Detailed design of the solution

pursuant to MIRRI Decree No. 401/2023 Coll.

Obliged entity	
Project name	
Person responsible for the project	
Project implementer	
Project owner	
Supplier	
Supplier's PM	

Document approval

Item	Name and surname	Organization	Job position	Date	Signature (or electronic approval)
Prepared by					

Document history

Version	Date	Changes	Name
R2v0.3	*30.05.2025*		

Contents

1 Abbreviations and terms used 5

2 Executive summary 6

2.1 Solution intent 6
2.2 Solution objectives 6
2.3 Measurable and performance indicators (KPI) 6
2.4 Solution scope 6
2.5 Outside the project scope 7
2.6 Brief description of the proposed solution 7
2.7 Summary of output M-06 of the proposed solution 7
2.7.1 Overview of end services – future state 7
2.7.2 Scope of information systems – future state 7
2.7.3 Overview of planned integrations of ISVS with supra-ministerial ISVS – common modules pursuant to Act No. 305/2013 on e-Government – future state 7
2.7.4 Overview of planned use of other ISVS (integrations) – future state 7
2.7.5 Application services for the delivery of end services – future state 8
2.7.6 Application services for integration – future state 8

3 Requirements analysis 8

3.1 Requirements analysis procedure 8
3.2 User research 9
3.3 Motivation architecture 9
3.3.1 9
3.4 Motivation Viewpoints 9
3.5 Catalogue of requirements and its mapping to the design of their solution 10
3.6 Legislation, guidelines and methodological materials taken into account in the solution 11
3.7 Acceptance criteria 11

4 Business architecture 11

4.1 Business architecture 11
4.2 BPMN - Business process model 11
4.3 Use cases 11
4.3.1 Actors 11
4.3.2 General use cases 14
4.3.2.1 Use case diagrams ..... **Chyba! Záložka nie je definovaná.**
4.3.2.2 UC_00 - Initialization and configuration of SOCaaS ..... **Chyba! Záložka nie je definovaná.**
4.3.2.3 UC_01 - Monitor security events ..... **Chyba! Záložka nie je definovaná.**
4.3.2.4 UC_02 - Visualize and report security data ..... **Chyba! Záložka nie je definovaná.**
4.3.2.5 UC_03 - Manage detection and correlation rules 25
4.3.2.6 UC_04 – Notify and escalate a security incident 26
4.3.2.7 UC_05 – Resolve and close a security incident 28
4.3.2.8 UC_06 – Evaluate incidents and improve detection and awareness 31
4.3.2.9 UC_07 – Automate incident response 33
4.3.2.10 UC_08 – Proactive threat hunting 34
4.3.2.11 UC_09 – Manage Threat Intelligence and indicators of compromise 36
4.3.2.12 UC_10 – Manage compliance, metrics and management reporting 37
4.3.2.13 UC_11 – Manage assets and the customer environment 40
4.3.2.14 UC_12 – Manage vulnerabilities and security findings 42
4.3.2.15 UC_13 – Design and maintain the security architecture of the customer's SOCaaS environment 44
4.3.2.16 UC_14 – Customer portal and collaboration 45

4.3.2.17 UC_15 – Manage SOCaaS platform operations 49
4.3.2.18 UC_16 – Customize reports, exports and documents to the organization's needs 50
4.3.2.19 UC_17 – Manage users and roles in the SOCaaS platform 52
4.4 Business architecture 54
4.4.1 Identification of end-service design owners (UX) 54
4.4.2 Specification of end-customer needs 54
4.4.3 User interface prototype design 54
4.4.4 Screen visual design 54
5 Data architecture 55
6 Application architecture 55
6.1 Application architecture 55
6.2 Aplication Viewpoints 55
6.3 Solution design of application services 58
6.3.1 Existing application interfaces 58
6.3.2 Discontinued application interfaces 58
6.3.3 New application interfaces 58
6.4 Integrations between systems 58
6.4.1 List of existing integrations without change 58
6.4.2 List and description of proposed integrations 58
6.4.3 List and description of required integration interfaces 58
6.4.4 Design of the application interfaces provided by the system (module) 58
7 Technological architecture 58
7.1 Technological architecture 59
7.2 Technology Viewpoint INNOSOC Common 59
7.3 Technology Viewpoint Asset Inventory 59
7.4 Technology Viewpoint Kubernetes 61
7.5 Technology Viewpoint OpenStack 74
7.6 Technology Viewpoint OpenStack External 76
7.7 Technology Viewpoint Reporting 76
7.8 Technology Viewpoint Vulnerability Scanner 79
7.9 Technology Viewpoint Ticketing System / Incident Recording and Management 79
7.10 Technology Viewpoint Knowledge Portal for the SOC Team 82
7.11 Technological components of the solution and their relationship to application components 84
7.11.1 Management of access, accounts and keys of operating environments 84
7.11.2 Containerization and orchestration of cloud services 84
7.11.3 Specification of the use of government cloud services 84
7.12 Description of the test environment 84
7.13 Description of the production environment 84
8 Service level (SLA) and performance requirements 84
9 Software licences and source code 84
10 Ensuring availability, backup and recovery of the solution 85
10.1 Ensuring availability 85
10.2 Backup 86
10.3 Recovery of the solution 86
11 Security - addressing security requirements 86
12 Data migration 89
12.1 Data migration procedure 89
12.2 Data migration verification 89

13 Implementation and deployment schedule, dependencies 89
13.1 Project phases and timeline - consolidated overview 89
13.2 Dependencies and solution design 89
13.3 Requirements for the Testing and Production Deployment phases 89
14 Open issues 89
15 Annexes 89

1 Abbreviations and terms used

ABBREVIATION	DESCRIPTION
API	Application Programming Interface
API GW	Process integration and data integration module - access part
CEP	ÚPVS central electronic registry module
CI/CD	Continuous integration/Continuous deployment – a series of steps for correct software deployment - Continuous integration and delivery
CUET	Central Official Electronic Notice Board
e-Government	Electronical Government, digitalization of public administration
eDesk	ÚPVS electronic communication mailbox module
eForm	ÚPVS electronic forms module
G2G	Process integration and data integration module - communication part
GDPR	General Data Protection Regulation
GovNET	Secure government network administered by the National Agency for Network and Electronic Systems. / Cross-sectoral public administration information network
GUI	Graphical user interface
IAM	Identity Access Management - User identity management. / Authentication module within ÚPVS
IDSK	A set of rules for creating the content and functionality of electronic services in line with users' needs.
KAV	Consolidated analytical layer
MIRRI	Ministry of Investments, Regional Development and Informatization
MoSCoW	M - Must have, S - Should have, C - Could have, W - Won't have. A technique for prioritizing requirements in project management and software development
NASES	National Agency for Network and Electronic Services
OVM	Public authority - a state body, a territorial self-government body, a public-law institution, a body of professional self-government, and a natural person or legal person entrusted by law with the exercise of public authority
PO/FO	Legal person/Natural person
SkTalk/Sk-Talk	Standard for communication via ÚPVS pursuant to Section 11(g) of Decree No. 78/2020 Coll.
SLA	Service Layer/Level Agreement - Agreement on the level/ parameters / quality of service provision
SvM	Slovensko v mobile (Slovakia in the mobile)
PROD	Production environment
UAT	Test environment
UIR	Universal integration interface – an asynchronous interface for receiving and sending messages from and to ÚPVS
USR	Universal synchronous interface for making other synchronous services of ÚPVS modules accessible – from the service provider's perspective, it is an external bus interface
ÚPVS	Central Public Administration Portal, central portal
XML	Extensible Markup Language
WSDL	Web Services Description Language
ZEP/KEP	Guaranteed electronic signature / Qualified electronic signature
ŽS	Life situation

TERM/REFERENCE	DESCRIPTION
ArchiMate	A standardized modelling language used primarily for the representation, description and subsequent analysis of enterprise architecture.
Authentication	The process of identifying and verifying the identity of a user requesting a service or resource in an information system.
Authorization	Represents verification of the authorization of persons or objects to work with data.
Documentation for e-Notify	Documentation for the ÚPVS e-Notify module for integration partners is available on the Partner Framework Portal (PFP); access to the PFP portal is granted on request.
Integration manual for e-Notify	Integration manual for e-Notify with a description of services in the current version on the PFP portal eNOTIFY - All documents (gov.sk) (gov.sk)
KAV	KAV - Consolidated analytical layer | Ministry of Investments, Regional Development and Informatization of the Slovak Republic (gov.sk)
MetaIS/METAIS	Central metainformation system of public administration
Notification	A notification is an information signal or message sent to a user in order to alert them to an event, an activity or a need for intervention.
eGOV,e-Government	Electronical Government, digitalization of public administration
OpenID Connect (OIDC), OAuth2	RFC 6749 (The OAuth 2.0 Authorization Framework), a standard for authentication and determining the scope of permissions when using the REST protocol pursuant to the Decree on Standards 78/2020, Section 11(i)
Rules for publishing electronic services	Rules for publishing electronic services in the multichannel environment of public administration https://mirri.gov.sk/wp-content/uploads/2018/10/Pravidla_Publikovania_Sluzieb_v1_0-1.pdf
Representational State Transfer (REST)	Network communication protocol used in the provision of electronic services required for processing electronic submissions or for the successful completion and preparation of an electronic submission via the process integration and data integration module pursuant to Decree on Standards 78/2020, §11(a)
Backward compatibility	In technical practice, a property of a product consisting in the product being able to replace its predecessor in such a way that it remains capable of interoperating with products that were created for this older product. The newer product is thus able to fully replace its predecessor.
ŽS	Priority life situations | Ministry of Investments, Regional Development and Informatization of the Slovak Republic (gov.sk)

2 Management summary

2.1 Solution intent

2.2 Solution objectives

2.3 Measurable and performance indicators (KPI)

2.4 Solution scope

2.5 Outside the project scope

2.6 Brief description of the proposed solution

2.7 Summary of output M-06 of the proposed solution

2.7.1 Overview of end services – future state

KS code (from MetaIS)	KS name	KS user (G2C/G2B/G2G/G2A)	Life situation (+ code from MetaIS)	KS level of electronization

Table 1 Overview of end services – future state

2.7.2 Scope of information systems – future state

ISVS code (from MetaIS)	ISVS name	ISVS module (tick if the ISVS is a module)	IS VS status	IS VS type	Code of the parent ISVS (if the checkbox for ISVS module is ticked)

Table 2 Scope of information systems – future state

2.7.3 Overview of planned integrations of the ISVS with cross-government ISVS – common modules pursuant to Act No. 305/2013 on e-Government – future state

IS code	ISVS name	Common modules pursuant to Act No. 305/2013 on e-Government

Table 3 Overview of planned integrations of the ISVS with cross-government ISVS

2.7.4 Overview of planned use of other ISVS (integrations) – future state

ISVS code *(from MetaIS)*	ISVS name	Code of the integrated ISVS *(from MetaIS)*	Name of the integrated ISVS

Table 4 Overview of planned use of other ISVS (integrations)

2.7.5 Application services for the implementation of end services – future state

AS code *(from MetaIS)*	AS name	ISVS/ISVS module *(code from MetaIS)*	Application service implements KS *(KS code from MetaIS)*

Table 5 Application services for the implementation of end services

2.7.6 Application services for integration – future state

AS (MetaIS code)	AS name	Implemented by ISVS (MetaIS code)	Providing or Consuming	Integration via CAMP	Integration with third-party IS	SaaS	Integration with the provider's AS (MetaIS code)

Table 6 Application services for integration

3 Requirements analysis

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. Primarily, the part of the Architecture Development Method concerning the development of the architecture in all four architecture domains was adopted from the framework. The Architecture Development Method is a process with recommended steps and activities. Although 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 consecutive phases.

The first phase involves the analysis of requirements that define the expectations of users and stakeholders.

During the second phase, the requirements are transformed into a system architecture design that defines how the system will work. The design takes place 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 modules of the system 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 the functionality that covers the specified functional requirements will be deployed, and the first phase of

implementation will be checked and approved. In the second step, the remaining functionality will be implemented and integrated with the existing modules.

The next phase ensures that the system is tested. It is 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 are able to 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 optimization. The maintenance process must respect all legislative requirements, for example ensuring personal data protection and cybersecurity.

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

3.2 User research

3.3 Motivation architecture

3.3.1

3.4 Motivation Viewpoints

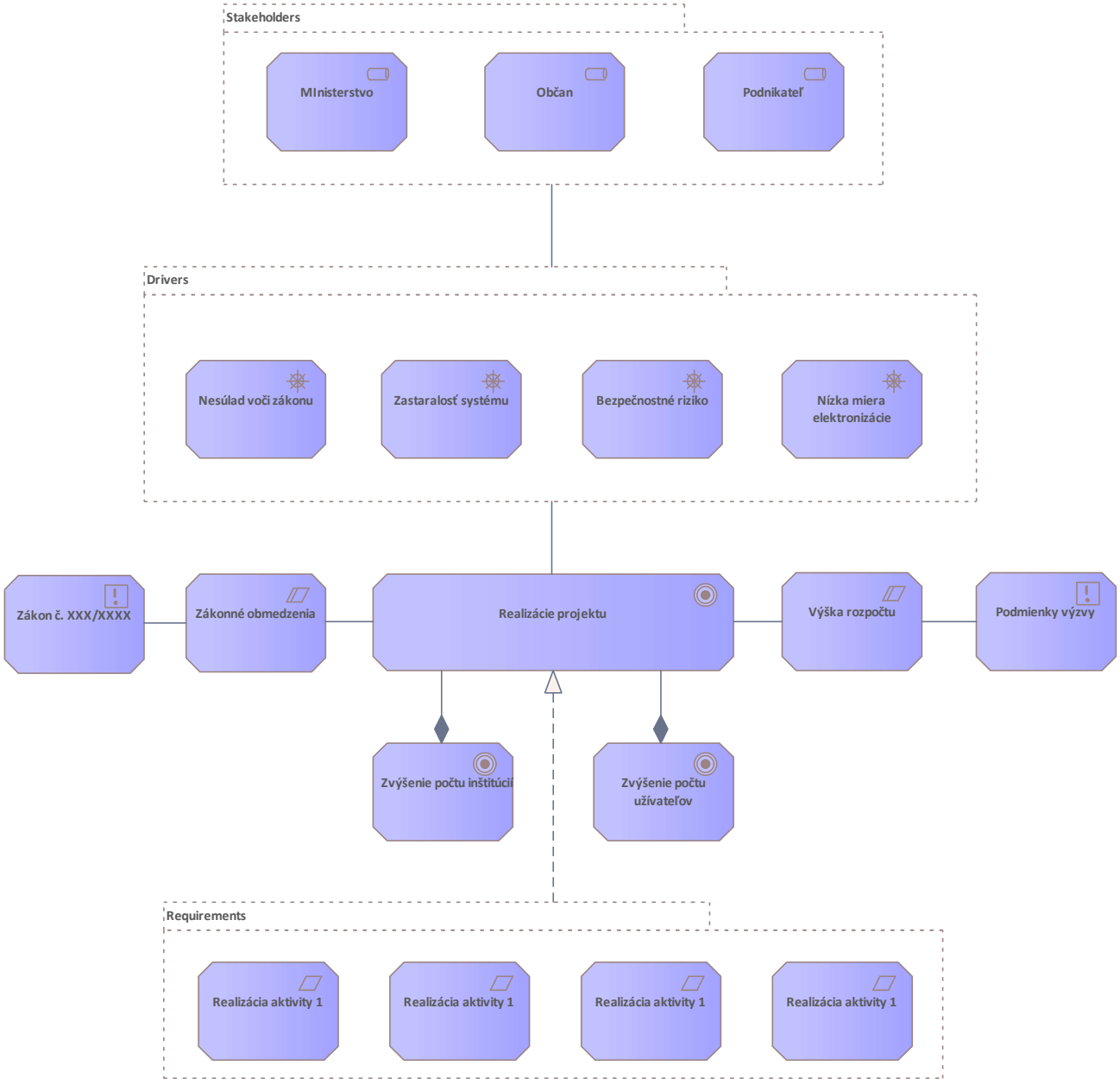


Figure 1 : Motivation Viewpoint

3.5 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. Based on the requirements catalogue, a Test Plan document has been created, which will serve as the basis for acceptance during UAT testing. The scope and plan of the tests are defined in the Test Strategy document and include:

- Functional testing (FAT) – on the supplier's side
- System and integration testing (supplier; NASES may participate)
- Load and performance testing
- Security testing
- User tests of the functional user interface (UX) and accessibility testing (may not be mandatory)
- User acceptance testing (UAT) – on the side of the contracting authority / recipient

The security tests will include penetration tests, from which a formal report will be produced.

Summary of the mapping of requirements to use cases and architecture objects:

3.6 Legislation, guidelines and methodological materials taken into account in the solution

3.7 Acceptance criteria

4 Business architecture

4.1 Business architecture

4.2 BPMN - Business process model

4.3 Use cases

The following chapter presents the individual user scenarios.

4.3.1 Actors

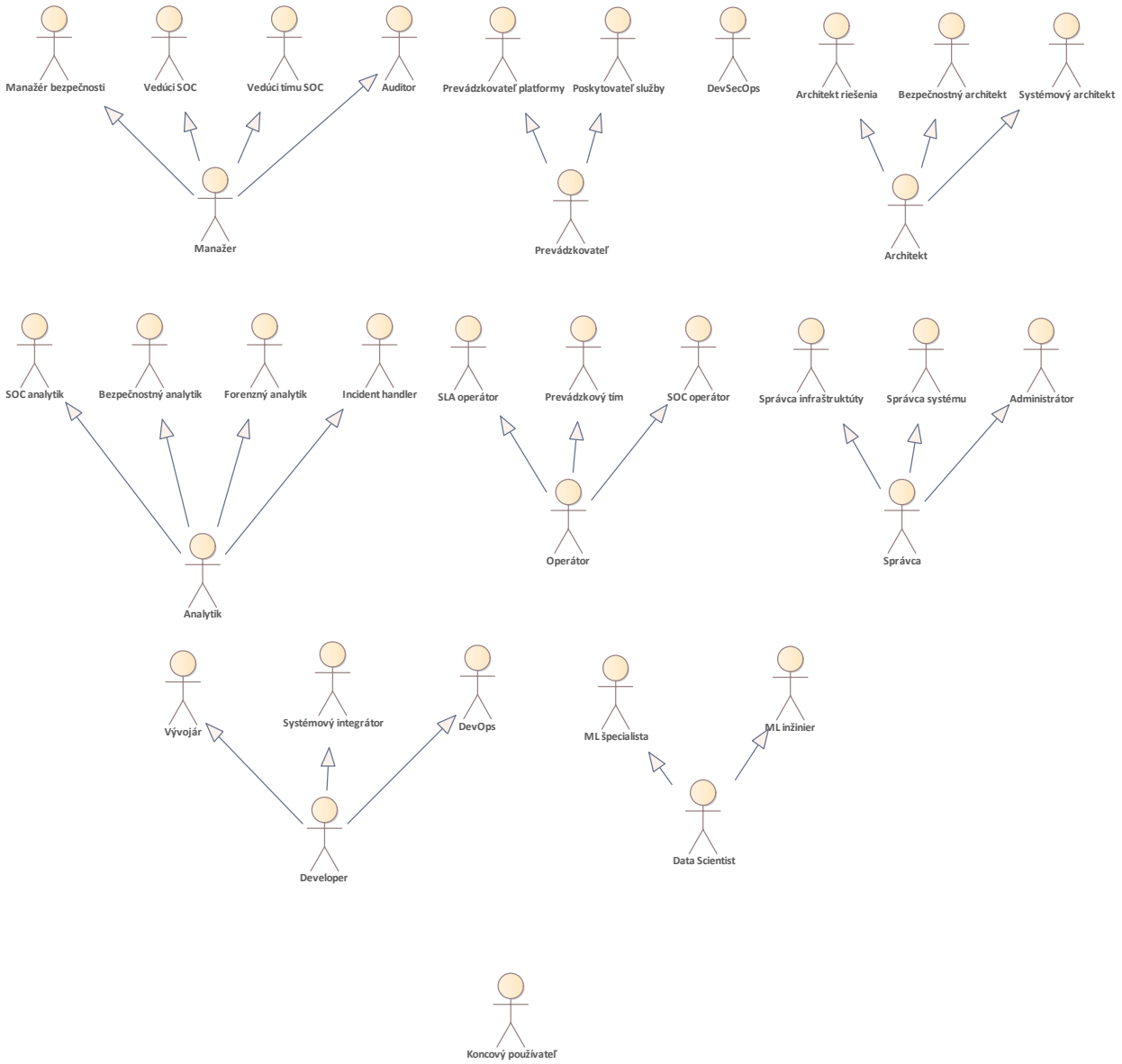


Figure 2 : Actors

Name of the actor/role	Description
Administrator	Responsible for: SIEM, log processing, data visualization, alerting, EDR/XDR, asset management, vulnerability scanner and security data sources.
Analyst	Responsible for areas such as: Log processing, SOAR, ML-based detection, Data visualization, 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 visualization, Alerting, EDR/XDR, CTI, Asset management, Vulnerability

Name of the actor/role	Description
	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 visualization, EDR/XDR, Secure 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.
End user	Responsible for areas such as: Data visualization.
Manager	Responsible for areas such as: SIEM, ML-based detection, data visualization, 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 visualization, 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.
Service operator	Responsible for areas such as: data visualization.
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.
Administrator	The administrator is a basic technical role responsible for the operation, maintenance and functioning of IT infrastructure and systems. It covers competencies 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 visualization, alerting, EDR/XDR, Asset management, Vulnerability scanner, Security data sources, Knowledge base, Ticketing, Incident recording and management, Reporting and System administrationdata visualization.

Name of the actor/role	Description
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, Data visualization, 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 visualization.
Developer	Responsible for areas such as: SIEM, SOAR, ML-based detection and Security data sources.

4.3.2 General use cases

4.3.2.1.1 UC_00 - Initialization and configuration of SOCaaS

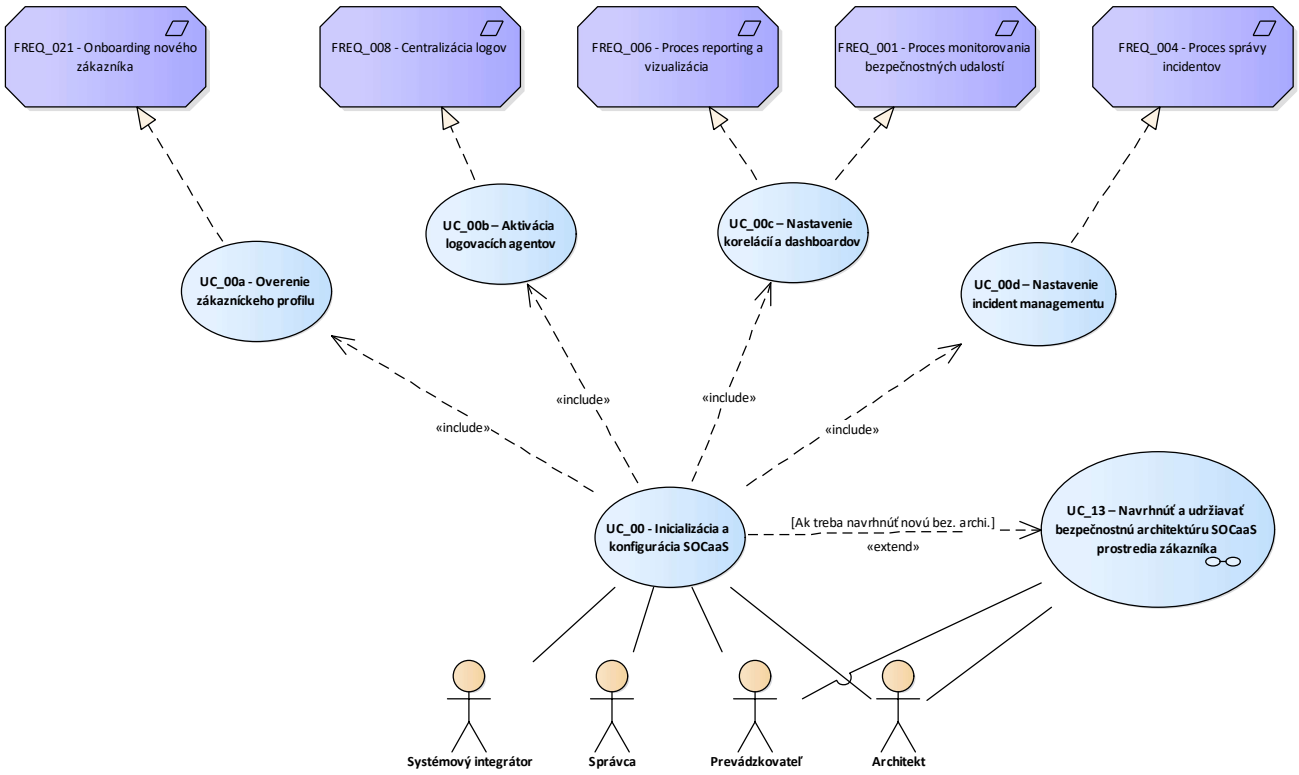


Figure 3 : UC_00 - Initialization and configuration of SOCaaS

Use case	UC_00 - Initialization and configuration of SOCaaS
Objective	The use case describes the initial initialization 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 SOC architect has configuration permissions.
Precondition	Basic network connectivity is available.
Postcondition	SOCaaS is successfully initialized and ready for monitoring and detection.
Postcondition	The basic modules are activated: logging, SIEM correlations, dashboards, incident management.

Basic Path	Basic Path
Step	**Activity**
1	The SOC architect starts the initialization 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 initialization tasks (registration of sources, activation of agents, creation of dashboards).
5	The architect confirms the configuration.
6	The system saves the configuration and reports successful completion.

ALT01 – Incomplete customer profile	Alternate
Step	**Activity**
1	The system detects missing data.
2	It notifies the architect to supply the missing data.
3	The process continues once the information has been supplied.

ALT02 – 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.

Use case	UC_00a - Verification of the customer profile
Objective	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 starts the profile verification.
2	The system loads the customer data.
3	The system checks the mandatory fields. The system checks the mandatory fields.
4	The verification is successful.

ALT01 – Incomplete data	Alternate

Step	Activity
1	the system requests completion.

ALT02 – The tenant lacks technical configuration	Alternate
Step	**Activity**
1	the system stops the process and notifies the administrator.

Use case	UC_00b – Activation of logging agents
Objective	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 starts the activation.
2	The system contacts the logging agents.
3	The availability of the API and certificates is verified.
4	The agents are activated.

ALT01 – The agent does not respond	Alternate
Step	**Activity**
1	the system reports an error and continues later.

ALT02 – Incorrect API keys	Alternate
Step	**Activity**
1	re-uploading of the keys is required.

Use case	UC_00c – Setting up correlations and dashboards
Objective	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	The correlation rules and dashboards are configured.

Basic Path	Basic Path
Step	**Activity**
1	The platform operator starts the set-up.
2	The system loads the correlation template.

3	The system creates dashboards for monitoring.
4	The configuration is saved.

ALT01 – The correlation rule is invalid	Alternate
Step	**Activity**
1	the system skips it and reports an error.

ALT02 – The dashboard cannot be created	Alternate
Step	**Activity**
1	the system creates only a minimal version.

Use case	UC_00d – Setting up incident management
Objective	The system activates the workflow for incident management and role assignment.
Precondition	The Ticketing or SOAR module is initialized.
Postcondition	The incident lifecycle (triage → analysis → closure) is ready.
Basic Path	Basic Path
Step	**Activity**
1	The platform operator starts the configuration.
2	The system loads the workflow templates.
3	The system creates incident categories and priorities.
4	It assigns roles (Analyst, Operator, Manager).

ALT01 – No user roles exist	Alternate
Step	**Activity**
1	the system stops the workflow and asks the administrator to add roles.

ALT02 – The ticketing module is unavailable	Alternate
Step	**Activity**
1	the system activates only the basic SOAR fallback.

4.3.2.1.2 UC_01 - Monitor security events

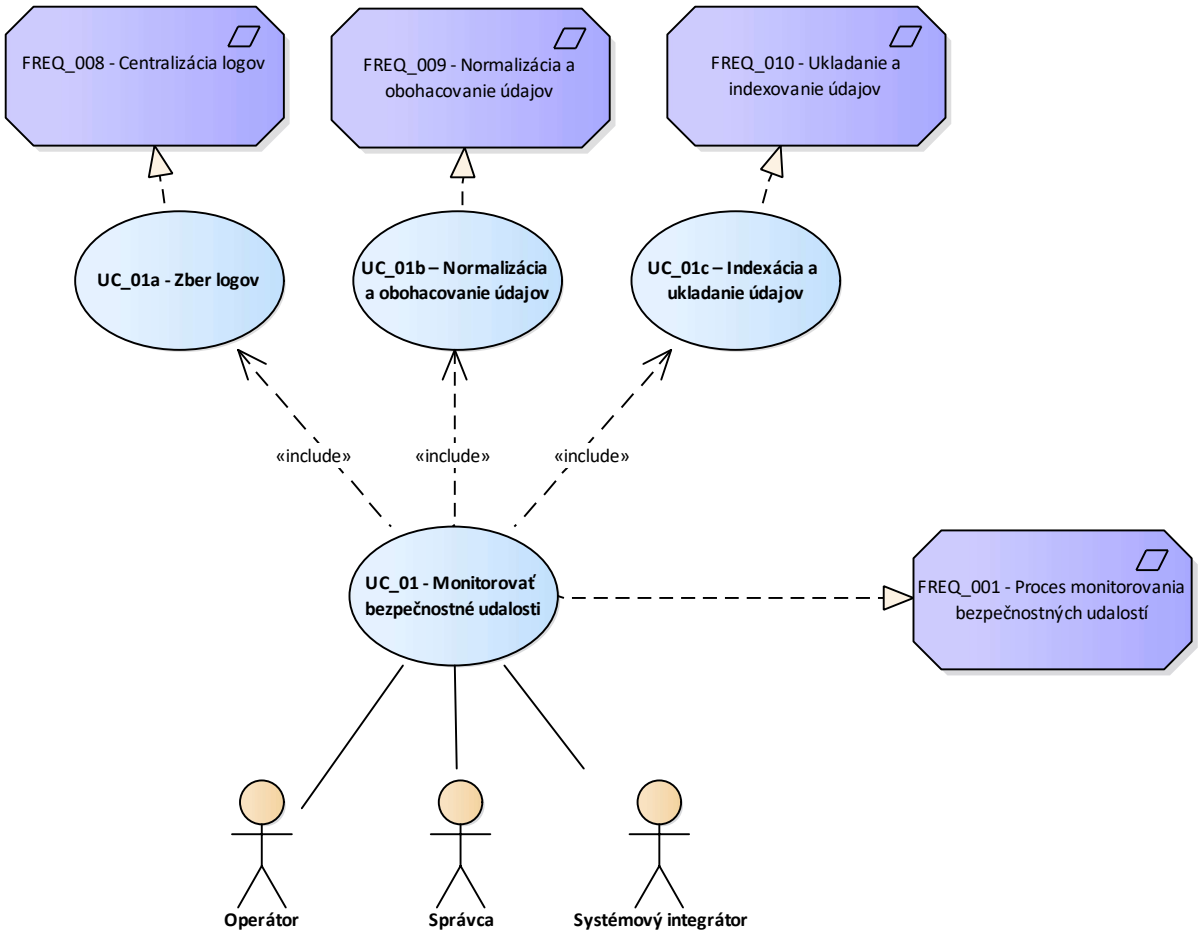


Figure 4 : UC_01 Monitor security events

Use case	UC_01 - Monitor security events
Objective	The use case describes the continuous monitoring of security events within SOCaaS. It includes the collection of events, their normalization, 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 (initialization) has been successfully completed.
Postcondition	The events are processed and passed on to the incident detection process.
Postcondition	The alerts are displayed to the operator.
Postcondition	Monitoring runs in real time.

Basic Path	Basic Path

Step	**Activity**
1	The operator monitors the control dashboard in SOCaaS.
2	UC_01a: The system receives events from logging sources.
3	UC_01b: The system normalizes and enriches the data.
4	UC_01c: The system indexes the stored data for fast searching.

5	SOCaaS generates alerts based on 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 → it activates UC_01x Incident escalation.
9	The Use Case ends – monitoring continues.

ALT01 – 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.

ALT02 – Overloaded SIEM or pipeline	Exception
Step	**Activity**
1	SOCaaS cannot keep up with processing the stream of events.
2	The system enables protective mechanisms (rate-limiting, buffering).
3	The operator is notified of the risk of event loss.

Use case	UC_01a - Log collection
Objective	It ensures the technical reception of logging 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 – normalization.
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 on to the normalization process.

ALT01 – Logs are not 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.

ALT02 – 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.

Use case	UC_01b – Data normalization and enrichment
Objective	The use case ensures the normalization of logs into a uniform format and their enrichment with information from CTI, the asset database or user data.
Precondition	A configuration of normalization rules exists.
Precondition	UC_01a (log collection) was successful.
Postcondition	The log is normalized and enriched.
Postcondition	The data are ready for storage and indexing.
Basic Path	Basic Path
Step	**Activity**
1	SOCaaS loads the received logs.
2	A normalization profile is applied according to the source type.
3	Data enrichment (IP reputation, hostname, user).
4	The resulting event is ready for indexing.

ALT01 – Unknown log type	Alternate
Step	**Activity**
1	the system uses a fallback normalization profile.

ALT02 – CTI feed missing	Alternate
Step	**Activity**
1	the system performs normalization without enrichment and marks the event as "unenriched".

Use case	UC_01c – Data indexing and storage
Objective	The use case handles the storage of normalized security events in storage and their indexing for fast searching and correlation mechanisms.
Precondition	Data normalization (UC_01b) has been completed.

Precondition	The storage and the cluster are available.
Postcondition	The events are stored in the SIEM database.
Postcondition	The indexes are ready for query systems.

Basic Path	Basic Path

Step	**Activity**
1	The system takes over the normalized data.
2	It writes them to storage (DB or distrib. index).
3	It creates/updates the indexes.
4	The event is available for correlation rules.

ALT01 – Storage unavailable	Alternate

Step	**Activity**
1	The system stores the data in a buffer and creates an alert.

ALT02 – Storage space exhausted	Alternate

Step	**Activity**
1	The system activates a clean-up or alerts operations.

4.3.2.1.3 UC_02 - Visualize and report security data

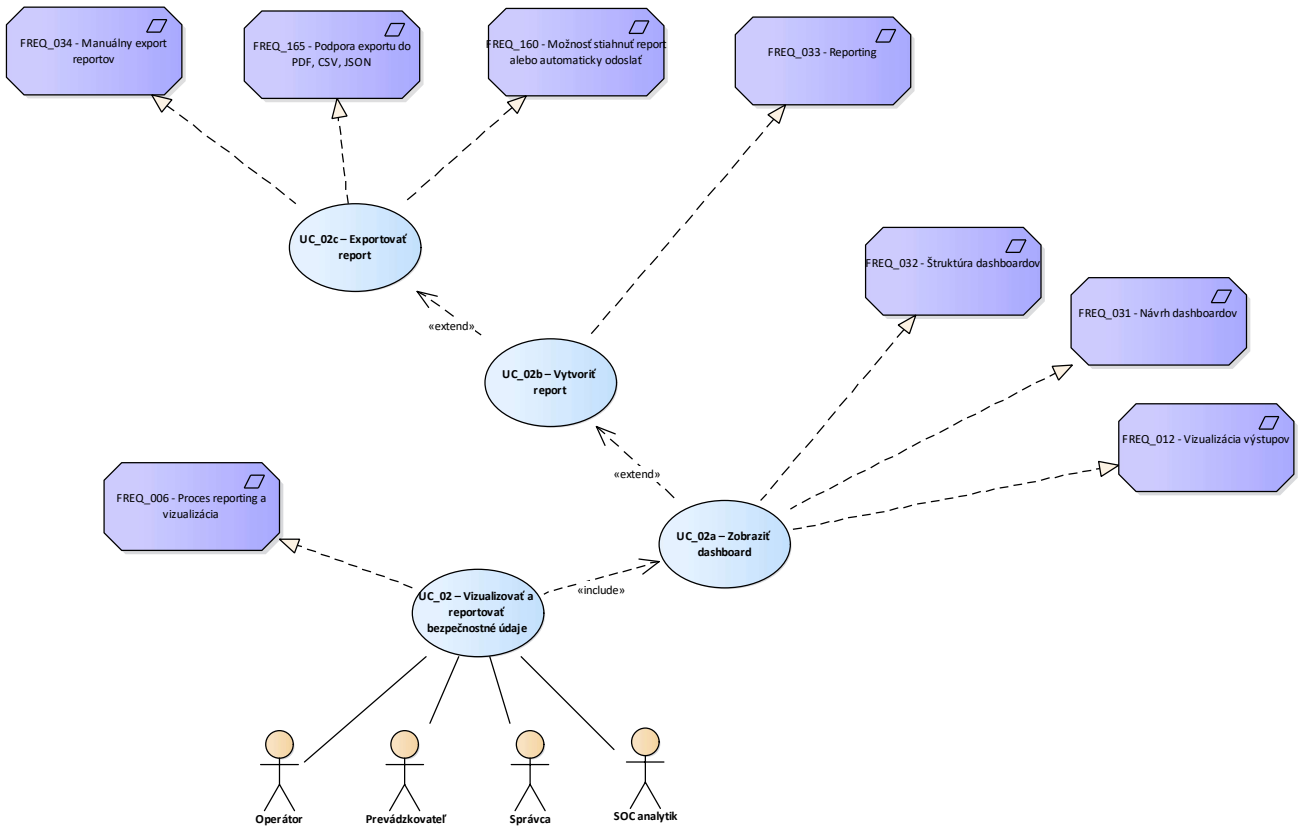


Figure 5 : UC_02 - Visualize and report security data

Use case	UC_02 – Visualize and report security data
Objective	The system enables SOC operators and the administrator to display security events using dashboards, filters, charts and tabular outputs. It also enables the generation of reports and the export of data for management, audit or customers.
Precondition	The dashboards are defined in the configuration (UC_00c).
Precondition	UC_01c – The data are indexed and available for visualization.
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 visualizations.

Basic Path	Basic Path

Step	**Activity**
1	The actor opens the “Visualization 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, 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).

ALT01 – Data unavailable	Alternate

Step	**Activity**
1	The dashboard returns an empty dataset.
2	The system displays a warning.
3	The actor may change the filtering criteria.

ALT02 – Incorrect permissions	Alternate

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

ALT03 – Report generation failed	Alternate

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

3	An actor (e.g. the Service operator) opens a dashboard outside their tenant.

Use case	UC_02a – Display a dashboard
Objective	The user opens a dashboard that visualizes security events, metrics and system states in real time.
Precondition	The system is initialized and configured (UC_00).
Precondition	All log sources are active and data are 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 authorized).

Basic Path	Basic Path

Step	**Activity**
1	The actor opens the Dashboards module.
2	The system loads the available dashboards according to the role.
3	The actor selects a dashboard to display.
4	The system loads and renders the visualizations (charts, heatmaps, tables).
5	The system updates the visualizations in real time or at defined intervals.

ALT_01 – Data does not load	Alternate

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

ALT_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 ask the administrator for access.

Use case	UC_02b – Create a report
Objective	The user creates a one-off report or saves a reporting template based on data from dashboards or events.
Precondition	The dashboards are available and the data are 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 a new report”.	
3	Selects: report type, time range, template or custom design.	
4	The system loads the necessary data.	
5	The user confirms the generation.	
6	The system generates the report and saves it in the report database.	

ALT_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 that it be supplied.	

ALT_02 – Data unavailable for the report		Alternate
Step	**Activity**	
1	The system displays the message “Data for the selected period are not available”.	
2	The actor selects a different period.	

Use case	UC_02c – Export a report
Objective	The user exports an existing report to PDF, CSV, JSON or PNG and can send it by email or via a webhook.
Precondition	There is a generated report (UC_02b).
Precondition	The user has the “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_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.

ALT_02 – Insufficient permissions	Alternate
Step	**Activity**
1	The system displays “Access Denied”.
2	It offers to request access.

4.3.2.2 UC_03 - Manage detection and correlation rules

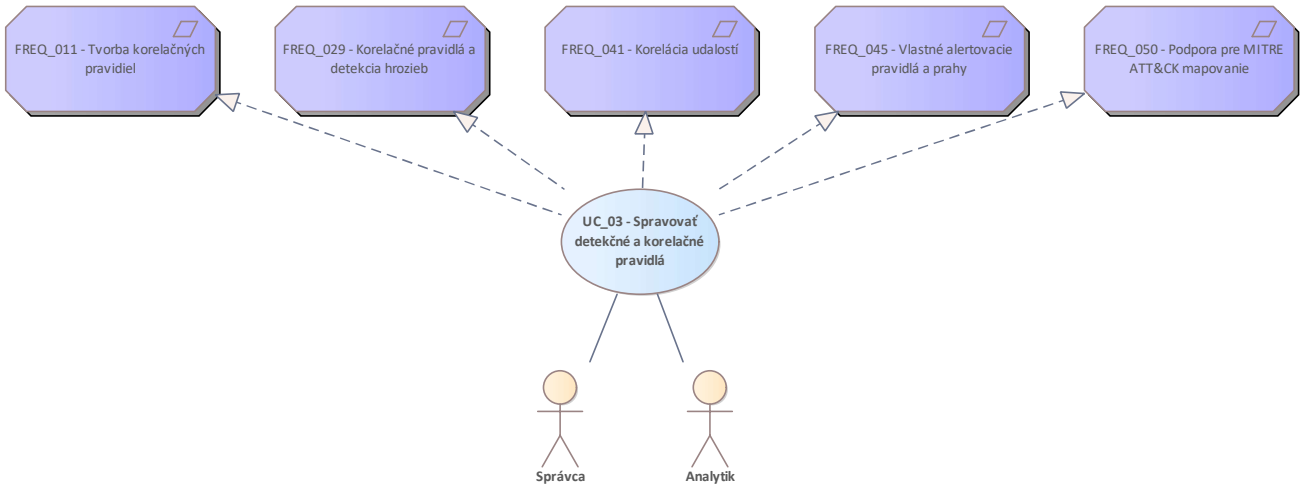


Figure 6 : UC_03 - Manage detection and correlation rules

Use case	UC_03 - Manage detection and correlation rules
Objective	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 based on 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 is created or an existing rule is modified, deactivated or deleted.
Postcondition	The rule is versioned, validated and ready for deployment to the production environment.
Postcondition	The rule is available for the evaluation of 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 severity, threat type and MITRE ATT&CK category to the rule.
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_1 – The rule is tested on historical data (debugging)	Alternate
Step	**Activity**
1	The analyst uses a tool for testing rules against historical logs.
2	The results are displayed; the rule is then modified or approved.

ALT_2 – 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 generates the underlying code of the rule.

4.3.2.3 UC_04 – Notify and escalate a security incident

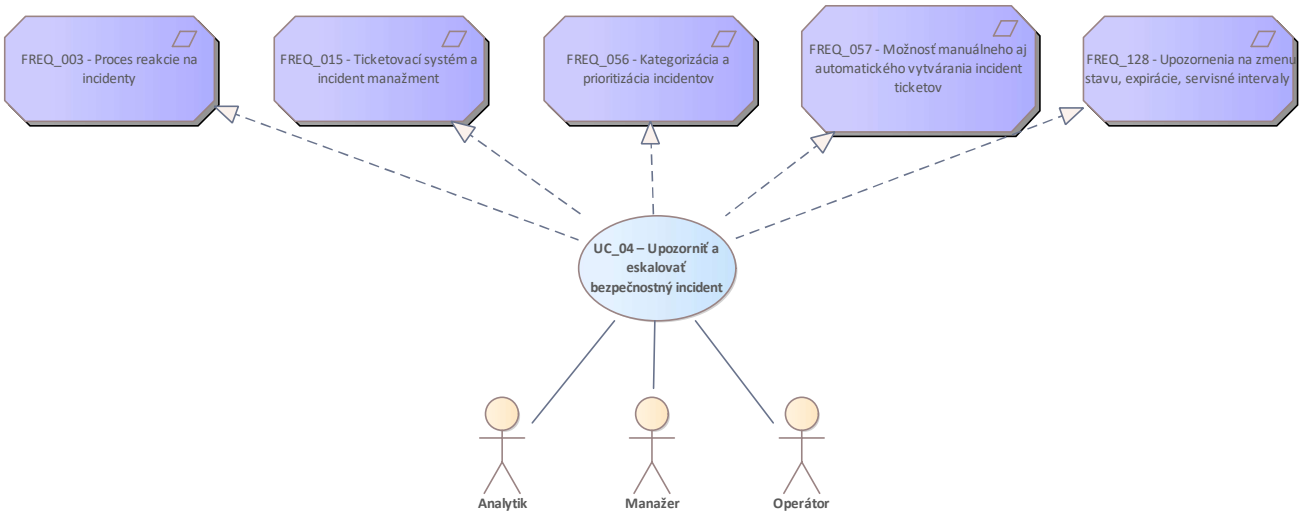


Figure 7 : UC_04 – Notify and escalate a security incident

Use case	UC_04 – Notify and escalate a security incident
Objective	After the system detects and creates a security incident (UC_01, UC_03), the objective 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 increasing 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
Precondition	The contact channels and the on-call schedule 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 unsuccessful attempt and the subsequent fallback are recorded in the incident history.
4	If all channels fail, the incident is flagged “notification unsuccessful” and the system automatically creates an internal ticket for the infrastructure administrator.

ALT_02 - The incident is not acknowledged within the SLA time - escalation	Alternate

Step	**Activity**
1	After the SLA interval has elapsed, the incident has not been acknowledged.
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).

4.3.2.4 UC_05 – Handle and close a security incident

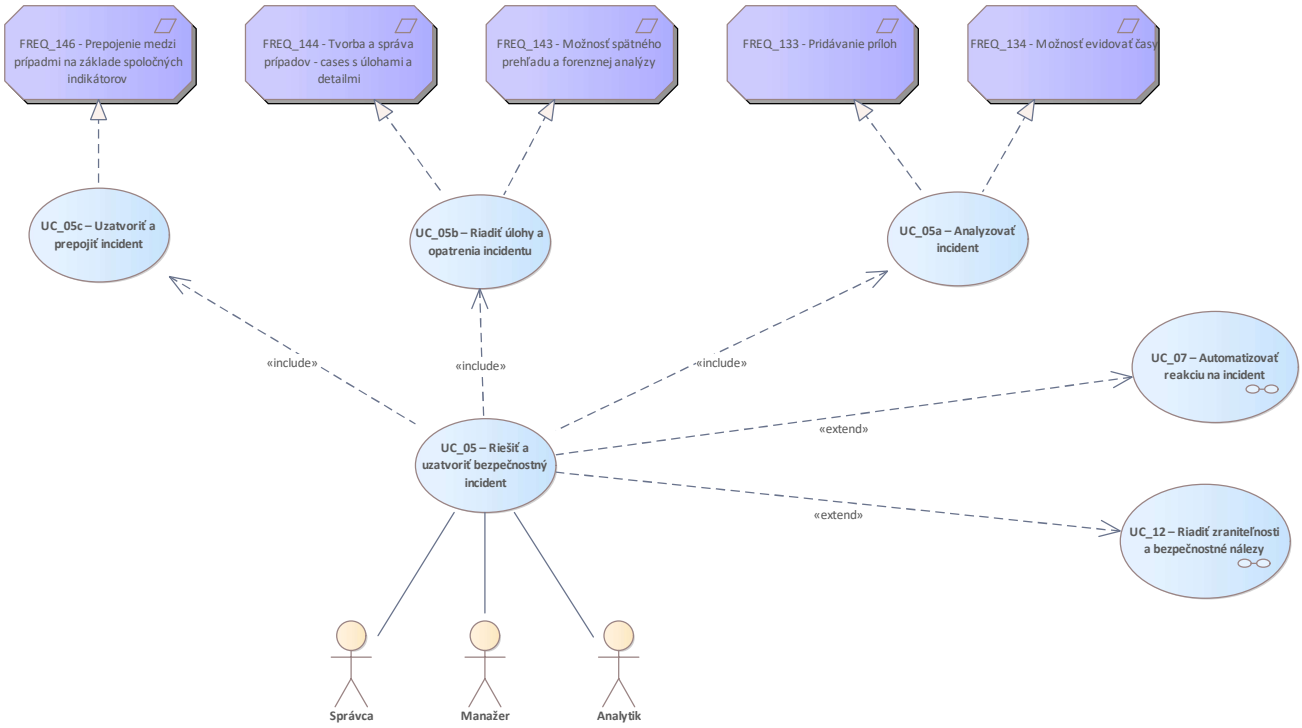


Figure 8 : UC_05 – Resolve and close a security incident

Use case	UC_05 – Handle and close a security incident
Objective	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 to the 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 supporting 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 Administrator
6	Tracking of times and the course of handling
7	Review and approval of the outcome
8	Closure of the incident
9	Preparation of supporting materials for lessons learned

ALT_ 01 – The incident is evaluated as a false positive	Alternate
Step	**Activity**
1	After the 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 Administrator 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.

Use case	UC_05a – Analyse an incident
Objective	The Analyst examines the incident in detail – reviews the timeline of events, logs and IOCs, adds evidence and builds 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 the 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	Based on 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 the logs are incomplete or missing; in the incident, they create a task for the Administrator to remedy this (supplementing logging, export from the source system) and continue

	working with what is available.

ALT_02 – The incident turns out to be a false positive	Alternate
Step	**Activity**
1	After the analysis, the Analyst concludes that it is not a real incident; they mark the incident as a “false positive”, briefly document the reason and UC_05 ends.

Use case	UC_05b – Manage incident tasks and measures
Objective	The Analyst and the Administrator 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 created list of tasks 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	Based on the analysis, the Analyst proposes specific corrective measures.
2	In the incident, creates tasks and assigns them to specific Administrators or teams.
3	Sets the priority and deadline for each task.
4	The Administrators continuously update the status of tasks according to the implementation of the measures.
5	The system records the times of task creation, start and completion.
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 Administrator finds that the proposed measure may cause an outage; they mark 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 Administrator updates the deadline and the reason in the task; the Analyst adjusts the incident priority or informs the Manager as needed.

Use case	UC_05c – Close and link an incident
Objective	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 with other related cases.
Precondition	The key tasks of the incident have been completed or a decision has been made 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 states (closed, false positive, risk accepted, part of a campaign).
Postcondition	The incident is linked with 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 that 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, asks the Manager to approve the closure.
7	The incident is moved to the archive and is ready for the “lessons learned” process.

ALT_01 – New related events detected during closure	Alternate

Step	**Activity**
1	During the review, the Analyst detects new suspicious events; they leave the incident open, create new tasks and return 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; they set it as subordinate to the parent case and adjust the final summary.

4.3.2.5 UC_06 – Evaluate incidents and improve detection and awareness

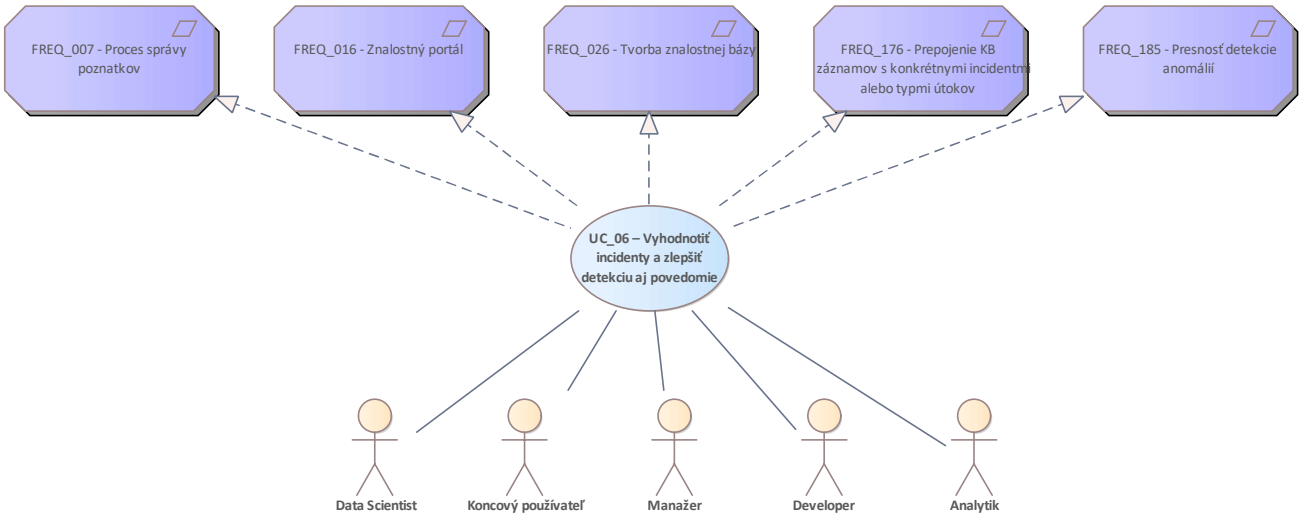


Figure 9 : UC_06 – Evaluate incidents and improve detection and awareness

Use case	UC_06 – Evaluate incidents and improve detection and awareness
Objective	After the incident has been resolved and closed (UC_05), this UC ensures that both the

	system and people learn from the incident: the incident is reviewed (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 processed “lessons learned” record
Postcondition	Adjustments to detection/ML have been proposed or carried out
Postcondition	Training / guidance for users has been prepared or updated

Basic Path	Basic Path

Step	**Activity**
1	Selection of incidents for evaluation
2	Preparation of supporting materials for evaluation
3	Post-incident meeting / review
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 proposed adjustments.
7	The knowledge base entry 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	Dissemination of knowledge
12	Conclusion of the UC

ALT_01 – Insufficient data for a quality evaluation	Alternate

Step	**Activity**
1	During the analysis it turns out that not all necessary logs or historical data are available; the team does carry out a basic lessons learned, but at the same time creates tasks to improve logging and data collection for the future.

ALT_02 – The proposed adjustments to detection/ML are rejected	Alternate

Step	**Activity**
1	The Data Scientist or Developer prepares a proposal for changes, but the Manager does not approve them on the basis of risks or costs; the recommendations are stored in the knowledge base and the changes may be reconsidered later.

ALT_03 – A need for targeted employee training has been identified		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.	

4.3.2.6 UC_07 – Automate incident response

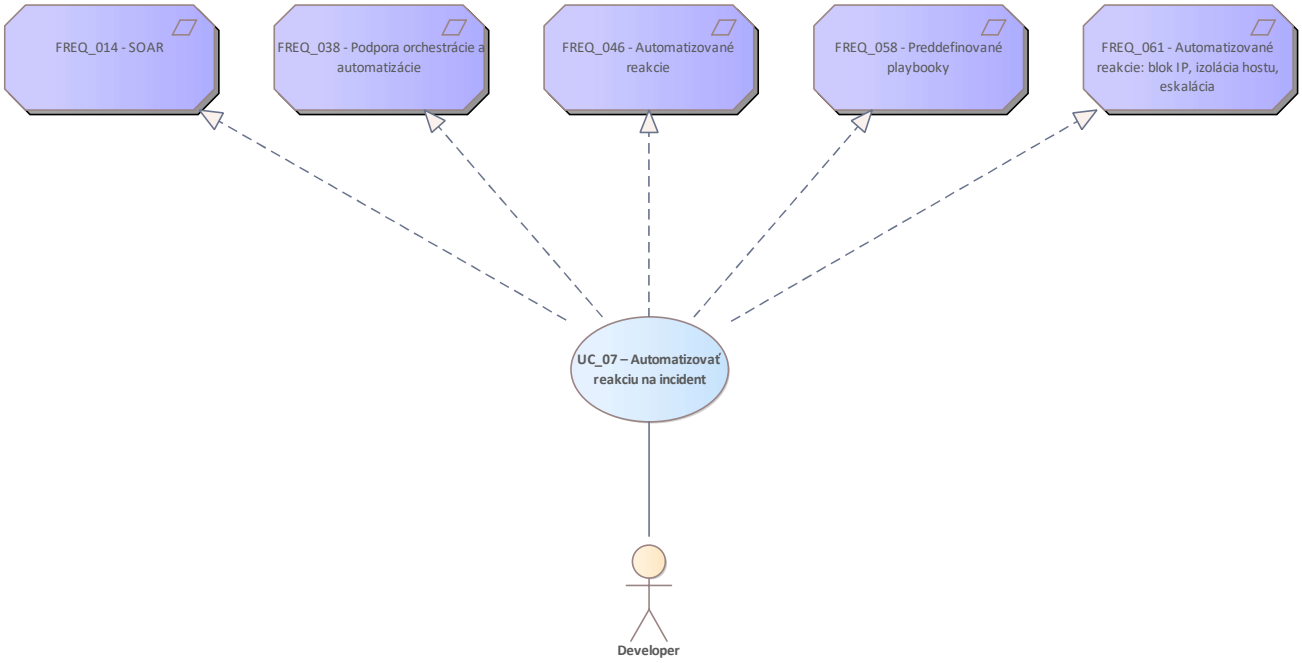


Figure 10 : UC_07 – Automate incident response

Use case	UC_07 – Automate incident response
Objective	UC_07 addresses the situation in which 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” status
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 status that reflects the outcome of the automated response
Basic Path	Basic Path
Step	**Activity**
1	The Analyst opens an incident in the “in progress” status 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 at low risk).
3	The system displays an overview of the planned playbook steps (e.g. IP block, 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 via integrations with the target systems and continuously records the status (success/failure).
6	After the playbook finishes, the system summarizes the outcome: which actions were successful, what failed and where manual intervention is needed.
7	The results of the executed actions and the playbook timeline are saved in the incident as part of the handling history.
8	Based on the outcome, the Analyst updates the incident status (e.g. containment done, 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 handling 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 executed manually or through partial automation.

4.3.2.7 UC_08 – Proactive threat hunting

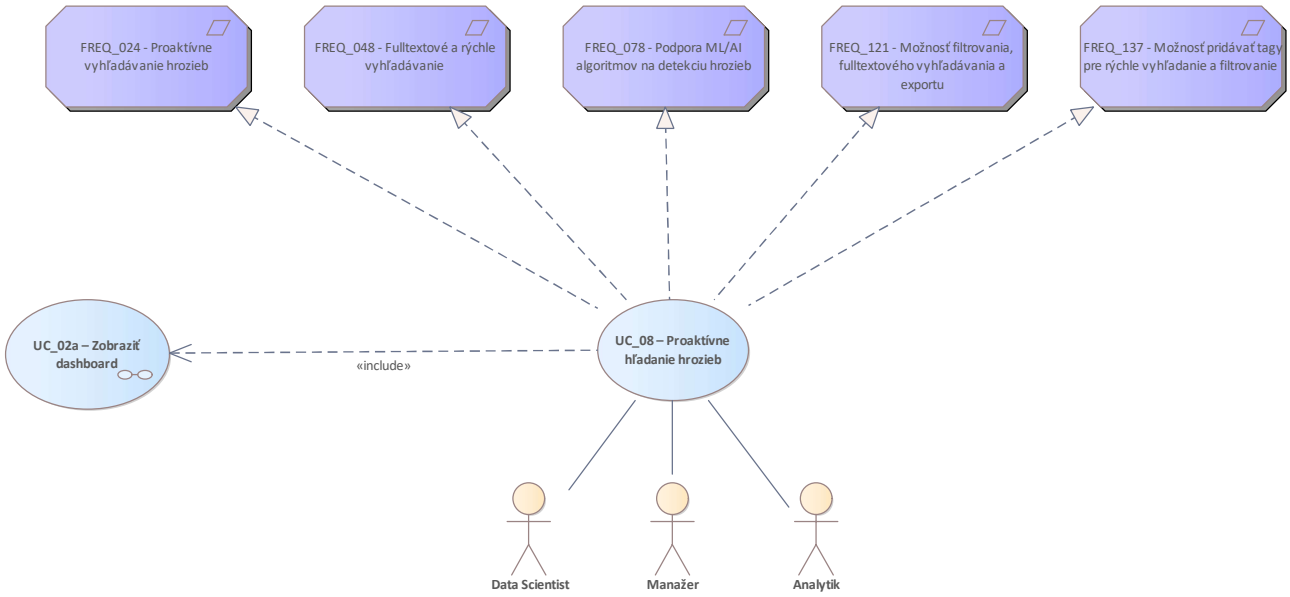


Figure 11 : UC_08 – Proactive threat hunting

Use case	UC_08 – Proactive threat hunting
Objective	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 an 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	The hunting scenarios are saved for reuse

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	Launching a hunting run over the data
5	Analysis of findings
6	Conversion of findings into incidents or enrichment of existing incidents
7	Saving of the hunting scenario
8	Documentation of results
9	Planning of repeated hunting

ALT_01 – Hunting reveals nothing significant	Alternate
Step	**Activity**
1	The hunting query does not return any suspicious findings, or all 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 an extensive 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 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 launched hunting query runs over a large volume of data and the systems evaluate it as too demanding; the system notifies the Analyst, stops or limits the query and offers the option to adjust the time range or granularity.

4.3.2.8 UC_09 – Manage Threat Intelligence and indicators of compromise

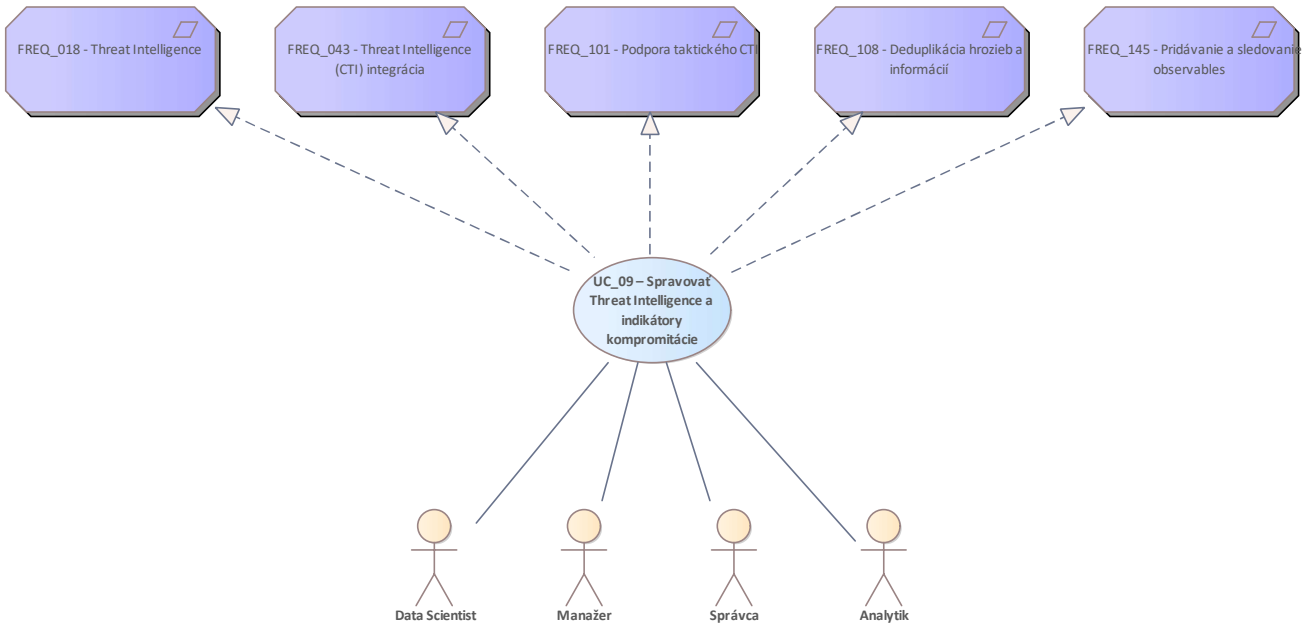


Figure 12 : UC_09 – Manage Threat Intelligence and indicators of compromise

Use case	UC_09 – Manage Threat Intelligence and indicators of compromise
Objective	UC_09 addresses the management of Threat Intelligence: the reception and integration of CTI feeds, work with indicators of compromise (IOCs), their deduplication and tracking over time. The aim is for the SOC to be able to centrally manage threats, enrich incidents and logs with TI information and keep its observables in order.
Precondition	TI sources are available
Precondition	Integrations with TI sources are configured

Precondition	The system has storage for TI and IOCs
Postcondition	TI data are loaded, normalized 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	Normalization 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 to 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 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 evaluated as outdated or unreliable	Alternate

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

4.3.2.9 UC_10 – Manage compliance, metrics and management reporting

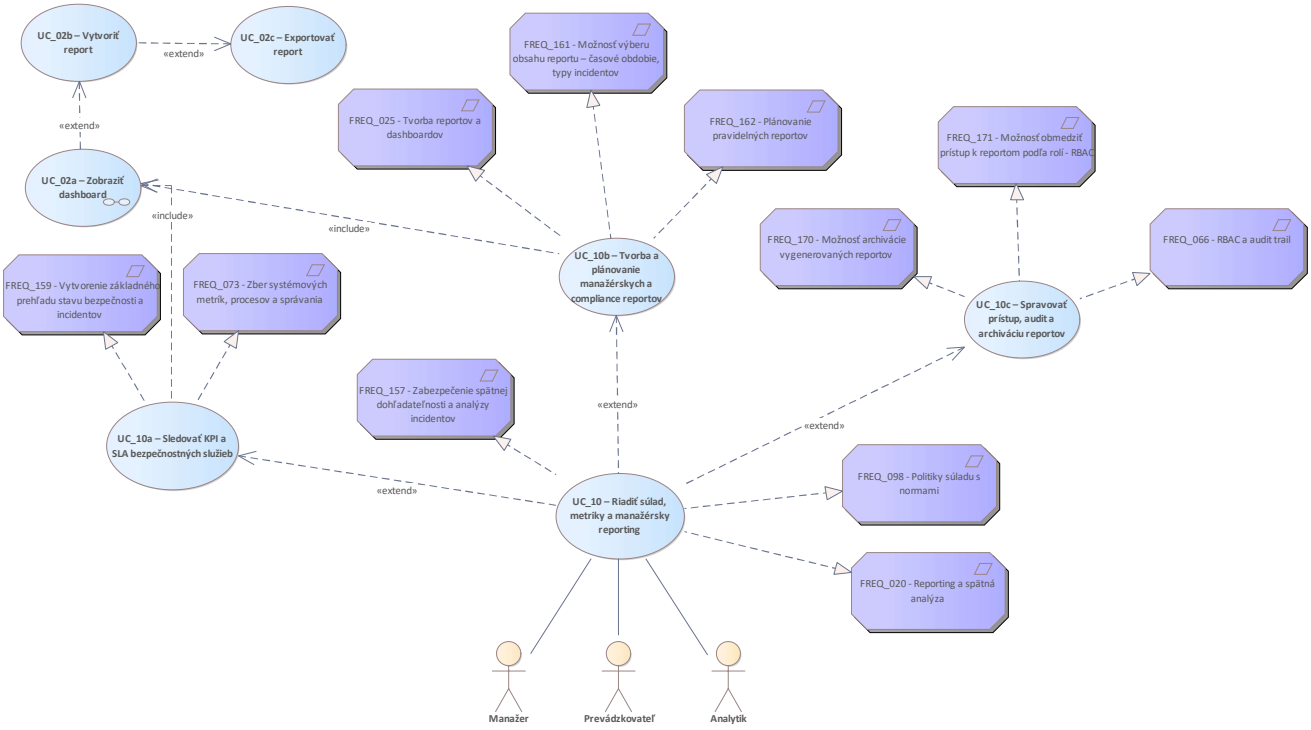


Figure 13 : UC_10 – Manage compliance, metrics and management reporting

Use case	UC_10 – Manage compliance, metrics and management reporting
Objective	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 trend), whether the system complies with standards/regulations, whether all important matters are traceable 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 is set (standards, internal guidelines)
Postcondition	The Manager and the Auditor have clear metrics and reports on the state of security and SOC services at their disposal
Postcondition	Identified problems in KPIs/SLAs or compliance are recorded and have assigned recommendations/measures
Postcondition	The created reports 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 the management overview is displayed, the system indicates that some metrics cannot be reliably calculated (missing data, incomplete logs); the Manager receives information on which parts are incomplete and can request remediation.

ALT_02 – Identified breach of an SLA or a 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 launches internal measures (e.g. a service quality incident, corrective measures).

Use case	UC_10a – Monitor KPIs and SLAs of security services
Objective	UC_10a extends UC_10 with detailed monitoring of KPIs, SLAs and the overall state of security. The Manager, the Auditor or the Service operator view, in the dashboards, trend charts, overviews by team/customer and the overall “health status” of SOCaaS.
Precondition	From UC_10 – the management overview is available.
Precondition	KPIs and SLAs are defined (e.g. MTTR, MTTD, % of incidents within the 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 a dashboard).
3	The Manager filters the metrics by period, team, incident type or customer.
4	The system displays trend charts, averages, SLA fulfilment percentages and comparisons with target values.
5	The Manager identifies problems (e.g. a specific team has long response times) and marks them for detailed analysis or corrective measures.

ALT_10a_01 – KPIs or SLAs are not yet 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 assigned data sources.

Use case	UC_10b – Creation and scheduling of management and compliance reports
Objective	UC_10b addresses practical work with reports for management and audits: creation, configuration of content, 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 according to needs.

Basic Path	Basic Path

Step	**Activity**
1	The Manager or the Auditor opens the “Reports” section within UC_10.
2	Selects 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, 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 a report) and UC_02c (Export a 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 can decide to generate the report anyway or to adjust the scope.

Use case	UC_10c – Manage access, audit and archiving of reports
Objective	UC_10c addresses the security and audit aspect 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 purposes of future audits.

Basic Path	Basic Path

Step	**Activity**
1	The Manager or the Auditor opens the management of reports and access.
2	For individual report types, sets which roles or specific users have access to them.
3	The system applies RBAC when reports are displayed and downloaded.
4	With 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 standard interface.

ALT_01 – Attempt to access a report without authorization	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 the Auditor.

4.3.2.10 UC_11 – Manage the customer's assets and environment

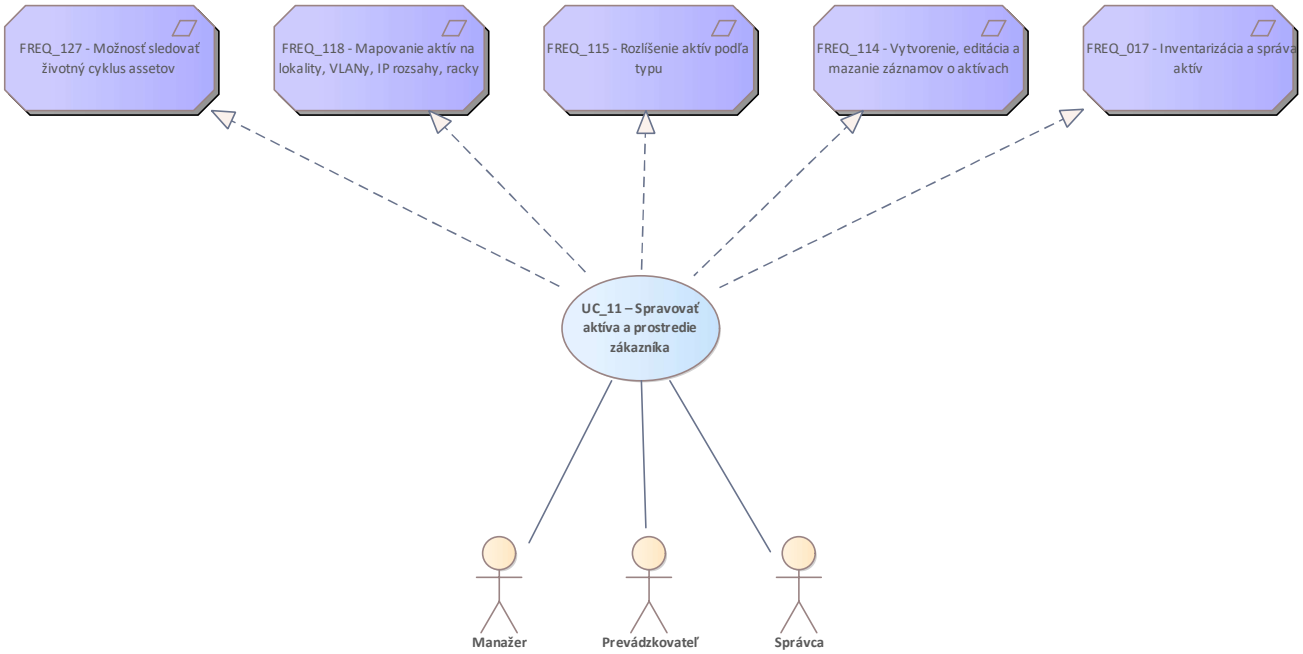


Figure 14 : UC_11 – Manage assets and the customer environment

Use case	UC_11 – Manage the customer's assets and environment
Objective	The objective of UC_11 is for SOCaaS to know what it is actually protecting – all relevant assets of the customer (servers, applications, networks, cloud services...) and their relationships. It makes it possible to record assets, distinguish their types, map them to locations/VLANs/IPs and track their life cycle so that incidents, threats and vulnerabilities can be tied 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 at its disposal.
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 Service operator opens the “Assets” module for a specific customer.
2	Creates a new asset record and enters its basic data (name, description, owner, criticality).Creates a new asset record and enters its basic data (name, description, owner, criticality).
3	Selects 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 Service operator sets the status of the asset in its life cycle (e.g. “in operation”, “being deployed”, “decommissioned”).

7	The system saves the asset record and makes it available to other modules (incident mgmt, TI, vuln mgmt, reporting).
8	With any further 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 Service operator selects 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 – Duplicate asset detected	Alternate
Step	**Activity**
1	When saving a new asset, the system detects 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.

4.3.2.11 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 prioritization according to risk, to the planning and tracking of corrective measures.
The aim is to have a comprehensive overview of where we have security gaps, what is most important to fix and within what time.

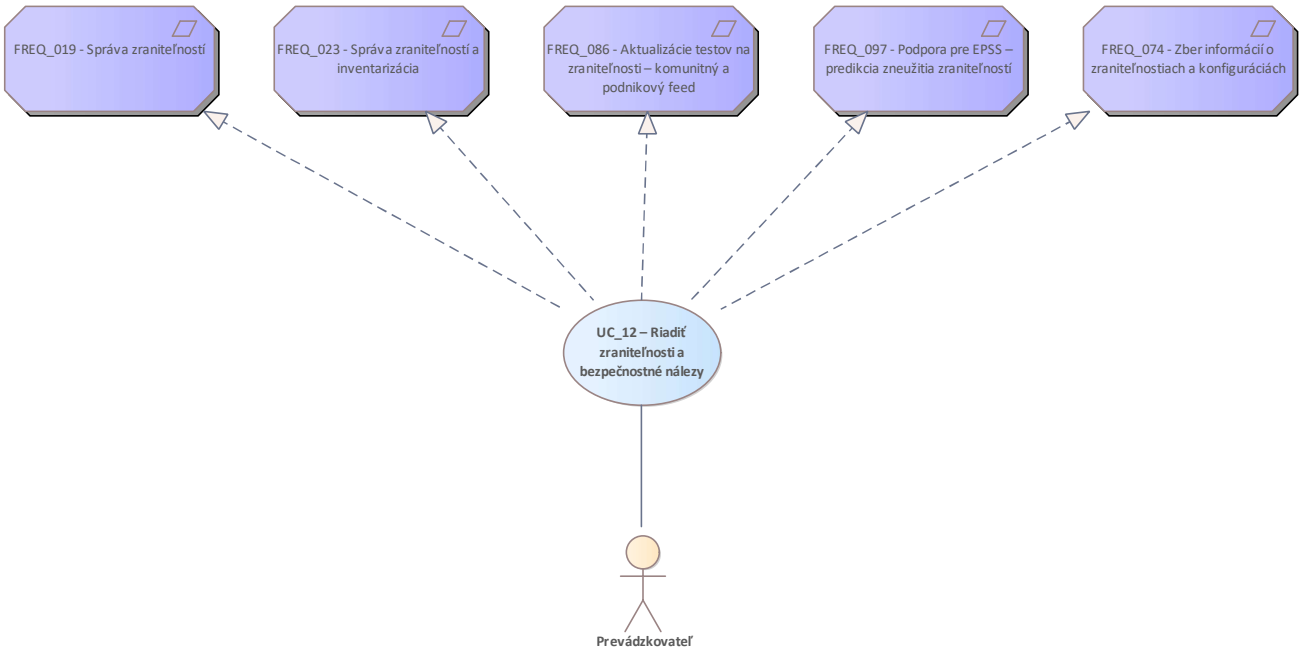


Figure 15 : UC_12 – Manage vulnerabilities and security findings

Use case	UC_12 – Manage vulnerabilities and security findings
Objective	UC_12 covers vulnerability management in SOCaaS: from receiving the outputs of vulnerability scanners, through their assignment to specific assets (UC_11) and prioritization according to risk, to the planning and tracking of corrective measures. The aim is to have a comprehensive overview of where we have security gaps, what is most important to fix and within what time.
Precondition	Vulnerability scan results or other security findings exist
Precondition	Assets are recorded in the system
Precondition	A basic vulnerability prioritization 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 (unfixed / in progress / fixed / risk accepted) is tracked and traceable.

Basic Path	Basic Path

Step	**Activity**
1	Import or entry of vulnerabilities
2	Normalization 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 corrective measures
7	Re-scan or status verification
8	Update of vulnerability status
9	Reporting and linking with 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 limits, no vendor patch exists, high impact on operations); after discussion, the Manager decides whether the risk is temporarily accepted or compensating measures are introduced (network segmentation, increased monitoring). The status of the vulnerability is set to “risk accepted” or “compensated”.

ALT_12_02 – Duplicate or obsolete finding detected	Alternate

Step	**Activity**
1	When importing scan results, the system detects 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 – The vulnerability leads to the opening of an incident	Alternate

Step	**Activity**
1	The Analyst evaluates a specific vulnerability (e.g. a critical, actively exploited one) as an acute threat; they create a security incident for it, which begins to be handled according to UC_05, while UC_12 is used to track the long-term remediation.

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

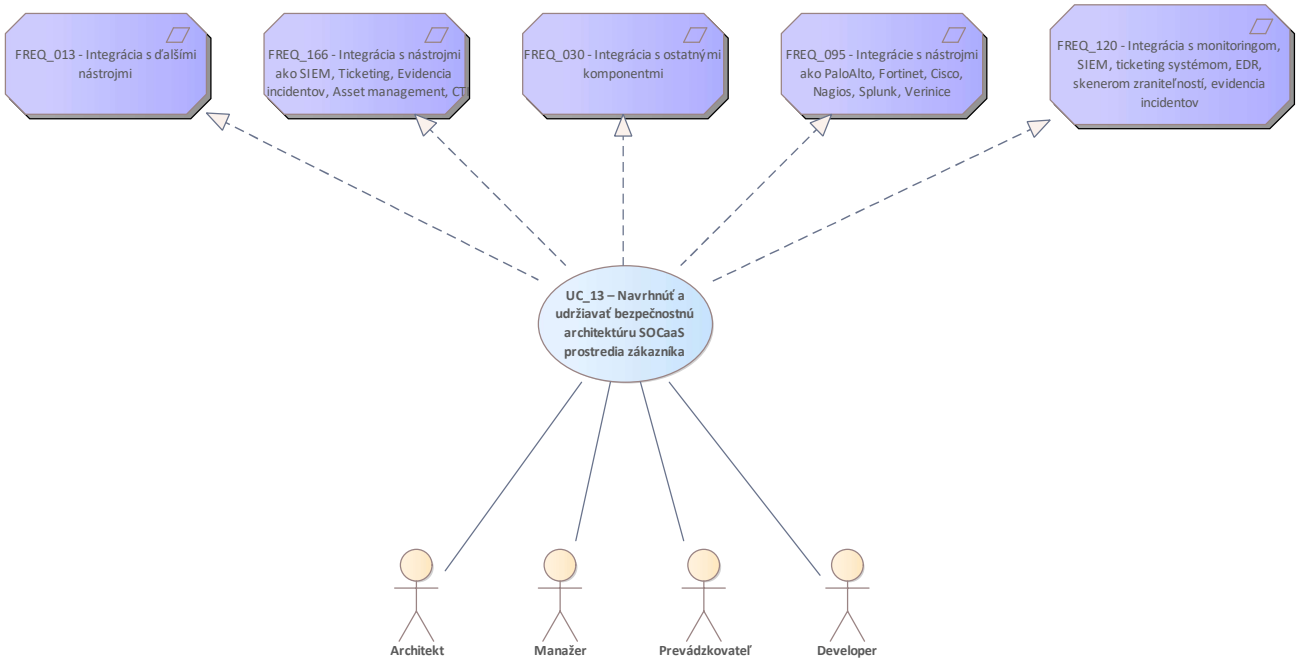


Figure 16 : UC_13 – Design and maintain the security architecture of the customer's SOCaaS environment

Use case	UC_13 – Design and maintain the security architecture of the customer's SOCaaS environment
Objective	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 the data flow (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), which serves as the 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 exists for connecting the customer's environment to SOCaaS (topology, integration points, data flow).
Postcondition	Integration channels are defined (which system → by what means → into which SOCaaS module).
Postcondition	Changes to the architecture are traceable (versions, who approved them, when) and usable as input for onboarding and operations.
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 patternsDesign of data flows and integration patterns
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's 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 narrowed scope of monitoring 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 Service 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.

4.3.2.13 UC_14 – Customer portal and collaboration

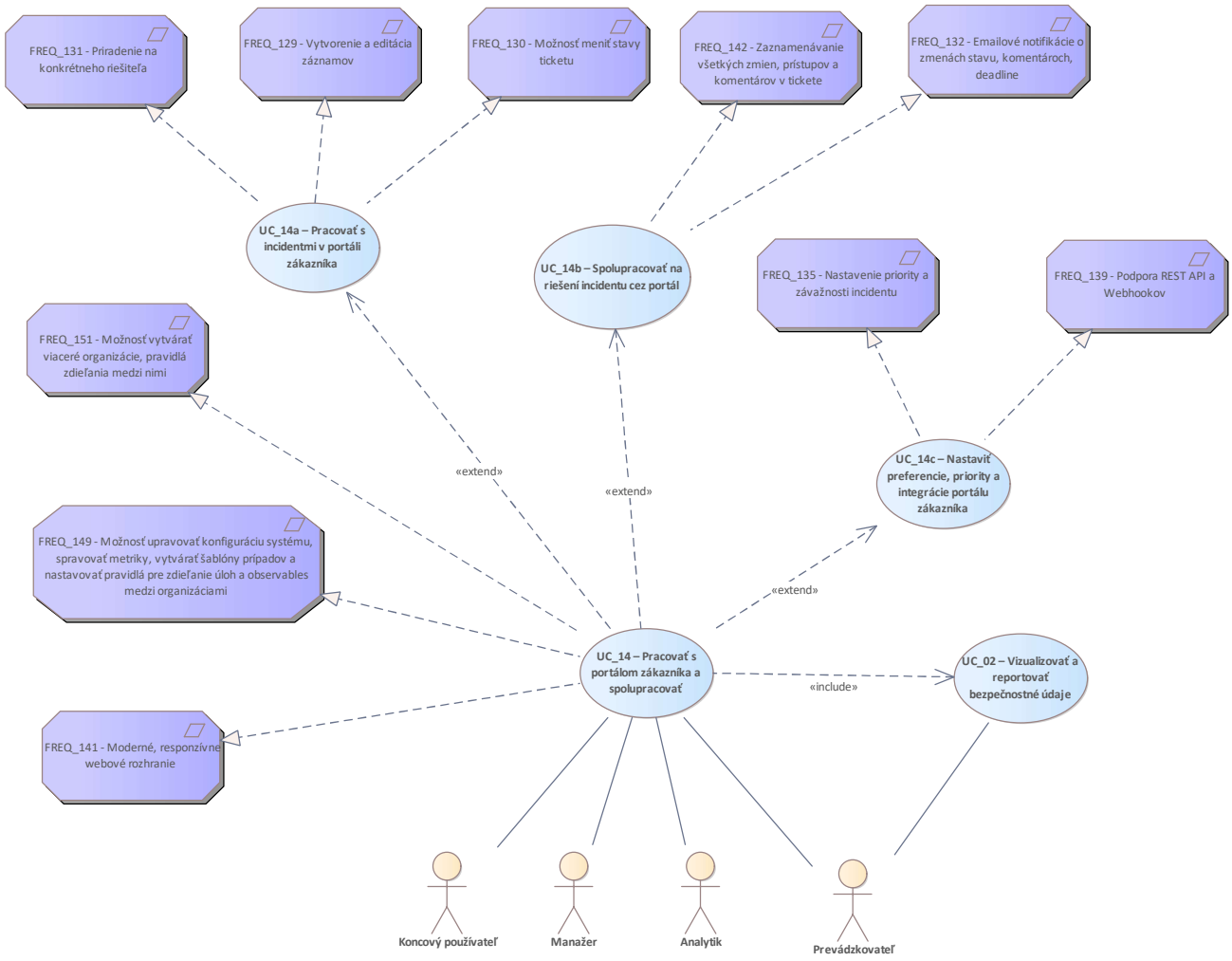


Figure 17 : UC_14 – Work with the customer portal and collaborate

Use case	UC_14 – Work with the customer portal and collaborate
Objective	UC_14 addresses the customer's view of SOCaaS – via a web portal, where the customer: sees their incidents, KPIs, reports and assets, can collaborate with the SOC (comments, attachments, approval of measures), can customize the portal for multiple organizations/units. The portal is a modern, responsive, multi-tenant web interface and also supports more advanced settings (templates, sharing, integration hooks in the sub-UCs).
Precondition	The customer has been onboarded (UC_00) and has at least one organization / tenant created.
Precondition	Accounts are created for users and roles are assigned (e.g. "Customer security officer").
Precondition	The portal is accessible via a secured 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).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 organizations and the settings are tailored to the needs of the specific organization.

Basic Path		Basic Path
Step	**Activity**	
1	The end user logs in to the customer portal (SSO / name + password / MFA according to the policy).	
2	The system displays the home overview – a dashboard with the status of incidents, basic KPIs and notifications	
3	The user selects the organization/tenant on behalf of which they want to work (if they have more than one).	
4	In the menu, the user chooses whether they want to handle incidents, view reports or manage settings and notifications.In the menu, the user chooses 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 organization is displayed; the user can click through to the detail	
6	When “Reports” is selected, the user can view or download the management reports that are available for their organization	
7	When “Settings” is selected, the user can (depending on their role) change notifications, preferences and the organizational 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 organization.	

ALT_14_01 – The user has no organization assigned		Alternate
Step	**Activity**	
1	After login, the system detects that the account is not linked to any organization; it displays information about the missing assignment and recommends contacting the Service operator or the Manager.	

ALT_14_02 – Restricted view according to role		Alternate
Step	**Activity**	
1	The user has only a role with restricted rights (e.g. “read-only”); the portal hides some action elements (editing, approval) and allows only viewing of incidents and reports.	

Use case	UC_14a – Work with incidents in the customer portal
Objective	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 organization.
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, the user can add information (description, impacts, contact person) or change the incident status (e.g. from “New” to “In progress”).
5	The user can assign an incident to a specific resolver or team (e.g. "SOC L1", "SOC L2", "customer's internal IT").
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 “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 – 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.

Use case	UC_14b – Collaborate on incident resolution via the portal
Objective	UC_14b extends UC_14 with a collaboration layer for incident resolution: comments between the customer and the SOC, e-mail notifications about 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 are 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 synchronized to the portal and displayed to the customer as another comment.
6	All comments, status changes and accesses can be viewed retrospectively in the incident history.

ALT_14b_01 – Notifications are disabled for a specific user	Alternate

Use case	UC_14c – Set preferences, priorities and integrations of the customer portal

Objective	UC_14c extends UC_14 with a configuration part for the customer: setting the priorities and severity of incidents from the customer's perspective, setting up 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 organization.
Precondition	At least basic incident priorities and categories are defined for the organization.
Postcondition	Preferred incident priorities/severities are defined for the given organization.
Postcondition	Integration channels are configured (if necessary) – e.g. Webhooks for notifications to the internal ticketing system.

Basic Path	Basic Path

Step	**Activity**
1	The Manager opens the “Organization settings” section in the portal.
2	In the “Incidents” section, they set the priority/severity mapping (e.g. how SOC priorities are to be reflected in the customer's internal priorities).
3	In the “Integrations” section, 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 alerts the Manager to this and either saves the integration as inactive or requires the parameters to be corrected.

4.3.2.14 UC_15 – Manage the operation of the SOCaaS platform

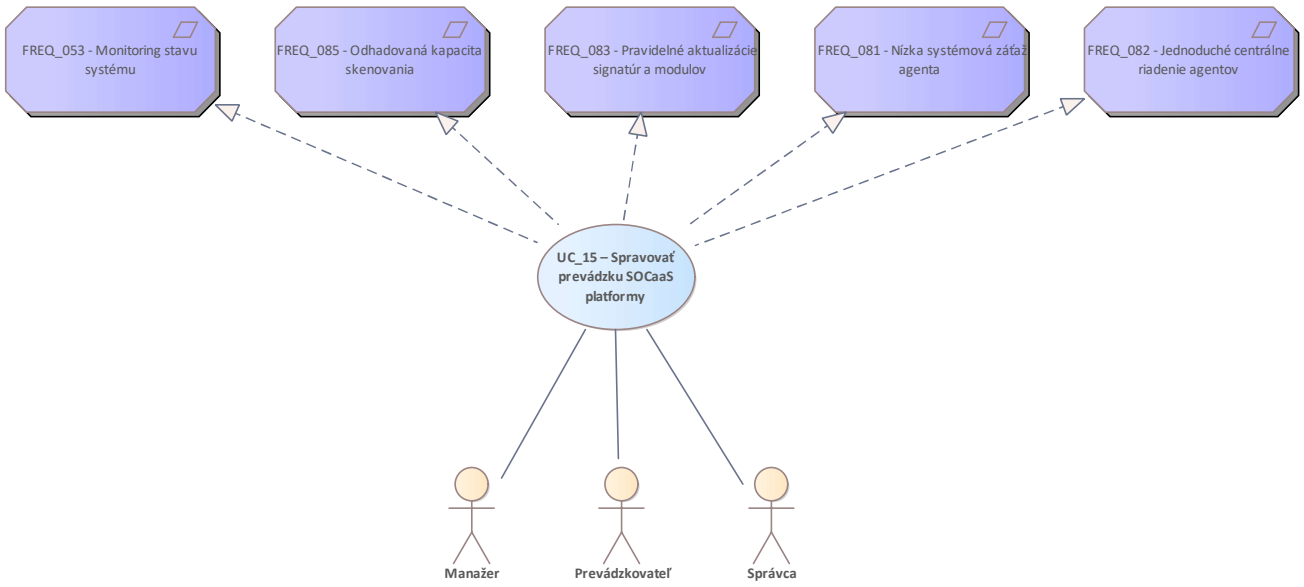


Figure 18 : UC_15 – Manage SOCaaS platform operations

Use case	UC_15 – Manage the operation of the SOCaaS platform
Objective	UC_15 addresses the technical operation of the SOCaaS platform so that: it is stable, performant and available, agents and sensors do not place 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, to be able to handle the expected volume of data and not to jeopardize the performance of the customer's systems.
Precondition	The SOCaaS platform is deployed in the production environment (on-prem/cloud/SaaS).
Precondition	Monitoring tools and mechanisms exist for tracking the status of the platform and the agents.
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 performed 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 optimization.

Basic Path	Basic Path

Step	**Activity**
1	The Operator monitors the status of the SOCaaS platform in the operational dashboard (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 grows, 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. outside peak hours) in order to minimize 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 scope.
6	If necessary, the Operator adjusts the agent configuration (estimates of scanning frequency, 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 detected	Alternate

Step	**Activity**
1	Monitoring shows that the agents or the platform itself place 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 optimization.

ALT_15_02 – Failure of signature or module update	Alternate

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

4.3.2.15 UC_16 – Customize reports, exports and documents to the organization's needs

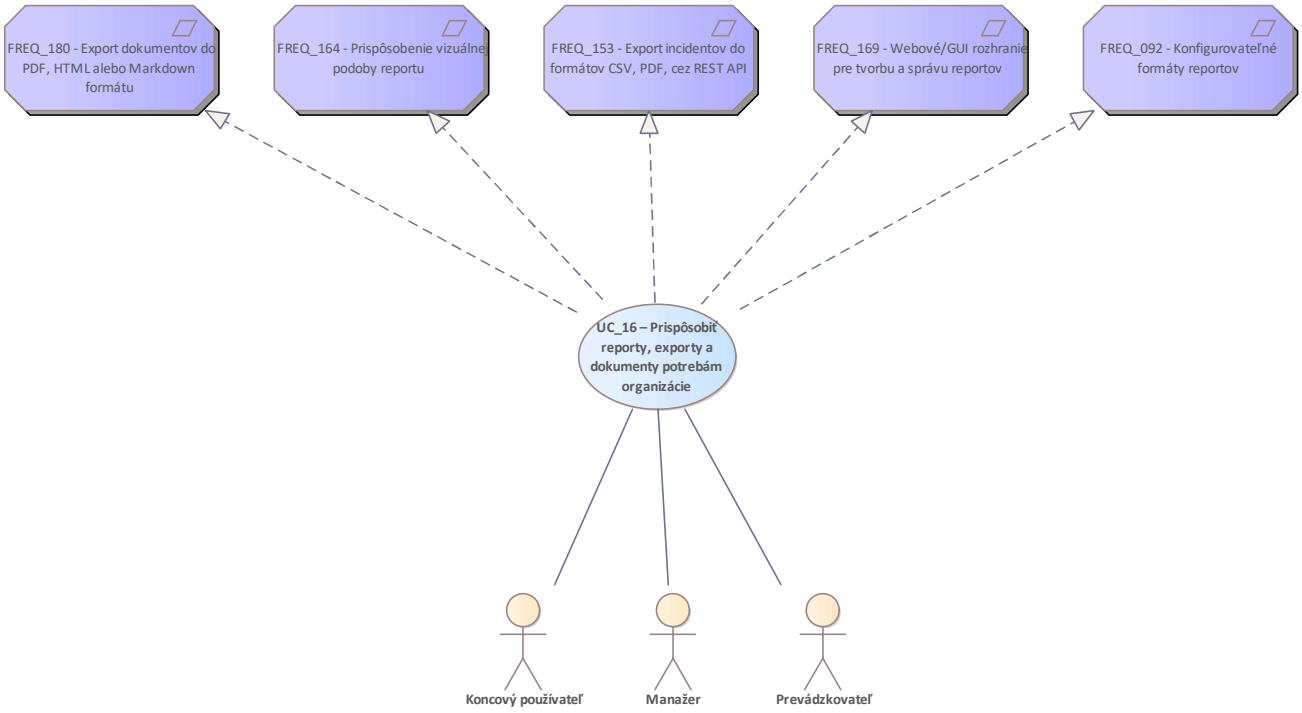


Figure 19 : UC_16 – Customize reports, exports and documents to the organization's needs

Use case	UC_16 – Customize reports, exports and documents to the organization's needs
Objective	UC_16 addresses the configuration of the form of SOCaaS outputs: the appearance of reports according to the corporate identity (logo, colours, header), export formats (CSV, XML, PDF, HTML, Markdown), export of incidents and documents for further processing or archiving. It builds on UC_10b (which addresses the content and scheduling of reports) by enabling detailed settings of presentation and export so that the outputs fit into the customer's environment and tools.
Precondition	There are defined reports and documents that can be generated (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 organizations (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	The Manager or Auditor opens the “Report and export settings” section in the portal.
2	They select the type of output they want to customize (e.g. “management report”, “incident overview”, “knowledge document”).

3	In the Appearance section, they select or modify the template:
4	In the Export formats section, they select the formats in which 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 preferred export formats for the given type of report/document.
7	The next time a report is generated or an incident is displayed, the system allows one of the configured exports to be selected directly (e.g. “Download as PDF”, “Export to CSV”, “Export document to HTML”).
8	Exported files can be downloaded by the user or sent to other systems (e.g. via REST API), according to the organization'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 alerts them to this and offers 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 aborts 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 organization 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.

4.3.2.16 UC_17 – Manage users and roles in the SOCaaS platform

UC_17 addresses the administration of users and roles within SOCaaS:

creating and modifying accounts,

assigning roles (SOC, customer, auditor…),

setting up access via RBAC,

adding custom attributes where applicable.

It is a system/administrative UC that supports all other UCs by determining who can see and do what in the platform.

Figure 20 : UC_16 – Customize reports, exports and documents to the organization's needs

Use case	UC_16 – Customize reports, exports and documents to the organization's needs
Objective	UC_16 addresses the configuration of the form of SOCaaS outputs: the appearance of reports according to the corporate identity (logo, colours, header),

	export formats (CSV, XML, PDF, HTML, Markdown), export of incidents and documents for further processing or archiving. It builds on UC_10b (which addresses the content and scheduling of reports) by enabling detailed settings of presentation and export so that the outputs fit into the customer's environment and tools.
Precondition	There are defined reports and documents that can be generated (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 organizations (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	The Manager or Auditor opens the “Report and export settings” section in the portal.
2	They select the type of output they want to customize (e.g. “management report”, “incident overview”, “knowledge document”).
3	In the Appearance section, they select or modify the template:
4	In the Export formats section, they select the formats in which 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 preferred export formats for the given type of report/document.
7	The next time a report is generated or an incident is displayed, the system allows one of the configured exports to be selected directly (e.g. “Download as PDF”, “Export to CSV”, “Export document to HTML”).
8	Exported files can be downloaded by the user or sent to other systems (e.g. via REST API), according to the organization'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 alerts them to this and offers 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 aborts 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 organization 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.

4.4 Business architecture

4.4.1 Identification of the design owners of end services (UX)

Name and surname	Job position	Telephone	E-mail

Table 7 : Identification of the design owners of end services (UX)

4.4.2 Specification of the end customer's needs

Description of the user need	Reference to the research from which the need arises	Proposed solution to the need	Quantity	Impact on customer experience	Technical complexity	Business benefit	Priority

Table 8 : Specification of user needs

Explanation of **Chyba! Neznámy argument prepínača.** (table cells from left to right):

- Description of the user need: a statement of a specific user need in the first person that does not describe the solution.
- Reference to the research from which the need arises: describes the specific interviews with respondents in which the requirement was expressed.
- Proposed solution to the need: one of the possible solutions that has been quantitatively validated.
- Quantity: output of the questionnaire in which users rated the proposed solution to the need on a scale from 1 to 10, where 1 is "It is not important to me" and 10 is "It is very important to me."
- Impact on customer experience: determined by a combination of qualitative and quantitative information from the research and rated on the MoSCoW[1] scale.
- Technical complexity is the input from the technical solution provider.
- Business benefit is the input from the business sponsor.
- Priority is the resulting importance of implementing this solution after considering the columns Impact on customer experience, Technical complexity and Business benefit.

4.4.3 User interface prototype design

4.4.4 Screen visual design

[1] https://en.wikipedia.org/wiki/MoSCoW_method

5 Data architecture

The following chapters describe the design of the entities of the module's persistence layer.
Each entity contains:

- Definition of the entity
- List of basic attributes
- Attribute name
- Data type
- Mandatory – the attribute must be populated (not null)
- Description – explanation of what the attribute represents, if this is not obvious from the name, or a description of the validation

6 Application architecture

The application architecture of the CNM module is fully modelled in the Sparx Enterprise Architect (EA) tool in ArchiMate 3.0 notation.

The following chapter shows the application architecture diagram with descriptions of the objects, which were generated from EA as an output.

All descriptions of application architecture objects, such as application services, application components, functional components and data objects, are part of the model in EA, recorded in the object's "Description" (note).

Since the output for the modelling of the DNR application architecture is generated from EA, where a given application architecture object is used more than once, its description is identical.

6.1 Application architecture

6.2 Aplication Viewpoints

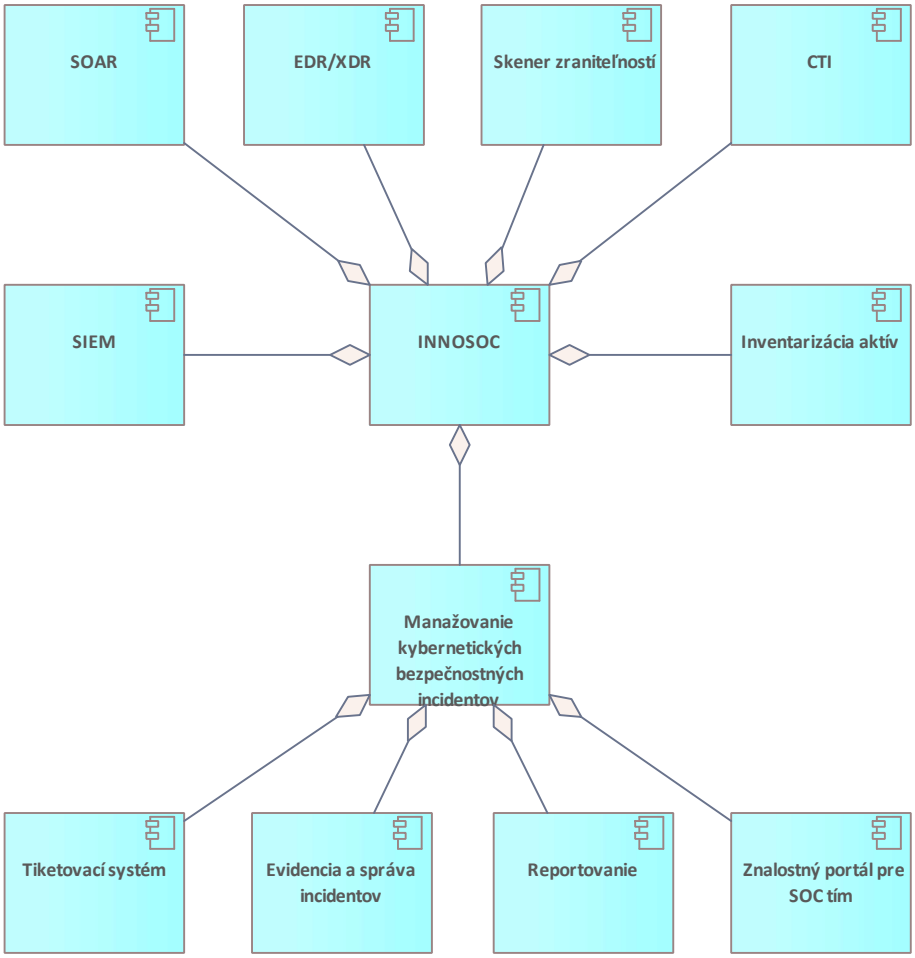


Figure 21 : Aplication Viewpoints Components

The diagram presents a high-level (HL) list of the solution components.

Description of the elements used:

CTI

CTI (Cyber Threat Intelligence) is a key building block of the SOCaaS architecture, as it provides the capability to enrich detection rules, correlation mechanisms and playbooks with external threat information – 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 of threat hunting and forensic analysis processes,
• Prioritization of incidents according to threat context.

For the solution design, open-source platforms were chosen that cover various aspects of CTI – from IOC collection through visualization to STIX/TAXII compatibility.

EDR/XDR

EDR (Endpoint Detection and Response) and XDR (Extended Detection and Response) are the fundamental technological pillars for threat detection at the endpoint level. In the SOCaaS environment, they serve for the collection and correlation of, and response to, events that would otherwise escape traditional perimeter detections. 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 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

with relevant threats, systems or historical cases. Ideally, it also supports advanced data enrichment, auditing, multi-tenancy and the possibility of retrospective investigation of incidents.

INNOSOC

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 organization's infrastructure. In the SOCaaS environment, it plays an even more significant role, as it enables centralized management, categorization, monitoring and tracking of the security status of the infrastructure across different customers.

Management of cybersecurity incidents

Effective handling of cyber incidents is a key pillar of the SOCaaS architecture. It includes not only the recording of events itself, but a comprehensive cycle of their registration, 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 indispensable pillar of every SOCaaS solution. Within the INNOSOC architecture, four subcomponents have been identified which together form an integrated system for the management, registration, documentation and handling of incidents:

1. Ticketing system
2. Incident registration and management
3. Reporting
4. Knowledge portal for the SOC team

Reporting

The reporting component within the SOCaaS architecture serves as the central means for aggregating, visualizing and distributing 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 organization or clients. Such a reporting system must cover:

• Generation of metrics for incidents, alerts, SLAs and KPIs,
• Overviews of trends over time and assessment 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 provides 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 weaknesses in the infrastructure, assess the status of updates, configurations and settings, and provide information for further prioritization and planning of corrective measures. The scanner 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 incident responses, orchestrating security tools and supporting analysts' decision-making. In SOCaaS, it is a fundamental element enabling the scaling of response capacities without the need to increase staff numbers. It is indispensable especially when processing large volumes of alerts, where it reduces the MTTR and MTTC metrics.

Ticketing system

The ticketing system serves as the central communication and organizational 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

A 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 the standardization 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.

6.3 Solution design of application services

6.3.1 Existing application interfaces

The CNM project envisages retaining the services of the eNotify module - SendNotification, SendNotificationUsingTemplate. For this purpose, a LegacyService service will be created, implementing an interface that receives asynchronous messages in the structure of SendNotification and SendNotificationUsingTemplate requests. A more detailed description is given in the following subchapter.

6.3.2 Discontinued application interfaces

6.3.3 New application interfaces

6.4 Integrations between systems

6.4.1 List of existing integrations without change

6.4.2 List and description of proposed integrations

6.4.3 List and description of required integration interfaces

6.4.4 Design of the application interfaces provided by the system (module)

7 Technological architecture

The chapter describes the physical architecture and defines the requirements for the test and production environments.

The proposed operation is planned on two identical environments differing only in the amount and type of resources:

- **PROD** environment – production environment with real data and operation of the public portal.
- **TEST** environment – testing, training, pre-production environment. The entire TEST environment is accessible via VPN for privileged users.

7.1 Technological architecture

7.2 Technology Viewpoint INNOSOC Common

7.3 Technology Viewpoint Asset Inventory

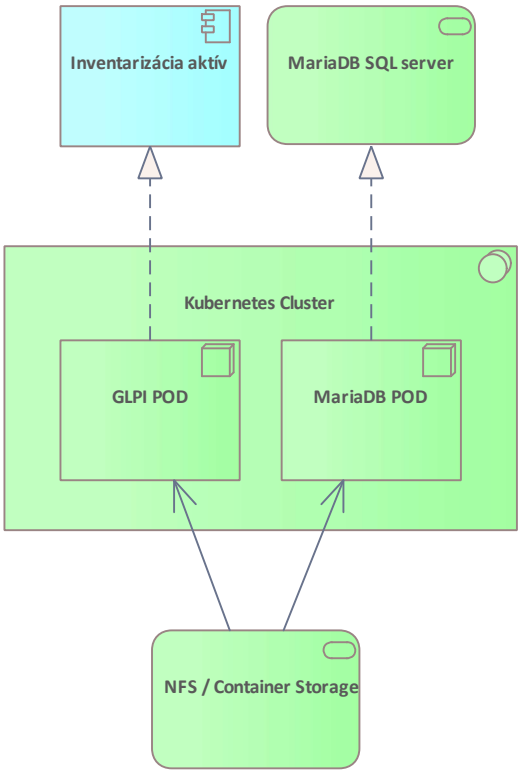


Figure 22 : Technology Usage Viewpoint Asset Inventory alt 1

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 containerized 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 organization's infrastructure. In the SOCaaS environment, it plays an even more significant role, as it enables centralized management, categorization, monitoring and tracking of the security status of the infrastructure across different customers.

NFS / Container Storage

Represents the NFS service or another Container Storage

Kubernetes Cluster

Represents the Kubernetes Cluster

MariaDB SQL server

Represents the technology service of the relational MariaDB SLQ database.

GLPI POD

Represents the GLPI POD for asset inventory.

MariaDB POD

Represents the MariaDB POD for the relational SQL DB.

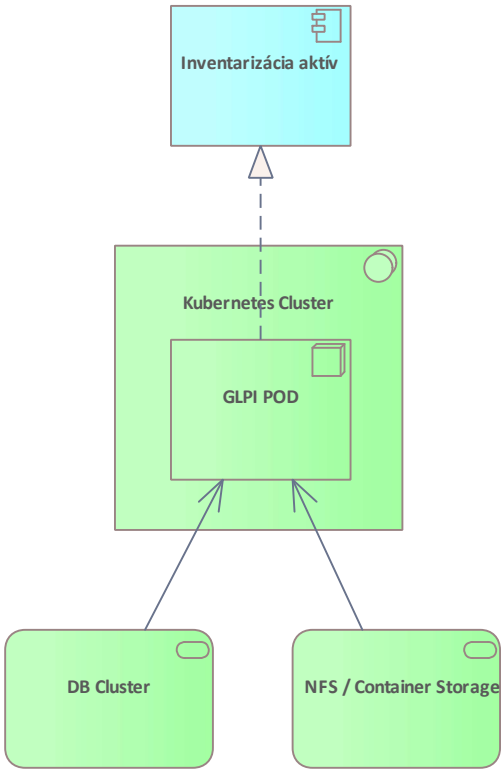


Figure 23 : Technology Usage Viewpoint Asset Inventory alt 2

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 centralized 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 organization's infrastructure. In the SOCaaS environment, it plays an even more significant role, as it enables centralized management, categorization, monitoring and tracking of the security status of the infrastructure across different customers.

DB Cluster
Represents the clustered relational database service

NFS / Container Storage
Represents the NFS service or another Container Storage

Kubernetes Cluster
Represents the Kubernetes Cluster

GLPI POD
Represents the GLPI POD for asset inventory.

7.4 Technology Viewpoint Kubernetes

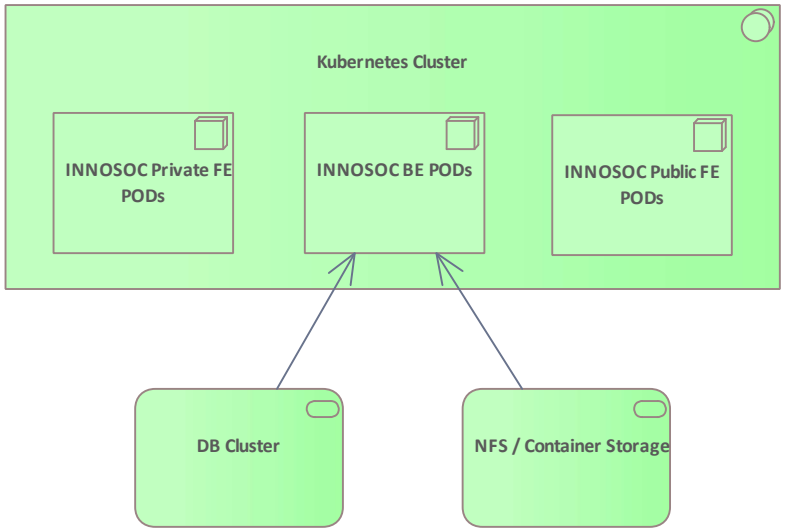


Figure 24 : Technology Usage Viewpoint CNM and Containers maping

The diagram shows the assignment of the individual PODs to the corresponding application components.

Description of the elements used:

DB Cluster
Represents the clustered relational database service

NFS / Container Storage
Represents the NFS service or another Container Storage

Kubernetes Cluster

Represents the Kubernetes Cluster

INNOSOC BE PODs
Represents the CNM BE PODs

INNOSOC Private FE PODs
Represents the CNM FE Private PODs

INNOSOC Public FE PODs
Represents the CNM FE Public PODs

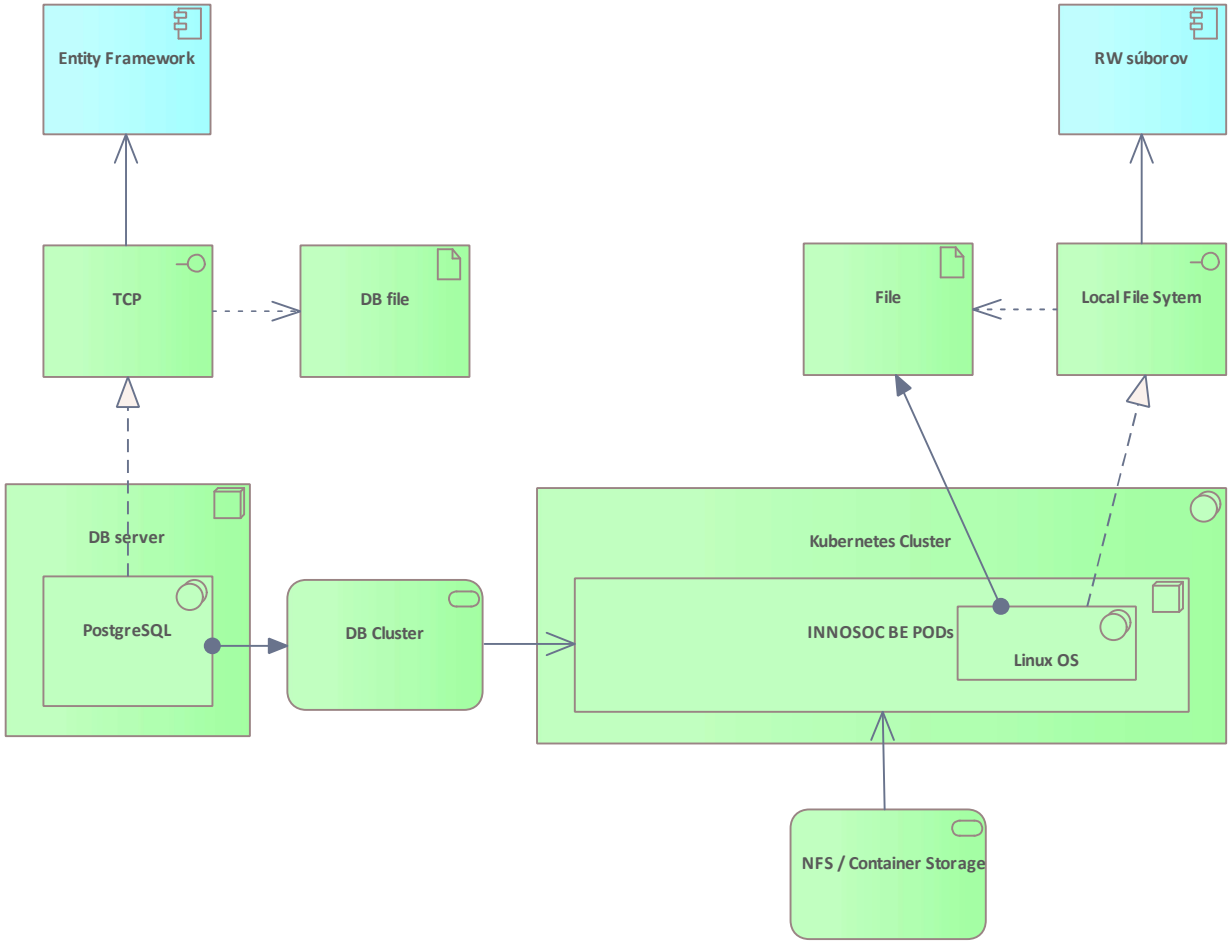


Figure 25 : Technology Usage Viewpoint CNM BE, DB and File System

The diagram represents and describes in more detail the use of the persistence layer for the needs of the BIT NC and CNM 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 the clustered relational database service

NFS / Container Storage
Represents the NFS service or another Container Storage

DB file
DB file - a file representing the MariaDB database file

DB server
Represents the MariaDB server

File
File - a file representing data such as attachments, images, documents and others

Entity Framework
Represents the Entity Framework application module

File RW
Represents the RW module for files

Local File Sytem
Local file system providing work with files, implemented by the operating system under which the BE module is run

TCP
Represents client-server communication using the TCP protocol

Kubernetes Cluster
Represents the Kubernetes Cluster

INNOSOC BE PODs
Represents the CNM BE PODs

Linux OS

PostgreSQL
Represents the PostgreSQL relational database

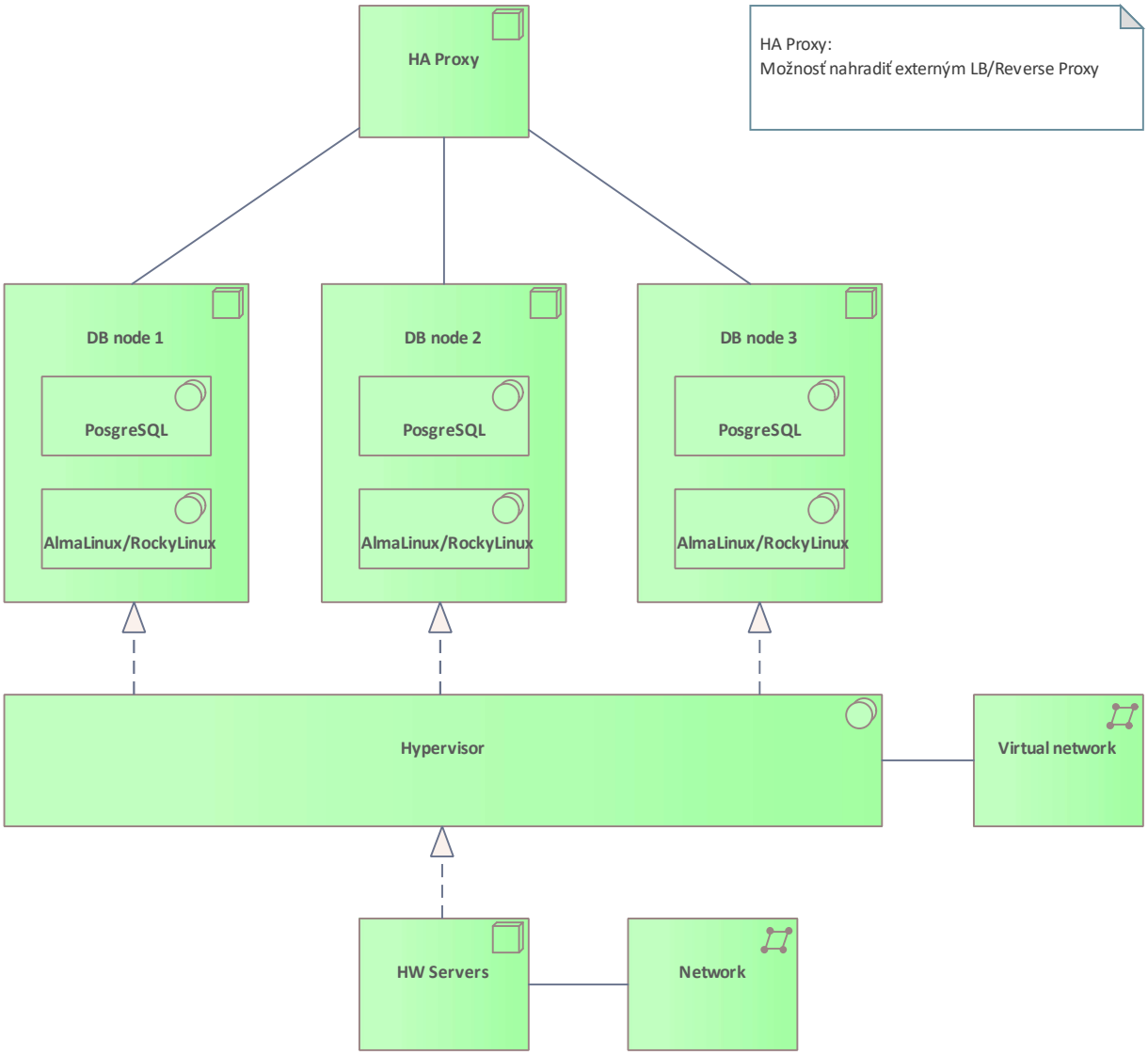


Figure 26 : Technology Viewpoint DB Cluster

The diagram shows the technological architecture of the PostgreSQL Cluster.

Description of the elements used:

DB node 1
Represents Node 2 of the MariaDB Galera Cluster

DB node 2
Represents Node 2 of the MariaDB Galera Cluster

DB node 3
Represents Node 3 of the MariaDB Galera Cluster

HA Proxy
Represents an internal reverse proxy built on HA Proxy

HW Servers
Represents the HW server infrastructure

Hypervisor
Represents a Type 1 or Type 2 hypervisor

Network

Represents the network/subnet of the HW infrastructure

Virtual network
Represents the virtual network/subnet of the hypervisor

Note
HA Proxy:
Can be replaced by an external LB/Reverse Proxy

AlmaLinux/RockyLinux
Represents the AlmaLinux/RockyLinux Linux distribution

PosgreSQL
Represents the PosgreSQL relational database

AlmaLinux/RockyLinux
Represents the AlmaLinux/RockyLinux Linux distribution

PosgreSQL
Represents the PosgreSQL relational database

AlmaLinux/RockyLinux
Represents the AlmaLinux/RockyLinux Linux distribution

PosgreSQL
Represents the PosgreSQL relational database

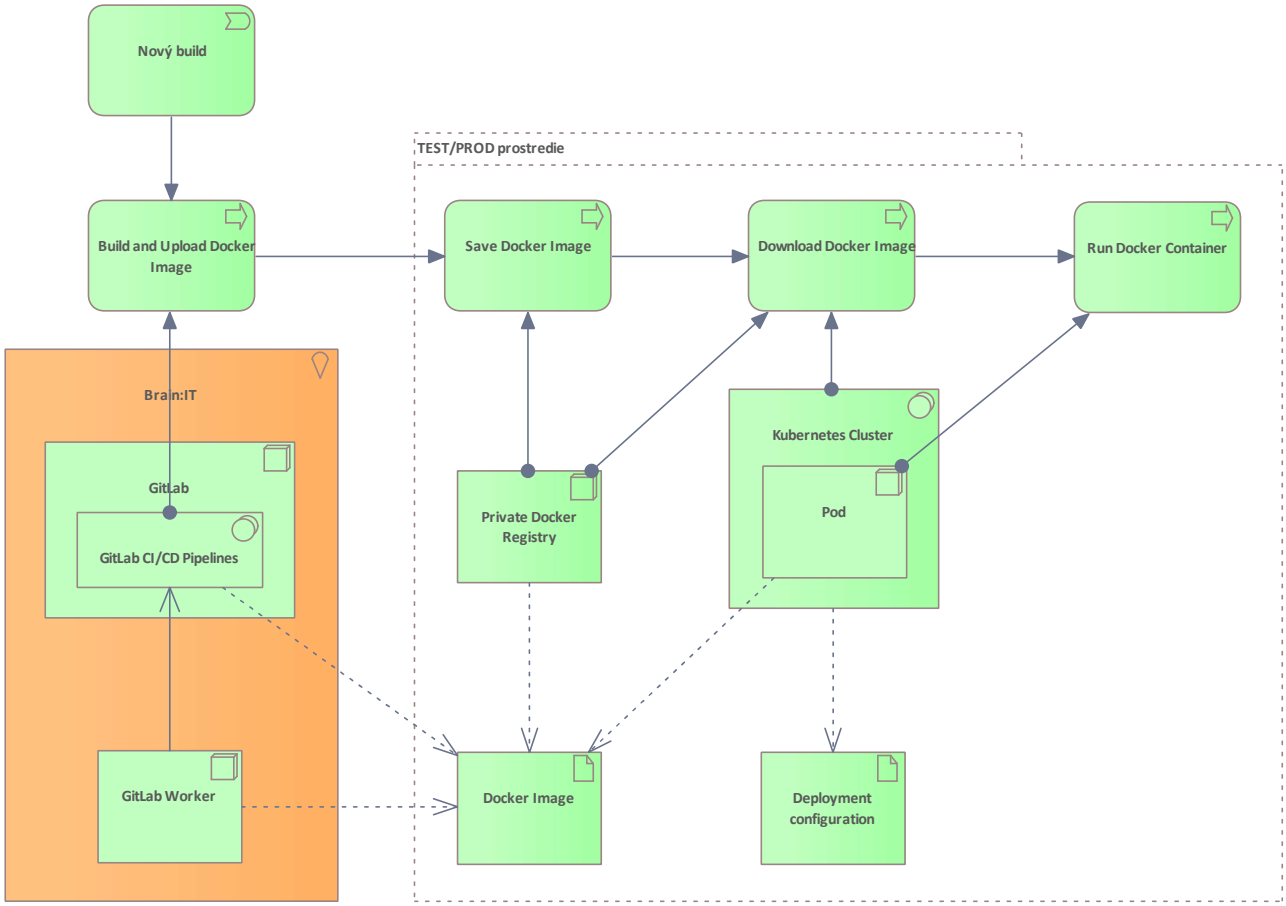


Figure 27 : Technology Viewpoint Development CI/CD TEST/PROD

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 is applicable to the case of in-house packages; for other existing ones, the public DockerHub source is used.

Description of the elements used:

Build and Upload Docker Image
Build process and upload of the Docker Image to the Docker Registry

Download Docker Image
Process of storing the Docker Image in the Docker Registry

New build
Launch of a new build process

Run Docker Container
Process of launching a Docker Container

Save Docker Image
Process of storing the Docker Image in the Docker Registry

TEST/PROD environment
Represents the internal DEV and TEST environment

Brain:IT
Brain:IT site

Private Docker Registry
Represents a private Docker Registry

Deployment configuration
Representation of the Kubernetes Deployment artefact.

Docker Image
Representation of the Docker Image artefact.

GitLab
Represents a GitLab instance.

GitLab Worker
Represents a GitLab Worker instance

Kubernetes Cluster
Represents the Kubernetes Cluster

GitLab CI/CD Pipelines
GitLab CI/CD Pipelines are a series of tasks providing compilation, tests, configuration, deployment and other activities within GitLab.

Pod
Represents a Pod instance in which a new Docker Image is run in the form of a Docker Container

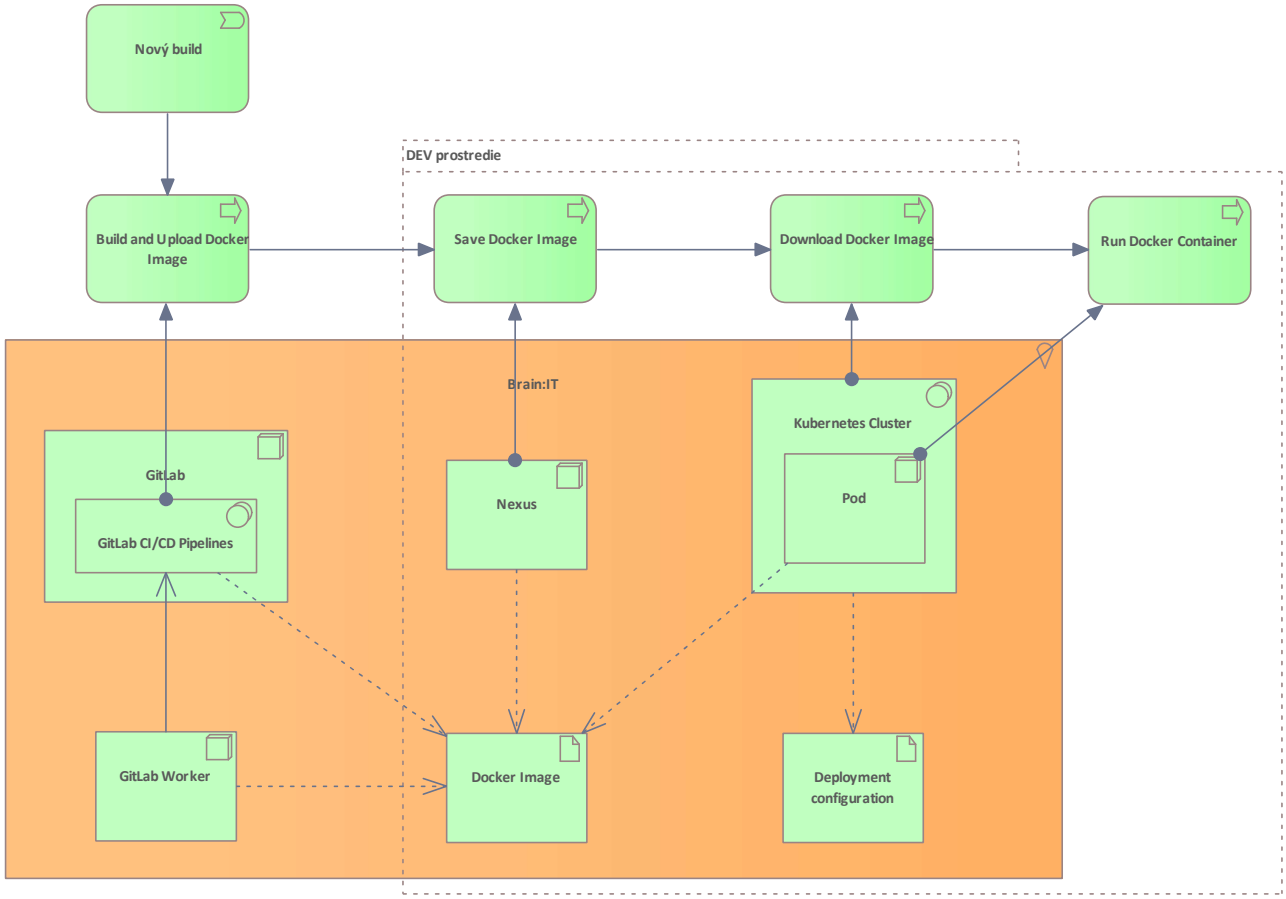


Figure 28 : Technology Viewpoint Developmnent CI/CD DEV

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 is applicable to the case of in-house packages; for other existing ones, the public DockerHub source is used.

Description of the elements used:

Build and Upload Docker Image
Build process and upload of the Docker Image to the Docker Registry

Download Docker Image
Process of storing the Docker Image in the Docker Registry

New build
Launch of a new build process

Run Docker Container
Process of launching a Docker Container

Save Docker Image
Process of storing the Docker Image in the Docker Registry

DEV environment
Represents the internal DEV and TEST environment

Brain:IT
Brain:IT site

Deployment configuration
Representation of the Kubernetes Deployment artefact.

Docker Image
Representation of the Docker Image artefact.

GitLab
Represents a GitLab instance.

GitLab Worker
Represents a GitLab Worker instance

Kubernetes Cluster
Represents the Kubernetes Cluster

Nexus
Represents a private Docker Registry

GitLab CI/CD Pipelines
GitLab CI/CD Pipelines are a series of tasks providing compilation, tests, configuration, deployment and other activities within GitLab.

Pod
Represents a Pod instance in which a new Docker Image is run in the form of a Docker Container

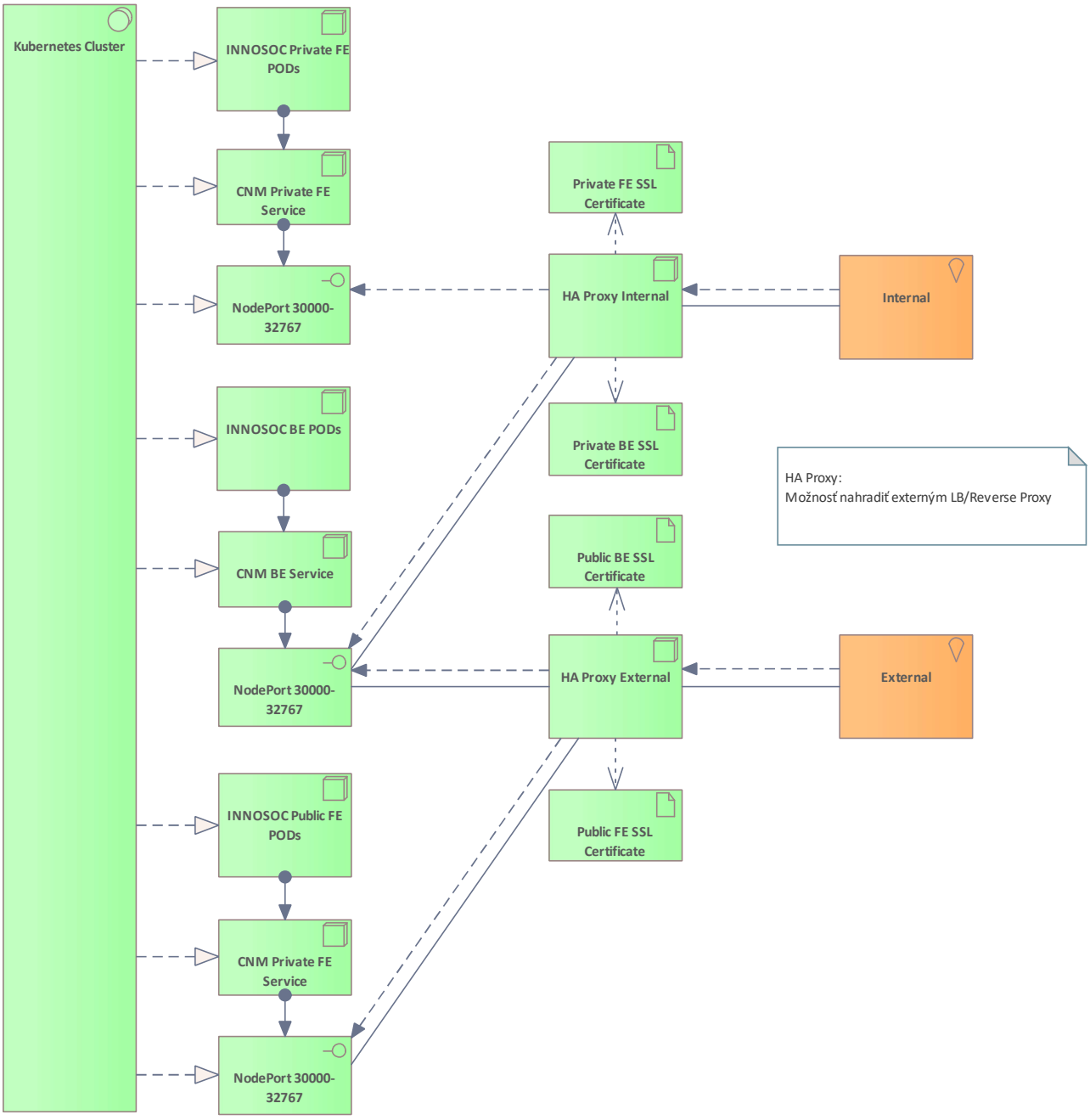


Figure 29 : Technology Viewpoint Interconnect

The diagram shows the deployment and interconnection of exposed Kubernetes services and the LB/Reverse Proxy for two zones.

Description of the elements used:

CNM BE Service
Represents CNM BE Service

CNM Private FE Service
Represents the CNM FE Private Service

CNM Private FE Service
Represents the CNM FE Private Service

External

Represents the external Internet environment

HA Proxy External
Represents an external reverse proxy built on HA Proxy

HA Proxy Internal
Represents an internal reverse proxy built on HA Proxy

Internal
Represents the internal BRAIN:IT environment

Private BE SSL Certificate
Represents a valid private SSL certificate for HTTPS purposes

Private FE SSL Certificate
Represents a valid SSL certificate for HTTPS purposes

Public BE SSL Certificate
Represents a valid public SSL certificate for HTTPS purposes

Public FE SSL Certificate
Represents a valid public SSL certificate for HTTPS purposes

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:
Can be replaced by an external LB/Reverse Proxy

Kubernetes Cluster
Represents the Kubernetes Cluster

INNOSOC BE PODs
Represents the CNM BE PODs

INNOSOC Private FE PODs
Represents the CNM FE Private PODs

INNOSOC Public FE PODs
Represents the CNM FE Public PODs

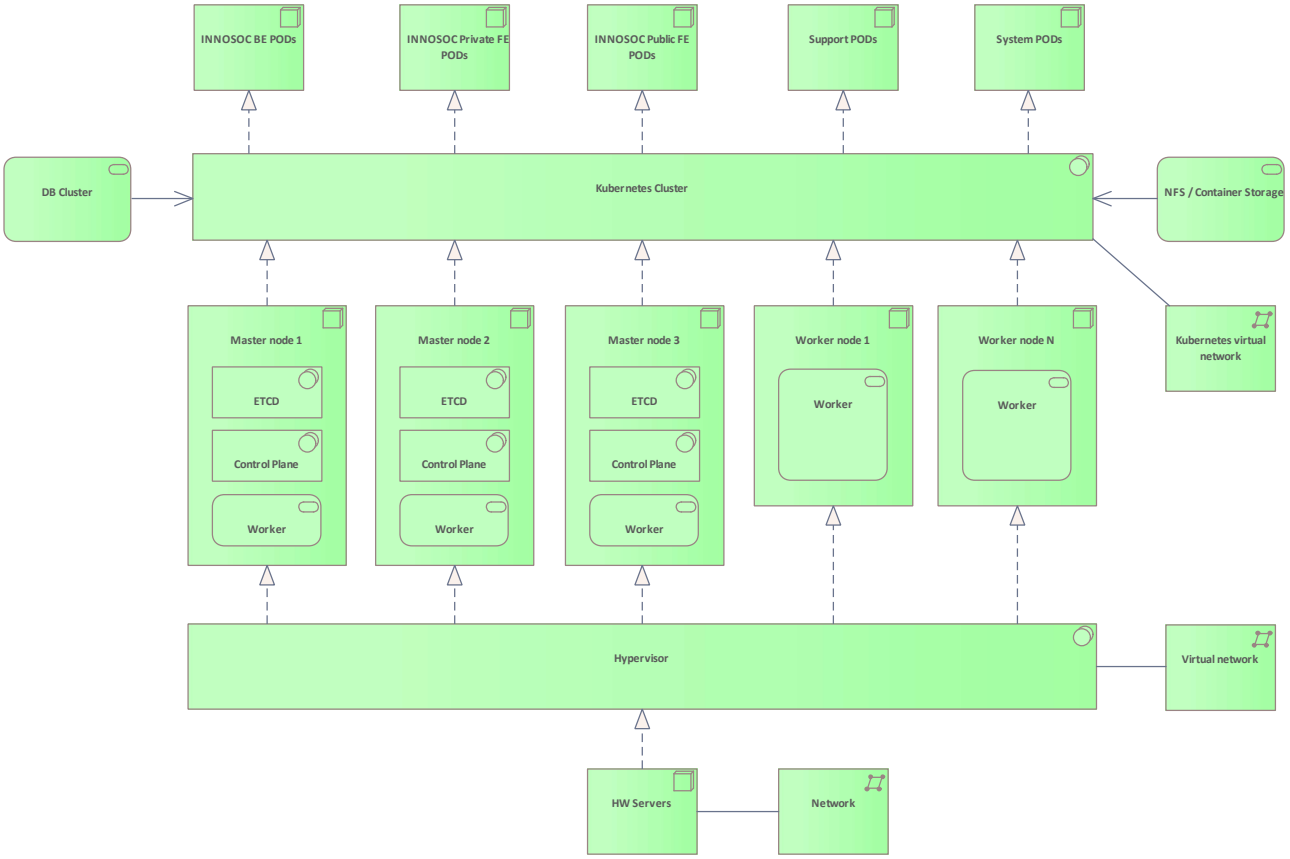


Figure 30 : Technology Viewpoint Kubernetes Cluster

Represents the role and service of the Worker node .

Description of the elements used:

DB Cluster
Represents the clustered relational database service

NFS / Container Storage
Represents the NFS service or another Container Storage

HW Servers
Represents the HW server infrastructure

Hypervisor
Represents a Type 1 or Type 2 hypervisor

Kubernetes virtual network
Represents the virtual network/subnet of the Kubernetes Cluster

Master node 1
Represents the Master node of the Kubernetes cluster

Master node 2
Represents the Master node of the Kubernetes cluster

Master node 3
Represents the Master node of the Kubernetes cluster

Network
Represents the network/subnet of the HW infrastructure

Support PODs
Represents the PODs of supporting services

System PODs
Represents the PODs of system services

Virtual network
Represents the virtual network/subnet of the hypervisor

Worker node 1
Represents a Worker node of the Kubernetes cluster

Worker node N
Represents a Worker node of the Kubernetes cluster

Kubernetes Cluster
Represents the Kubernetes Cluster

INNOSOC BE PODs
Represents the CNM BE PODs

INNOSOC Private FE PODs
Represents the CNM FE Private PODs

INNOSOC Public FE PODs
Represents the CNM FE Public PODs

Worker
Represents the role and service of the Worker node

Control Plane
Represents the Control Plane for managing the Kubernetes cluster

ETCD
Represents the ETCD database

Worker
Represents the role and service of the Worker node

Control Plane
Represents the Control Plane for managing the Kubernetes cluster

ETCD
Represents the ETCD database

Worker

Control Plane
Represents the Control Plane for managing the Kubernetes cluster

ETCD
Represents the ETCD database

Worker

Worker

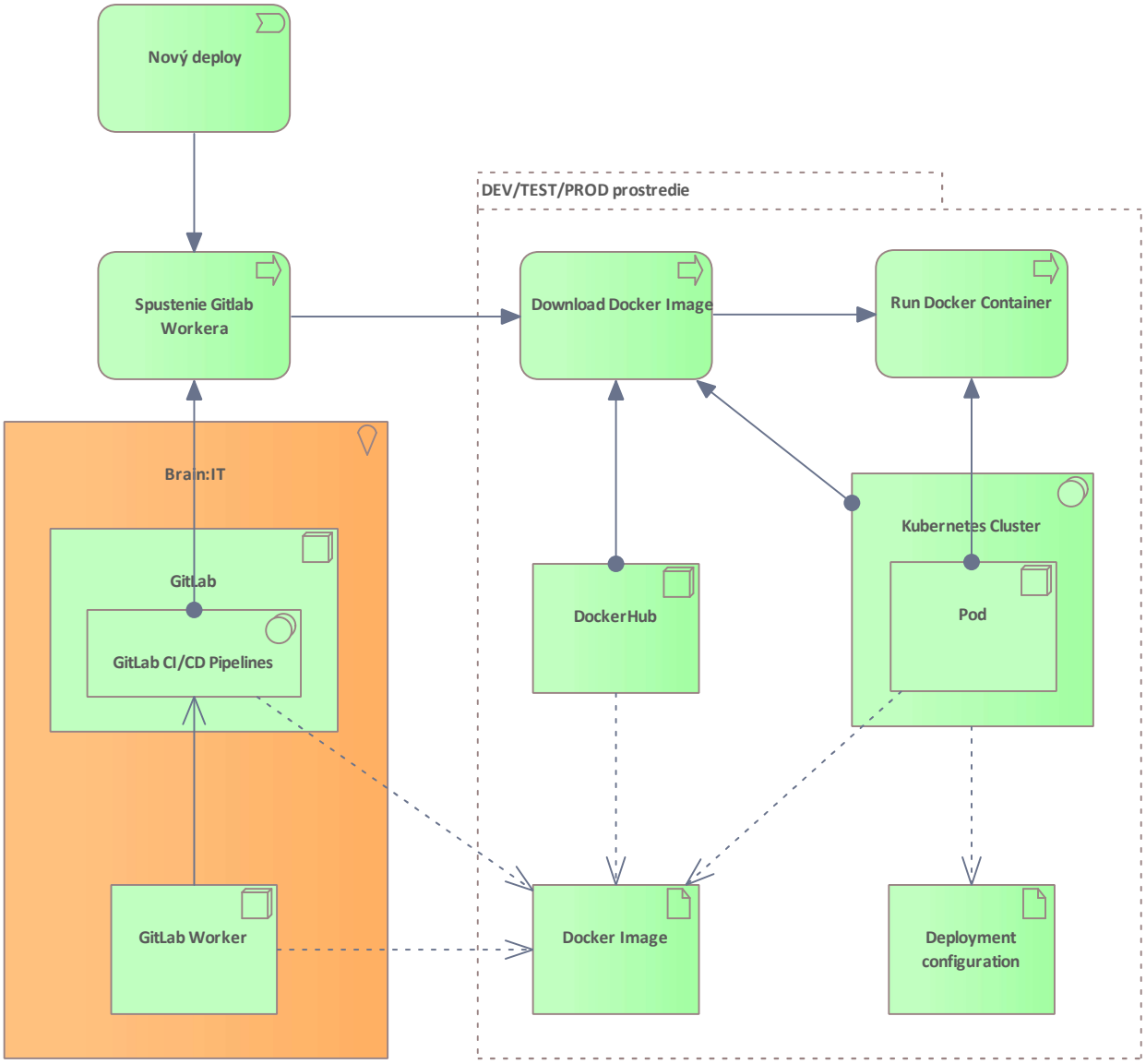


Figure 31 : Technology Viewpoint standard CI/CD DEV/TEST/PROD

Description of the elements used:

Download Docker Image
Process of storing the Docker Image in the Docker Registry

New deploy
Launch of a new deploy process

Run Docker Container
Process of launching a Docker Container

Starting the Gitlab Worker
process of launching the Gitlam Worker to deploy a new version of the package.

DEV/TEST/PROD environment
Represents the internal DEV and TEST environment

Brain:IT
Brain:IT site

DockerHub
Represents the public Docker Registry DockerHub

Deployment configuration
Representation of the Kubernetes Deployment artefact.

Docker Image
Representation of the Docker Image artefact.

GitLab
Represents a GitLab instance.

GitLab Worker
Represents a GitLab Worker instance

Kubernetes Cluster
Represents the Kubernetes Cluster

GitLab CI/CD Pipelines
GitLab CI/CD Pipelines are a series of tasks providing compilation, tests, configuration, deployment and other activities within GitLab.

Pod
Represents a Pod instance in which a new Docker Image is run in the form of a Docker Container

7.5 Technology Viewpoint OpenStack

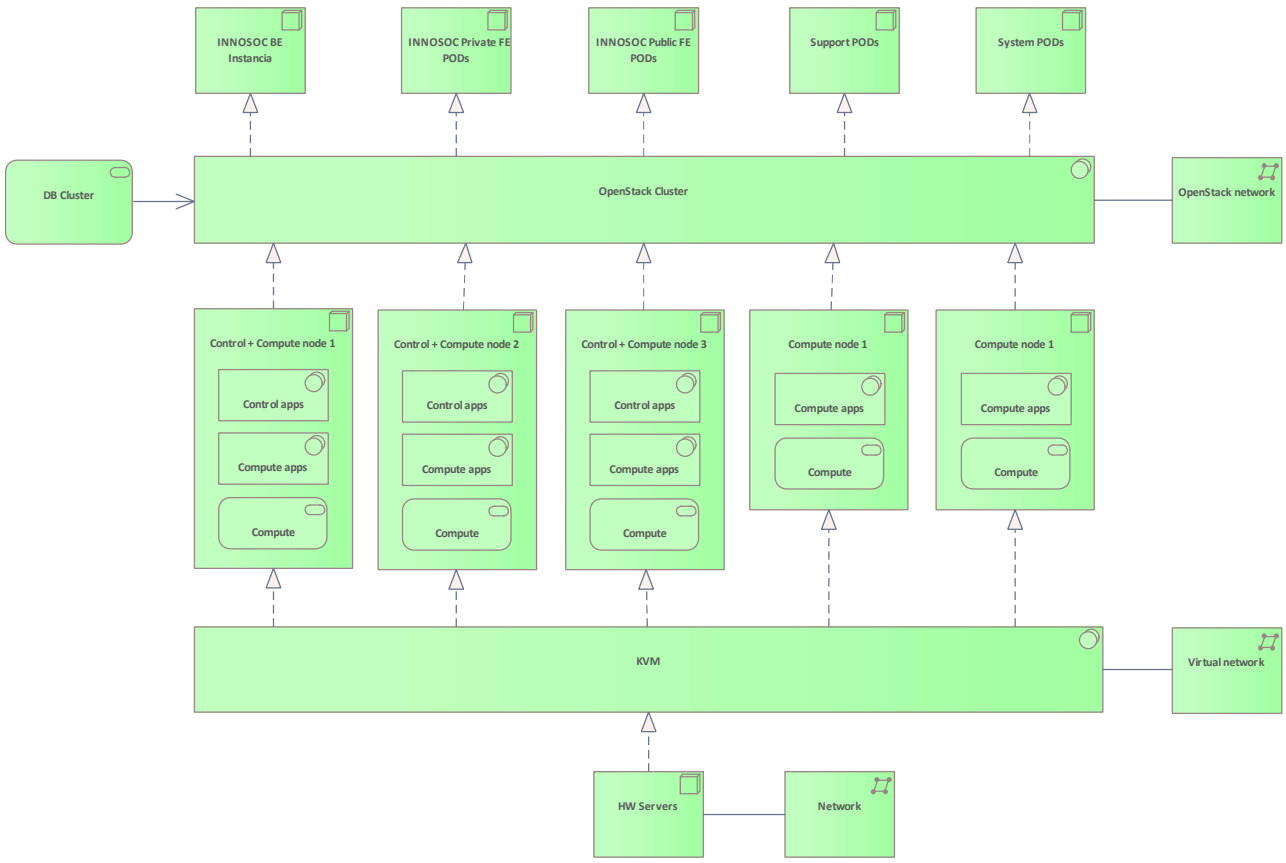


Figure 32 : Technology Viewpoint OpenStack Cluster

Description of the elements used:

DB Cluster
Represents the clustered relational database service

Compute node 1
Represents a Compute node of the OpenStack cluster

Compute node 1
Represents a Compute node of the OpenStack cluster

Control + Compute node 1
Represents Control and Compute node 1 of the OpenStack cluster

Control + Compute node 2
Represents Control and Compute node 2 of the OpenStack cluster

Control + Compute node 3
Represents Control and Compute node 3 of the OpenStack cluster

HW Servers
Represents the HW server infrastructure

KVM
Represents a type 1 KVM hypervisor.

Network
Represents the network/subnet of the HW infrastructure

OpenStack Cluster
Represents the OpenStack Cluster

OpenStack network
Represents the network/subnet of the OpenStack cluster, specifically:
Corporate network
OpenStack Tenant Data Network
OpenStack Internal Admin / API Subnet

Support PODs
Represents the PODs of supporting services

System PODs
Represents the PODs of system services

Virtual network
Represents the virtual network/subnet of the hypervisor

INNOSOC BE Instance
Represents an OpenStack VM instance or a container

INNOSOC Private FE PODs
Represents the CNM FE Private PODs

INNOSOC Public FE PODs
Represents the CNM FE Public PODs

Compute
Represents the role and service of the Compute node

Compute apps
Represents the set of Compute applications

Control apps
Represents the set of applications for the Control node required for managing the cluster

Compute
Represents the role and service of the Compute node

Compute apps
Represents the set of Compute applications

Compute
Represents the role and service of the Compute node

Compute apps
Represents the set of Compute applications

Control apps
Represents the set of applications for the Control node required for managing the cluster

Compute
Represents the role and service of the Compute node

Compute apps
Represents the set of Compute applications

Control apps
Represents the set of applications for the Control node required for managing the cluster

Compute
Represents the role and service of the Compute node

Compute apps
Represents the set of Compute applications

7.6 Technology Viewpoint OpenStack External

7.7 Technology Viewpoint Reporting

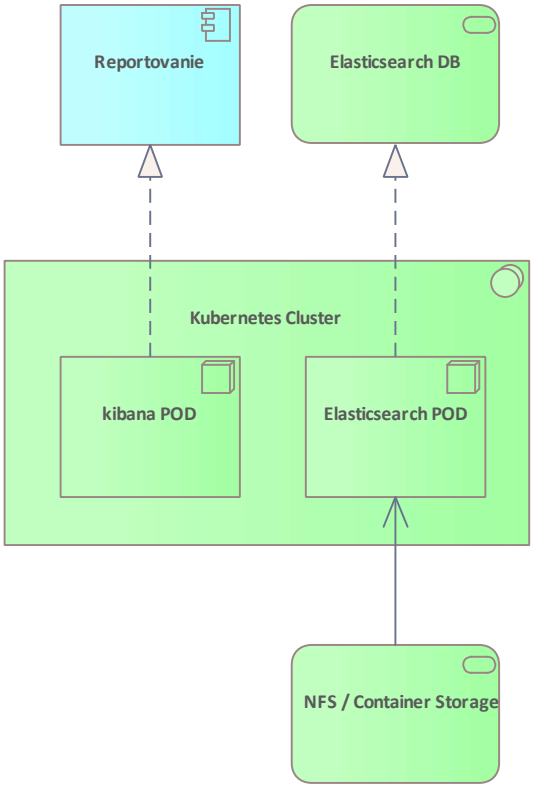


Figure 33 : Technology Usage Viewpoint Reporting alt 1

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 containerized version of the NoSQL database for Elasticsearch.

Description of the elements used:

Reporting
The reporting component within the SOCaaS architecture serves as the central means for aggregating, visualizing and distributing 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 organization or clients. Such a reporting system must cover:
• Generation of metrics for incidents, alerts, SLAs and KPIs,
• Overviews of trends over time and assessment 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 the NFS service or another Container Storage

Kubernetes Cluster
Represents the Kubernetes Cluster

Elasticsearch DB
Represents the technology service of the NoSQL Elasticsearch database.

Elasticsearch POD
Represents the Elasticsearch POD for the NoSQL database.

kibana POD
Represents the Kibana POD for Reporting.

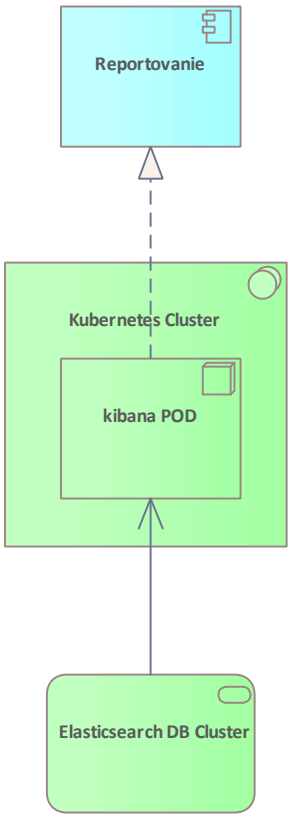


Figure 34 : Technology Usage Viewpoint Reporting alt 2

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 centralized DB cluster.

Description of the elements used:

Reporting
The reporting component within the SOCaaS architecture serves as the central means for aggregating, visualizing and distributing 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 organization or clients. Such a reporting system must cover:
• Generation of metrics for incidents, alerts, SLAs and KPIs,
• Overviews of trends over time and assessment 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 the Kubernetes Cluster

Elasticsearch DB Cluster
Represents the technology service of the NoSQL Elasticsearch database in the form of a cluster.

kibana POD
Represents the Kibana POD for Reporting.

7.8 Technology Viewpoint Vulnerability Scanner

7.9 Technology Viewpoint Ticketing system / Incident registration and management

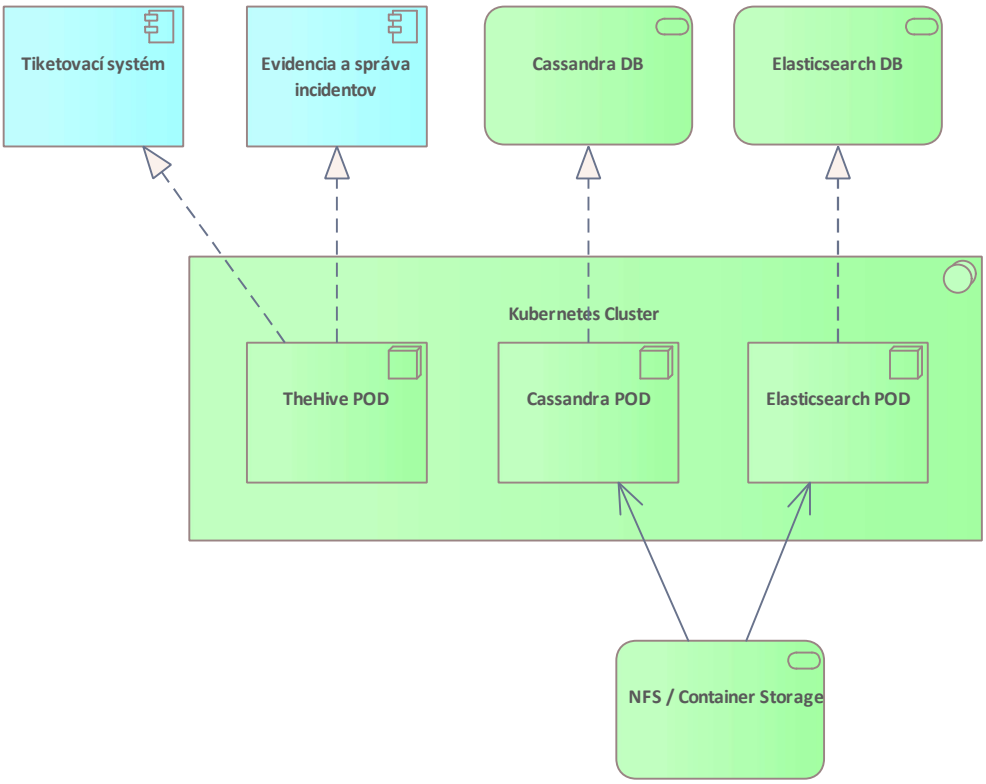


Figure 35 : Technology Usage Viewpoint Ticketing System / Incident Recording and Management alt 1

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 a containerized version 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 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 with 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 organizational 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 the NFS service or another Container Storage

Kubernetes Cluster
Represents the Kubernetes Cluster

Cassandra DB
Represents the technology service of the NoSQL Cassandra database.

Elasticsearch DB
Represents the technology service of the NoSQL Elasticsearch database.

Cassandra POD
Represents the Apache Cassandra POD for the NoSQL database.

Elasticsearch POD
Represents the Elasticsearch POD for the NoSQL database.

TheHive POD
Represents the TheHive POD for asset inventory.

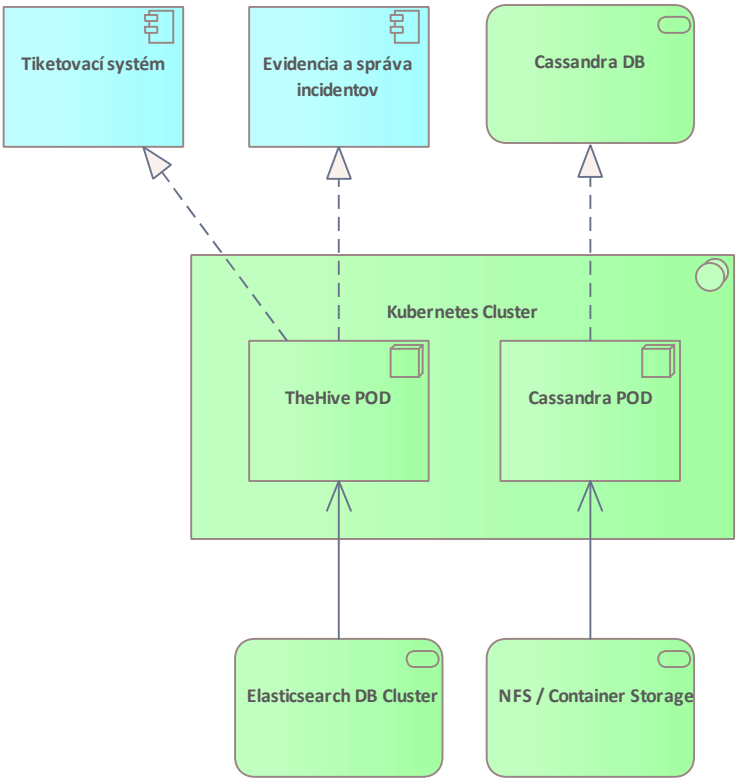


Figure 36 : Technology Usage Viewpoint Ticketing System / Incident Recording and Management alt 2

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 centralized DB cluster. The design envisages a containerized 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 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 with 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 organizational 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 the NFS service or another Container Storage

Kubernetes Cluster
Represents the Kubernetes Cluster

Cassandra DB
Represents the technology service of the NoSQL Cassandra database.

Elasticsearch DB Cluster

Represents the technology service of the NoSQL Elasticsearch database in the form of a cluster.

Cassandra POD
Represents the Apache Cassandra POD for the NoSQL database.

TheHive POD
Represents the TheHive POD for asset inventory.

7.10 Technology Viewpoint Knowledge Portal for the SOC Team

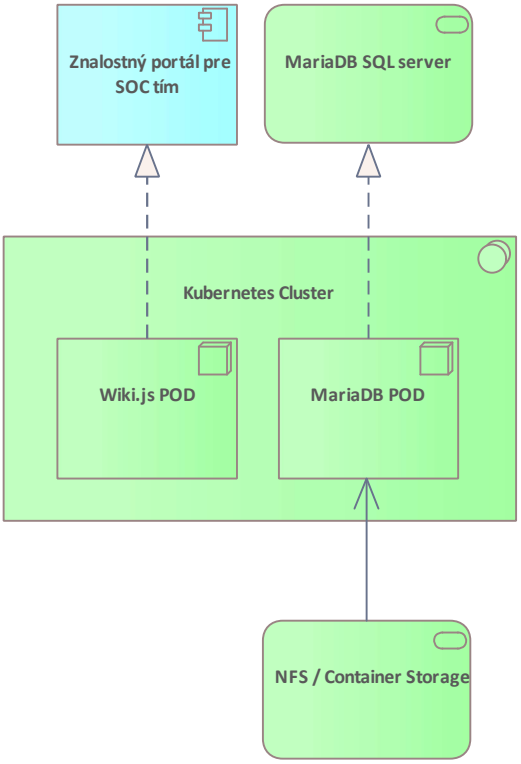


Figure 37 : Technology Usage Viewpoint Knowledge Portal for the SOC Team alt 1

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 containerized version of the SQL server.

Description of the elements used:

Knowledge portal for the SOC team
A 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 the standardization 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 the NFS service or another Container Storage

Kubernetes Cluster
Represents the Kubernetes Cluster

MariaDB SQL server
Represents the technology service of the relational MariaDB SLQ database.

MariaDB POD
Represents the MariaDB POD for the relational SQL DB.

Wiki.js POD
Represents the Wiki.js POD for the Knowledge portal for the SOC team.

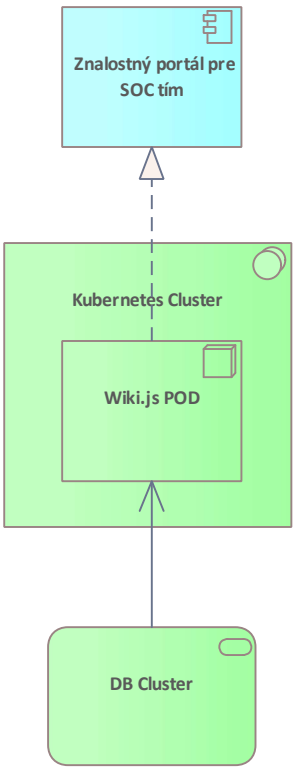


Figure 38 : Technology Usage Viewpoint Knowledge Portal for the SOC Team alt 2

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 centralized DB cluster.

Description of the elements used:

Knowledge portal for the SOC team

A 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 the standardization 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 the clustered relational database service

Kubernetes Cluster
Represents the Kubernetes Cluster

Wiki.js POD
Represents the Wiki.js POD for the Knowledge portal for the SOC team.

7.11 Technological components of the solution and their relationship to application components

7.11.1 Management of access, accounts and keys of operating environments

7.11.2 Containerization and orchestration of cloud services

7.11.3 Specification of the use of government cloud services

7.12 Description of the test environment

Given in chapter 7.1.

7.13 Description of the production environment

Given in chapter 7.1.

8 Service level (SLA) and performance requirements

9 Software licences and source code

List of SW licences that will be used during development:

No.	Name of the SW used	Type of the SW used	SW licence
....			

10 Ensuring availability, backup and recovery of the solution

The description of the method for ENSURING AVAILABILITY, Backup and recovery of the solution is precisely set out and addressed in the valid **Contract for the Provision of Information System Support and Maintenance Services**, which takes effect on the date of acceptance of the Final Acceptance Protocol for the delivered work.

10.1 Ensuring availability

The basic pillar of the availability of critical systems in a cloud environment is system clustering, which ensures 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 (HA) of systems by means of 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 restart of the system after a hardware or software crash or a system fault.
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 to repair hardware or network faults whose occurrence caused part of the system to stop functioning, thus preparing it for another potential outage. If the designed "redundancy" of the system were not sufficient to ensure the functioning of the system 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 restore systems to operation using data backups as well.
4. Replication of data to another location for the resilience of the system (computer centre) 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, system configuration, operation records and the databases themselves.

All important central components of the system and infrastructure are deployed in production operation multiple times as needed, so that the failure of any single component does not jeopardize 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

The PostgreSQL database needs to be backed up. It is not necessary to deal with file backup.
The method of backup and recovery is the responsibility of the contracting authority; the recommended backup frequency is once every 12 hours with a retention policy of one week.

10.3 Recovery of the solution

11 Security - addressing security requirements

Secure development principles
The requirements laid down in this document must be met for all components that are used during development and all artefacts that are considered products of development.
The system will be developed in accordance with internationally recognized recommendations and best practices in the field of secure software development. The 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 advisable 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 software development and delivery, in particular:
- introduction of vulnerabilities as a result of poor programming techniques,
- introduction of vulnerabilities as a result of using insecure and unsupported frameworks,
- introduction of undesirable 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 its 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 an appropriately chosen security standard and/or methodology. During the implementation of the proposed solution, the software is verified during development against well-known vulnerabilities and security recommendations, which form part of secure development practice to prevent the occurrence of the most common security risks for web applications.

By components used during development we mean in particular:

- workstations used in development
- virtualization environment
- servers used in development (development, test)
- network environment used in development, including home networks
- printers and other active elements
- physical environment in which development takes place, including home offices
- supporting 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 security best practice principles, including secure administration, configuration, patching and hardening of workstations and servers, anti-malware protection, segregation of physical and logical environments, account policies, credential management 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.
Endpoints 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.
The computer will not automatically connect to Wi-Fi networks other than the protected networks used in development. The Wi-Fi SSID to which a developer's computer is connected must have access controlled by WPA2 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 devices (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 that filters dangerous and malicious content. All communication from computers used for development that are located outside protected internal networks (for example, when working from home) 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 data centre and server room premises.

All configurations must be protected against unauthorized 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 except for disaster recovery. All accounts must be protected by SSH keys or, in exceptional cases where necessary, by a sufficiently strong password or passphrase with a high level of entropy.
On all systems used for development, only the minimum necessary set of software required for the specific activity may be installed.

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 central user management and identity management, known as IAM (Identity and Access Management), using oAuth/SAML with the possible use of 2FA, on the basis of which 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 for reading only, not for writing (unless the service and process themselves require it).

The web content and all logs generated by it will be located on a separate hard disk or on a logical partition other than that of the OS and the web server. In the case of containerization, the logging method will be adapted in line with the orchestration platform concerned.
To meet the security requirements associated with the development of the project, a suitable tool will be used that provides continuous code quality checks, validation of the secure life cycle and checks that detect bugs, duplicated code and security holes.

Securing communication

All communication channels will be known and described. The individual system components will be separated according to layers that perform a specific role.
The basic layers are:
- Access layer
- Application layer
- Database layer

The individual layers will be located in separate, logically isolated network segments or independently 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 orchestration platform rules, and in addition monitoring and filtering will be possible.
The administration and monitoring interfaces of network devices and servers are not accessible directly from the external network, the internet. The 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 ensured by a valid, trusted certificate issued for the domain on which the portal in question is available. For this purpose, 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 credentials) must be transmitted exclusively via an encrypted channel.
The web server will support TLS 1.2, TLS 1.3 and later. By default, it will not support the old versions SSLv2, SSLv3, TLS 1.0 and TLS 1.1, unless otherwise required.
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 the elimination of 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 central user management based on oAuth/SAML and, where applicable, 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 utilization 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 via a secure 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 storage 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 of 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, the 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 central user management and identity management, known as IAM (Identity and Access Management), using oAuth/SAML with the possible use of 2FA, on the basis of which 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 will be performed within the solution.

12.1 Data migration procedure

12.2 Data migration verification

13 Implementation and deployment schedule, dependencies

13.1 Project phases and timeline - consolidated overview

13.2 Dependencies and solution design

13.3 Requirements for the Testing and Production Deployment phases

The work will be implemented as a project managed in accordance with the PRINCE2 methodology. The delivery of the solution will go through the basic project stages within the meaning of Decree No. 401/2023 on project management. The outputs of each stage will be approved by the Project Steering Committee.

14 Open issues

15 Annexes

List of DNR annexes:

1. Catalogue of requirements and mapping to their implementation in the proposed solution
2. List of figures