D2.2: DevSecOps and ML/Data- Ops methodology specification

Revision: v.1.0

Work package	**WP2**
Task	Task 2.3 & Task 2.4
Due date	31/05/2025
Submission date	02/06/2025
Deliverable lead	Netcompany-Intrasoft SA (INTRA)
Version	1.0
Authors	Filodamos Papanatsios (INTRA) Charalampos Marantos (INTRA) Matej Posinković (BEYOND) David Campo (FIWARE)
Reviewers	Fernando Lopez (FIWARE) Ilaria Bortone (MEDITECH)

www.cognets.eu

Grant Agreement No.: 101135930
Call: HORIZON-CL4-2023-DATA-01.

Topic: HORIZON-CL4-2023-DATA-01-04
Type of action: HORIZON Research and Innovation Actions

Abstract	This document details the methodology and requirements of DevSecOps and MLOps into the CoGNETs framework, focusing on CI/CD and ML/Data-Ops processes for seamless interoperability across the PUCs key domains. It also provides a comprehensive tools ecosystem and guidelines to streamline automation and adoption of these methodologies.
Keywords	DevSecOps, MLOps, CI/CD pipeline, interoperability, system orchestration, automated workflow, data integration, tools ecosystem, operational guidelines.

Document Revision History

Version	Date	Description of change	List of contributor(s)
V0.1	20/03/2025	Table of Contents	INTRA
V0.2	30/04/2025	Input provided in Sections 2 and 3	INTRA
V0.3	06/05/2025	Input provided in Sections 1, 2 and 5	INTRA
V0.4	08/05/2025	Input provided in Section 3.2.4	BEYOND
V0.5	09/05/2025	Input provided in Sections 2 and 3	INTRA and BEYOND with additional input from PUC providers - CERTH, HMU, FHG, SIEMENS, AVL.
V0.6	12/05/2025	Input provided in Section 3 (3.2.2 and 3.2.3)	FIWARE
V0.7	15/05/2025	Updates in Sections 2, 3 and 5	INTRA
V0.8	19/05/2025	Pre-final version for internal review	INTRA
V0.81	27/05/2025	Review comments integration	MEDITECH, CERTH
V0.82	27/05/2025	Review comments integration	FIWARE
V0.9	29/05/2025	Reviewers comments addressed (for 2nd round of review)	INTRA
V1.0	30/05/2025	Final version for submission	INTRA

DISCLAIMER

Funded by the European Union

Project funded by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

Funded by the European Union (CoGNETs, 101135930). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

This work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI).

COPYRIGHT NOTICE

Project funded by the European Commission in the Horizon Europe Programme		
Nature of the deliverable:	**R**	
Dissemination Level		
PU	*Public, fully open, e.g. web (Deliverables flagged as public will be automatically published in CORDIS project's page)*	✓
SEN	*Sensitive, limited under the conditions of the Grant Agreement*	
Classified R-UE/ EU-R	*EU RESTRICTED under the Commission Decision No2015/ 444*	
Classified C-UE/ EU-C	*EU CONFIDENTIAL under the Commission Decision No2015/ 444*	
Classified S-UE/ EU-S	*EU SECRET under the Commission Decision No2015/ 444*	

* R: Document, report (excluding the periodic and final reports)
DEM: Demonstrator, pilot, prototype, plan designs
DEC: Websites, patents filing, press & media actions, videos, etc.
DATA: Data sets, microdata, etc.
DMP: Data management plan
ETHICS: Deliverables related to ethics issues.
SECURITY: Deliverables related to security issues
OTHER: Software, technical diagram, algorithms, models, etc.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

EXECUTIVE SUMMARY

This document constitutes the Deliverable 2.2, "DevSecOps and ML/Data-Ops Methodology Specification". It provides a detailed analysis and strategy for the implementation and integration of DevSecOps and MLOps within the CoGNETs framework. CoGNETs provides advanced CI/CD and ML/Data-Ops processes that ensure interoperability and scalability across the Pilot Use Case domains (manufacturing, mobility, and healthcare). The document presents a tool ecosystem for automated workflows, orchestration and system monitoring. Additionally, it includes operational guidelines and user documentation to facilitate the effective adoption of DevSecOps and MLOps principles.

Based on the above, the following key topics are covered:

- CoGNETs DevSecOps and MLOps methodology and specifications.
- Implementation/Integration strategies and CI/CD tools and environments.

The Deliverable 2.2. mainly presents the outcomes of the activities performed in Task 2.3 "Specification of DevSecOps and ML/Data-Ops methods for developments" and part of the work of Task 2.4 "Lab-based programming testbed (basis for developments and experiments)" up to month 12. This work is part of the Work Package (WP) 2 "System requirements, architecture, and developments roadmap". Some of the components presented in the document are being developed in WPs 3, 4 and 5.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

TABLE OF CONTENTS

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

LIST OF FIGURES

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

LIST OF TABLES

Funded by the European Union

Project funded by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

ABBREVIATIONS

API	Application Programming Interface
CB	Context Broker
CI	Continuous Integration
CD	Continuous Development/Deployment
CPU	Central Processing Unit
CSV	Comma-Separated Values
D	Deliverable
DAG	Directed Acyclic Graphs
DevSecOps	Development, Security, and Operations
DTDL	Digital Twins Definition Language
ELK	Elasticsearch, Logstash & Kibana
GUI	Graphical User Interface
IoT	Internet of Things
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
LMU	Local ML Unit
ML	Machine Learning
NGSI	Next Generation Service Interface
NGSI-LD	NGSI – Linked Data
Ops	Operations
PUC	Pilot Use Case
RAM	Random Access Memory
RDF	Resource Description Framework
REST	Representational State Transfer
SCM	Source Code Management
SQL	Structured Query Language
SSO	Single Sign On

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

TMS	Thermal Management System
UI	User Interface
URI	Uniform Resource Identifier
VM	Virtual Machine
YAML	YAML Ain't Markup Language

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

1 INTRODUCTION

1.1 PURPOSE AND SCOPE

This document provides an overview of the innovative CoGNETs DevSecOps methodology and specifications. It includes a detailed integration strategy, focusing on a robust framework that enhances and supports the CoGNETs' complex ML/Data operations. The methodology relies on the CI/CD and DevSecOps paradigms, augmented with mechanisms supporting the CoGNETs' complex ML functions. This ensures seamless interoperability and security across the various PUC domains, which include manufacturing, mobility, and healthcare.

The purpose of this deliverable is to define and document both the CoGNETs DevSecOps and MLOps methodologies, along with their specifications. It outlines the integration strategies employed and details the CI/CD tools and environments utilized in the process. The document begins with the presentation of the CoGNETs DevSecOps implementation and integration strategy, describing the integrated CI/CD stack. Afterwards, the ML/Data-Ops methodology is presented, illustrating the interoperability framework developed within the context of CoGNETs. In addition, it presents a detailed mapping between PUCs sub-processes and PROJECT ML/Data-Ops components, such as INTRA's Streamhandler and FIWARE's Context Broker. Finally, the document covers the operational guidelines for utilizing the introduced integration pipeline, including user documentation and best practices.

1.2 DELIVERABLE STRUCTURE

The document is structured as follows:

- **Section 1** serves as the introduction, presenting the purpose and scope of the work included in the document and its relationship with the rest of the CoGNETs work packages and tasks.

- **Section 2** outlines the CoGNETs DevSecOps Implementation and Integration Strategy, beginning with partner engagement in defining design choices and proceeding with a detailed description of the integrated CI/CD stack. It introduces the tools ecosystem and provides an overview of source code management, continuous integration processes, registry, monitoring procedures, interfaces and implementation patterns. Section 2 also covers the integration pipeline's operational guidelines, including user documentation and team enablement strategies, offering DevSecOps principles and best practices.

- **Section 3** defines the ML/Data-Ops specifications. The presented methodology describes the interoperability framework being developed in the context of CoGNETs and its components, including INTRA's Streamhandler and FIWARE Context Broker. Finally, the mapping between PUCs sub-processes and CoGNETs ML/Data-Ops components is presented in detail.

- **Section 4** focuses on the security aspects and mechanisms of the system, outlining CoGNETs' use of a self-managed API Gateway to enhance component communication with strict request rules, centralized authentication, and uniform HTTPS, alongside secure API interactions via Keycloak, data filtering to protect sensitive information, and real-time error and security monitoring through Sentry server.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

- **Section 5** concludes this document, summarises the achievements and presents the next steps in the DevSecOps and MLOps development process, towards the first stage of PUC demos in the CoGNETs testbed.

1.3 INTERRELATIONS WITH OTHER WORK PACKAGES

The content of this Deliverable has important connections with various Work Packages in the CoGNETs project. It is primarily based on WP2 ("System requirements, architecture, and developments roadmap"), focusing on Task 2.3, which covers DevSecOps and ML/Data-Ops methods, and Task 2.4, which establishes the programming testbed.

The components presented in this document are closely linked to ongoing work in Work Packages 3 (“Integrated models and mechanisms for collaborative smart nodes”) and 4 (“Middleware infrustructure for dynamic IoT-to-Cloud swarm computing”). The DevSecOps and ML/Data-Ops methods outlined here provide a strong basis for integrating these other development efforts. The work introduced in this deliverable, will facilitate the design of a collaborative platform that incorporates input from all CoGNETs components and PUCs.

This collaboration across Work Packages, based on the DevSecOps and ML/Data-Ops methodology and specifications described in this document, will lead to two main demonstration phases: an initial demo at M18, that will be described in D2.3 "Lab-based testbed deployment" and a more detailed demonstration in WP5 (“System integration, use cases deployment, and validation”). The foundations presented in this document, along with the rest of the WP2 deliverables, will support the final integration of data and tools across the project, ensuring a unified validation of the CoGNETs innovations.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

2 DEVSECOPS IMPLEMENTATION AND INTEGRATION STRATEGY

DevSecOps is a set of practices that combines software development (Dev), security (Sec), and IT operations (Ops). Building on DevOps, which enhances collaboration between development and operations to speed up software delivery using continuous integration and deployment (CI/CD), automation, and iterative improvement, DevSecOps adds a focus on integrating security into every phase of the development lifecycle. This approach accelerates the development process, ensures continuous delivery of high-quality software and incorporates security measures, allowing organizations to deliver robust applications efficiently.

The following sub-sections present the CoGNETs DevSecOps specifications and strategy. More specifically, starting with the partners engagement during the early phase of the project, the CoGNETs integration methodology is presented, including the CI/CD pipeline that includes all the necessary tools that offer source code management, continuous integration and system monitoring features.

2.1 PARTNER ENGAGEMENT

From the beginning of the project, as also referred to in D2.1 [1], the key partners and end-users were actively engaged in defining and developing of the DevSecOps and ML/Data-Ops methodology and specifications. Collaborative documents were created and shared with the partners (technology and PUCs providers) to collect the details and requirements systematically.

The information collected includes:

- Components type
- Software requirements
- Hardware requirements
- Component access
- Use-case
- Summary of the component

Following the initial spreadsheets shared at the beginning of the project and as we reached a more mature phase after defining the first architecture views, a complementary questionnaire regarding the CI/CD stack was prepared and shared with the CoGNETs Consortium. This information not only engaged and coordinated the partners but was also very important for defining the characteristics and design choices of the DevSecOps implementation. One crucial question is the accessibility of the CoGNETs components. Figure 1 summarizes the results. According to this study, we indicate three main accessibility categories:

- Private – Only accessible to members, which was the most popular one
- No access – only containers and images
- Public – accessible to everyone outside the project

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

While the choices of being fully private, meaning only accessible to the component's owner and neither code nor container, had no answers. As a result, the integration strategy has to support the three categories of the CoGNETs components, with regard to their accessibility.

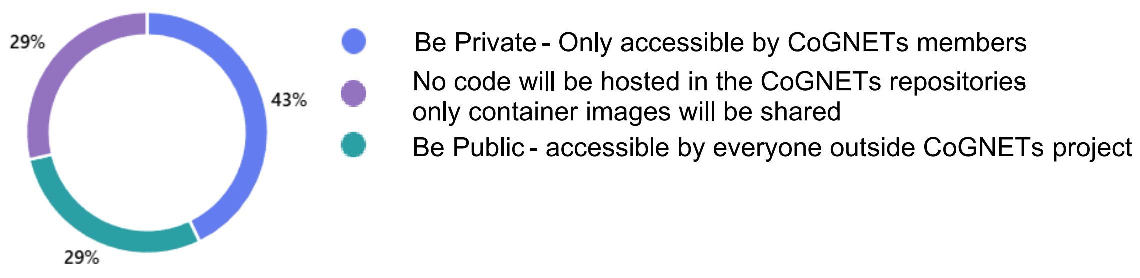


Figure 1 Accessibility of the individual CoGNETs components

Table 1 presents an indicative example of the initial list of software components planned to be integrated using the CI/CD stack presented in this document, as they were collected by the CoGNETs partners during the project's first months.

Table 1 Indicative CoGNETs software elements that constitute the CoGNETs components integrated using the introduced CI/CD stack

Organization	WP/Task	Software Elements
CERTH	T4.3	Orion Context Broker, IoT Agents, Keycloak
FHG	WP5	ROS2/DDS, OPC UA
FIWARE	T4.1	Dockerfiles, Helm charts, Open API, Python, Go, Java, Node.js
Meditech	T4.2	Not Available Yet
K3Y	T4.4	Application Framework: Python/FastAPI
NCSRD	T3.3, T3.4, T4.2	Not Available Yet
HMU	T4.4	Application Framework

2.2 COGNETS INTEGRATION: CI/CD STACK

The CoGNETs middleware consists of numerous web services that are continuously updated to meet the ongoing requirements of the node swarm it supports. In light of this, INTRA has developed a comprehensive Continuous Integration (CI) and Continuous Delivery (CD) software suite designed to streamline the software delivery process for all technical partners in the CoGNETs consortium.

Funded by the European Union
Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

Before we provide a detailed exposition of all the involved software modules we will introduce the concept of the CI/CD practices.

CI/CD is a modern software development practice aimed at automating the processes of building, testing, and deploying code. **Continuous Integration** involves developers frequently merging code changes into a shared online repository, where automated builds and tests are run to detect issues early. Once the deployed code is free of any problems discovered in the CI stage, the **Continuous Deployment** stage can begin. This final step delivers validated code to production or staging environments, ensuring faster and more reliable software releases ideally with virtually zero downtime. Together, CI/CD improves code quality, reduces integration risks, and accelerates delivery cycles in software development.

2.2.1 CI/CD Tools Ecosystem

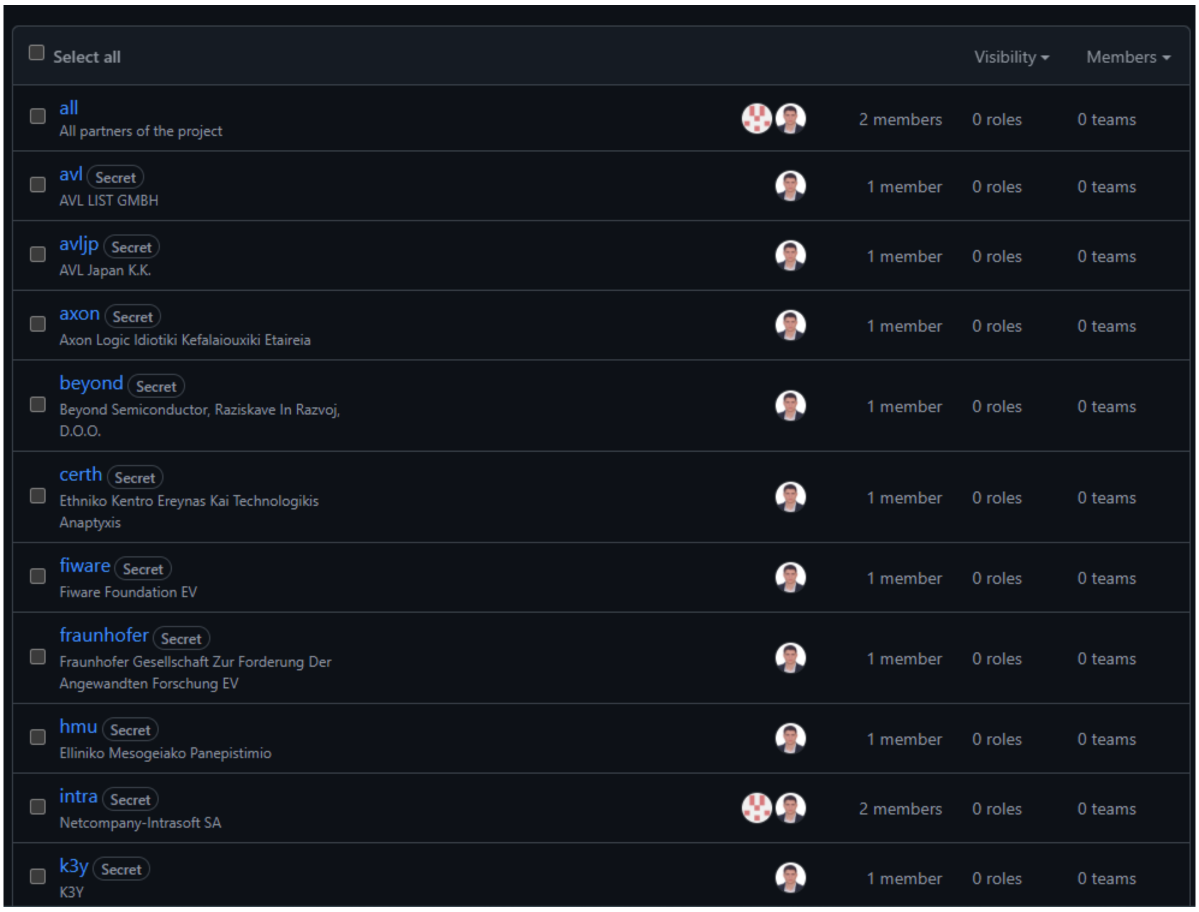
A collection of modern software tools is utilised to implement the CoGNETs CI/CD process. Our approach is based on setting up a **Single Sign-On** (SSO) environment, allowing developers to access all CI/CD tools without needing to re-authenticate after their initial login. All tools are hosted on INTRA's cloud premises and shared with the consortium's technical partners through a simple portal UI which allows easy navigation. The development of this portal is currently in progress.

2.2.1.1 Source Code Management (SCM)

For SCM we are using GitHub [2], which offers features such as issue tracking, code reviews, and wikis. Moreover, a GitHub organisation (cognets-eu, see Figure 2) has been setup with separate teams for each software module that will use it for SCM. The central organisation URL can be found in https://github.com/cognets-eu.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI



Select all | Visibility | Members

all All partners of the project	2 members	0 roles	0 teams
avl Secret AVL LIST GMBH	1 member	0 roles	0 teams
avljp Secret AVL Japan K.K.	1 member	0 roles	0 teams
axon Secret Axon Logic Idiotiki Kefalaiouxiki Etaireia	1 member	0 roles	0 teams
beyond Secret Beyond Semiconductor, Raziskave In Razvoj, D.O.O.	1 member	0 roles	0 teams
certh Secret Ethniko Kentro Ereynas Kai Technologikis Anaptyxis	1 member	0 roles	0 teams
fiware Secret Fiware Foundation EV	1 member	0 roles	0 teams
fraunhofer Secret Fraunhofer Gesellschaft Zur Forderung Der Angewandten Forschung EV	1 member	0 roles	0 teams
hmu Secret Elliniko Mesogeiako Panepistimio	1 member	0 roles	0 teams
intra Secret Netcompany-Intrasoft SA	2 members	0 roles	0 teams
k3y Secret K3Y	1 member	0 roles	0 teams

Figure 2 CoGNETs GitHub organization

2.2.1.2 Continuous Integration

As mentioned in the introductory section 2.2, **Continuous Integration** (CI) is a software development practice in which developers frequently integrate code changes into a shared repository, typically several times a day. Each integration is automatically verified by running builds and automated tests to quickly detect errors and improve code quality. CI helps teams identify issues early in the development cycle, reduce integration problems, and deliver software faster and reliably. Additionally, it enforces a consistent coding style, which is essential when large software engineering teams are involved in a project.

In CoGNETs the software that orchestrates this process is the **Jenkins** [3] automation server. Jenkins automatically scans for and pulls the latest code from each repository, then it builds the associated container and runs any available tests. If those tests pass, Jenkins triggers code analysis through **SonarQube** [4]**.** If no code quality issues are identified, Jenkins will then build a container image and either deploy it to the CoGNETs container image registry (described in the next section) or directly deploy it on the relevant servers, depending on the nature of the software module.

The CoGNETs Jenkins web UI is available in https://jenkins.cognets.rid-intrasoft.eu to monitor this process at will. Each CoGNETs component has its folder on the Jenkins dashboard (Figure 3) where the corresponding pipelines will be created. Partners will be able to view and contribute to all folders and pipelines specific to the components they are working on.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

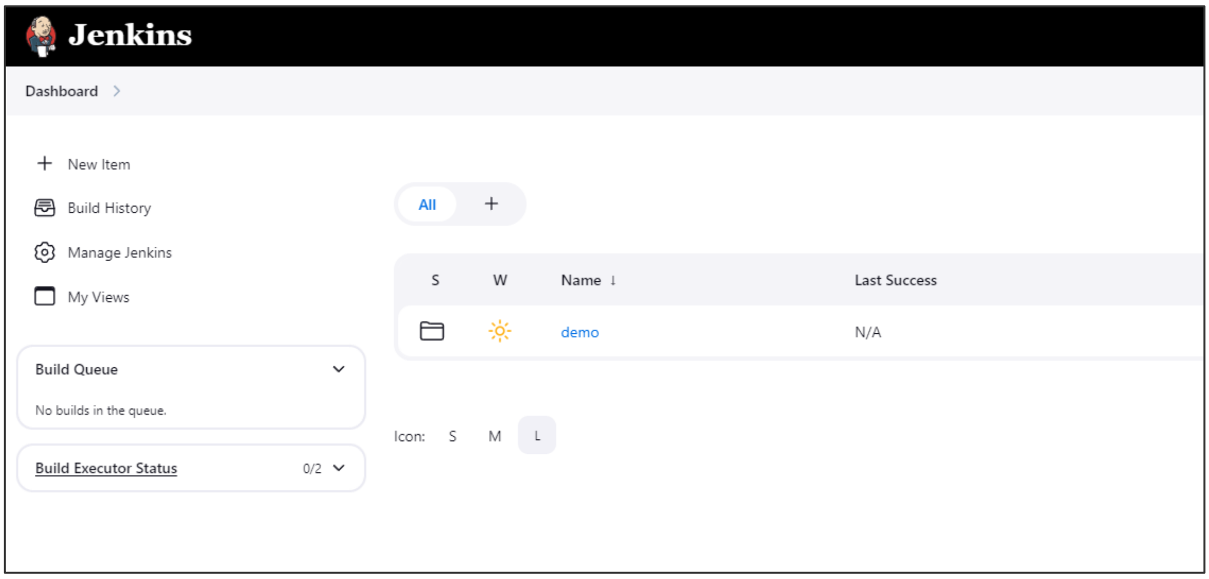


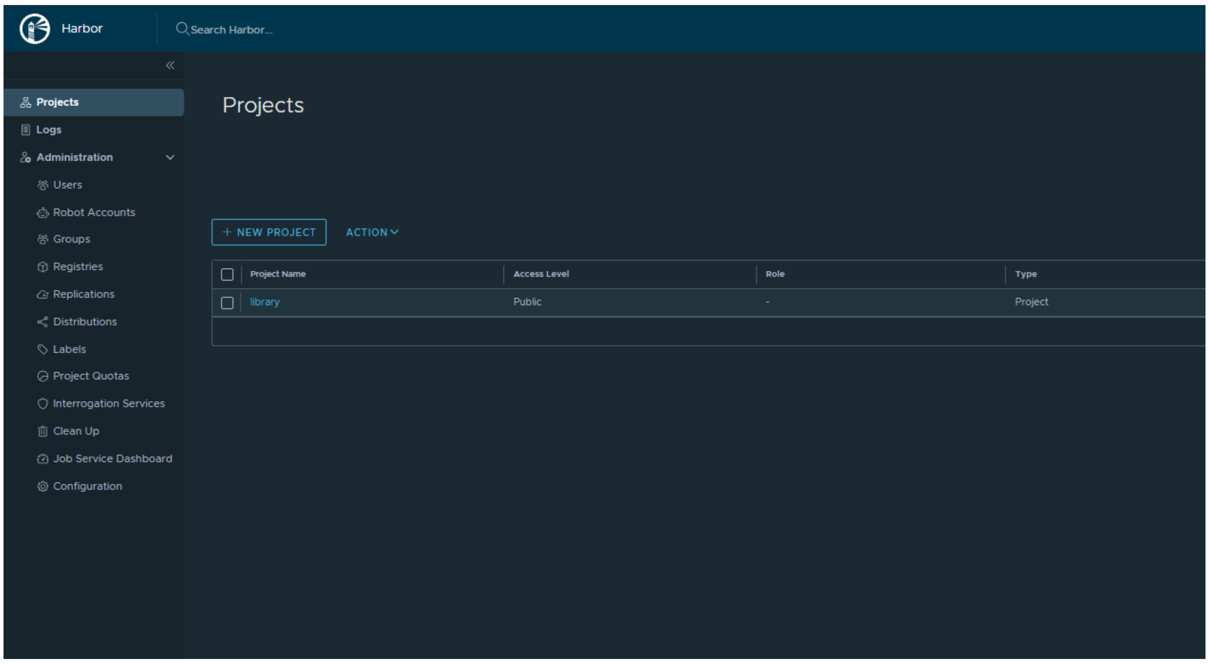
Figure 3 Jenkins dashboard

2.2.1.3 Container Registry

In CoGNETs all software modules will be deployed with the use of containerization. Containerization is a lightweight virtualization method that packages an application and its dependencies into a single, isolated unit called a container. One of the main advantages of containers is their ability to ensure consistency across different environments by behaving the same way regardless of the underlying system (e.g., a developer's machine, a test server, or a production server). Unlike traditional virtual machines (VMs), which can be resource-intensive when running multiple instances on a single host, containers efficiently share the host system's operating system, maximizing resource usage and reducing startup time. This approach enhances portability, scalability, and simplifies deployment.

To orchestrate the registration and organization of container images, CoGNETs uses the **Harbor** [5] Container Registry. Harbor provides a secure, centralized location for storing Docker container images accessible from all CoGNETs servers and nodes. Images will be pushed to Harbor via the CoGNETs CI/CD pipeline. It is described later in detail in section 2.2.1.6. The registered images can be browsed through the Harbor portal, presented in Figure 4, which is hosted in https://harbor.cognets.rid-intrasoft.eu/https://harbor.cognets.rid-intrasoft.eu.

Funded by the European Union
Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI



Project Name	Access Level	Role	Type
library	Public	-	Project

Figure 4 Harbor portal

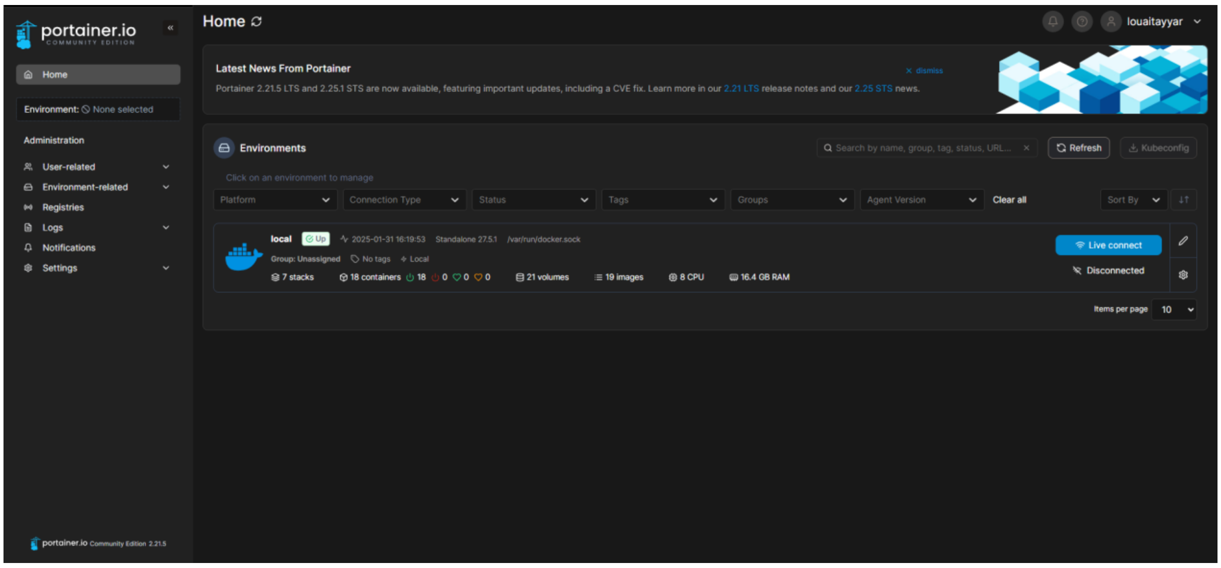
Each component will have its own project on Harbor and each partner will be able to only see the components for which they permission to access.

2.2.1.4 Container Management and Monitoring

CoGNETs is an extensive, distributed software system where numerous software modules are deployed through containerization. In such an environment the easy monitoring and management of containers is of paramount importance. To that end, CoGNETs utilizes **Portainer** [6], a lightweight, open-source management web UI that simplifies the deployment and management of Docker containers. Portainer provides a user-friendly web interface that allows for easy monitoring, configuration, and control of container resources. Partners with the appropriate permissions can access the CoGNETs Portainer instance: https://portainer.cognets.rid-intrasoft.eu. The CoGNETs Portainer is presented in Figure 5.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI



Latest News From Portainer

Portainer 2.21.5 LTS and 2.25.1 STS are now available, featuring important updates, including a CVE fix. Learn more in our 2.21 LTS release notes and our 2.25 STS news.

local Up 2025-01-31 16:19:53 Standalone 27.5.1 /var/run/docker.sock
Group: Unassigned No tags Local
7 stacks 18 containers 18 0 0 0 21 volumes 19 images 8 CPU 16.4 GB RAM

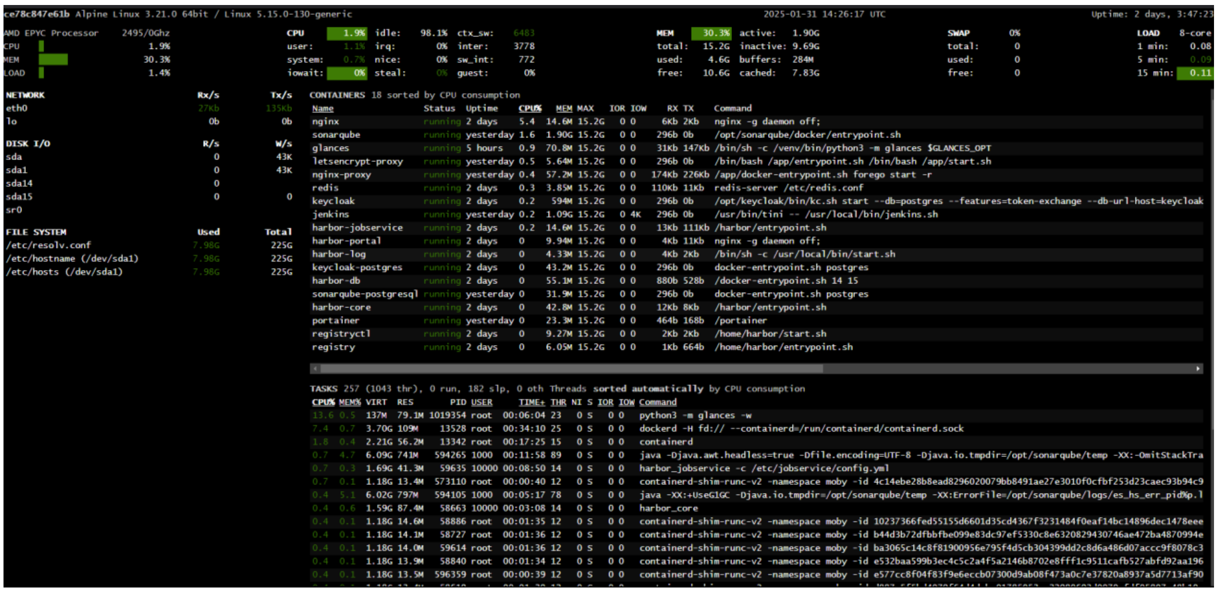
Figure 5 CoGNETs Portainer for monitoring, configuration and control of containers

2.2.1.5 System Monitoring

System monitoring in a massive distributed system such as CoGNETs is crucial. In addition to Portainer, mentioned in the previous section, which monitors containers and their resources, we utilise Glances [7] to monitor hardware resources in real time in all CoGNETs main nodes (excluding the edge computing nodes). Special alerts and metrics will be setup to proactively notify administrators of usage spikes (CPU, RAM or storage) and network outages, as presented in Figure 6.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI



```
ce78c847e61b Alpine Linux 3.21.0 64bit / Linux 5.15.0-130-generic                    2025-01-31 14:26:17 UTC                                         Uptime: 2 days, 3:47:23

AMD EPYC Processor    2495/0Ghz         CPU      1.9%  idle:     98.1%  ctx_sw:   6483          MEM     30.3%  active:    1.90G              SWAP      0%              LOAD    8-core
CPU  |                    1.9%          user:    1.1%  irq:         0%  inter:    3778          total:  15.2G  inactive:  9.69G              total:     0              1 min:    0.08
MEM  |||||               30.3%          system:  0.7%  nice:        0%  sw_int:    772          used:    4.6G  buffers:   284M               used:      0              5 min:    0.09
LOAD |                    1.4%          iowait:    0%  steal:       0%  guest:      0%          free:   10.6G  cached:    7.83G              free:      0              15 min:   0.11

NETWORK                  Rx/s      Tx/s  CONTAINERS 18 sorted by CPU consumption
eth0                     27Kb     135Kb  Name               Status  Uptime    CPU%  MEM MAX    IOR IOW    RX TX     Command
lo                         0b        0b  nginx              running 2 days    5.4  14.6M 15.2G  0 0      6Kb 2Kb   nginx -g daemon off;
                                         sonarqube          running yesterday 1.6  1.90G 15.2G  0 0     296b 0b    /opt/sonarqube/docker/entrypoint.sh
DISK I/O                  R/s       W/s  glances            running 5 hours   0.9  70.8M 15.2G  0 0     31Kb 147Kb /bin/sh -c /venv/bin/python3 -m glances $GLANCES_OPT
sda                         0       43K  letsencrypt-proxy  running yesterday 0.5  5.64M 15.2G  0 0     296b 0b    /bin/bash /app/entrypoint.sh /bin/bash /app/start.sh
sda1                        0       43K  nginx-proxy        running yesterday 0.4  57.2M 15.2G  0 0    174Kb 226Kb /app/docker-entrypoint.sh forego start -r
sda14                       0            redis              running 2 days    0.3  3.85M 15.2G  0 0    110Kb 11Kb  redis-server /etc/redis.conf
sda15                       0         0  keycloak           running 2 days    0.2   594M 15.2G  0 0     296b 0b    /opt/keycloak/bin/kc.sh start --db=postgres --features=token-exchange --db-url-host=keycloak
sr0                                      jenkins            running yesterday 0.2  1.09G 15.2G  0 4K    296b 0b    /usr/bin/tini -- /usr/local/bin/jenkins.sh
                                         harbor-jobservice  running 2 days    0.2  14.6M 15.2G  0 0     13Kb 111Kb /harbor/entrypoint.sh
FILE SYSTEM              Used     Total  harbor-portal      running 2 days    0    9.94M 15.2G  0 0      4Kb 11Kb  nginx -g daemon off;
/etc/resolv.conf         7.98G     225G  harbor-log         running 2 days    0    4.33M 15.2G  0 0      4Kb 2Kb   /bin/sh -c /usr/local/bin/start.sh
/etc/hostname (/dev/sda1) 7.98G    225G  keycloak-postgres  running 2 days    0    43.2M 15.2G  0 0     296b 0b    docker-entrypoint.sh postgres
/etc/hosts (/dev/sda1)   7.98G     225G  harbor-db          running 2 days    0    55.1M 15.2G  0 0     880b 528b  /docker-entrypoint.sh 14 15
                                         sonarqube-postgresql running yesterday 0  31.9M 15.2G  0 0     296b 0b    docker-entrypoint.sh postgres
                                         harbor-core        running 2 days    0    42.8M 15.2G  0 0     12Kb 8Kb   /harbor/entrypoint.sh
                                         portainer          running yesterday 0    23.3M 15.2G  0 0     464b 168b  /portainer
                                         registryctl        running 2 days    0    9.27M 15.2G  0 0      2Kb 2Kb   /home/harbor/start.sh
                                         registry           running 2 days    0    6.05M 15.2G  0 0      1Kb 664b  /home/harbor/entrypoint.sh

                                         TASKS 257 (1043 thr), 0 run, 182 slp, 0 oth Threads sorted automatically by CPU consumption
                                         CPU% MEM% VIRT  RES       PID USER     TIME+ THR NI S IOR IOW Command
                                         13.6 0.5  137M  79.1M 1019354 root  00:06:04 23   0 S   0 0   python3 -m glances -w
                                         7.4  0.7  3.70G 109M    13528 root  00:34:10 25   0 S   0 0   dockerd -H fd:// --containerd=/run/containerd/containerd.sock
                                         1.8  0.4  2.21G 56.2M   13342 root  00:17:25 15   0 S   0 0   containerd
                                         0.7  4.7  6.09G 741M   594265 1000  00:11:58 89   0 S   0 0   java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=/opt/sonarqube/temp -XX:-OmitStackTra
                                         0.7  0.3  1.69G 41.3M   59635 10000 00:08:50 14   0 S   0 0   harbor_jobservice -c /etc/jobservice/config.yml
                                         0.7  0.1  1.18G 13.4M  573110 root  00:00:40 12   0 S   0 0   containerd-shim-runc-v2 -namespace moby -id 4c14ebe28b8ead8296020079bb8491ae27e3010f0cfbf253d23caec93b94c9
                                         0.4  5.1  6.02G 797M   594105 1000  00:05:17 78   0 S   0 0   java -XX:+UseG1GC -Djava.io.tmpdir=/opt/sonarqube/temp -XX:ErrorFile=/opt/sonarqube/logs/es_hs_err_pid%p.l
                                         0.4  0.6  1.59G 87.4M   58663 10000 00:03:08 14   0 S   0 0   harbor_core
                                         0.4  0.1  1.18G 14.6M   58886 root  00:01:35 12   0 S   0 0   containerd-shim-runc-v2 -namespace moby -id 10237366fed55155d6601d35cd4367f3231484f0eaf14bc14896dec1478eee
                                         0.4  0.1  1.18G 14.1M   58727 root  00:01:36 12   0 S   0 0   containerd-shim-runc-v2 -namespace moby -id b44d3b72dfbbfbe099e83dc97ef5330c8e6320829430746ae472ba4870994e
                                         0.4  0.1  1.18G 14.0M   59614 root  00:01:36 12   0 S   0 0   containerd-shim-runc-v2 -namespace moby -id ba3065c14c8f81900956e795f4d5cb304399dd2c8d6a486d07accc9f8078c3
                                         0.4  0.1  1.18G 13.9M   58840 root  00:01:34 12   0 S   0 0   containerd-shim-runc-v2 -namespace moby -id e532baa599b3ec4c5c2a4f5a2146b8702e8fff1c9511cafb527abfd92aa196
                                         0.4  0.1  1.18G 13.5M  596359 root  00:00:39 12   0 S   0 0   containerd-shim-runc-v2 -namespace moby -id e577cc8f04f83f9e6eccb07300d9ab08f473a0c7e37820a8937a5d7713af90
```

Figure 6 CoGNETs DevSecOps monitoring (using Glances)

2.2.1.6 CI/CD Pipeline Overview

For various reasons (e.g. privacy, licensing, confidential industrial techniques, etc.) some partners may choose not to share actual source code within the CoGNETs GitHub organization, opting instead to share only pre-built container images. Conversely, other partners will choose to host their source code within CoGNETs. This distinction leads to two separate CI/CD pipelines, as outlined below.

It is important to note that, regardless of the approach partners choose, all CoGNETs software components utilising the CI/CD pipeline will have an associated code repository in the CoGNETs GitHub organization. The difference is that the repositories for components that do not share actual source code will contain only integration and deployment specific content.

- Source Code-Based Workflow

  The source code will be hosted within the CoGNETs GitHub organisation in this scenario. The Integration flow is as follows.

  - A new code commit is pushed, or a merge request is made on a CoGNETs repository.

  - Via a GitHub webhook, the Jenkins orchestration server is triggered automatically. Subsequently, the associated Jenkinsfile (file created for each code repository in the CogNets organization) is being read and the included integration pipeline stages are executed. These stages, depending on the module, include:

    - Static Code Analysis via SonarQube: This step ensures that no poor-quality code is pushed by checking for known security vulnerabilities and verifying adherence to programming best practices.

    - Code styling & linting checks. This step is particularly important for programming languages not strongly statically typed such as Python, Ruby,

Funded by the European Union
Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

or JavaScript. Similar to SonarQube, it ensures that no poor-quality code is being pushed and that the code conforms to a pre-agreed coding style, which is essential in large team software engineering projects.

    - If the code repository contains unit tests (maintained by the owner of the repository) these are executed.

  - Once the integration stages have been completed without any errors then the deployment stages are executed. This involves a simpler series of steps where:
    - Container images are pushed to the CoGNETs container registry.
    - Containers are deployed on the relevant target computing nodes as defined in the corresponding Jenkins files.
  - Once all stages have been executed, administrators are notified whether the whole pipeline was successfully integrated and deployed. In all scenarios, the CI/CD process's outcome can be monitored in detail through the Jenkins Web GUI.

- Container Image-Based Workflow

  Several CoGNETs technical partners cannot share source code within the project's GitHub organisation due to privacy and licensing reasons. Therefore, our CI/CD pipeline is adjusted to accommodate this situation.

  In this scenario, a GitHub code repository is still necessary for each software module to host the information required for its deployment. This includes the associated Jenkins file and any other relevant configuration files.

  The integration flow in this scenario is as follows:

  - A new commit is pushed.
  - Static analysis tools can be run locally.
  - If the previous step is successful, the associated container image is built and uploaded to the CoGNETs image repository (Harbor).
  - Jenkins is notified of a new image (or a new version of an existing image) and deploys the container on the relevant computing nodes.

2.3 USER DOCUMENTATION AND BEST PRACTICES

Comprehensive user documentation designed to assist CoGNETs partners in integrating various complex tools into the CoGNETs DevSecOps environment, supports the CI/CD workflows. This documentation serves as a crucial resource, offering step-by-step guidance and best practices for seamless integration. It is continuously updated and enhanced to incorporate new insights, additional tools and feedback gathered from the Consortium. This ensures the documentation remains current and relevant, reflecting the evolving landscape of DevSecOps practices.

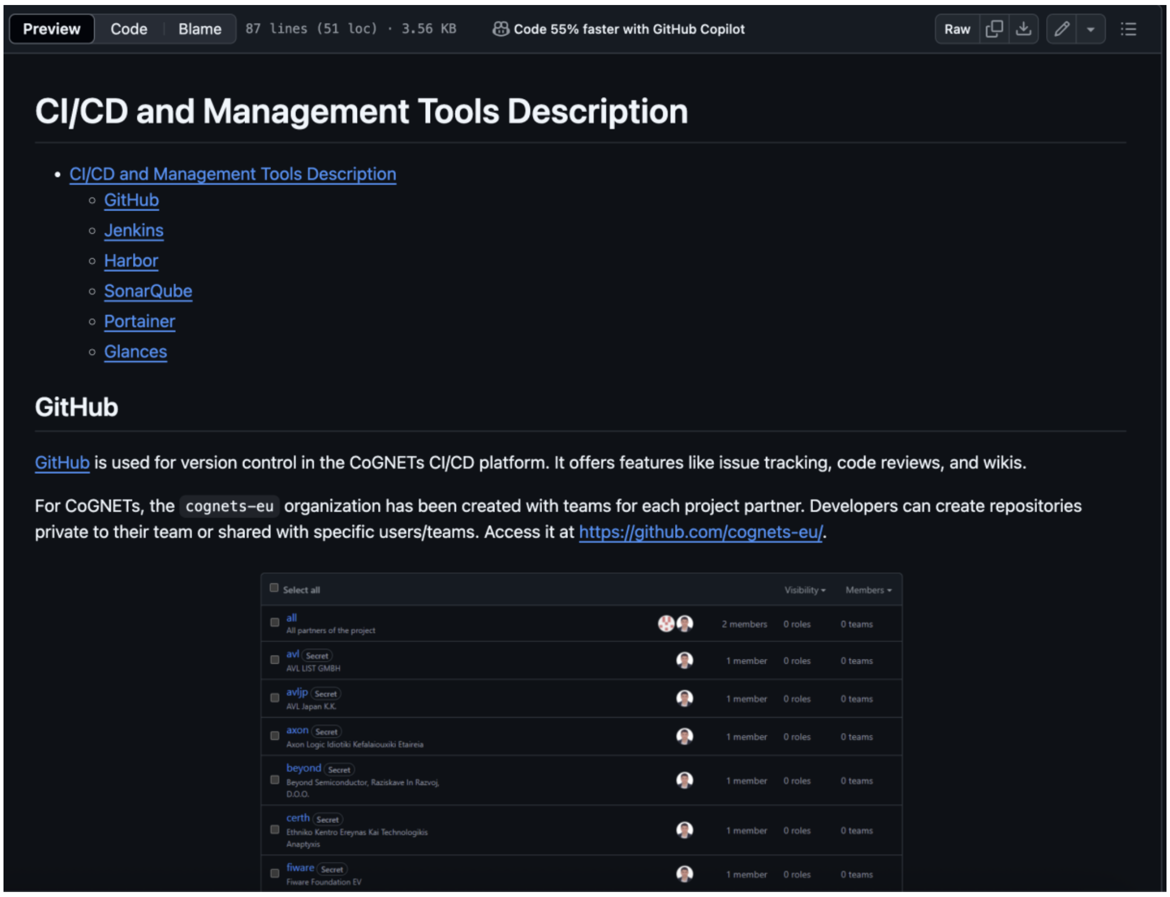
The CI/CD documentation is divided into the following categories:

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

- **CI/CD Tools**: This category presents a comprehensive list of all available CI/CD tools, each accompanied by a brief introduction and the corresponding web GUI URL (see Figure 7). This resource aims to help users understand the functionalities of each tool and how to access them.

  Link: https://github.com/cognets-eu/ci-cd-documentation/blob/main/cicd_tools.md


Preview
Code
Blame
87 lines (51 loc) · 3.56 KB
Code 55% faster with GitHub Copilot
Raw

CI/CD and Management Tools Description

- CI/CD and Management Tools Description
  - GitHub
  - Jenkins
  - Harbor
  - SonarQube
  - Portainer
  - Glances

GitHub

GitHub is used for version control in the CoGNETs CI/CD platform. It offers features like issue tracking, code reviews, and wikis.

For CoGNETs, the `cognets-eu` organization has been created with teams for each project partner. Developers can create repositories private to their team or shared with specific users/teams. Access it at https://github.com/cognets-eu/.

Select all		Visibility	Members
all All partners of the project	2 members	0 roles	0 teams
avl Secret AVL LIST GMBH	1 member	0 roles	0 teams
avljp Secret AVL Japan K.K.	1 member	0 roles	0 teams
axon Secret Axon Logic Idiotiki Kefalaiouxiki Etaireia	1 member	0 roles	0 teams
beyond Secret Beyond Semiconductor, Raziskave In Razvoj, D.O.O.	1 member	0 roles	0 teams
certh Secret Ethniko Kentro Ereynas Kai Technologikis Anaptyxis	1 member	0 roles	0 teams
fiware Secret Fiware Foundation EV	1 member	0 roles	0 teams

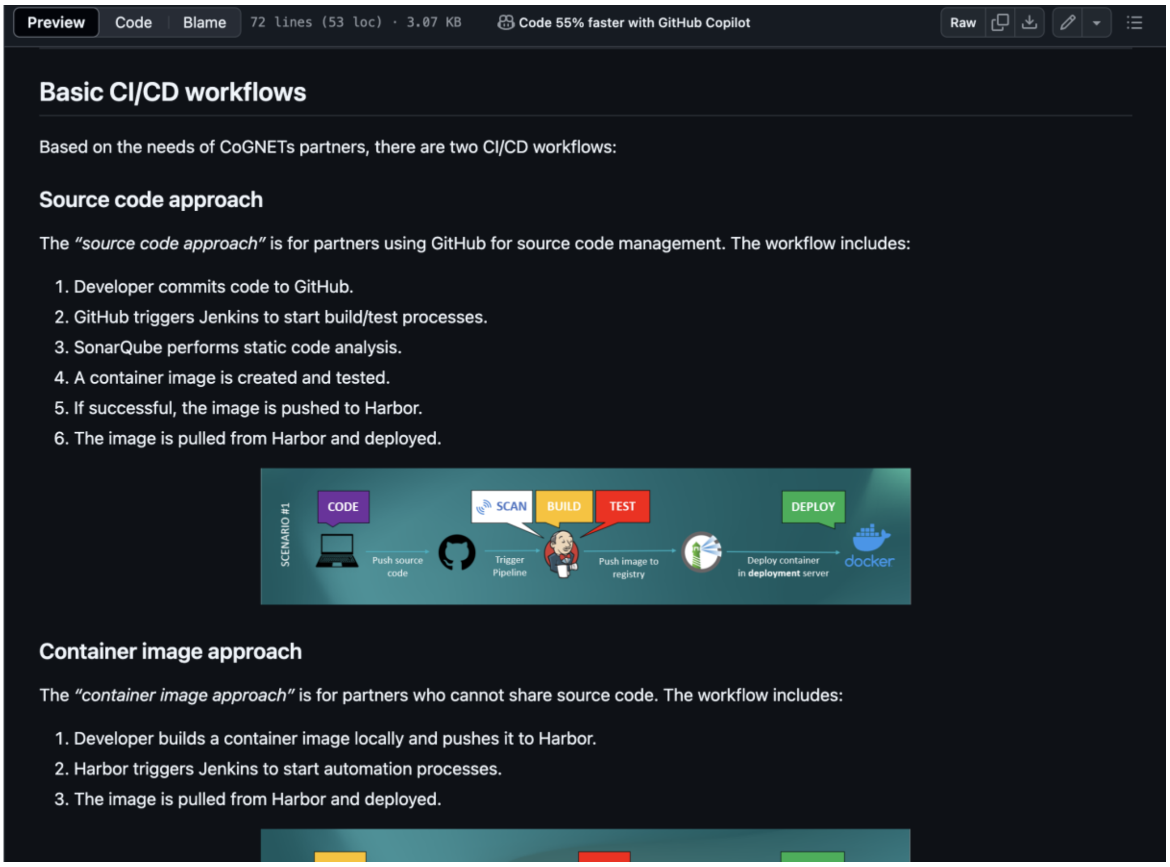
Figure 7 CoGNETs CI/CD tools user documentation

- **CI/CD Workflows**: This category provides an overview of the various CI/CD pipelines available, detailing how the previously mentioned tools are utilized within these

Funded by the European Union
Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

workflows (see Figure 8). This resource aims to enhance understanding of the integration and functionality of each tool in the CI/CD process.

Link: https://github.com/cognets-eu/ci-cd-documentation/blob/main/cicd_workflows.md


Preview
Code
Blame
72 lines (53 loc) · 3.07 KB
Code 55% faster with GitHub Copilot
Raw

Basic CI/CD workflows

Based on the needs of CoGNETs partners, there are two CI/CD workflows:

Source code approach

The *"source code approach"* is for partners using GitHub for source code management. The workflow includes:

1. Developer commits code to GitHub.
2. GitHub triggers Jenkins to start build/test processes.
3. SonarQube performs static code analysis.
4. A container image is created and tested.
5. If successful, the image is pushed to Harbor.
6. The image is pulled from Harbor and deployed.

Container image approach

The *"container image approach"* is for partners who cannot share source code. The workflow includes:

1. Developer builds a container image locally and pushes it to Harbor.
2. Harbor triggers Jenkins to start automation processes.
3. The image is pulled from Harbor and deployed.

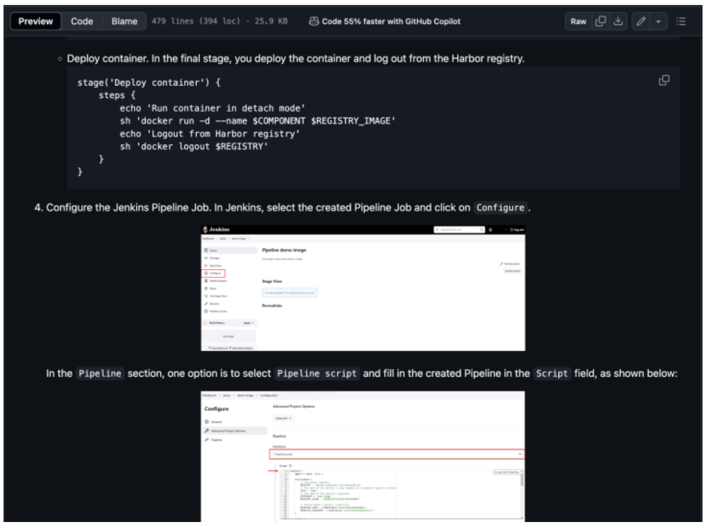
Figure 8 CoGNETs CI/CD workflows user documentation

- **CI/CD User Manual**: This category serves as the core of the CoGNETs CI/CD documentation, providing comprehensive integration and deployment steps. Each step is presented in detail, complemented by multiple screenshots and relevant code snippets where necessary (see Figure 9). This resource is designed to guide users through the CI/CD process effectively.

  Link: https://github.com/cognets-eu/ci-cd-documentation/blob/main/user_manual.md

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI



Preview
Code
Blame
479 lines (394 loc) · 25.9 KB
Code 55% faster with GitHub Copilot
Raw

- Deploy container. In the final stage, you deploy the container and log out from the Harbor registry.

```
stage('Deploy container') {
    steps {
        echo 'Run container in detach mode'
        sh 'docker run -d --name $COMPONENT $REGISTRY_IMAGE'
        echo 'Logout from Harbor registry'
        sh 'docker logout $REGISTRY'
    }
}
```

4. Configure the Jenkins Pipeline Job. In Jenkins, select the created Pipeline Job and click on `Configure`.

In the `Pipeline` section, one option is to select `Pipeline script` and fill in the created Pipeline in the `Script` field, as shown below:

Figure 9 CI/CD User Manual

Funded by the European Union
Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

3 COGNETS ML/DATA-OPS METHODOLOGY SPECIFICATION

3.1 COGNETS COMPONENTS SPECIFICATIONS

Before exploring the individual MLOps components in details, it is important to first present a high-level overview of each unit.

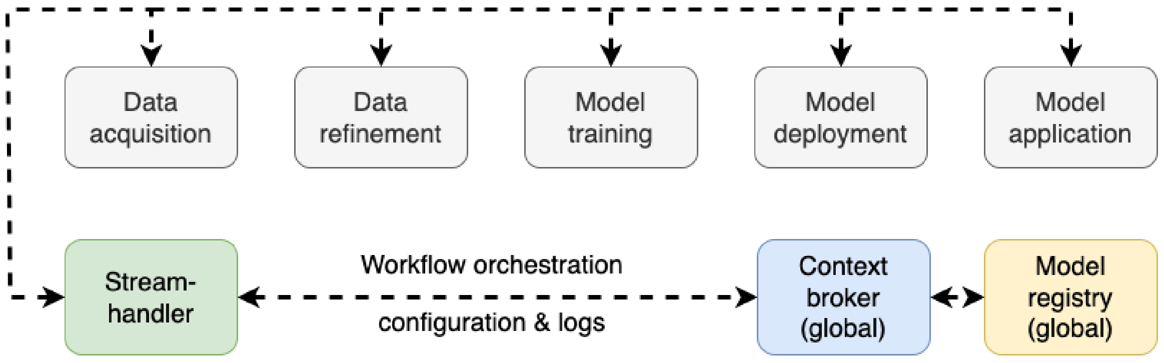


Figure 10 A simplified high level MLOps schema.

The MLOps high level schema (Figure 10) contains:

- The standard ML model creation pipeline, made of (1) data acquisition (2) data refinement and feature vectors creation (3) model training (4) model deployment and (5) model application.

  These modules represent essential steps in the ML model development process and are generally self-explanatory. The data acquisition module collects all relevant data, while the data refinement stage standardizes this data into a data representation format suitable for training. The training module then develops the ML model, which is stored in the model registry. Finally, the model is deployed and made available for use by the application service.

- INTRA's Streamhandler functions as an orchestrator.

  In modern MLOps, the modules (shown in grey in Figure 10) are typically interconnected and orchestrated using a streaming messaging service. While other methods of connecting modules are possible, the benefits offered by a modern streaming service have made it a crucial component of contemporary MLOps frameworks, including CoGNETs MLOps framework.

- FIWARE Context Broker (CB) functions as a data handler.

  Finally, a FIWARE Context Broker oversees the data flow, handling several key tasks, including retrieving and storing information as described Section 3.2.3 through the use of open Data Models. The FIWARE Context Broker can operate as a centralized unit or be localized within the CoGNETs node, acting as a Distributed Storage System.

Funded by the European Union
Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

3.2 COGNETS ML/DATA-OPS METHODOLOGY

The CoGNETs middleware consists of two primary components: the StreamHandler platform and the FIWARE Context Broker.

StreamHandler provides MLOps services to edge nodes which communicate with one another and with Streamhandler through the FIWARE Context Broker. Both components are distributed, ensuring resilience against service disruptions. Streamhandler modules and FIWARE Context Broker instances interact via real-time message streaming via Kafka messages. The exchanged messages are based on well-defined data structures which both Streamhandler and the FIWARE Context Broker define below.

3.2.1 INTRA's Streamhandler

The CoGNETs MLOps framework is powered by the Streamhandler platform, a high throughput distributed messaging, workflow and storage system. Streamhandler is built around the following key components:

- **Apache Kafka** [8] is utilised to handle and distribute MLOps requests through message streaming.
- **Apache Airflow** [9] is employed to define, execute and monitor the MLOps pipelines.
- **Streamhandler Workers** are responsible for consuming messages, preprocessing metadata, and deploying models.
- **MLFlow StreamHandler Model Registry** serves as a model repository management tool responsible for storing, reviewing and pulling ML Models for CoGNETs.
- **StreamHandler ELK logging stack** used for logging all MLOps transactions.

Figure 11 presents the Streamhandler's components and their roles within the CoGNETs MLOps framework. In this figure, components directly related to the Streamhandler are highlighted in blue, the FIWARE Context Broker is highlighted in yellow, and the PUC's cluster components, specifically the IoT nodes, are highlighted in green.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

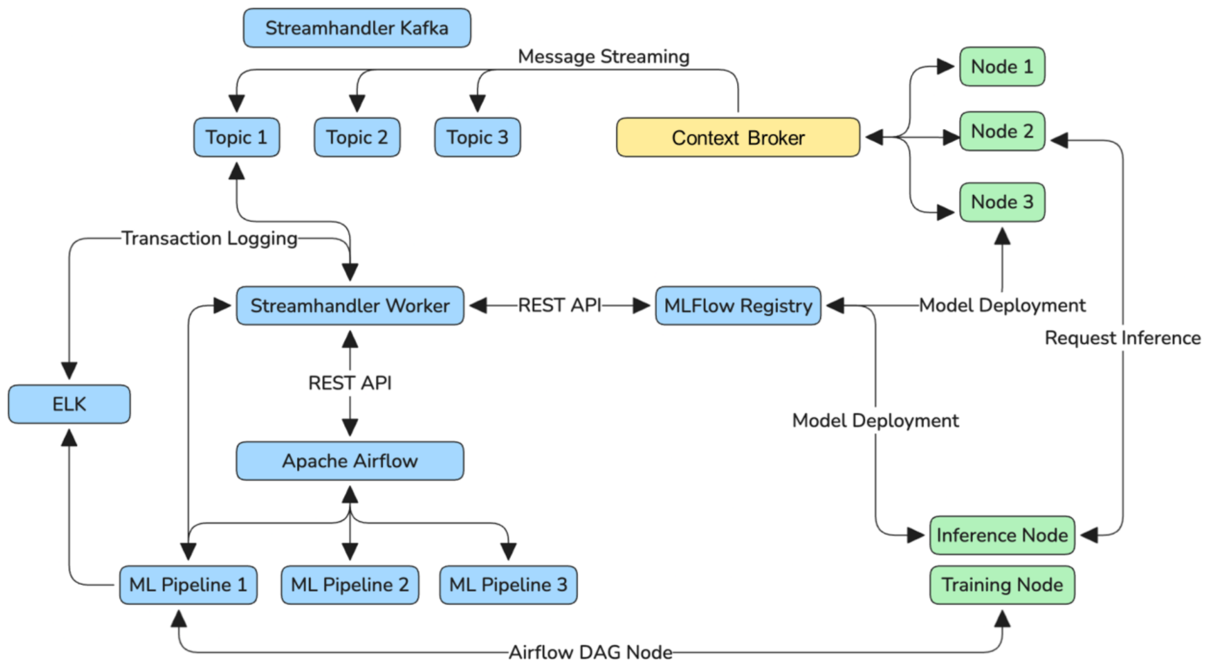


Figure 11 Streamhandler's components in CoGNETs MLOps framework

3.2.1.1 Streamhandler Message Streaming

The purpose of message streaming component is to facilitate the reception of requests and updates for MLOps-related tasks. These tasks will originate from the CogNETs nodes, and they include, but is not limited to, the following actions:

- Requesting the training of a model.
- Pulling a pre-trained model.
- Uploading or distributing a pre-trained model.
- Requesting the upload or distribution of a dataset used for training models.
- Requesting access to a dataset used for training models.

Streamhandler's Message Streaming is based on Apache Kafka, making it distributed and resilient, with the capacity to scale efficiently to handle millions of messages per second. The above presented actions are organized by defining relevant Kafka topics to which Streamhandler Workers are connected, enabling them to receive and process messages. In Apache Kafka, topics categorize streams of messages, with producers publishing data to these topics and consumers subscribing to retrieve and process the messages efficiently.

3.2.1.2 Apache Airflow

Apache Airflow is an open-source workflow management platform that enables robust workflow design by encoding MLOps pipelines as python scripts. This approach allows for fine-grained control over workflows, leveraging the full scripting capabilities of Python to define complex paths and easily clone or extend workflows. Moreover, its standard GUI provides valuable tools for workflow creation, management, and monitoring.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

In Airflow, MLOps pipelines and workflows are defined as Directed Acyclic Graphs (DAGs) written in pure Python. This approach guarantees that each workflow ends in one of a set of well-defined states, resulting in robust and efficient execution. Moreover, multiple tasks can be reliably chained together, enabling the creation of complex pipelines. This capability is particularly crucial within the CoGNETs middleware, as a unified MLOps stack must accommodate diverse requirements.

3.2.1.3 Streamhandler Workers

Streamhandler workers serve as context pre-processors, implementing the required business logic to bridge the gap between real-time message streaming and the execution of MLOps pipelines. Once a message is consumed, a worker's task is to decide which MLOps pipeline to execute and how to handle each result. Once a pipeline completes, the worker is notified, and appropriate messages are sent to the relevant Kafka topics. This ensures that requester nodes are informed of the pipeline's outcome and act accordingly.

From a software stack point of view, Streamhandler Workers are implemented as lightweight microservices using the FastAPI [10] Python framework. Streamhandler provides simple easily extensible worker templates, in the form of docker images, streamlining the creation of new workers. Moreover, these workers include pre-defined HTTP request handlers, as presented in Figure 12, that automatically log any request made, ensuring proper logging without interfering with the software development process.

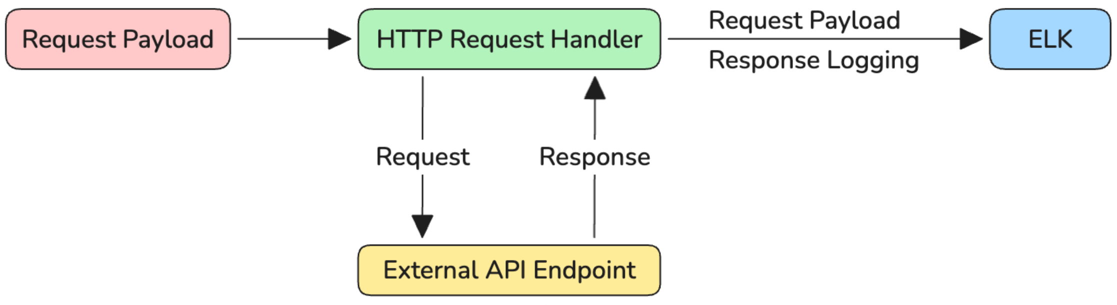


Figure 12 HTTP Request Handler with auto ELK logging

3.2.1.4 MLFlow Streamhandler Model Registry

The **MLFlow Streamhandler Model Registry** is responsible for storing and managing all ML models utilized within the CoGNETs framework. For example, once an ML training pipeline is complete, the resulting model and its associated metadata are stored to MLFlow, where they can be reviewed, distributed and managed. A Streamhandler worker service facilitates the integration between Apache Airflow and MLFlow Streamhandler Model Registry.

3.2.1.5 Streamhandler ELK logging stack

For, traceability, debugging and monitoring Streamhandler employes an instance of the ELK stack, which consists of Elasticsearch, Logstash and Kibana.

A dedicated Streamhandler Worker service is responsible for consuming logging messages and sequentially passing them to Logstash. Here, the logs are parsed and pre-processed before being sent to Elasticsearch for efficient storage. This streamlined flow ensures that all logging data is systematically organized and easily accessible.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

Kibana, along with the Elasticsearch API, is utilized to retrieve and review structured logging data, providing powerful visualization tools and dashboards. This integration allows developers and operators to monitor system performance, identify issues in real-time, and maintain comprehensive traceability of all operations within the CoGNETs network. By leveraging the ELK stack, Streamhandler enhances its ability to deliver reliable and transparent MLOps processes.

Given the diverse use cases that the CoGNETs middleware aims to cover, specialized MLOps-related **data indexes** will be dynamically created to categorize logging data according to their associated MLOps pipelines. This dynamic indexing allows for more granular organization and retrieval of logs, facilitating easier analysis and insights.

Based on these data indexes, custom data-collection views will be developed in Kibana, offering users an intuitive interface to explore and understand logging data. These views will enable users to filter, visualize, and analyse logs relevant to specific MLOps workflows, enhancing the overall monitoring and debugging experience.

The Streamhandler worker is responsible for ensuring that the appropriate data indexes are used for each logging entry, maintaining consistency and accuracy across the system. This architecture not only improves traceability but also empowers teams to quickly identify bottlenecks or issues within their MLOps pipelines, ultimately leading to more efficient operations and better decision-making.

3.2.2 The FIWARE Context Broker

The FIWARE Context Broker (CB), specifically the Orion-LD Context Broker implementation, functions as an open-source module designed to manage the lifecycle of context information. It adheres to both the FIWARE NGSIv2 specification and is compliant with ETSI NGSI-LD. ETSI NGSI-LD is formally defined as an Application Programming Interface (API) for managing context information, where context represents a collection of attribute values that characterize Digital Twins. As an open standard, all services implementing ETSI NGSI-LD API, overseen by the FIWARE Foundation, are inherently open source and available on platforms like GitHub.

The Context Broker is widely utilized across various domains to enhance operational efficacy. Its applications span Digital Twin frameworks, smart city dashboards, optimization of energy grids, traffic management, synchronization of Edge-Cloud computing, and environmental monitoring. It is particularly valuable in use cases that demand rapid responses to context changes. Characterized by a RESTful interface, the FIWARE Context Broker supports fundamental context information management operations, including the creation, update, deletion, and retrieval of entities.

It also allows clients to register external context providers and subscribe to changes in context information. These subscriptions can be configured with fine-grained filters, enabling asynchronous notifications delivered via HTTP callbacks or message brokers. Architecturally, it employs an entity-attribute-value model, featuring support for nested metadata which allows for the representation of complex Digital Twin structures and relationships within systems-of-systems.

The publish-subscribe mechanism offers flexibility in defining subscription rules based on criteria such as entity types, attribute conditions, or metadata filters. Notifications can be channelled through HTTP callbacks or integrated message streams like Apache Kafka. By default, the Orion-LD CB leverages MongoDB for real-time data persistence. While it primarily stores the "current state" of entities by replacing old attribute values upon update, NGSI-LD compliant brokers are also designed to store the temporal evolution of entities, which is crucial for applications like Machine Learning.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

Historical data handling has traditionally been managed by other FIWARE Generic Enablers such as Cygnus, Quantum Leap, and STH Comet, which subscribe to context changes to maintain historical archives, support analytics, and integrate with third-party storage solutions.

A significant strength of the FIWARE Context Broker is its adherence to open standards, FIWARE NGSIv2 and ETSI NGSI-LD APIs, which guarantees broad interoperability with other FIWARE Generic Enablers and external frameworks, promoting vendor-neutral integration. It supports "Distributed Operations" also known as Federation, allowing multiple NGSI-LD context brokers to interconnect and function cohesively, providing automatic distributed storage mechanism. This is facilitated through "Registrations", where one broker can inform others about the information it holds, enabling queries for that information to be routed correctly and potentially creating a network of brokers connecting different smart domains.

The effectiveness of this federation significantly relies on the availability of the Data Models defined in the Smart Data Models (SDM) program. The broker also provides context hosting services for NGSI-LD, including the management of @context definitions. An @context in JSON-LD serves primarily as a key-value list for expansion and compaction, allowing different applications to use aliases for the same terms. NGSI-LD supports various payload formats for JSON, including Normalized (default), Concise, and Simplified (key-values), offering flexibility for both machine and human readability.

NGSI-LD attributes can include sub-properties like unitCode for units and observedAt for timestamps. Querying entities can be performed using various parameters such as id, type, idPattern, attrs, pick/omit, scopeQ, q (using the NGSI-LD Query Language), geo-query, and lang. The modular, containerized architecture of the Context Broker supports horizontal scalability, making it suitable for high-throughput and large-scale enterprise deployments. It is further supported by a vibrant open-source community and recognized as a CEF Building Block for data spaces, benefiting from continuous improvements, professional support, and strict compliance with specifications.

3.2.3 Smart data models

The Smart Data Models (SDM) is a collaborative program jointly led by the FIWARE Foundation, TM Forum, IUDX, and OASC. It focused on developing a comprehensive catalogue of harmonised, domain-specific reference data models for context information management. The core objective is to enable genuine data interchange between organizations by providing these data models under open licenses, following the principles of agile standardization. The program aims to offer data models suitable for Digital Twins and Data Spaces.

These data models facilitate semantic interoperability, ensuring consistent data schemas across diverse smart applications. Their use mitigates the need for custom adapter development and reduces integration complexity, particularly in Digital Twin and data marketplace scenarios. They underpin cross-vendor solutions across various sectors, including but not limited to Smart Cities, Industry 4.0, Smart Agriculture, and Energy Management.

These data modes establish a shared vocabulary that enables seamless data sharing and sophisticated analytics across disparate systems. While compatible with the FIWARE platform and the NGSIv2 and NGSI-LD standards, the models can also be exported into several other formats. They assist cities in selecting and opening data across different solutions and support various data models from different standards. Their adoption helps define databases, applications, or systems to be compatible with other solutions for both internal and external use.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

The initiative's models are used or recommended by numerous organizations and initiatives, including DSBA, Microsoft, IUDX (India), Atos, OASC, and AWS.

Each Data Model is defined by four fundamental components: a machine-readable JSON Schema, human-readable documentation, a canonical URI for identification, and example payloads formatted for both NGSIv2 and NGSI-LD. The data models are hosted on GitHub and accessible via the https://smartdatamodels.org website, providing a centralized repository for discovering, contributing to, and managing the lifecycle of data models. All models are provided under a royalty-free license, promoting maximum adoption and innovation.

The JSON Schema serves as the single source of truth for the model definitions. These definitions can be exported into multiple representations including JSON, JSON-LD, CSV, SQL, YAML, DTDL, with plans for forthcoming RDF formats. Models progress through defined maturity stages -incubation, harmonization, and official release- via collaborative reviews and validation against real-world use cases to ensure their robustness and relevance.

The data models are structured into subjects, represented by GitHub repositories and domains, representing vertical sectors. There are thirteen identified domains, including Smart Cities, Smart Energy, Smart Environment, Smart Agrifood, Smart Water, Smart Health, Smart Manufacturing, and Smart Logistics, among others. The Smart Cities domain features many data models, covering concepts like ZEB (Zero Energy Building) with models for AirConditionerTerminal, BuildingZEB, Room, Sensor, Window, and others.

Smart Manufacturing includes models for Predictive Maintenance, such as AIPrediction, MaintenanceRequest, and ServiceTechnician, to name a few. Other domains or subjects include AAS (Asset Administration Shell) with a TimeSeries model, and Smart Robotics with OPC UA models like MachineTool. As of the last updates, there are over 1,200 data models, including official ones and those in the queue, distributed across these domains, with specific numbers cited for Smart Energy (+420), Smart Cities (+150), and Smart Sensoring (+130).

The initiative boasts over 120 participating organizations and more than 180 individual contributors. Publication and debugging processes are largely automated, enabling rapid updates. Documentation is available in eight languages.

The data models are based on real use cases and adhere open standards. Mapping of standards and ontologies is performed under specific conditions, requiring the original source to be open licensed (allowing free use, modification, and sharing with credit) and adopted by the market with available examples. The models can be customized to address local needs and are compatible with linked data. For handling units, NGSIv2 uses metadata, while NGSI-LD recommends the UnitCode sub-property.

The initiative operates under principles of agile standardization, including:

- "Do not reinvent the wheel".
- "Normalize real cases".
- "Be open".
- "Don't be overly specific".
- "Flat, not Deep".
- "Sustainability is key".

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

The initiative supports different collaboration tracks for users, contributors, developers, researchers, and stakeholders. Users can find data models via lists and a search tool on the website, which allows searching by properties, data models, or descriptions. Contribution is managed through Pull Requests on the GitHub repositories, with the incubated repository recommended for drafting new data models. Support is available through various channels, including live sessions, Slack, email, and GitHub issues.

An improved testing service allows multiple data model tests in one run in one go, offering extended documentation, local or remote execution, detailed error comments, and warnings. The roadmap for 2025 includes priorities such as evolving the pysmartdatamodels package[1] for validation and drafting data models, improving contribution tools, implementing new NGSI types like languageProperty, updating contexts, exporting documentation to formats like ReadTheDocs and PDF, creating explanatory videos, adding dates to metadata, generating synthetic data, porting ontologies/regulations (like BRICK, BIM, OGC, OSM), and visualizing data model structures. The initiative is a crucial technical building block for data interoperability within Data Spaces.

3.2.4 Mapping between PUCs sub-processes and CoGNETs ML/Data-Ops

MLOps schemas typically provide a standard framework for machine learning development. However, CoGNETs presents unique requirements that must be thoughtfully considered to design an effective solution. To begin this process, the schema developed by FIWARE, depicted in Figure 13, was utilized as a foundational reference. This schema offered a structured starting point for adapting and integrating the specific needs of CoGNETs MLOps. The distinct characteristics of CoGNETs were carefully analysed to ensure that the final MLOps design effectively addresses its requirements while maintaining alignment with established MLOps practices.

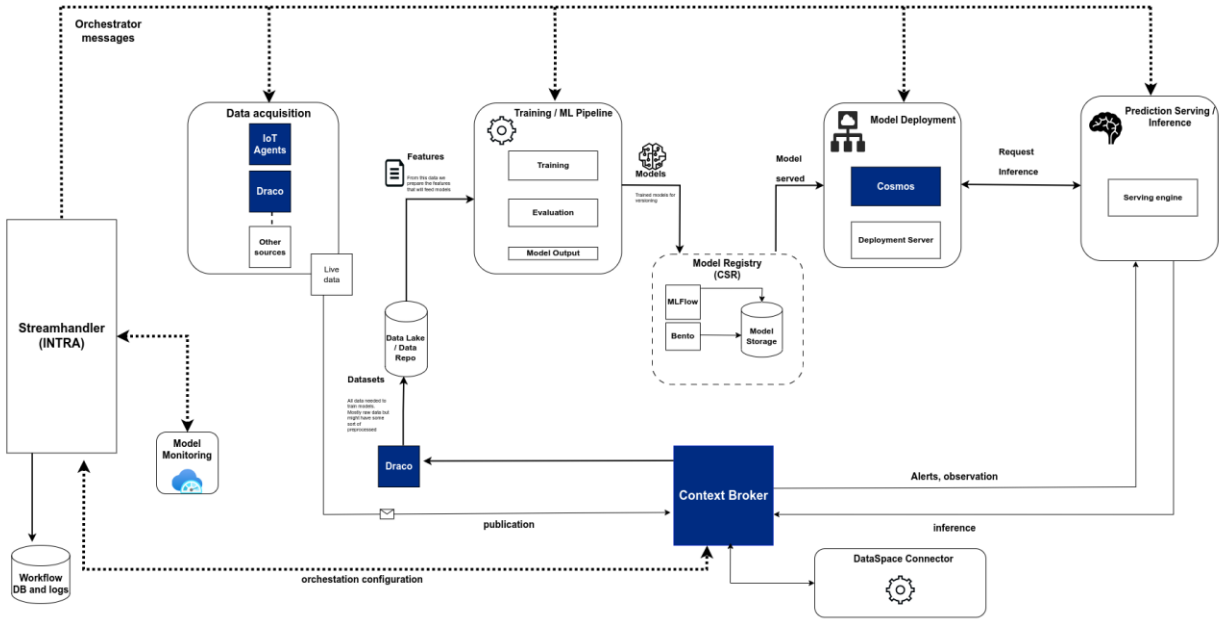


Figure 13 Initial MLOps schema.

To adapt the schema to the CoGNETs' needs, we first grouped the schema elements into various high-level sections, as presented in Figure 14. To enhance clarity, we separated the

[1]https://pypi.org/project/pysmartdatamodels

Funded by the European Union
Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

CoGNETs middleware (denoted in red in the image below) from the MLOps section, which has been divided into “custom MLOps” (purple section) and the central units (green sections).

While the CoGNETs middleware (red section) is self-explanatory, the reason for separating purple and green sections -and the difference between the two sections- is that the purple section is part of the PUC’s control, while the units within the green section is an integral and centralized element shared among all PUCs. In other words, the purple section can be perceived as a black box, operated for the purposes of each PUC (which may or may not rely on CoGNETs middleware), while green section provides core unit, connecting PUCs MLOps sections to the rest of the CoGNETs framework.

Clearly, CoGNETs prioritises a decentralized architecture; ideally, there would be no central units. However, certain restrictions necessitate that some information be stored centrally. These include the model registry as part of the CoGNETs middleware, a Streamhandler and a swarm-level Context Broker.

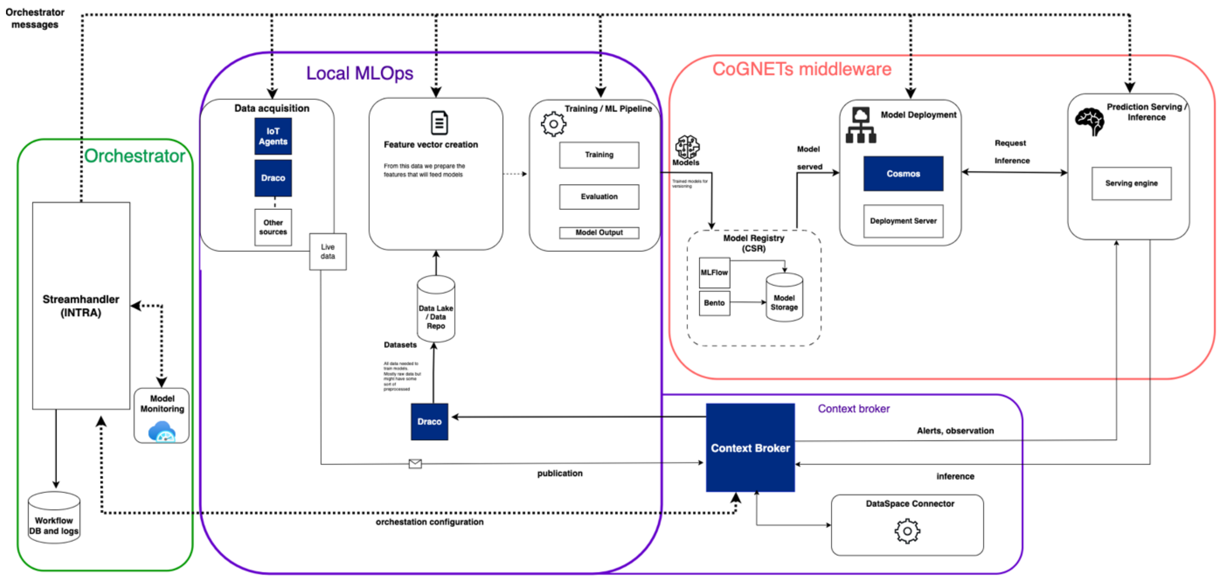


Figure 14 MLOps in relation to CoGNETs and other elements.

When describing the MLOps section, highlighted in purple, it is essential to focus on the ultimate objective, which is creating a ML model, regardless of the methodology employed. This underlying premise serves as a critical factor when determining the most effective approach to the MLOps schema. By prioritizing the final goal, informed decisions can be made to ensure the schema aligns with the CoGNETs goals. This involves considering whether to:

- Implement a unified procedure applicable to all PUCs, thereby standardizing the entire process or
- Allow each PUC to manage its own MLOps environment, requiring standardization only at the stage of deploying ML models to CoGNETs. This deployment is the point at which PUCs begin interacting with CoGNETs.

Funded by the European Union

Project funded by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation

Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

The first approach would be more rigorous for the PUCs, as it imposes certain guidelines that may not be necessarily for custom MLOps development. For example, the use of common models (e.g., Smart Data Models – see Section 3.2.3) would require each PUC not only to familiarize themselves with these data models but also to create new ones if suitable options do not exist. In other words, the first approach would potentially introduce additional overhead for PUC developers.

The second approach, which has since been adopted, grants the PUCs the freedom to develop their own MLOps. From the perspective of CoGNETs scalability, this approach is more effective as it minimizes requirements. The concept is presented in Figure 15.

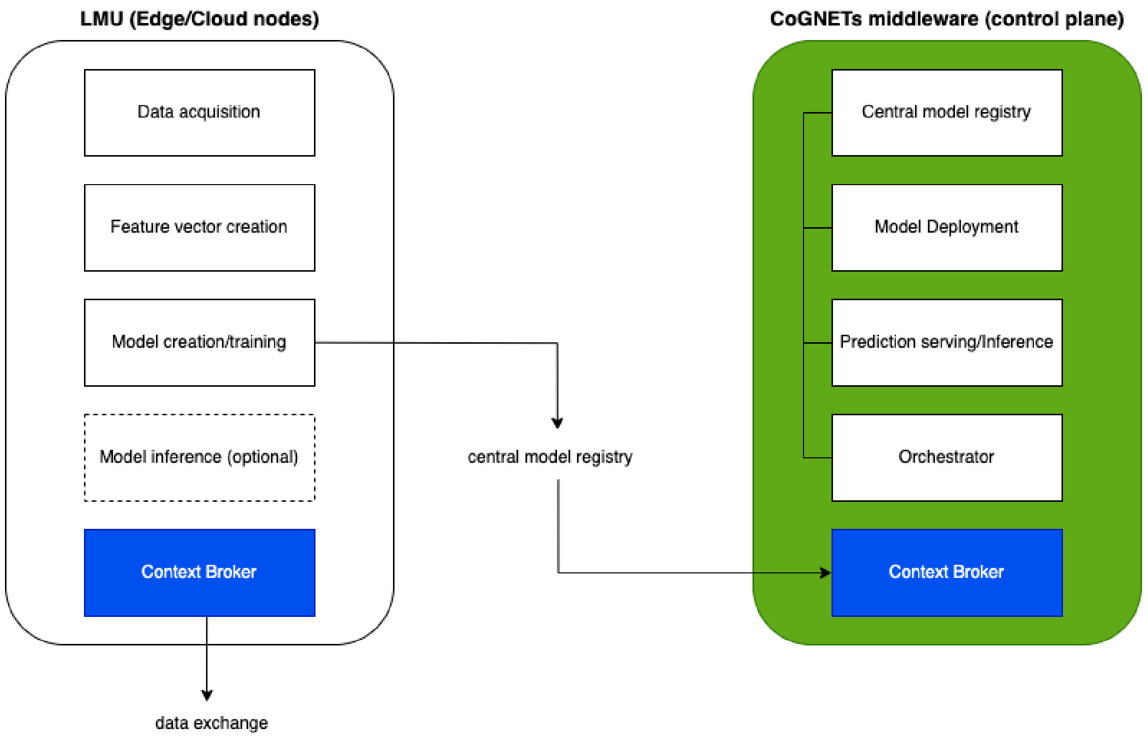


Figure 15 LMU (Local ML Unit) vs CoGNETs middleware.

Funded by the European Union
Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

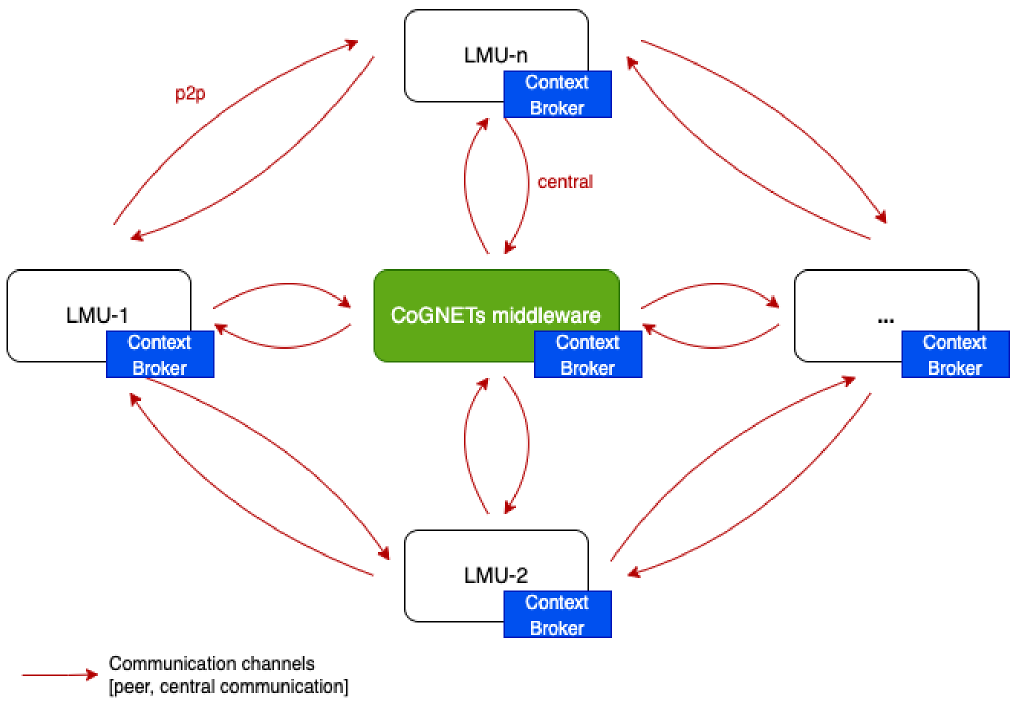


Figure 16 Deployment and interaction between LMUs (Local ML Unit) and CoGNETs middleware.

Figure 16 illustrates the process of creating ML models on edge devices, which subsequently utilize CoGNETs middleware to support a range of operations, including inference, federated learning, and resource access. The red arrows between LMUs (Local ML Units) indicate the ability of peer communication (as a part of decentralized architecture) next to expected LMU-middleware communication. This integration allows for efficient task management directly on edge devices, optimizing both performance and resource utilization.

In the remainder content of this section, we provide a detailed explanation of how the CoGNETs MLOps schema is applied to each of the CoGNETs PUCs. Additionally, we outline the correspondence between the aforementioned MLOps components and the subcomponents of the PUCs.

3.2.4.1 PUC1 Manufacturing

The industrial Pilot Use Case (PUC1) will adopt the MLOps framework as detailed at the beginning of Section 3.2.4. As it does not necessitate demanding data privacy measures, as for example PUC3 does due to the handling of highly sensitive information (see Section 3.2.4.3), PUC1 affords greater flexibility in its model-building strategies. This flexibility enables PUC1 to explore diverse methodologies, such as employing federated learning across distributed sources, or aggregating data collected from multiple robots to construct models. These approaches allow for solutions that optimize performance and adaptability within the industrial context.

Consequently, because PUC1 does not face stringent data privacy or other constraints, not many details or explanations are needed at this stage. This lack of complexity in PUC 1 requirements permits a more streamlined initial deployment, focusing on rapid experimentation and iteration to refine and enhance the models effectively.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

3.2.4.2 PUC2 Mobility

The PUC2 use case is positioned within the mobility sector, specifically focusing on electric vehicles and optimising battery consumption. Maintaining optimal cabin temperature consumes substantial energy resources, thereby reducing vehicle range. For instance, at very low temperatures, a substantial amount of energy is used for heating.

Automobile companies are actively tackling this challenge. To regulate cabin temperature, the vehicle first consults the Thermal Management System (TMS), represented as a lab model or the vehicle's digital twin (depicted as the blue cloud in Figure 17). If a discrepancy between the model and real conditions is detected, the vehicle uses pre-trained AI models, known as surrogate models (shown as the purple cloud in the image below). When these pre-trained models prove insufficient, edge training is conducted, where CoGNETs plays a crucial role.

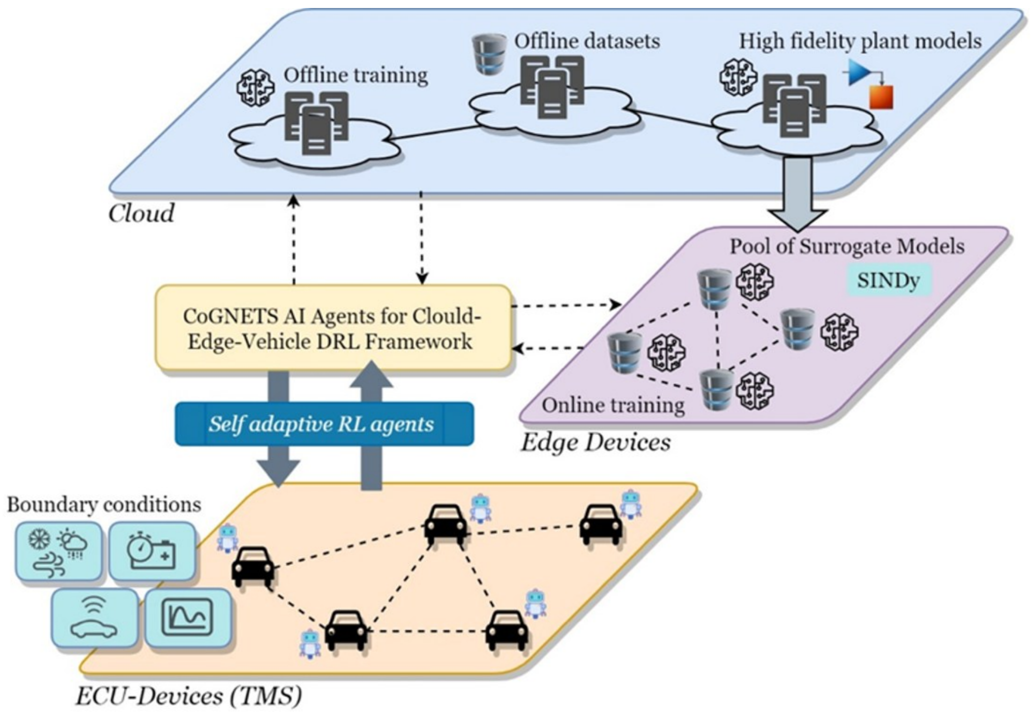


Figure 17 Various levels and approaches when tackling vehicle cabin temperature.

In the PUC2 use case, the nodes are vehicles. CoGNETs serves two key purposes in this scenario:

- Facilitating resource sharing.
- Enabling access to edge models of other vehicles.

The MLOps schema for this use case is adapted as presented in Figure 18.

Funded by the European Union
Project funded by
Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

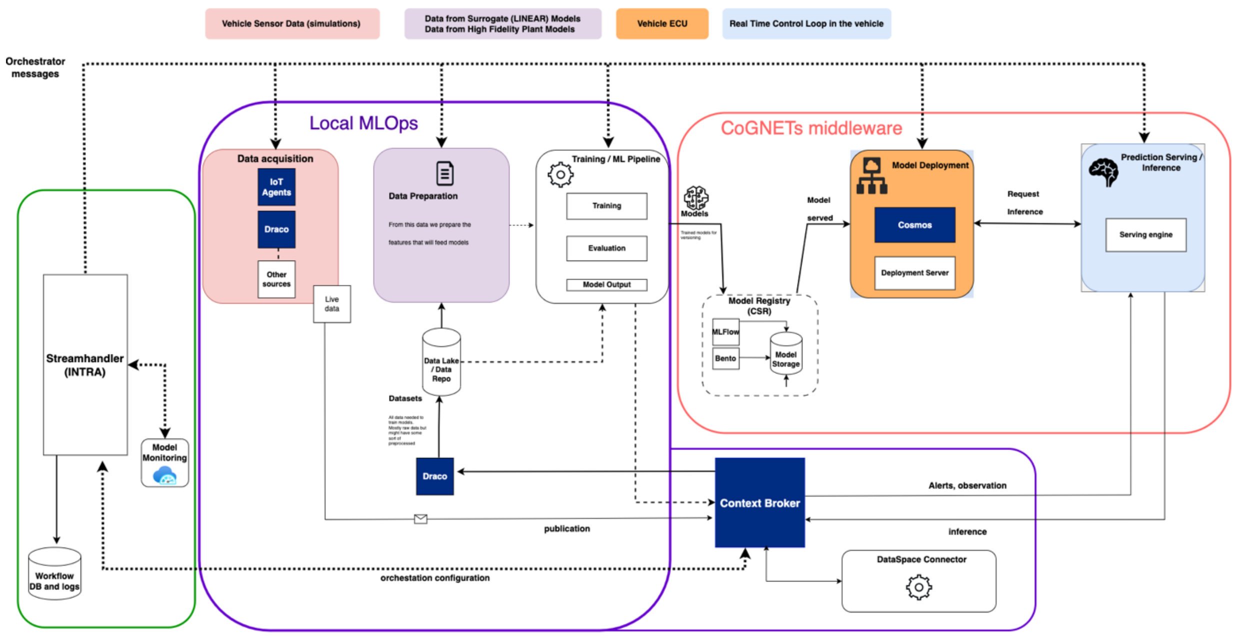


Figure 18 MLOps adapted to PUC2

Figure 18 shows the data processing workflow. Initially, data are collected from the vehicle sensors. This data is enriched with inputs from digital twins and surrogate models to provide additional context. Once enriched, the data are sent for training to develop the edge ML models that facilitate efficient, real-time decision-making within the vehicles.

3.2.4.3 PUC3 Healthcare

Minor adaptations are necessary to support the health use case presented in this sub-section. This use case involves multiple health departments, such as those treating patients with specific heart conditions. Each department operates its own local computer, functioning as a node within the CoGNETs framework. In addition to participating in the shared resources of the node swarm, there is a specific requirement that must be addressed for PUC3.

Due to the sensitive nature of the data involved in this type of applications, PUC3 nodes cannot share data across different nodes. To address this limitation, federated learning has been proposed as a solution. As indicated in Figure 19, local ML models are built on each node and then shared through a central model registry with the other nodes. The swarm-level context broker acts as a gateway, allowing local nodes to access all ML models.

Figure 19 illustrates the PUC3 CoGNETs swarm. In this figure, the resource-sharing mechanism of the CoGNETs middleware is highlighted in red, the nodes (or LMUs) are shown in purple, and the central units are represented in green. The PUC3 MLOps will utilize the MLOps framework outlined at the beginning of Section 3.2.4.

Funded by the European Union
Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

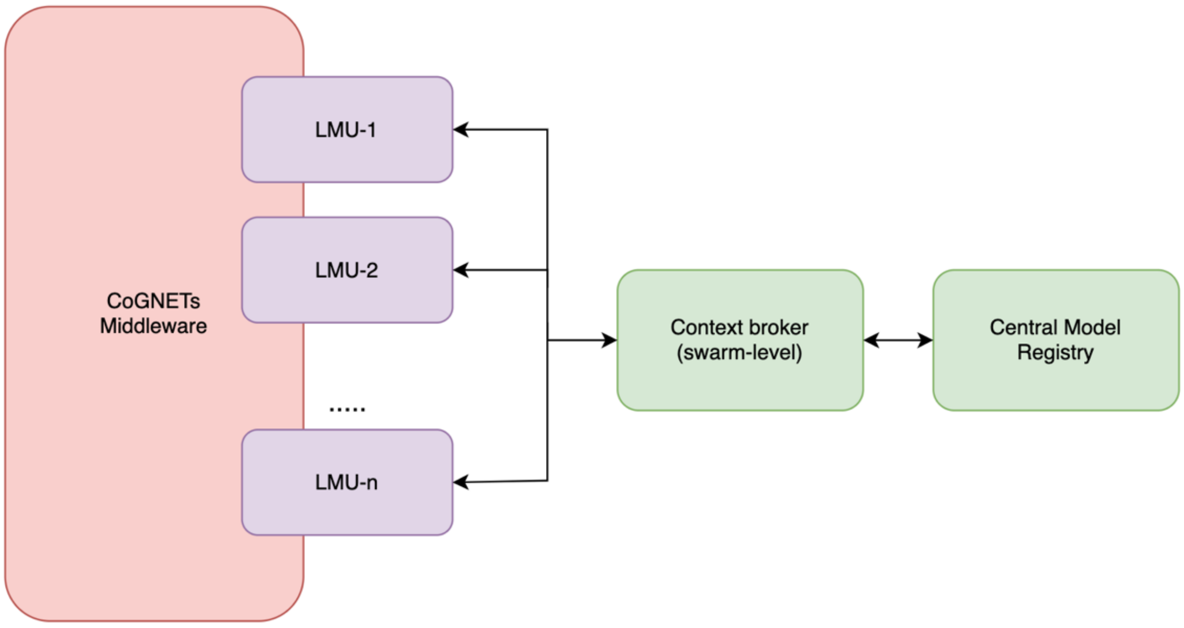


Figure 19 PUC3 CoGNETs swarm.

Funded by
the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs,
Education and Research EAER
State Secretariat for Education,
Research and Innovation SERI

4 SECURITY THROUGHOUT THE SYSTEM

CoGNETs uses a self-managed API Gateway to facilitate the communication among all components within the system. This approach enables the easy implementation of the following features:

- Strict rules for incoming requests, allowing only connections that originate from the API Gateway.
- Centralized API token-based authentication at the API Gateway level, ensuring that only authorized requests propagate through the CoGNETs middleware.
- Uniform enforcement of HTTPS connections. The API Gateway serves all its connected endpoints over HTTPS. Therefore, the SSL handling and maintenance is done centrally and the clients communicating over the gateway benefit from this since all API calls are done over HTTPS automatically.
- Simplified service discovery, as all endpoints are centrally listed and accessible.

API communication between CoGNETs services will be secured through a token-based authentication system, using the Keycloak Identity and Access Management Platform.

CoGNETs will log all inter-service transactions, which often contain sensitive or private information. To prevent any sensitive data from being permanently logged, Streamhandler Workers implement a rules-based data filtering process that removes such information during the logging pipeline.

Additionally, each Streamhandler service and node within the integrated testbed platform will be monitored by centrally deployed instance of Sentry server (configured on-premises). This server is responsible for collecting real-time bug & exception traces, as well as monitoring usage spikes that may indicate security breach attempts. This setup enables timely detection and management of runtime errors, which is especially important for CoGNETs services that do not use statically compiled languages (Python and JavaScript).

Finally, for each captured exception, real-time notifications are sent to the relevant development teams. These notifications may include emails, instant messages, and SMS alerts. The exception capture flow is presented in Figure 20.

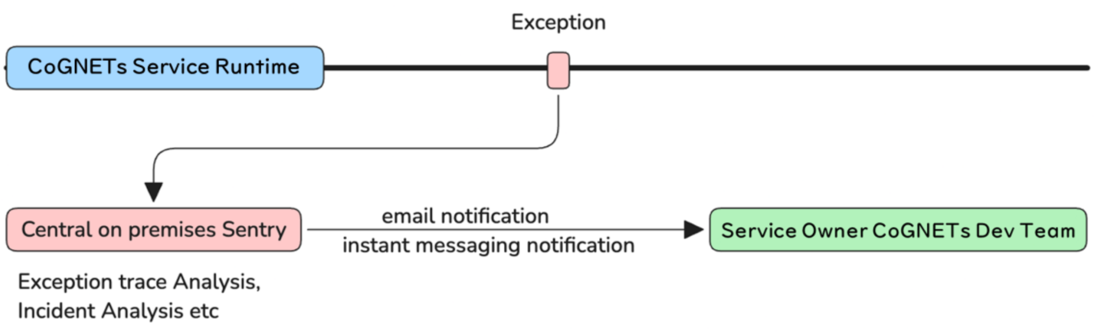

Exception
CoGNETs Service Runtime
Central on premises Sentry
email notification
instant messaging notification
Service Owner CoGNETs Dev Team
Exception trace Analysis,
Incident Analysis etc

Figure 20 Exception capture flow

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

5 CONCLUSIONS AND FUTURE WORK

In conclusion, this document details significant progress in advancing data interoperability and integration within the CoGNETs project by outlining the DevSecOps specifications and constructing an innovative ML/Data-Ops methodology with robust practices. These efforts establish a strong basis for seamless component integration within CoGNETs, provide guidelines for developing the CoGNETs testbed, and offer user manuals to support ongoing and future developments.

5.1 SUMMARY OF ACHIEVEMENTS

The achievements of the work presented in this document are summarised as follows:

- Specifications for incorporating new data interoperability paradigms were defined through the design of an advanced ML/Data-Ops framework utilizing the FIWARE Context Broker and the INTRA Streamhandler.
- DevSecOps and integration practices were enabled by design. The integration methodology and the CI/CD stack outlined in this document lay the groundfloor for CoGNETs partners to integrate their advanced IT, AI, and security components into the CoGNETs solution, mitigating potential risks with each new software delivery.
- Specifications and design principles for developing the CoGNETs testbed and realizing the architecture described in D2.1 [1] were established.
- A user manual and best practices were provided to the CoGNETs development team, ensuring a clear understanding and facilitating future project developments.

5.2 FUTURE STEPS

This sub-section outlines the future steps and directions for the work presented in this document:

- **Integration of ML/Data-Ops and Tools for CoGNETs Testbed Development:** This document details the methodology and specifications for CoGNETs DevSecOps and the ML/Data-Ops, providing a foundation for the ongoing development of the lab-based programming testbed. This initiative has already started and will be further enhanced through the active involvement of all CoGNETs components and PUCs. This collaborative effort will be crucial for validating the project's innovations and experiments, leading up to the initial demos scheduled for M18, which will be documented in D2.3 "Lab-based Testbed Deployment". Building on the work established in this document, along with the contributions from the remaining Tasks and Deliverables within WP2, the final phase of data integration and tools unification will be facilitated, paving the way for the second stage of demonstrations in WP5.
- **Training Sessions and Hands-On Workshops**: In the coming months, the focus will be on organizing additional training sessions and hands-on workshops. These initiatives aim to engage CoGNETs partners more deeply, ensuring they have a comprehensive understanding of the processes and can provide valuable feedback to optimize the CI/CD stack and MLOps components. This coordinated effort seeks to align all partners for the seamless integration of their solutions and components into the established CI/CD stack

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

and the project’s testbed. Moreover, this process will yield critical feedback to enhance the CI/CD pipeline, facilitating and automating functional, modular, and inter-modular testing. After receiving feedback, further workshops will be organized, including sessions on the MLOps framework.

- **MLOps Best Practices and Documentation**: As mentioned in Section 2.3 the CoGNETs DevSecOps pipeline is supported by a detailed user manual available in the GitHub repository[2]. Similar documentation will be provided for the ML/Data-Ops schema, designed to assist the developments within the CoGNETs framework. This effort aims to enhance the overall outcomes, resulting in a robust solution that supports and empowers IoT, Edge, and Cloud devices to optimise data processing and service provisioning.

[2] https://github.com/cognets-eu

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

REFERENCES

[1] "D2.1 Reference system architecture and validation planning of the implementation scenarios," CoGNETs Consortium.

[2] GitHub, [Online]. Available: https://github.com/.

[3] Jenkins, [Online]. Available: https://www.jenkins.io.

[4] SonarQube, [Online]. Available: https://www.sonarqube.org.

[5] Harbor, [Online]. Available: https://goharbor.io.

[6] Portainer, [Online]. Available: https://www.portainer.io/.

[7] Glances, [Online]. Available: https://nicolargo.github.io/glances/.

[8] Apache Kafka, [Online]. Available: https://kafka.apache.org/.

[9] Apache Airflow, [Online]. Available: https://airflow.apache.org/.

[10] FastAPI, [Online]. Available: https://fastapi.tiangolo.com/.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI