Cluster

Represents a clustered relational database service. A clustered relational database service is a distributed database system that provides relational database services through multiple interconnected database nodes (clusters), thereby ensuring high availability, scalability, redundancy and fault tolerance.

Kubernetes Cluster

Represents a Kubernetes Cluster. A Kubernetes Cluster is a distributed computing environment used for the orchestration of containerised applications. It ensures the deployment, scaling, monitoring, updating and management of containers (most commonly Docker or compatible OCI containers) across a group of physical or virtual servers. In a Kubernetes cluster, computing resources (CPU, memory, storage, network) are abstracted and managed as a unified system, which enables automated management of the application life cycle.

Wiki.js POD

The Wiki.js POD for the Knowledge Portal for the SOC team is a container unit deployed in a Kubernetes cluster that hosts the Wiki.js application – a modern and powerful system for managing documentation and knowledge. This POD serves as the central knowledge portal for the cybersecurity (SOC) team, where information on procedures, incidents, analyses and internal processes is stored, managed and shared with an emphasis on security, auditability and access rights.

8 Service Level (SLA) and Performance Requirements

Given the nature of the project and PoC testing, basic availability is envisaged, and within the To-Be design a high-availability system is envisaged.

8.1 Support Levels and Helpdesk

The Helpdesk will be delivered through 3 support levels with the following designations:

- **L1 support** (direct customer contact) – the initial level of support, which is responsible for resolving basic problems and requests of end users and other services requiring a basic level of technical support. The basic function of 1st-level support is to gather information, perform a basic analysis and determine the cause of the problem and its classification. Typically, level L1 handles straightforward and simple problems and basic diagnostics, verification of the availability of individual infrastructure layers (network, operating, visualisation, application, etc.) and basic user problems (typically a forgotten password), verification of SW and HW settings, etc.

- **L2 support** (escalation of requests from L1) – a selected group of guarantors with knowledge of the system – resolution teams with deeper technological knowledge of the given area. Resolvers at the L2 Support level do not communicate directly with the end user, but are responsible for providing cooperation to the 1st-level support resolvers in resolving an escalated report, which among other things also includes a retrospective
check and a more detailed analysis of the identified data handed over by the 1st-level support resolvers. The outcome of such a check may be confirmation, clarification or reassessment of the report, depending on the needs. The primary objective of resolvers at the L2 Support level is to bring the Report under control as soon as possible and subsequently resolve it – with the possibility of escalation to a higher support level – L3 Support.
- **L3 support** (escalation of requests from L2) – provided by the Provider on the basis of the support contract – 3rd-level support represents the highest level of support for resolving the most difficult Reports, including performing in-depth analyses and resolving extreme cases.

The availability of L1, L2 and L3 support for the infrastructure is in accordance with the selected SLA.

Incidents will be reported to the helpdesk:

- Via the incident management tool provided by the Provider, or, subject to prior agreement, also
- By electronic mail/e-mail (with a delivery receipt request set)
- For duplicate notification in the case of a critical-priority problem, also by telephone so that resolution of the incident can begin as soon as possible

By means of the incident management tool, the Provider will ensure online reporting of service events with the possibility of tracking the status of their resolution, and will ensure access to the record of reported service events.

8.2 Proposed SLA Parameters and Incident Resolution

An incident is deemed to be an error of the IS, i.e. behaviour contrary to the operating and user documentation of the IS. An error that occurred outside the IS environment, e.g. an outage in the provision of a specific Government Cloud service or of the communication infrastructure, is not deemed to be an incident.

Categorisation of incident urgency:

Categorisation of incident urgency	Incident severity	Description of incident urgency
A	**Critical**	Critical errors that cause a complete failure of the system as a whole, where no part of it can be used and the required output cannot be provided from the IS.
B	**High**	Errors and deficiencies that cause a partial failure of the system and prevent the use of part of the system.
C	**Medium**	Errors and deficiencies that cause partial restrictions in the use of the system.

D	**Low**	Cosmetic and minor errors.

Designation of incident severity:

Designation of incident severity	Impact	Description of impact
1	**catastrophic**	catastrophic impact, direct financial impact or loss of data,
2	**significant**	significant impact or loss of data
3	**minor**	minor impact or loss of data

The incident priority is calculated as a combination of impact and urgency in accordance with ITIL v3 best practices, as shown in the following incident priority matrix:

Incident priority matrix		**Impact**		
		Catastrophic - 1	Significant - 2	Minor - 3
Urgency	Critical - A	1	2	3
	High - B	2	3	3
	Medium - C	2	3	4
	Low - D	3	4	4

Response times:

Designation of incident priority	Response time(1) from reporting of the incident to the start of resolution of the incident	Time to final resolution of the incident from reporting of the incident (DKVI) (2)	Reliability (3) (number of incidents per year)
1	30 min.	4 hours	2
2	1 hr.	6 hours	5
3	8 hrs.	12 hours	10
4	8 hrs.	Resolved and deployed within planned releases	

1. The response time is the time between the reporting of an incident by the contracting authority (including users of the IS who are not in an employment relationship with the contracting authority) to the L3 helpdesk and its acceptance for resolution.
2. Restoration of standard operation – the time between the reporting of an incident by the contracting authority and the resolution of the incident by the successful tenderer (until the functionality of the environment is fully restored again). The time to final resolution of an incident from its reporting by the

contracting authority is counted throughout the whole day. This time does not include the time needed for the necessary cooperation of the contracting authority, if such cooperation is required to resolve the incident. Where necessary, the successful tenderer is entitled to request approval of the incident resolution from the contracting authority.

3. The maximum number of incidents per calendar year is the maximum number of incidents of a specific priority. Duplicate or technically related incidents (submitted within one working day, during working hours of 8 hours) are considered as one incident.
4. Incidents reported by the contracting authority to the successful tenderer within the test environment i. Have priority 3 or lower ii. Relate exclusively to the availability of the test environment iii. An incident relating to the functionality currently being tested is not considered an incident in the test environment.

The above SLA parameters will not be applied to the following services:

- System support services on request (beyond the flat fee)
- Services for implementing application changes resulting from legislative and methodological changes (beyond the flat fee)

Separate delivery parameters will be agreed for these services.

8.3 Proposed Planned Availability of the Infrastructure

Availability is a term from the field of security management in an organisation. Availability means that data are accessible at the moment they are needed. A breach of availability is referred to as undesired destruction or unavailability. Availability is usually expressed as a percentage of time in a given period, usually per year.

Cloud service

Description	Parameter	Note
Operating hours	24 hours	From 00:00 to 23:59 throughout the year
Service interventions	8 hours	From 22:00 to 06:00 on working days Service interventions are always carried out at times of low demand for the service. At hours other than times of low demand, only after approval by the competent persons of the Contracting Authority.
	8 hours	From 22:00 to 06:00 on non-working days and public holidays Service interventions are always carried out at times of low demand for the service. At hours other than times of low demand, only after approval by the competent persons of the Contracting Authority.
Availability of the IS production environment	99,9%	• 99.9% of 24/7/365, i.e. the max. annual outage is 8h 41m 38s The maximum daily outage is 1m 26s The maximum weekly outage is 10m 4.8s The maximum monthly outage is 43m 28s The maximum quarterly outage is 2h 10m 24s The maximum annual outage is 8h 41m 38s • Such a period is always deemed to be the time from 0.00 to 23.59 on working days of the week. • Unavailability of the IS is counted from the reporting of an incident by the Contracting Authority during the Provider's support availability hours (i.e. reporting of the incident to L3 within the hours of the SLA selected from the offer). Service windows and planned outages of the IS are not counted towards IS availability.

9 Software Licences and Source Code

Component name	Application	Licence	Primary deployment
ELK Stack	SIEM	MIT License	Docker / OpenStack (PoC)
Shuffle	SOAR	GNU Affero General Public License	Docker / OpenStack (PoC)
Wazuh	EDR	GPLv2	Docker / OpenStack (PoC)
Greenbone Community edition	Vulnerability scanner	Creative Commons Attribution Share Alike 4.0 International	Docker / OpenStack (PoC)
OpenCTI	CTI	Apache 2.0 license	Docker / OpenStack (PoC)
MISP	CTI	AGPL-3.0 license	Docker / OpenStack (PoC)
GLPI	Asset management	GPL-3.0	Docker / OpenStack (PoC)
NetBox	Asset management / IPAM/DCIM	Apache 2.0 license	Docker / OpenStack (PoC)
DFIR IRIS	Ticketing	LGPL-3.0 license	Docker / OpenStack (PoC)
DFIR IRIS	Incident recording and management	LGPL-3.0 license	Docker / OpenStack (PoC)
Kibana (ELK)	Reporting	Apache 2.0-license	Docker / OpenStack (PoC)
Wiki.js	Knowledge base	AGPL-v3 License	Docker / OpenStack (PoC)

10 Ensuring Availability, Backup and Recovery of the Solution

10.1 Ensuring Availability

The basic pillar of the availability of critical systems in a cloud environment is system clustering, which ensures the automatic failover of an application to a standby node (virtual server) in the event of a failure of the primary node and the execution of all operations associated with it. Critical applications or network services are implemented in high availability (High Availability – HA) systems using a software cluster..., which represents the standard in the field of high availability.

The reliability of critical systems is implemented at two (up to four) levels:

1. High availability of the system through the use of software clustering. This ensures a fast, operator-independent start-up of the system after a hardware or software crash or system failure.

2. Cyclical (and automatic) data backup (system logs, application data, the applications themselves...). In this way we protect the system against any loss of valuable data.

3. The purpose of a service intervention is to return the system to its original state and repair the hardware or network faults that caused part of the system to stop working, thus preparing it for another potential outage. Should the designed “redundancy” of the system be insufficient to keep the system functioning in the event of larger-scale outages, or in the case of outages caused by incorrect operation (administrator errors, etc.), it is sometimes necessary to bring systems back into operation using data backups as well.

4. Replication of data to another location to make the system (computer centre) resilient against large-scale outages, natural disasters, etc. However, this matter is fully ensured by the cloud solution provider.

Backups are created of the server application environment, the system configuration, operational records and the databases themselves.

All important central components of the system and infrastructure are deployed in production operation with multiple instances as needed, so that the failure of any single component does not cause the failure or unavailability of certain system services.

All internal network interfaces of workstations are protected by a local firewall, have an access control policy set with ports blocked by default (default deny all), and communication is permitted only for designated connections according to the communication matrix.

10.2 Backup

Given the nature of the project, only basic testing of backup is envisaged, namely at the level of the databases and the file system used by the individual components.

10.3 Recovery of the Solution

Given the nature of the project, only basic testing of recovery is envisaged, namely at the level of the databases and the file system used by the individual components.

11 Security – Addressing Security Requirements

Secure development principles

The requirements set out in this document must be met for all components used during development and all artefacts that are considered products of development.

The system will be developed in accordance with internationally recognised recommendations and good practices in the field of secure software development. Development activities will reflect the requirements defined within the CISA Secure by Design initiative (https://www.cisa.gov/resources-tools/resources/secure-by-design) as well as the OWASP Developer Guide methodology (https://owasp.org/www-project-developer-guide/release/).

By products of development we mean in particular:

- High-level Design (HLD), Low-level Design (LLD), Entity Relationship Diagram (ERD), process diagrams and the like
- software source code
- executable programs that are part of the delivery
- configurations
- database schemas
- test data sets

It is appropriate to divide the security requirements into three parts:

- Requirements for the development environment
- Requirements for the developed product itself – the application
- Requirements for the production environment in which the product is operated.

The aim of these measures is to reduce the likelihood and impact of threats associated with the development and delivery of software, in particular:

- introduction of vulnerabilities as a result of poor programming techniques,
- introduction of vulnerabilities as a result of using unsafe and unsupported frameworks,
- introduction of undesired functionality into the code, whether through negligence or intentionally,
- delay in delivery due to an outage of development environments,
- breach of personal data protection principles due to their improper handling,
- introduction of vulnerabilities due to incorrect configuration of environments,
- criminal liability as a result of copyright infringement,
- leakage of sensitive data, loss of sensitive data,
- inability to secure evidence in the event of a security incident, both in development and in operation, and the like.

During the development of the project, it is necessary to prevent and avoid the introduction of the most common implementation errors, vulnerabilities and risks by adhering to a secure development life cycle, best practices and a suitably chosen security standard and/or methodology. When implementing the proposed solution, the software is verified during development against well-known vulnerabilities and security recommendations, which form part of adherence to secure development in order to prevent the most common security risks for web applications.

By components used during development we mean in particular:

- workstations used in development
- virtualisation environment
- servers used in development (development, test)
- network environment used in development, including home networks
- printers and other active elements
- the physical environment in which development takes place, including home offices
- support services used in development such as communication tools (MS Teams, Jira, email...)
- development tools (editors, compilers)
- development frameworks
- middleware (web servers, proxy servers, application servers)
- source code and configuration repositories (github)
- backup applications/services/systems and backed-up data

authentication mechanisms (AD, AAD, LDAP, Radius, Kerberos, SAML, cloud access security broker and the like)

Securing the development environment

Development follows the principles of best security practice, including secure administration, configuration, patching and hardening of workstations and servers, anti-malware protection, segregation of physical and logical environments, account policies, management of login credentials and secure login procedures, e.g. in accordance with ISO/IEC 27001.

All devices used in development will be protected by a firewall, both at the network perimeter (the interface between networks) and locally by means of a host firewall that permits only necessary connections. Only devices that are needed for development may be connected to the networks in which development takes place.

End stations on which development takes place must not be accessible to or used by other persons. In the case of computers, automatic screen locking protected by a password will be ensured and required to be set.

The computer will not automatically connect to Wi-Fi networks other than the protected networks used in development. The Wi-Fi SSID to which the developer's computer is connected must have WPA2-controlled access and a complex password, ideally authentication via WPA2 Enterprise. Where a physical LAN is used, access protection must be used.

All devices used in development, including the OS of network and security elements (routers, switches, firewalls, Wi-Fi access points), must have current security patches installed and the firmware version must be at least N-1.

All network communication to and from the internet must pass through a device filtering dangerous and malicious content. All communication from computers used for development that are located outside the protected internal networks (for example, in a home office) must pass through the corporate VPN gateway, which inspects and filters dangerous content.

Only legal, supported and up-to-date software will be used in development. The disks of all portable devices (laptops) must be encrypted. It is also necessary to encrypt the disks of non-portable workstations and servers that are not located in sufficiently protected premises of data centres and server rooms.

All configurations must be protected against unauthorised changes. All configuration changes will be automatically detected and reported.

Only personal accounts may be used to log in to systems. Personal accounts used for ordinary work must be separate from personal accounts for administration. The use of generic and shared accounts (root, administrator) is prohibited, with the exception of disaster recovery. All accounts must be protected using SSH keys or, in exceptional cases where necessary, by a sufficiently strong password or passphrase with a high degree of entropy.

Only the minimum necessary set of software required for the specific activity may be installed on all systems used for development.

All test data, sensitive data, passwords and the like are stored in defined locations.

End-user management – The rights themselves, whether for access to information, permissions to work with data or access to individual modules and functionalities, will be governed by permissions defined in the central user management and identity management, the so-called IAM (Identity and Access Management), using oAuth/SAML with the possible use of 2FA, on the basis of which the individual user roles will be assigned.

Ensuring integrity

Validation of all user inputs and actions will be performed on the server side and, for suitable operations, also on the client side. All user inputs must be checked on the server side.

Both the web server process and the backend database process must be configured to run under a unique user account with a limited set of privileges.

The web server will be configured so that files with web content are accessible to the process belonging to the web server service read-only, but not for writing (unless required by the service and process themselves).

Web content and all logs it creates will be located on a separate hard disk or on a different logical partition from the OS and the web server. In the case of containerisation, the logging method will be adapted in line with the given orchestration platform.

To ensure the security requirements associated with the development of the project, a suitable tool will be used that provides continuous code quality inspection, validation of the secure life cycle and checks that reveal bugs, duplicated code and security holes.

Securing communication

All communication channels will be known and described. The individual components of the system will be separated into layers fulfilling a specific role.

The basic layers are:

- Access layer
- Application layer
- Database layer

The individual layers will be placed in separate, logically isolated network segments or separately executable blocks, or by means of defining access on the basis of access groups, with communication between them permitted on the basis of firewall rules or rules of the orchestration platform, and in addition monitoring and filtering will be possible.

The administration and monitoring interfaces of network elements and servers are not accessible directly from the external network or the internet. Administration and monitoring interfaces are accessible only from pre-specified networks (e.g. the administrator network, a designated internal network).

The web server will be externally accessible exclusively via the HTTPS protocol. In line with best practices, the web portal is not to be accessed externally via HTTP.

The identity of the web portal will be secured by a valid, trusted certificate issued for the domain on which the portal in question is available. For these purposes, a certificate and a technical certificate from an internal CA may be used. The web portal must not use untrusted or expired SSL/TLS certificates towards the external environment.

Data that are sensitive in terms of integrity or confidentiality must be transmitted only via an encrypted SSL/TLS connection. Sensitive data (especially login credentials) must be transmitted exclusively via an encrypted channel.

The web server will support TLS 1.2, TLS 1.3 and newer. By default, it will not support the old versions SSLv2, SSLv3, TLS 1.0 and TLS 1.1, unless required otherwise.

The web server is not to support ciphers with a key shorter than 112 bits and a block shorter than 64 bits. The web server must not support (they must be disabled) the HTTP methods OPTIONS, TRACK and TRACE.

The web application will, to the greatest possible extent, take into account and apply recommendations for eliminating well-known vulnerabilities according to best practices.

The basic logical topology of the system solution is shown within the physical architecture.

Access control

Access control to the application uses IAM, which assigns user roles on the basis of permissions defined in the central user management based on oAuth/SAML and possibly 2FA.

Access control between individual zones or subnets is ensured by means of firewalls or another security mechanism.

System logging and collection of audit records

Logging and monitoring of the system will be ensured by technology supplied by the Contracting Authority. It is recommended to set appropriate metrics for monitoring resource utilisation and capacity for each monitored device/system.

System monitoring and logging should include alerts that should be provided to the system administrator in cases such as: storage capacity approaching full, abnormally high system load, an unusual security event detected.

Logs and audit records will contain specific information on the date, time and time zone used, should be transmitted over a secured channel, and the granularity of logging will correspond to the requirements of the

given service. Log retention should be set so as to prevent the logging location from filling up and not to affect the stability of resources.

Logging is subject to a centrally defined retention policy that determines the retention period for individual types of log records depending on their nature (e.g. operational, security, audit). Retention periods will be configurable and set in accordance with legislative requirements and the operational needs of the Contracting Authority. The retention policy recommended by the supplier is at least 1 month for logging and at least one year for audit logs. After the retention period has expired, records will be automatically archived or securely deleted.

User Management Methodology

The rights themselves, whether for access to information, permissions to work with data or access to individual modules and functionalities, will be governed by permissions defined in the central user management and identity management, the so-called IAM (Identity and Access Management), using oAuth/SAML with the possible use of 2FA, on the basis of which the individual user roles will be assigned. IAM will also be used to define the way in which individual users may interact with the system (CRUD operations).

12 Data Migration

No data migration is envisaged within the proposed solution. The solution is designed as a new environment with no requirement to transfer historical data from original or external systems. The subject of the design is the creation of the target architecture and a functional solution for the collection, processing, retention and evaluation of data generated only during the operation of the platform.

For this reason, this document does not include migration scenarios, transformation rules or mapping of source and target data structures. In the event of a future extension of the solution to integrate existing systems or take over historical data, a separate data migration analysis will need to be prepared.

12.1 Data Migration Procedure

Data migration will not be carried out within this solution. For this reason, no specific migration procedure, migration schedule or technical steps for data transfer are defined.

The initial population of the system will be carried out only in the form of a basic configuration initialisation of the environment, in particular by creating tenants, user roles, access permissions, integration links and the configuration parameters needed to start operation.

12.2 Verification of Data Migration

Since data migration is not part of the proposed solution, its verification will not be carried out either.
Verification will relate only to the correctness of the initialisation configuration, the availability of integration interfaces and the functionality of collecting new operational and security data after deployment of the solution.

13 Implementation and Deployment Schedule, Dependencies

The project schedule is set in accordance with the call and is not modified within the DNR.

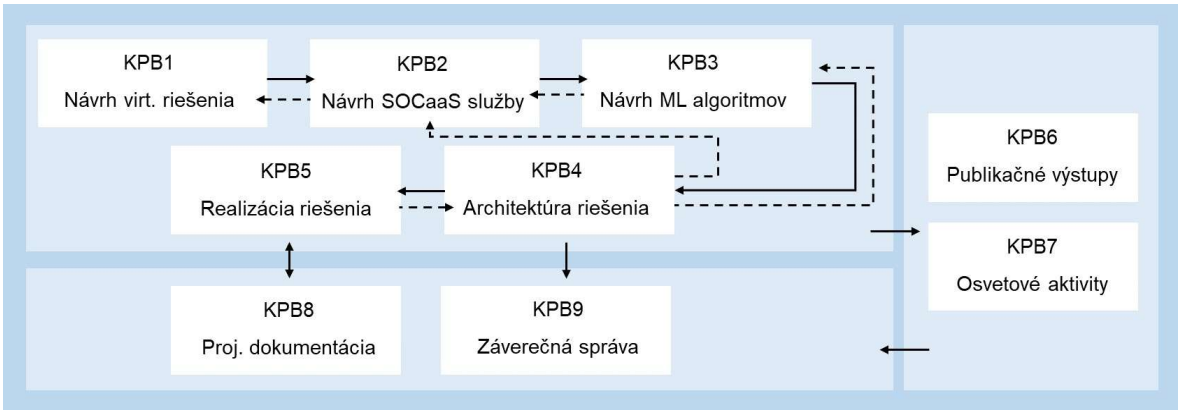
The structure and description of the project plan are based on the work packages defined in the project and their interconnections, as well as their links to the defined basic components of the delivered project output. To define the logical and temporal dependencies between the individual work packages (or packages decomposed one level lower in connection with the defined partial outputs, where this was necessary due to the parallel implementation of work packages), the Pert diagram technique was also used, with identification of the critical path and slack. To clearly capture the project timeline, a Gantt chart was compiled, showing the months of implementation of the individual work packages. We present the individual planned work packages summarised in the form of a list. In this way we support checking the adequacy of the planned work in relation to the scope of the project.

13.1 Work Packages

The entire project is broken down according to PRINCE2 into three main components through the application of the PBS technique. The components were subsequently structured into the appropriate number of work packages using a WBS. By applying the PRINCE2 methodology, we will thus ensure that the defined objectives (chapter 1.1) are met through the delivery of the defined project product, broken down into its main components. The defined work packages (WP – Work Package) then made it possible to group the individual activities needed to create the specified components. The project structure created in this way is captured graphically in the following diagram.

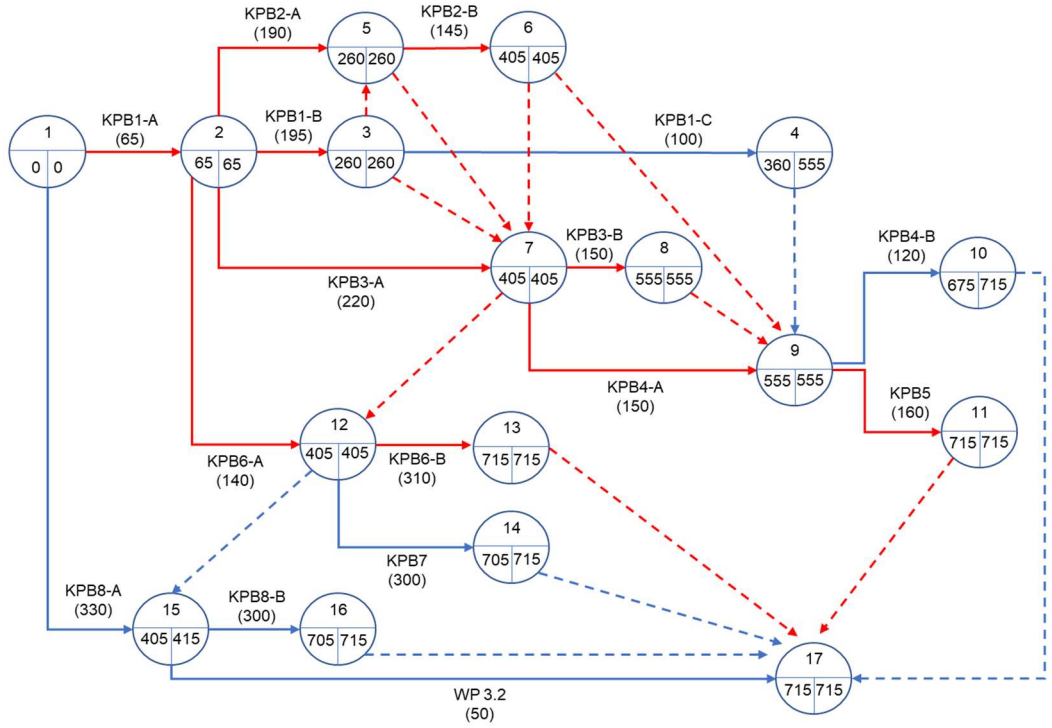


Mutual links are defined between the individual work packages, as shown in the following diagram. The feedback that is carried out within the activities included in the individual work packages is shown in the diagram by a dashed line.



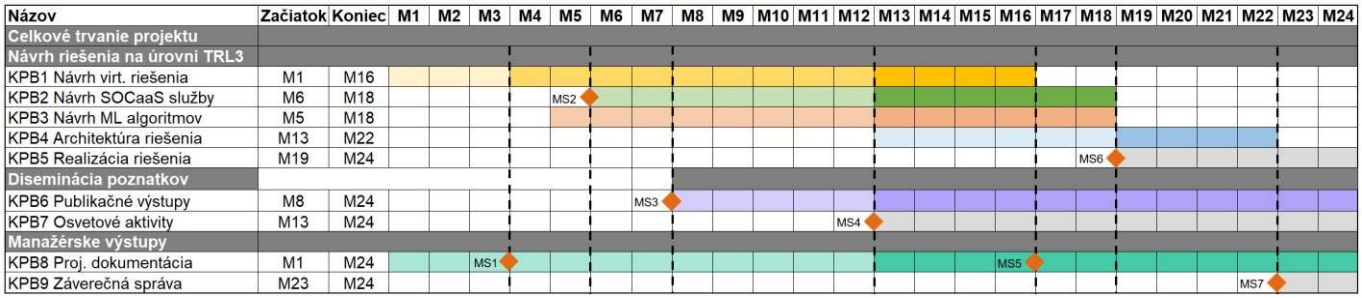
From the point of view of the combination of logical and temporal dependencies, the project is represented by the following Pert diagram. It is constructed as a result of a deeper decomposition of the work packages owing to the need to carry out certain activities in the project in parallel. The constructed Pert diagram already reflects the duration of the activities contained in whole or further decomposed work packages after performing the necessary calculations taking into account optimistic and pessimistic estimates of activity duration. The captured duration of activities is in days. We identified the predecessors for the individual work packages (and their deeper decomposition where necessary) and, together with the duration estimate, captured this information in the following table. In our arc-oriented version of the diagram, the arcs represent the individual work packages and the numbered nodes contain information describing the earliest possible and latest permissible start of the individual work.

Activity name		Predecessors	Duration [days]
KPB1	KPB1-A Virt. solution design	-	65
	KPB1-B Virt. solution design	KPB1-A	195
	KPB1-C Virt. solution design	KPB1-B	100
KPB2	KPB2-A SOCaaS service design	KPB1-A	190
	KPB2-B SOCaaS service design	KPB2-A, KPB1-B	145
KPB3	KPB3-A ML algorithm design	KPB1-A	220
	KPB3-B ML algorithm design	KPB1-B, KPB2-A, KPB3-A	150
KPB4	KPB4-A Solution architecture	KPB1-B, KPB2-B, KPB3-A	150
	KPB4-B Solution architecture	KPB1-C, KPB2-B, KPB3-B, KPB4-A	120
KPB5	KPB5 Solution implementation	KPB1-C, KPB2-B, KPB3-B, KPB4-A	160
KPB6	KPB6-A Publication outputs	KPB1-A	140
	KPB6-B Publication outputs	KPB1-B, KPB2-A, KPB3-A, KPB6-A	310
KPB7	KPB7 Awareness-raising activities	KPB6-A	300
KPB8	KPB8-A Proj. documentation	-	330
	KPB8-B Proj. documentation	KPB8-A	300
KPB9	KPB9 Final report	KPB6-A, KPB8-A	50



In the project we identify several critical paths (marked in red) describing the work packages whose implementation requires close attention. The work packages located on one of the branches of the critical path relate mainly to the initial parts of the analysis and design of the technical solution. It is therefore essential that we focus on precisely meeting the milestones and time limits of these activities. The success of the implementation of the subsequent stages depends on them. Ongoing monitoring of progress in the project and time reserves in the defined technical stages make it possible to apply corrective measures if necessary. A better view of the time requirements of the entire project and the temporal sequence of the work contained in the individual WPs is provided by the following Gantt chart.

13.2 Schedule



The chart captures both how the work will be carried out within the individual WPs to create the main components of the project and the placement of the defined milestones. The individual milestones are placed in the project so that it is possible within them to summarise the progress of the implementation of the technical stages of the project, while at the same time continuing the ongoing monitoring of indicators related to project management. This is shown by the dashed lines of the placed milestones. In this way, when parallel parts of the project need to be implemented, it will always be possible to summarise the necessary inputs from the previous stage, as well as to continuously evaluate actual progress in the project against the plan, which supports the achievement of the set objectives and the delivery of the required management outputs in the form of the interim and final project reports. The individual work packages are further described, with all the necessary particulars, in the form of tables.

13.3 Project Risks

Description of the implementation risk	Work package code	Proposed measures to minimise or eliminate the risk
R1 Insufficient objectivity in the preparation of designs	KPB1KPB2, KPB3 and KPB4	Diverse team composition; use of a combination of methods
R2 Failure to involve essential entities in decision-making on individual parts of the project	KPB5, KPB6 and KPB7	Project communication strategy; RACI matrix; contracts
R3 Failure to include an essential part of the solution in testing	KPB5	List of key parts of the solution
R4Overlooking significant trends when updating the knowledge base	KPB1, KPB2 and KPB3	Continuously updated project knowledge base

R5 Non-acceptance of the prepared publication outputs by relevant journals	KPB6	Use of best practice and experience from previous publication outputs
R6 Overlooking available technological solutions on the market	KPB3	Continuously updated project knowledge base

13.4 Dependencies

Within the project there are no dependencies on other projects or ongoing projects.

13.5 Requirements for the Testing and Deployment to Production Phases

All details regarding testing after deployment of the PoC will be provided in output V17.

14 Open Issues

Within the project and its current status, no open issues are recorded.

15 Annexes

List of DNR annexes:

1. Requirements catalogue and mapping to their implementation in the proposed solution