PROJECT

Multi-tenant cybersecurity operations centre designed
as an open cloud service with machine learning elements

V12 Business architecture

KPB4 – Solution architecture

type – documentation

month of submission – M22

Funded by the European Union
NextGenerationEU

RECOVERY AND RESILIENCE PLAN

GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC

Contents

1. Introduction 3
2. Application architecture 3
1.1. Aplication Viewpoints DELETE OLD 7
3. technological architecture 14
1.2. Technology Viewpoint Asset inventory 14
1.3. Technology Viewpoint Kubernetes 17
1.4. Technology Viewpoint OpenStack 32
1.5. Technology Viewpoint Reporting 35
1.6. Technology Viewpoint Ticketing system / Incident recording and management 38
1.7. Technology Viewpoint Knowledge portal for the SOC team 41
4. Conclusion 44

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

1. INTRODUCTION

The aim of the project is to develop a technologically modern, open and scalable solution that will be able to support organizations in effectively handling cyber threats. The project responds to the needs of small and medium-sized organizations in particular, which often lack sufficient expert capacity or the budget to build their own security centre. This document 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.

2. 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 presents the application architecture diagram with descriptions of the objects, which were generated from EA as output.

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

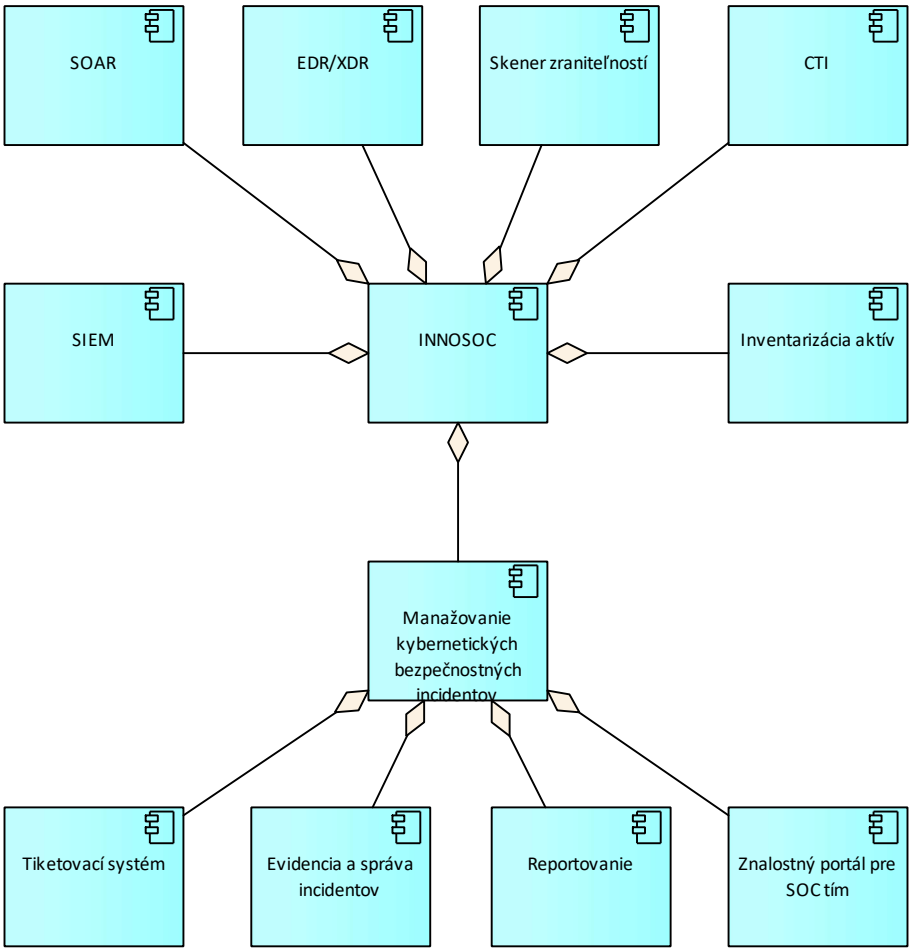


Figure 1 : Aplication Viewpoints Components

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.

1.1. Aplication Viewpoints DELETE OLD

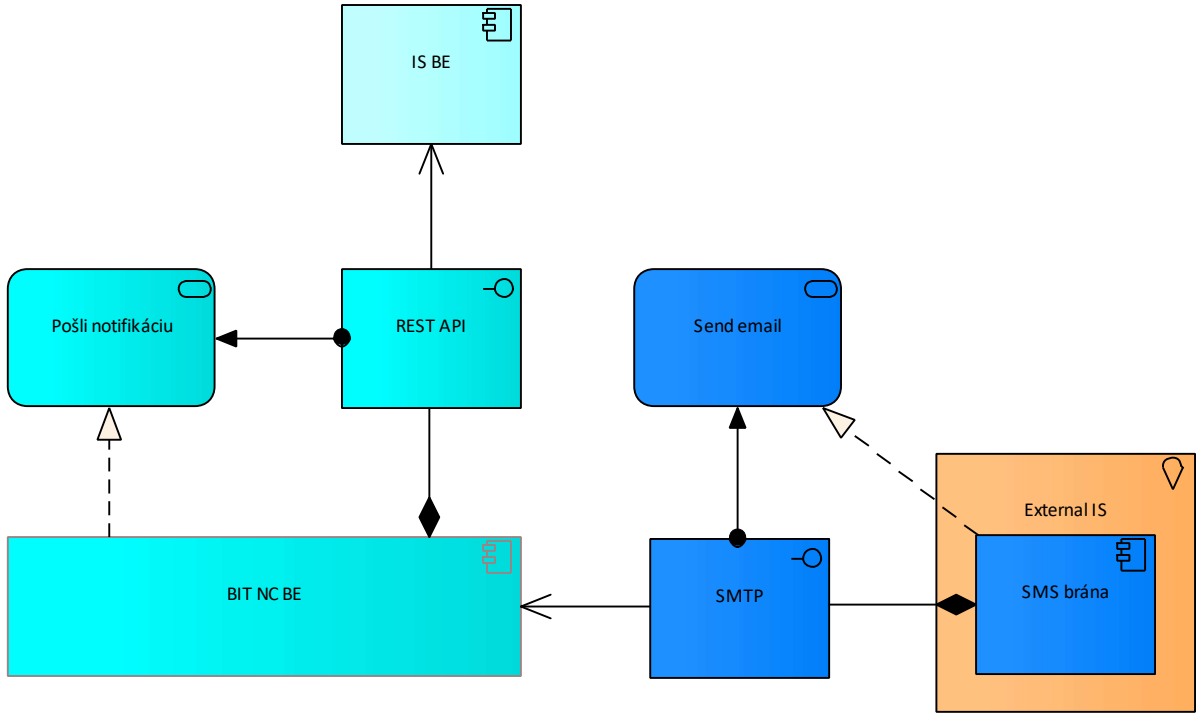


Figure 2 : Application Cooperation Viewpoint Empire and BIT NC

The diagram presents the integration of the IS BE with the notification-sending services of the BIT NC module using REST API, and the integration of BIT NC with the mail server using SMTP and its e-mail sending service.

Description of the elements used:

Send notification
Represents the HTTP POST, GET, PUT and DELETE methods of the service for sending a notification.

Send email
Service for sending an e-mail.

External IS
External information systems.

BIT NC BE
BIT NC BE represents the BE module of BIT Notify Center, which provides central management of notifications together with the corresponding BIT NC FE.

IS BE
The main BE component of the IS, which uses the services of NFQES Enterprise IAM and BIT

NC and provides an interface for the IS FE module.

REST API
Represents the REST API of the BIT NC module.

SMTP
SMTP service of the mail server. Port 587 for Secure SMTP, alternatively 465, and in the case of an internal mail server port 25 is also possible (not recommended).

SMS gateway
Represents an e-mail server connected using a username and password via SMTP.

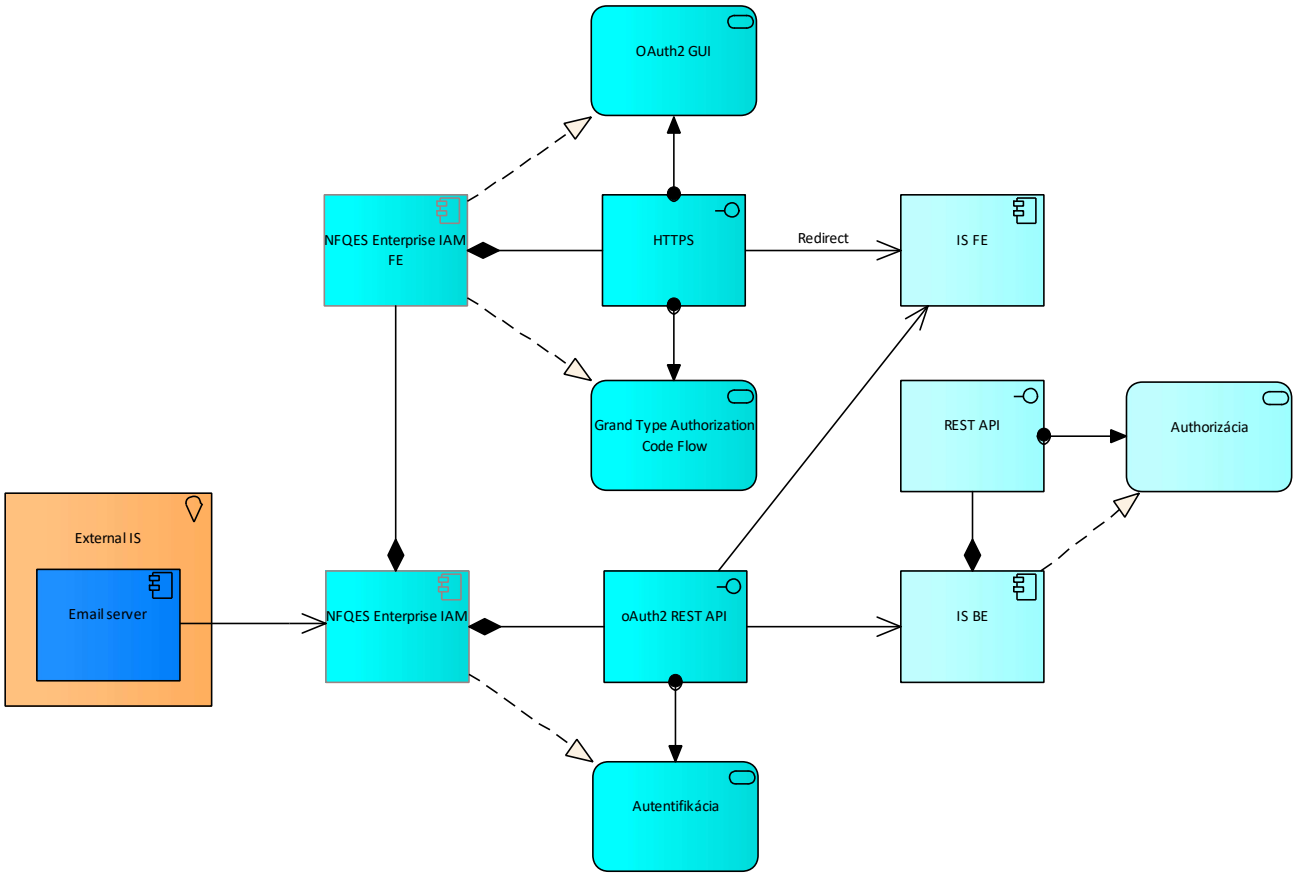


Figure 3 : Application Cooperation Viewpoint IS and BIT IAM

The diagram presents the mutual integration between the IS being built and the existing NFQES Enterprise IAM module.

Description of the elements used:

Authentication
Represents the HTTP POST, GET, PUT and DELETE methods of the service for Authentication.

Authorization
Represents the authorization application service.

Grand Type Authorization Code Flow
Grand Type - Authorization Code Flow.
Note - The second flow, which we are not implementing, is the Pasword Flow (signing in directly from the app - the client id and password must not leak).

OAuth2 GUI
Represents the HTTP pages of NFQES Enterprise IAM.

External IS
External information systems.

IS BE
The main BE component of the IS, which uses the services of NFQES Enterprise IAM and BIT NC and provides an interface for the IS FE module.

IS FE
The main FE component of the IS, which uses the services of IS BE and NFQES Enterprise IAM.

NFQES Enterprise IAM
NFQES Enterprise IAM represents the BIT Identity Access Management module, which provides authentication and role management with the corresponding NFQES Enterprise IAM FE.

NFQES Enterprise IAM FE
IAM FE represents the FE submodule of the NFQES Enterprise IAM FE module, which provides the basic screens required for the authentication flow.

HTTPS
Represents the HTTPS interface of the NFQES Enterprise IAM module.

oAuth2 REST API
Represents the oAuth2 REST API of the NFQES Enterprise IAM module.

REST API
Represents the REST API of the IS BE module.

Email server
Represents the accounting program into which data will be imported in an agreed format. This is a process integration.

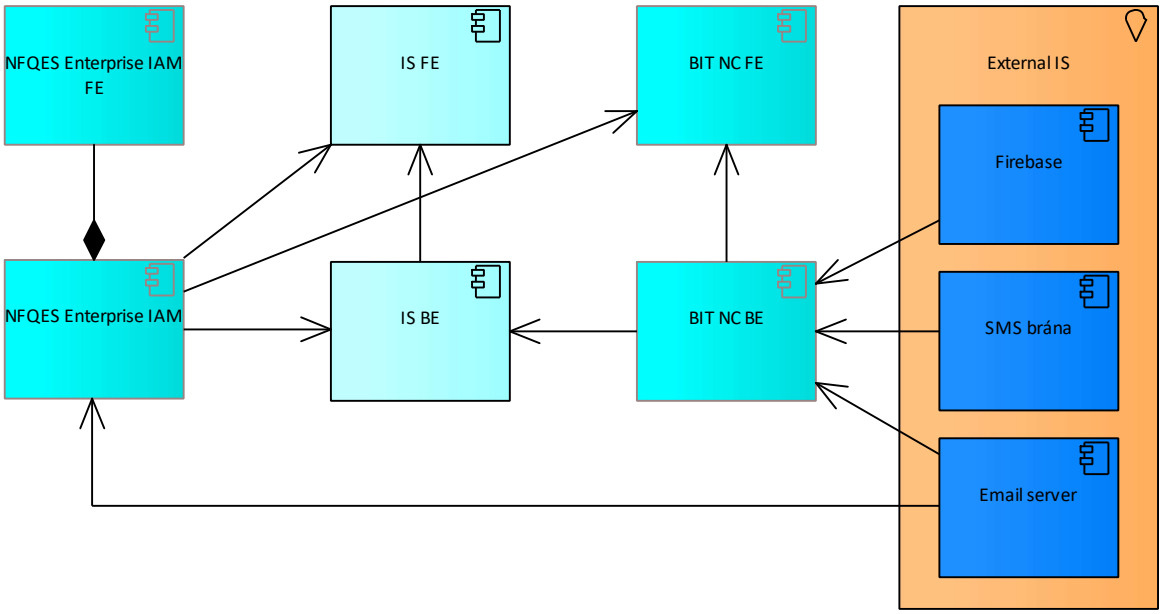


Figure 4 : Application Structure Viewpoint

The diagram presents a high-level architectural scheme of the interconnection of components.

Description of the elements used:

External IS
External information systems.

BIT NC BE
BIT NC BE represents the BE module of BIT Notify Center, which provides central management of notifications together with the corresponding BIT NC FE.

BIT NC FE
BIT NC FE represents a standalone FE module using the BIT NC module.

IS BE
The main BE component of the IS, which uses the services of NFQES Enterprise IAM and BIT NC and provides an interface for the IS FE module.

IS FE
The main FE component of the IS, which uses the services of IS BE and NFQES Enterprise IAM.

NFQES Enterprise IAM
NFQES Enterprise IAM represents the BIT Identity Access Management module, which provides authentication and role management with the corresponding NFQES Enterprise IAM FE.

NFQES Enterprise IAM FE
IAM FE represents the FE submodule of the NFQES Enterprise IAM FE module, which provides the basic screens required for the authentication flow.

Email server
Represents the accounting program into which data will be imported in an agreed format. This is a process integration.

Firebase
Represents the accounting program into which data will be imported in an agreed format. This is a process integration.

SMS gateway
Represents an e-mail server connected using a username and password via SMTP.

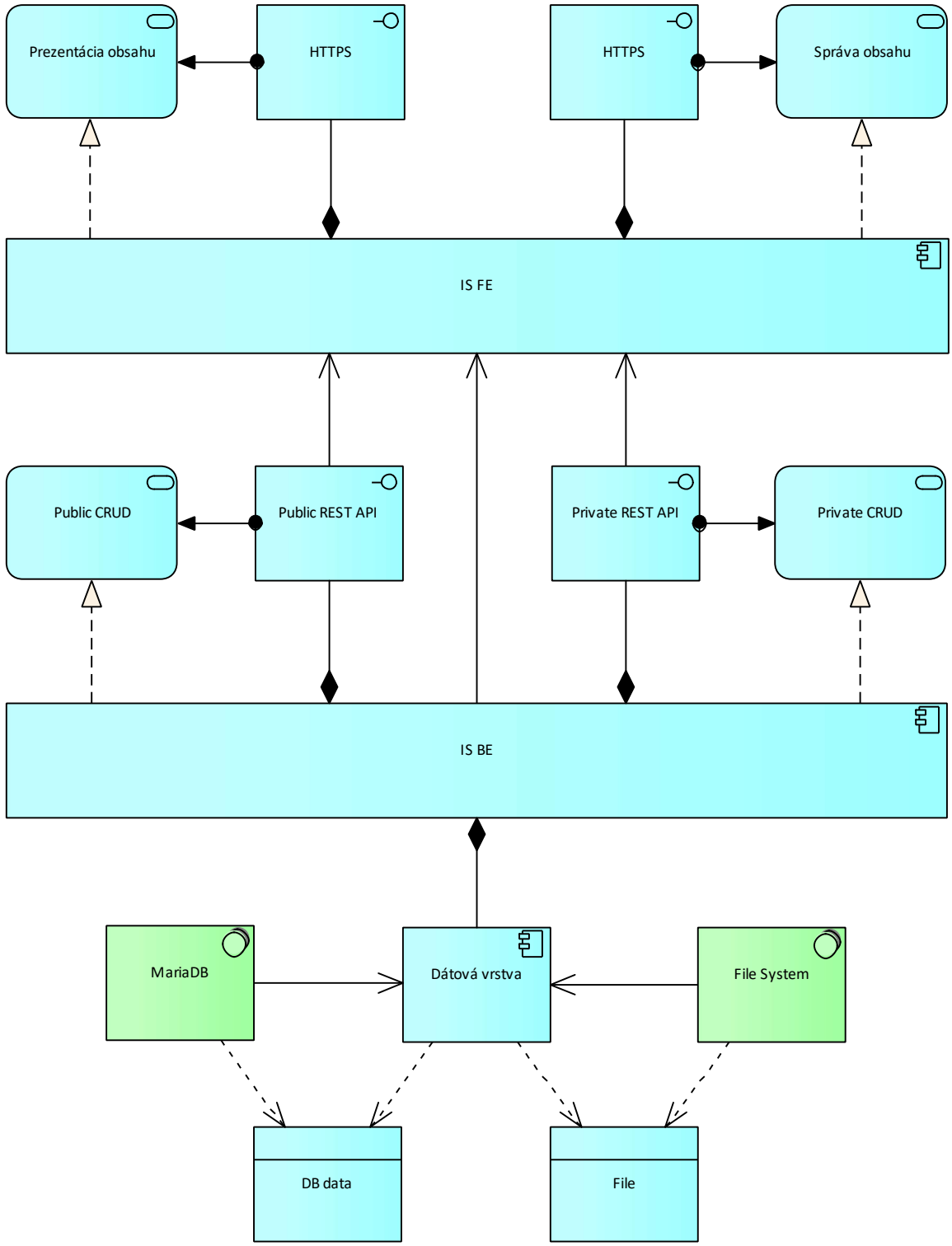


Figure 5 : Implementation and Deployment Viewpoint with bussines objects - Empire

The diagram presents the application architecture of the IS and its interconnection by means of REAST API interfaces. It also depicts the use of the persistence layer in the form of the relational SQL database MariaDB and the file system.

Description of the elements used:

Content presentation
Represents the content presentation application service.

Private CRUD
Represents the private HTTP POST, GET, PUT and DELETE methods of the service.

Public CRUD
Represents the public HTTP POST, GET, PUT and DELETE methods of the service.

Content management
Represents the content management application service.

DB data
Represents database data.

File
A standard file representing any required data such as images, documents and others.

File System
Local file system for standard work with files.

MariaDB
A relational SQL database providing the persistence layer for data other than files.

Data layer
Represents the component of the Empire BE module that performs operations on the persistence layer in the form of the file system, the database and possibly others.

IS BE
The main BE component of the IS, which uses the services of NFQES Enterprise IAM and BIT NC and provides an interface for the IS FE module.

IS FE
The main FE component of the IS, which uses the services of IS BE and NFQES Enterprise IAM.

HTTPS
Represents the HTTPS interface for accessing content.

HTTPS
Represents the HTTPS interface for content management.

Private REST API
Represents the private REST API of the WCM module.

Public REST API
Represents the public REST API of the WCM module.

3. 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.

1.2. Technology Viewpoint Asset Inventory

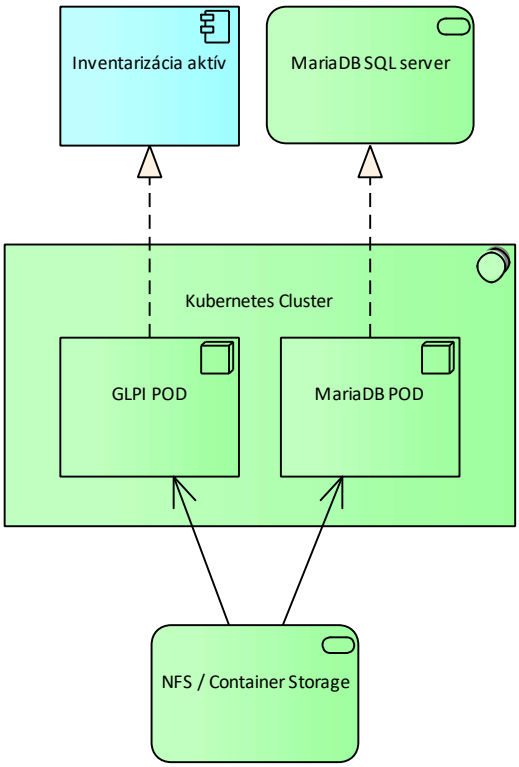


Figure 6 : Technology Usage Viewpoint Asset Inventory alt 1

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.

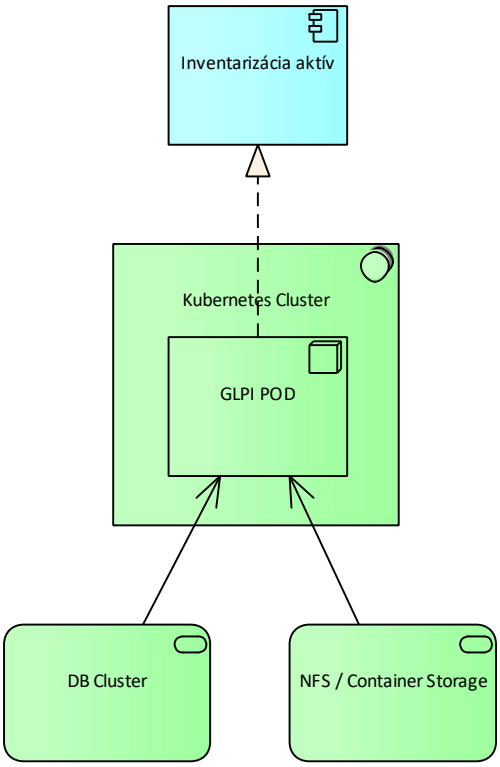


Figure 7 : Technology Usage Viewpoint Asset Inventory alt 2

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.

1.3. Technology Viewpoint Kubernetes

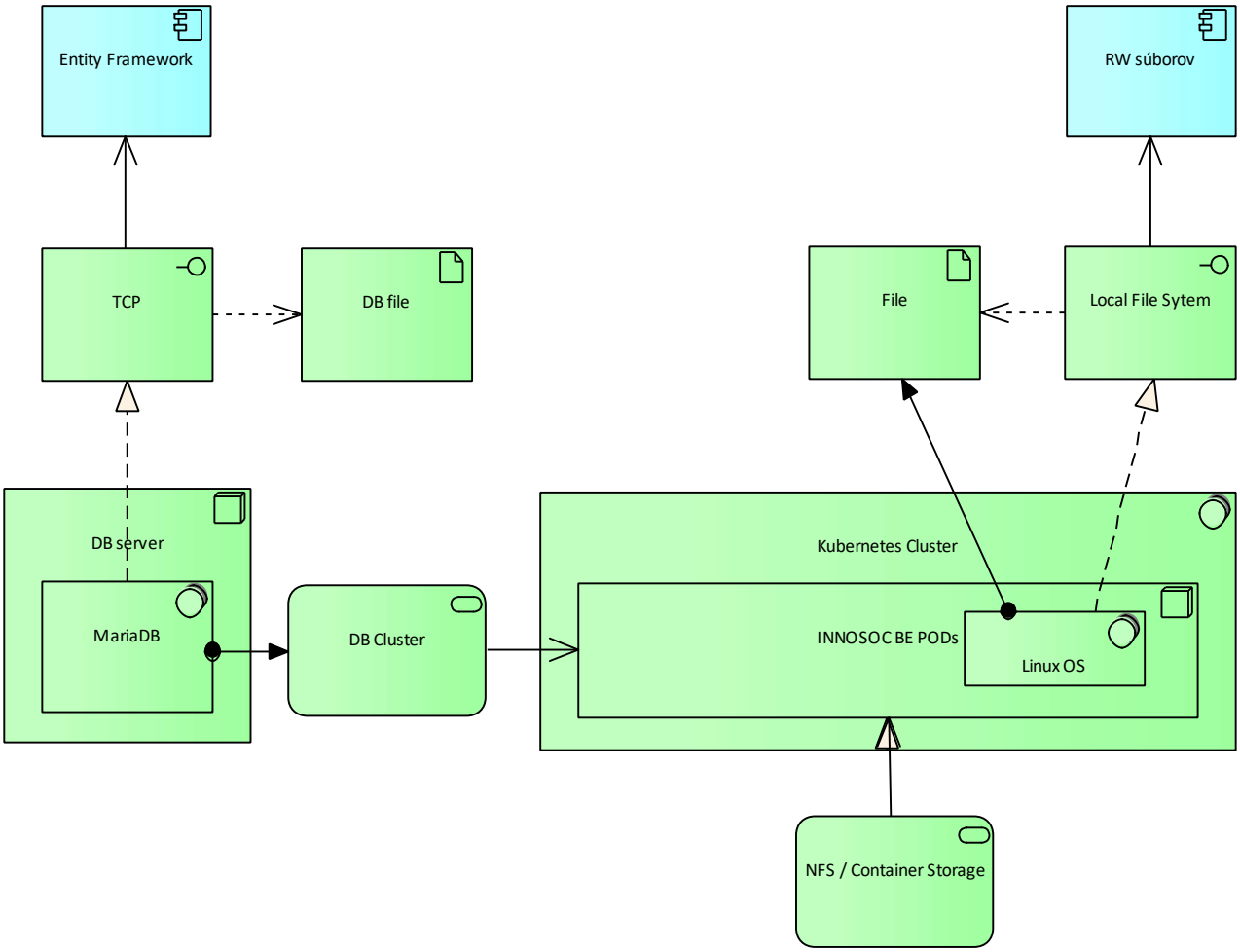


Figure 8 : Technology Usage Viewpoint CNM BE, DB and File System

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

MariaDB
Represents the MariaDB relational database

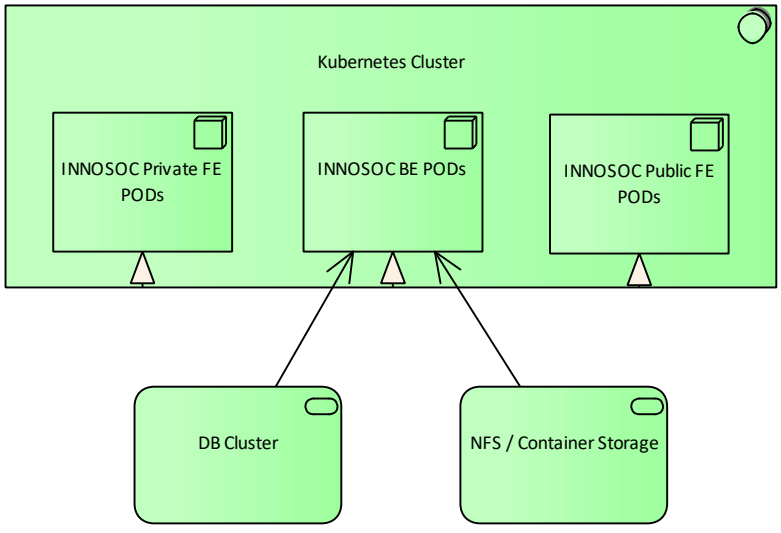


Figure 9 : Technology Usage Viewpoint CNM and Containers maping

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

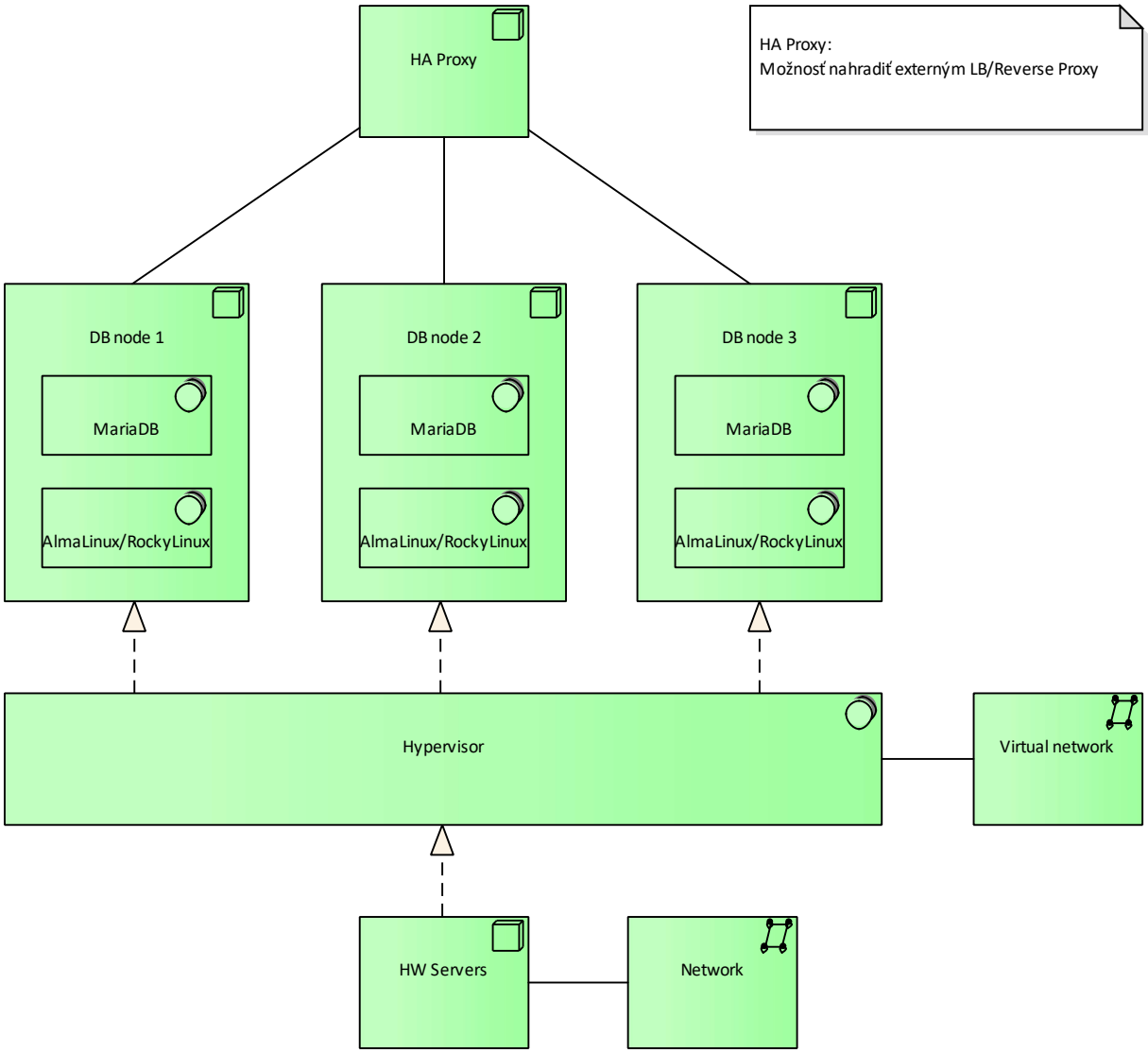


Figure 10 : Technology Viewpoint DB Cluster

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

MariaDB
Represents the MariaDB relational database

AlmaLinux/RockyLinux
Represents the AlmaLinux/RockyLinux Linux distribution

MariaDB
Represents the MariaDB relational database

AlmaLinux/RockyLinux
Represents the AlmaLinux/RockyLinux Linux distribution

MariaDB
Represents the MariaDB relational database

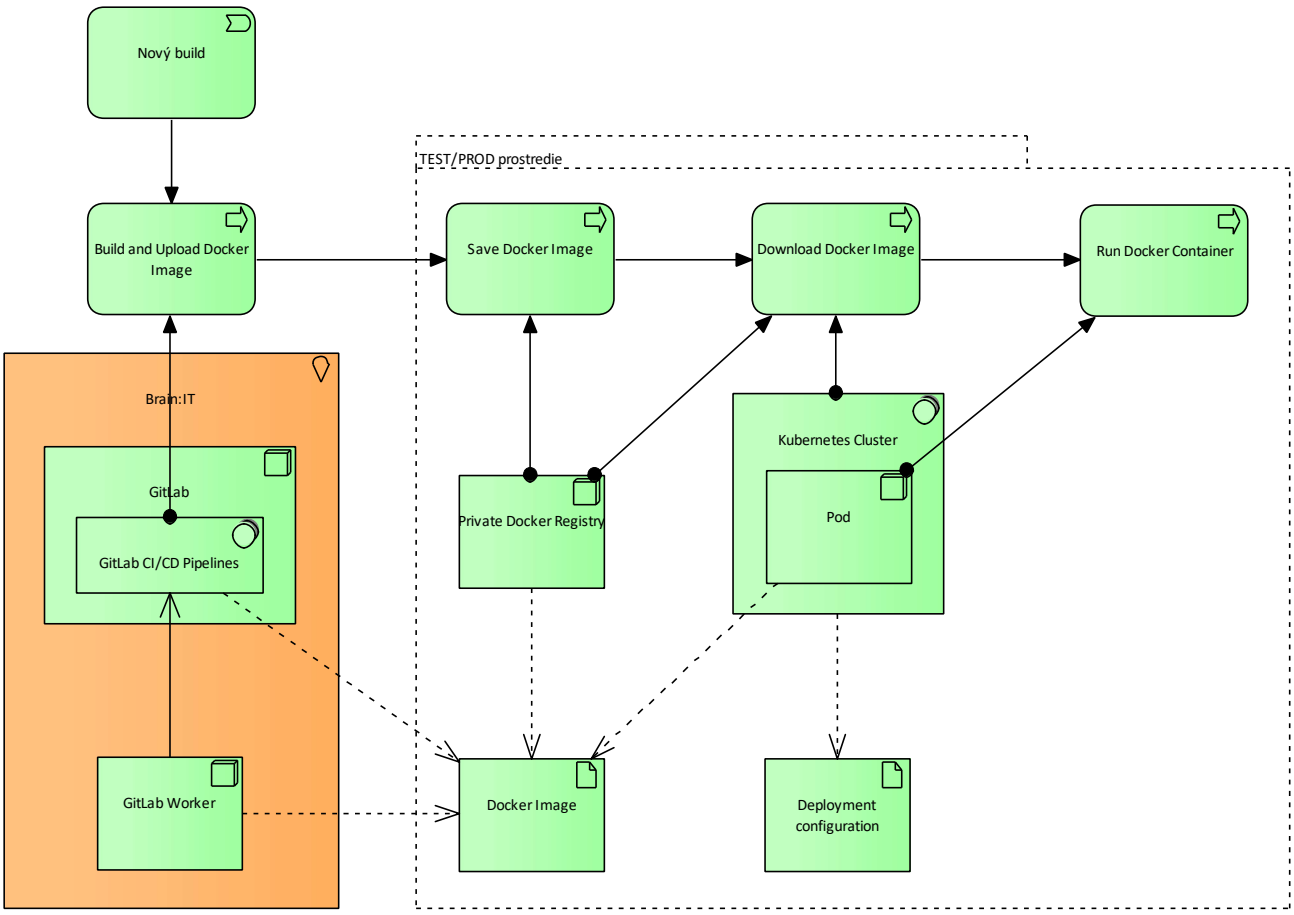


Figure 11 : Technology Viewpoint Development CI/CD TEST/PROD

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

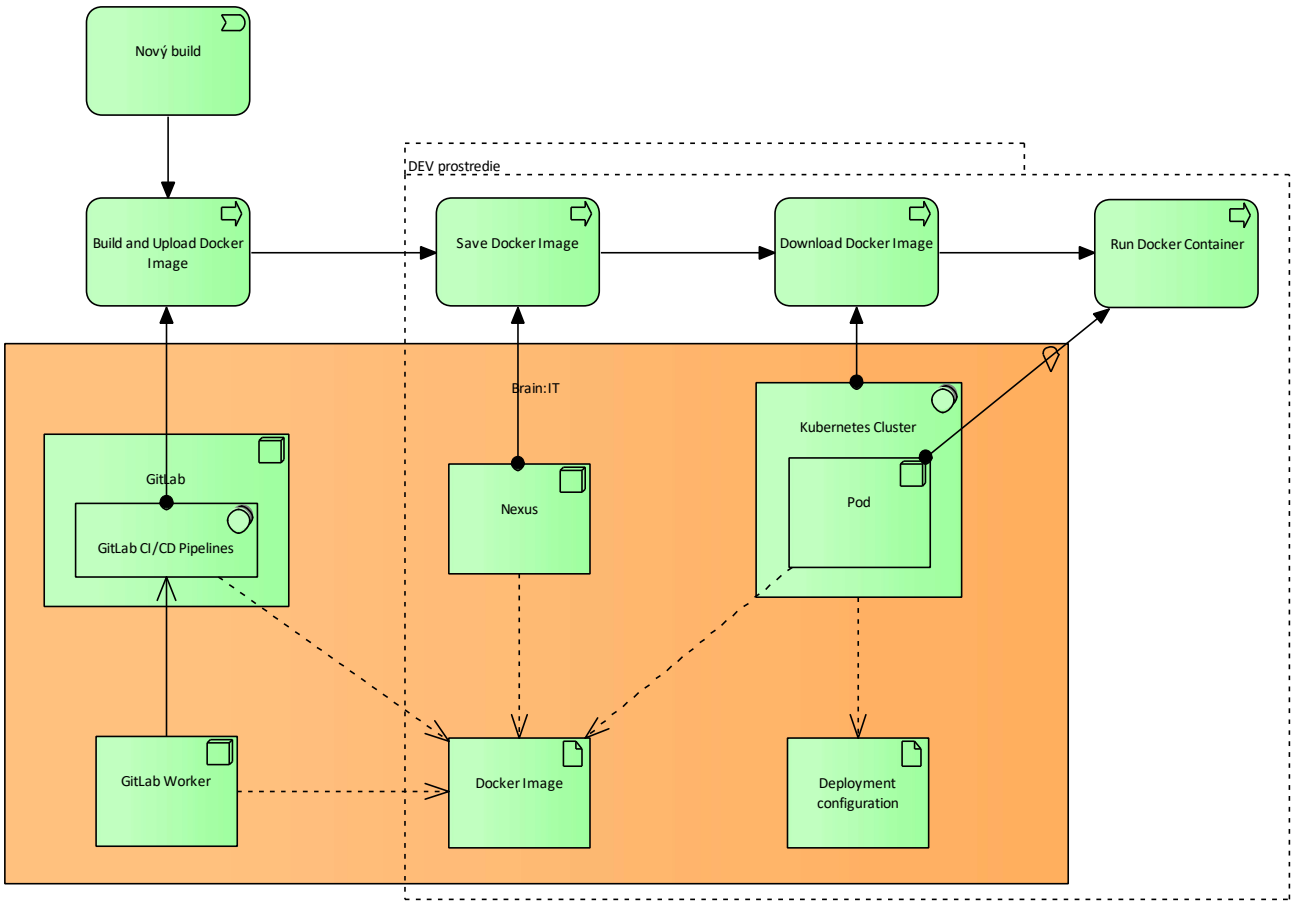


Figure 12 : Technology Viewpoint Developmnent CI/CD DEV

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

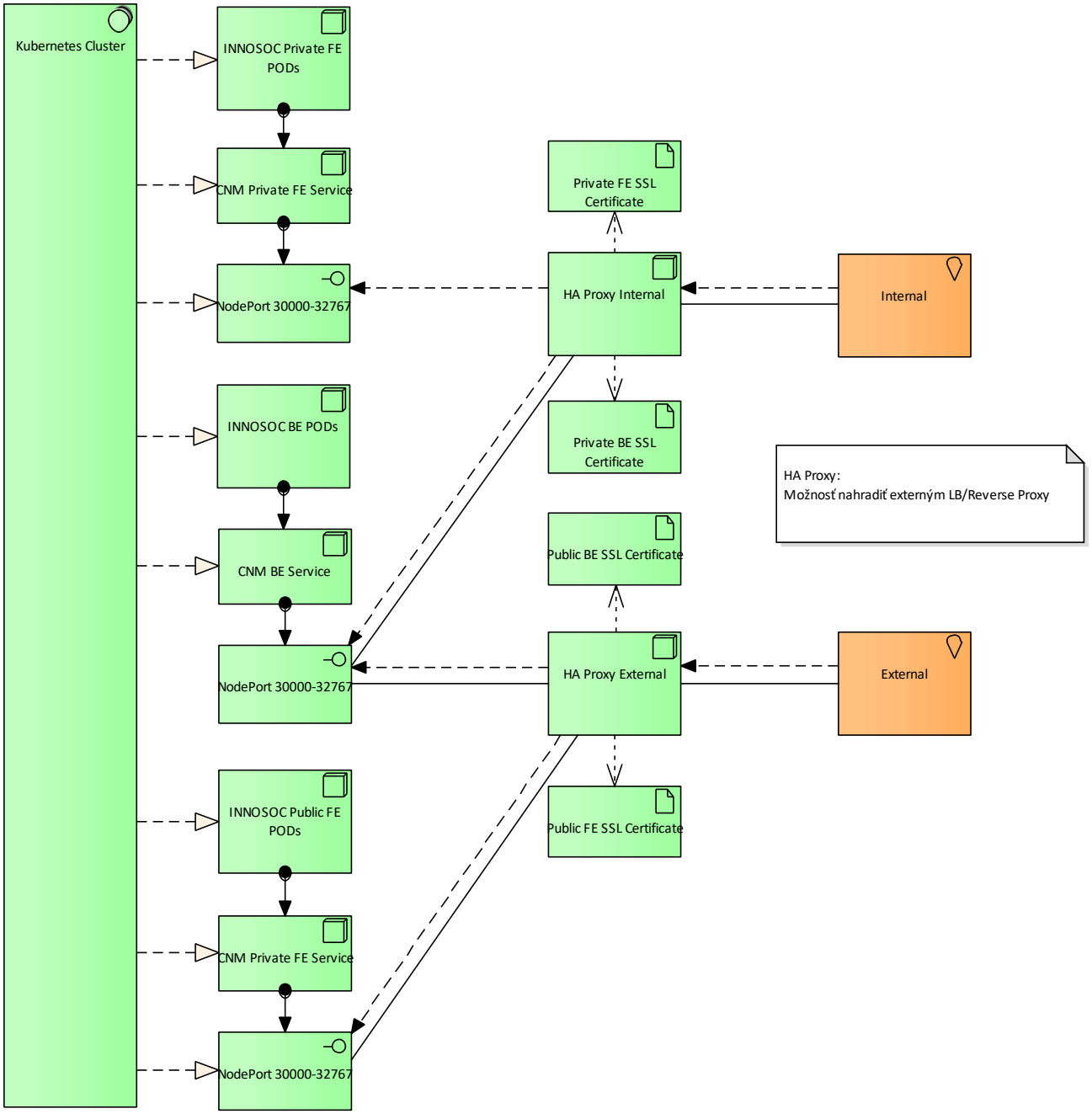


Figure 13 : Technology Viewpoint Interconnect

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 the 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: Option to replace with 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

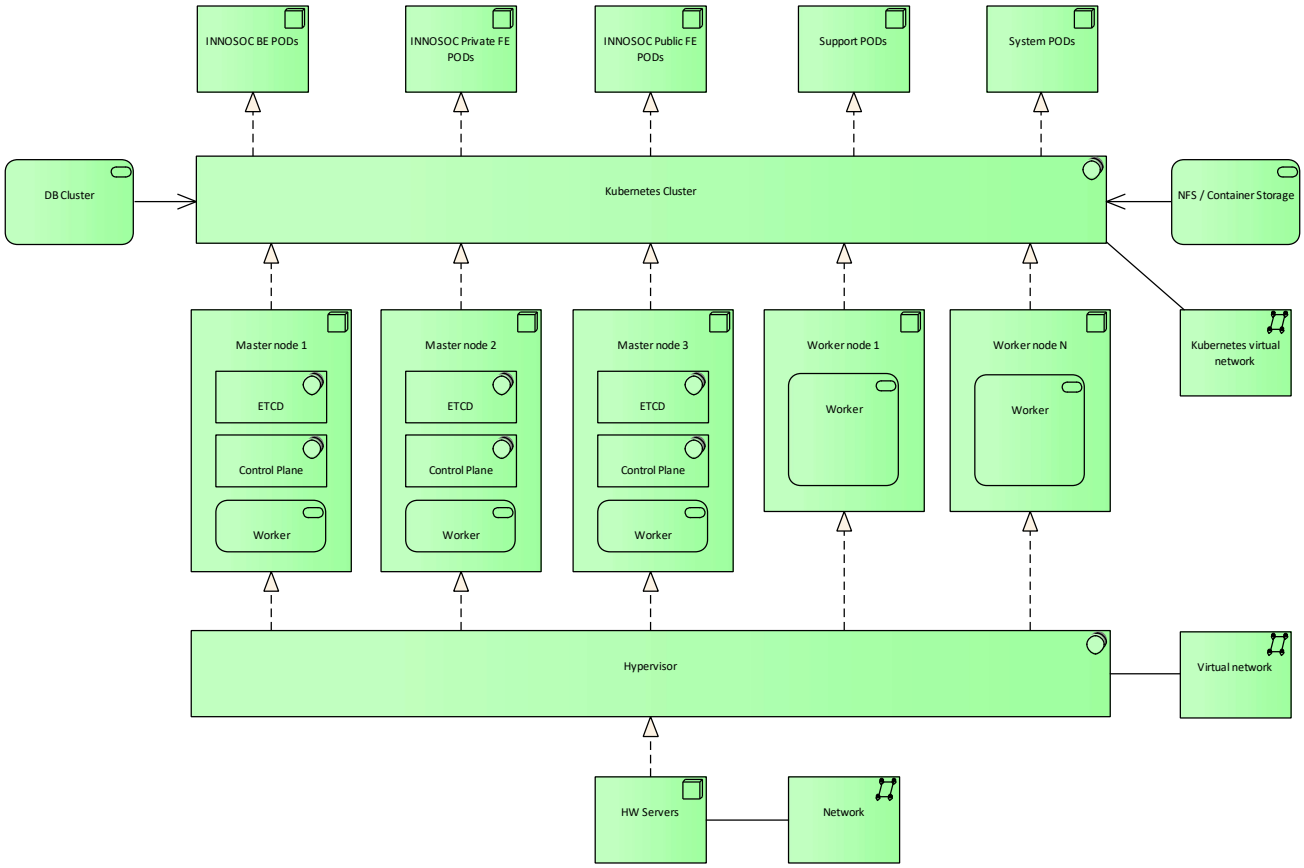


Figure 14 : Technology Viewpoint Kubernetes Cluster

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

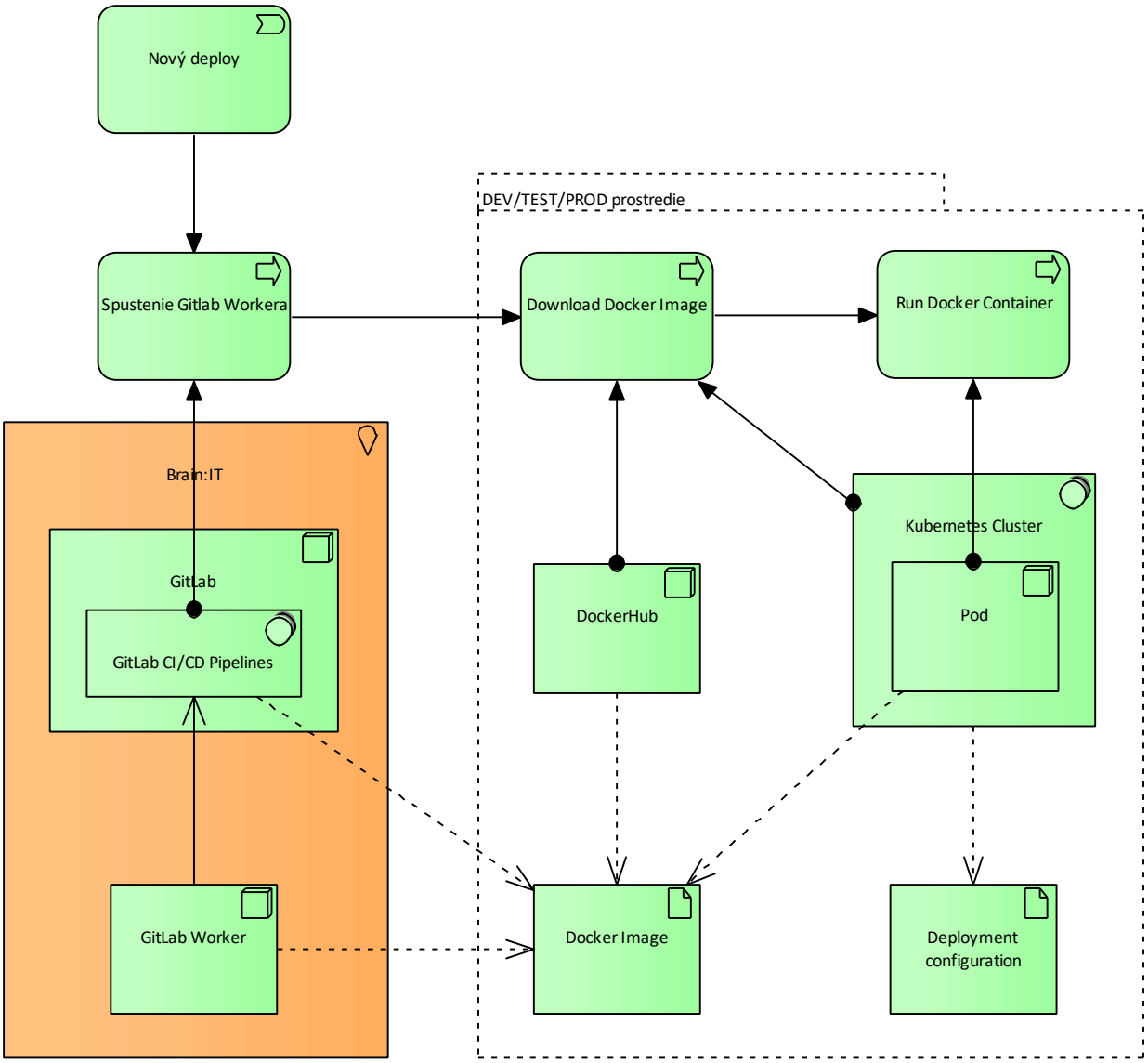


Figure 15 : Technology Viewpoint standard CI/CD DEV/TEST/PROD

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

1.4. Technology Viewpoint OpenStack

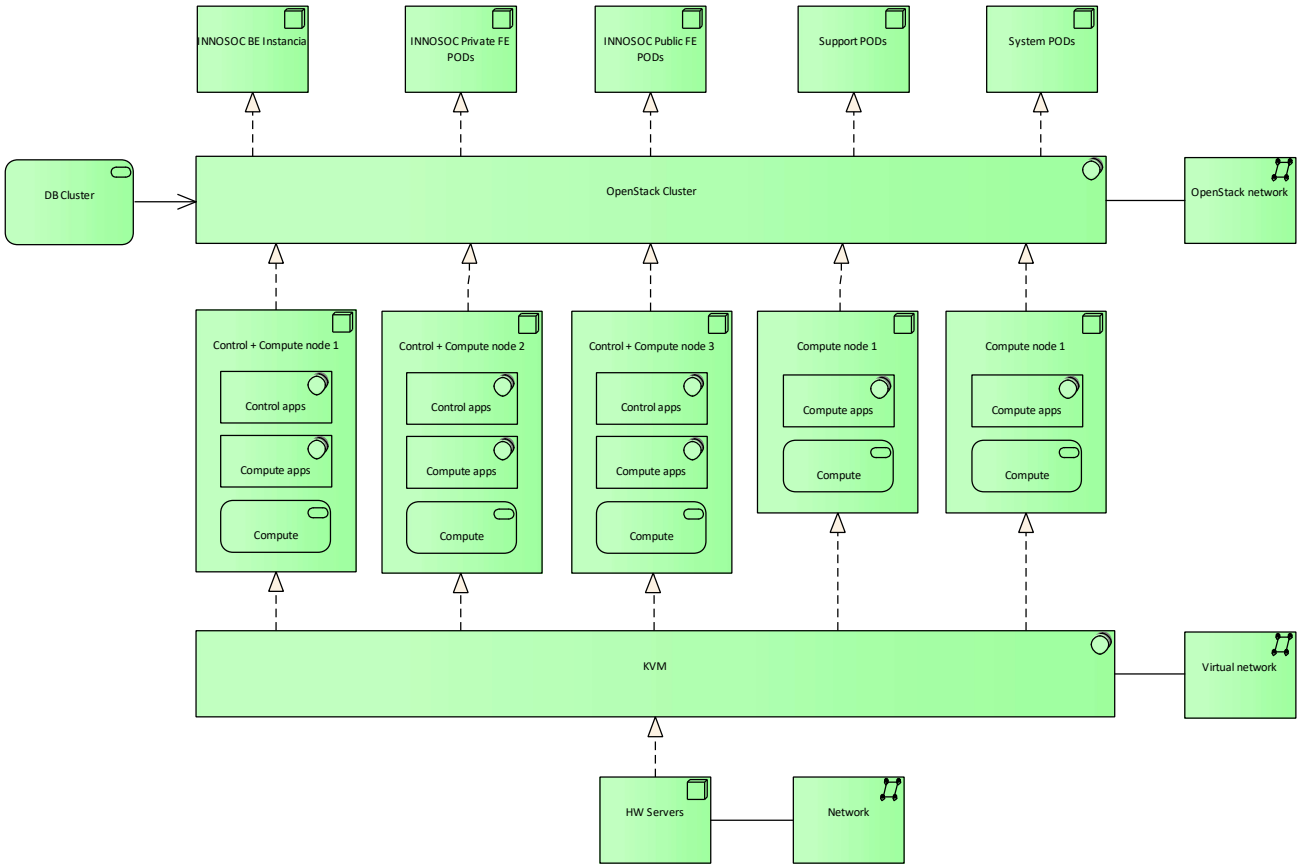


Figure 16 : Technology Viewpoint OpenStack Cluster

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

1.5. Technology Viewpoint Reporting

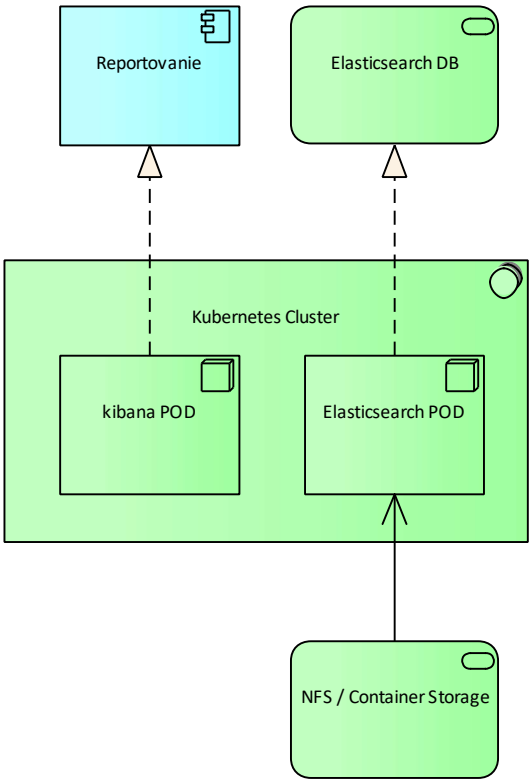


Figure 17 : Technology Usage Viewpoint Reporting alt 1

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.

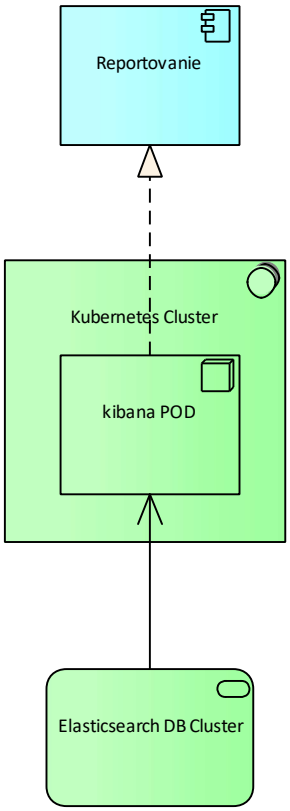


Figure 18 : Technology Usage Viewpoint Reporting alt 2

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.

1.6. Technology Viewpoint Ticketing system / Incident registration and management

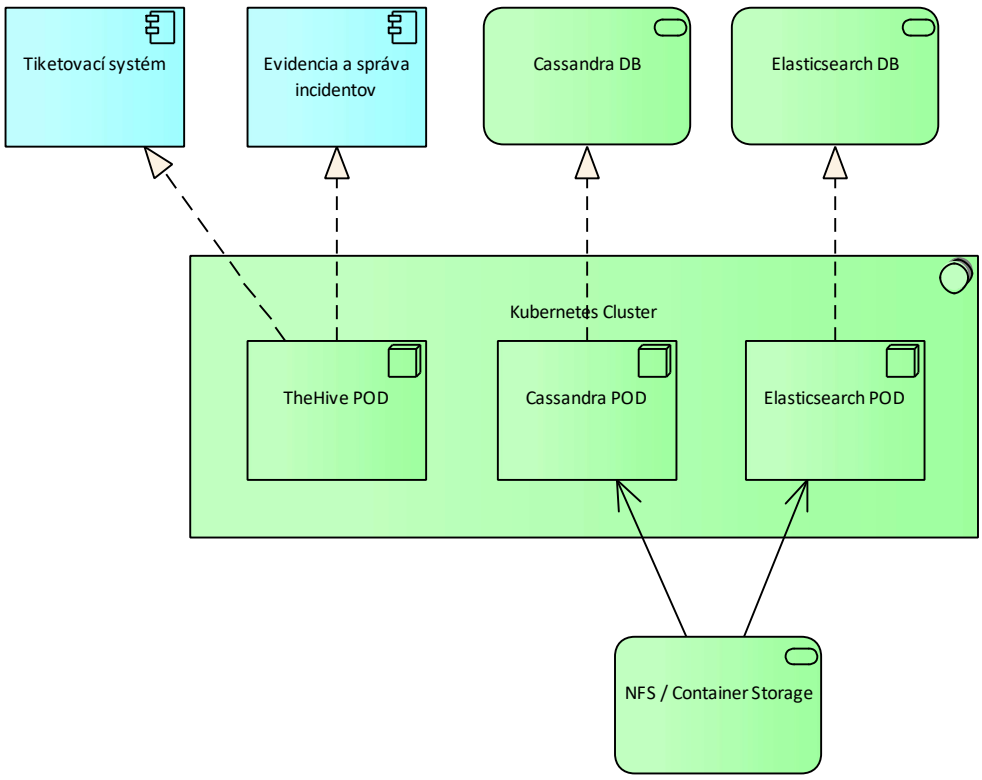


Figure 19 : Technology Usage Viewpoint Ticketing System / Incident Recording and Management alt 1

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.

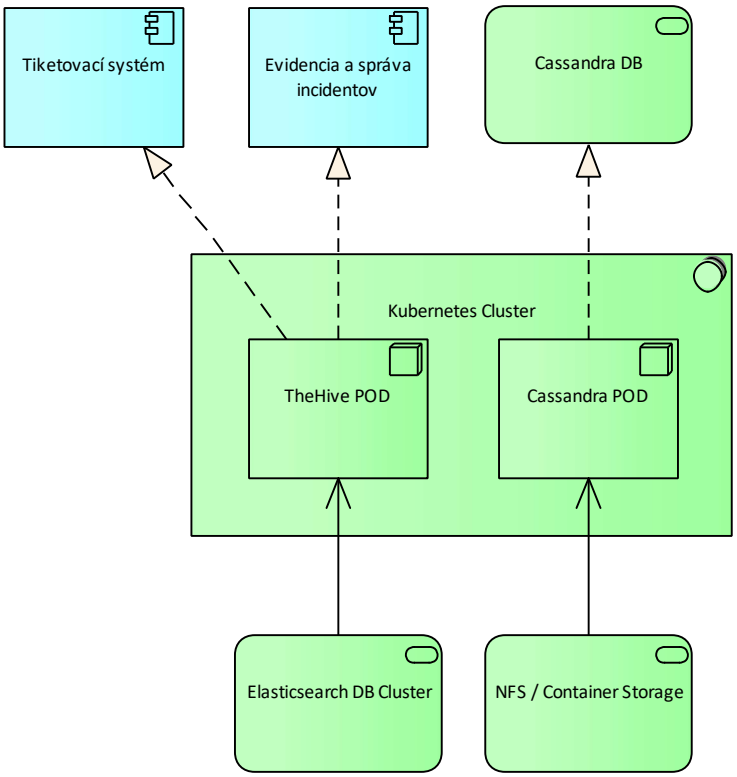


Figure 20 : Technology Usage Viewpoint Ticketing System / Incident Recording and Management alt 2

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.

1.7. Technology Viewpoint Knowledge Portal for the SOC Team

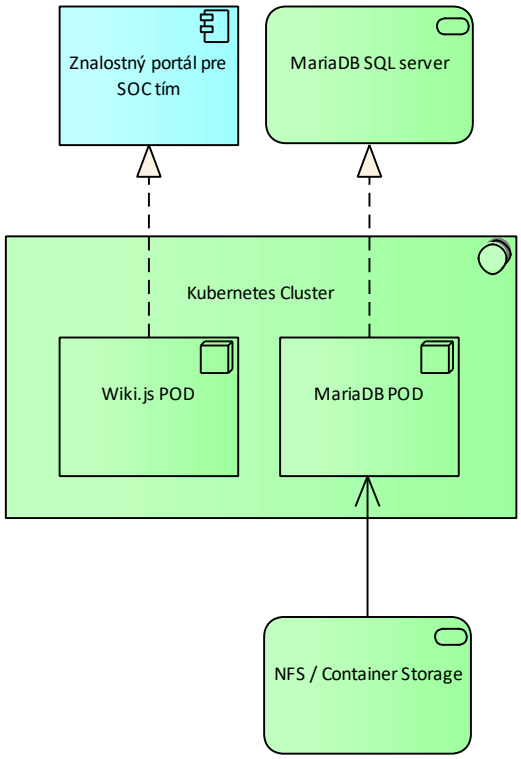


Figure 21 : Technology Usage Viewpoint Knowledge Portal for the SOC Team alt 1

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.

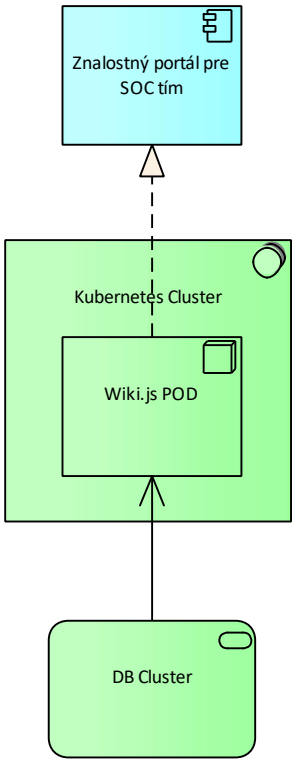


Figure 22 : Technology Usage Viewpoint Knowledge Portal for the SOC Team alt 2

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.

4. CONCLUSION

The solution proposed within the INNOSOC project focuses on building a multi-tenant cybersecurity operations centre (SOC) that will be provided as an open and flexible cloud service. Specifically, the work describes the elements used and their use within the technological architecture. By integrating elements of automation and advanced machine learning methods, in particular deep neural networks (DNN), the system's ability to effectively detect and predict anomalies in real time will be increased.

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