A Cognitive Serverless Framework for the Cloud-Edge Continuum

D2.5 COGNIT Framework - Architecture - e

Version 1.0

30 April 2025

Abstract

COGNIT is an AI-Enabled Adaptive Serverless Framework for the Cognitive Cloud-Edge Continuum that enables the seamless, transparent, and trustworthy integration of data processing resources from providers and on-premises data centers in the cloud-edge continuum, and their automatic and intelligent adaptation to optimise where and how data is processed according to application requirements, changes in application demands and behaviour, and the operation of the infrastructure in terms of the main environmental sustainability metrics. The aim of this incremental version of the COGNIT Framework Architecture report is to provide an overview of the Project's overall development status, offer a summary of the work done in the Fourth Research & Innovation Cycle (M22-M27), and identify the priorities for the Fifth and last Research & Innovation Cycle (M28-M33).

This project is funded by the European Union's Horizon Europe research and innovation programme under Grant Agreement 101092711 – SovereignEdge.Cognit

Deliverable Metadata

Project Title:	A Cognitive Serverless Framework for the Cloud-Edge Continuum
Project Acronym:	SovereignEdge.Cognit
Call:	HORIZON-CL4-2022-DATA-01-02
Grant Agreement:	101092711
WP number and Title:	WP2. Adaptive Cloud-Edge Serverless Framework Architecture
Nature:	R: Report
Dissemination Level:	PU: Public
Version:	1.0
Contractual Date of Delivery:	31/03/2025
Actual Date of Delivery:	30/04/2025
Lead Author:	Marco Mancini (OpenNebula
Authors:	Mirko Stojiljković (OpenNebula), Christophe Ponsard (CETIC), Monowar Bhuyan (UMU), Malik Bouhou (CETIC), Aritz Brosa (Ikerlan), Cristina Cruces (Ikerlan), Martxel Lasa(Ikerlan), Jean Lazarou (CETIC), Fátima Fernández (Ikerlan), Aitor Garciandia (Ikerlan), Torsten Hallmann (SUSE), Philippe Massonet (CETIC), Nikolaos Matskanis (CETIC), Deins Darquennes (CETIC), Daniel Olsson (RISE), Mikel Irazola (Ikerlan), Álvaro Puente (Ikerlan), Thomas Ohlson Timoudas (RISE), Paul Townend (UMU), Iván Valdés (Ikerlan).
Status:	

Document History

Version	Issue Date	Status[1]	Content and changes
0.1	22/04/2024	Draft	Initial Draft
0.2	24/04/2024	Peer-Reviewed	Reviewed Draft
1.0	30/04/2024	Submitted	Final Version

Peer Review History

Version	Peer Review Date	Reviewed By
0.1	24/04/2024	Joel Höglund (RISE)
0.1	24/04/2024	Alberto P. Martí (OpenNebula)

Summary of Changes from Previous Versions

First Version of Deliverable D2.5

[1] A deliverable can be in one of these stages: Draft, Peer-Reviewed, Submitted, and Approved.

Executive Summary

Deliverable D2.5, released at the end of the Fourth Research & Innovation Cycle (M27), is the first incremental version of the COGNIT Framework Architecture report (D2.4) in WP2 “Adaptive Cloud-Edge Serverless Framework Architecture”. This report provides an overview of the Project's overall development status and offers a summary of the work done in the Fourth Research & Innovation Cycle (M22-M27).

During the Fourth Research & Innovation Cycle (M22-M27), and in line with the expected contribution of each Software Requirement towards meeting the Project’s Milestones and global KPIs, the Project has focused its efforts on releasing the third version (release 3.0) of the COGNIT Framework, which is formed by these main components:

- **Device Client.** It allows to offload function executions on the cloud-edge continuum using a Serverless paradigm and then to perform different tasks such as function execution, transferring data from external sources and uploading data from the device itself.
- **COGNIT Frontend.** It is the main entry point for the COGNIT service. It authenticates the device client that would like to use the service and returns a token that can be used by the device for interacting with the COGNIT components. It provides different endpoints for uploading application requirements, storing functions and data, and it returns information about the most suitable resources where the device can offload the execution of the functions.
- **Edge Cluster.** It is a semi-autonomous complete environment with a dedicated software-defined resource set providing an abstraction layer enabling interoperability across the multi-provider cloud-edge infrastructure. Edge Clusters allow the execution of functions that are offloaded by the devices within a hardened and secure environment called **Serverless Runtime**.
- **Cloud-Edge Manager**. It is responsible for managing the highly distributed and heterogeneous resources of the cloud-edge continuum and the lifecycle of Serverless Runtimes within each Edge Cluster for the execution of the functions offloaded by the devices.
- **AI-Enabled Orchestrator**. It manages and optimises the lifecycle of Edge Clusters and the Serverless Runtimes within each Edge Cluster according to the application requirements, infrastructure and virtual resource usage, and energy-aware policies.

In connection with those components, the Project has delivered progress specifically in those software requirements needed to achieve the third release of the COGNIT Framework, and to allow the Use Cases to perform their own research and development activities and implement their own applications in this cycle (M22-M27), using the Device Client SDK.

The features of the release 3.0 of the COGNIT Framework include:

- Release 3.0 of the Device Client SDK (both for Python and C) that provides the interface to offload from the device the execution of Python functions using both Python and C clients.

- Release 3.0 of the COGNIT Frontend that is the main entry point for the COGNIT service. It authenticates the device that would like to use the service, returning an authentication token to the device. It also queries the AI-Enabled Orchestrator for the most suitable Edge Cluster for the device to offload the execution of the functions and returns an Edge Cluster context.
- Release 3.0 of the Edge Cluster Frontend Release that is responsible for executing functions that are sent by the Device Client and providing the results to the Device Client itself.
- Release 3.0 of the AI-Enabled Orchestrator responsible for managing and optimising the lifecycle of the Serverless Runtimes within each Edge Cluster, according to infrastructure and virtual resource usage, and energy-aware policies.

The present incremental report (Deliverable D2.5) includes a list of research and development priorities for the Fifth Research & Innovation Cycle (M28-M33).

Apart from the overview provided in this report, specific research and development activities performed in WP3 “Distributed FaaS Model for Edge Application Development” (related to the Device Client, the COGNIT Frontend, the Edge Cluster Frontend, and the Secure and Trusted Execution of Computing Environments) are described in detail in reports D3.4 “COGNIT FaaS Model - Scientific Report” and D3.9 “COGNIT FaaS Model - Software Source”, whereas those performed in WP4 “AI-Enabled Distributed Serverless Platform and Workload Orchestration” (related to the Cloud-Edge Manager, the AI-Enabled Orchestrator, and the Energy Efficiency Optimization in the Multi-Provider Cloud-Edge Continuum) are described in reports D4.4 “COGNIT Serverless Platform - Scientific Report” and D4.9 “COGNIT Serverless Platform - Software Source”; details about the COGNIT software integration and the verification scenarios of each software requirements can be found in report D5.5 “Use Cases - Scientific Report”.

During this cycle, the **OpsForge** tool (see Deliverable D5.8) has been updated for the automated deployment of the COGNIT Platform in a target infrastructure, being a public cloud provider such as AWS or an on-premise datacenter. Additionally, Deliverable D5.11 includes a demonstration of some of the capabilities of the COGNIT Framework using the COGNIT testbed.

This deliverable has been released at the end of the Fourth Research & Innovation Cycle (M27), and will be updated with a self-consistent release at the end of the last research and innovation cycle (M33).

Table of Contents

Abbreviations and Acronyms

AI	Artificial Intelligence
API	Application Programming Interface
ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge (From MITRE)
AWS	Amazon Web Services
CD	Continuous Delivery (Deployment)
CIA	Confidentiality Integrity Availability
DaaS	Data as a Service
FaaS	Function as a Service
FFD	First-Fit Decreasing
IAM	Identity and Access Management system
IP	Internet Protocol
ITSRM	Information Technology Security Risk Management Methodology
JSON	Javascript Object Notation
JWT	JSON Web Token
ML	Machine Learning
OBS	Open Build Service
REST	Representational State Transfer
SysML	System Modeling Language
SDK	Software Development Kit
UML	Unified Modeling Language
VM	Virtual Machine

Introduction

The version of the COGNIT Framework Architecture report (Deliverable D2.4), released in M21, included a summary of Use Cases requirements, an analysis of sovereignty, sustainability, interoperability, and security requirements, the design and architecture specifications of the COGNIT Framework (second version), and the methodology for verification.

This incremental version (Deliverable D2.5) is to provide an overview of the Project's overall development status, offer a summary of the work done in the Fourth Research & Innovation Cycle (M22-M27), and identify the priorities for the Fifth Research & Innovation Cycle (M28-M33).

D2.5 is a living document that is composed of an introduction and three main sections:

- Section 1 provides an overview of the Project's overall development status, including the current status of each Software Requirement towards completion, and an update on the expected contribution of each of them towards meeting the Project’s Milestones and global KPIs.
- Section 2 provides an up-to-date overview of the readiness and maturity level of each component of the COGNIT Framework Architecture, and describes which (and how) specific software requirements have been addressed during the Fourth Innovation Cycle (M22-M27).
- Section 3 provides a brief summary of research and development priorities for the Fifth Research & Innovation Cycle (M28-M33).
- Section 4 provides more details about the transversal security concern by updating the initial risk analysis and summarizing the recommendations at the security architecture level.

The document ends with a conclusion section.

1. Overall Development Status

This section provides an overview of the Project's overall development status.

1.1. Update of User Requirements

New User Requirements

No additional user requirements have been identified during the Fourth Research & Innovation Cycle (M22-M27).

1.2. Update of Software Requirements

Minor amendments to existing Software Requirements

Existing Software Requirements have not been subject to any minor amendments during the Fourth Research & Innovation Cycle (M22-M27).

Existing Software Requirements whose scope has been modified

Existing Software Requirements have not been subject to any modification during the Fourth Research & Innovation Cycle (M22-M27).

New Software Requirements

No additional Software Requirements have been defined during the Fourth Research & Innovation Cycle (M22-M27).

1.3. Software Requirement Progress

The table below shows the status of each Software Requirement towards completion, following a simple colour code: – for activities that have not started yet or are on hold because of dependencies, ↻ for activities in progress, and ✓ for completed activities:

Software Requirements	Cycle 1 (M4-M9)	Cycle 2 (M10-M15)	Cycle 3 (M16-M21)	Cycle 4 (M22-M27)	Cycle 5 (M28-M33)
Device Client					
SR1.1 Interface with COGNIT Frontend			↻	✓	–
SR1.2 Interface with Edge Cluster			↻	✓	–
SR1.3 Programming languages	↻	↻	↻	✓	–
SR1.4 Low memory footprint for constrained devices	–	–	–	✓	–
SR1.5 Security	–	–	↻	✓	–
SR1.6 Collecting Latency Measurements			–	↻	–
COGNIT Frontend					
SR2.1 COGNIT Frontend			↻	✓	–
Edge Cluster					
SR3.1 Edge Cluster Frontend			↻	✓	–
SR3.2 Secure and Trusted Serverless Runtimes	↻	↻	✓	–	–
Cloud-Edge Manager					
SR4.1 Provider Catalogue	–	–	–	–	–

SR4.2 Edge Cluster Provisioning	–	–	–	–	–
SR4.3 Serverless Runtime Deployment	↻	↻	✓	–	–
SR4.4 Metrics, Monitoring, Auditing	↻	↻	↻	↻	–
SR4.5 Authentication & Authorization	↻	–	↻	✓	–
SR4.6 Plan Executor			–	✓	–
AI-Enabled Orchestrator					
SR5.1 Building Learning Models	↻	↻	↻	✓	–
SR5.2 Smart Management of Cloud-Edge Resources	↻	↻	↻	↻	–
Secure and Trusted Execution of Computing Environments					
SR6.1 Advanced Access Control	↻	↻	↻	↻	–
SR6.2 Confidential Computing	↻	↻	↻	↻	–
SR6.3 Federated Learning	–	–	–	–	–

Table 2.1. Current status of each Software Requirement towards completion, per research & innovation cycle.

1.4. Global KPIs Progress

The table below shows in which Milestone each Software Requirement is expected to contribute to meet the Project's global KPIs. Here we are marking in green activities actually completed by Milestone 4, those that have already started but not complete yet, and those to be started later on during the Project:

MEASURABLE & VERIFIABLE RESULT	KPI	Device Client						COGNIT Frontend	Edge Cluster		Cloud-Edge Manager						AI-Enabled Orchestrator		Secure & Trusted Execution of Computing Environments		
		SR1.1	SR1.2	SR1.3	SR1.4	SR1.5	SR1.6	SR2.1	SR3.1	SR3.2	SR4.1	SR4.2	SR4.3	SR4.4	SR4.5	SR4.6	SR5.1	SR5.2	SR6.1	SR6.2	SR6.3
Deployment of large-scale, highly distributed data processing environments.	**[KPI1.1]** Framework tested to scale up to tens of thousands of distributed nodes.										4										
Adaptation by offering a scalable monitoring system that enables the use of AI/ML techniques and methods for the smart operation of cloud-edge environments.	**[KPI1.2]** New AI-enabled orchestration demonstrated through Validation Use Cases.													4				4			
Adaptation by enabling the elasticity of the infrastructure with automated provisioning of edge PoPs.	**[KPI1.3]** Full provision of edge PoPs in "1-click" and in less than 10 minutes.											4									
Deployment and operation of edge cloud environments incorporating infrastructure resources across the whole cloud-edge continuum.	**[KPI1.4]** Deployment of a geo-distributed Testbed with resources from on-premises datacenters, cloud provider, edge provider, 5G provider, and on-premises far edge.										4										
Ability to operate both the infrastructure layer and the virtualized resources using a single pane of glass.	**[KPI1.5]** Unified GUI, CLI, and API able to manage all resource layers across the continuum.										4	4	<								
Seamless access for end users.	**[KPI2.1]** Meet experience level agreements with dynamic needs without intervention of the user.						4									4		4			
Application developers can define the execution environment for the FaaS Runtimes.	**[KPI2.2]** Definition of Execution Context, Base Image, Communication Patterns, and Deployment Requirements for performance, cost, security and energy requirements.																				
Developers can change the deployment requirements of the FaaS Runtime according to execution flow.	**[KPI2.3]** Migration across edge PoPs to meet application needs.															4					
Early and continuous security assurance of FaaS Runtimes.	**[KPI2.4]** Security processes and controls automated and integrated into the runtime lifecycle following a DevSecOps approach.									<											
Efficient orchestration of serverless workloads across the Continuum.	**[KPI3.1]** Backward compatibility with virtualization tech (e.g. KVM, Firecracker) and containers (e.g. K8s).										4		<								
Able to migrate workloads in geo-distributed edge clouds, combining edge/cloud resources.	**[KPI3.2]** Migrate workloads across the Continuum with minimal downtime.											4				4					
Intelligent self-adaptation of FaaS Runtime to changes in application behavior and data variability.	**[KPI3.3]** Automatic scale up/down of the microVM running the FaaS Runtime.													4		4					
Use of confidential and trusted computing for secure computation of private data at the edge/cloud.	**[KPI3.4]** Use case demonstrating isolation of sensitive data as it's being processed.																			4	4
Dynamic load balancing to adapt to changes in application needs and cloud-edge infrastructure.	**[KPI4.1]** Scalable, multi-variate AI-enabled modeling and optimization techniques.																	4			
Intelligent adaptation of cloud-edge continuum management.	**[KPI4.2]** New methods to create and retrain ML-based predictive behavioral forecasts for all major management considerations faced by cloud-edge continuum environments.																				
Hierarchical multi-constraint resource management layer to minimize environmental impact.	**[KPI4.3]** Reduction of at least 10% energy consumption compared with manual edge deployment.																	4			
Orchestration mechanisms to enforce coherent global security and governance policies.	**[KPI4.4]** Able to automate a European security policy on a multi-provider edge deployment.																		4		

Table 2.2. Expected contribution of each Software Requirement towards meeting the Project's global KPIs, by Milestone.

2. Work Done in Fourth Research & Innovation Cycle (M22-M27)

During the Fourth Research & Innovation Cycle (M22-M27) the Project has mostly focused on those software requirements needed to achieve Milestone 4 by M27 and to release an updated version of the COGNIT Framework.

During this cycle, the tool **OpsForge** has been updated in order to allow the user to deploy the main components of the COGNIT Framework: COGNIT Frontend, Cloud Edge-Manager, AI-Enabled Orchestrator in order to setup a Cloud-Edge infrastructure to manage Serverless Runtimes needed to offload the function execution from the devices. Furthermore, a new version of the Device Client SDK has been released in order to support both Python and C languages.

The features of the 3.0 release of the COGNIT Framework include:

- The **Device Client** SDK for both Python and C provides the interface to define the application requirements, to request the most suitable Edge Cluster to offload the execution of Python functions from the device.
- The **COGNIT Frontend** provides a REST API for the authentication of the device client, for uploading application requirements, storing functions and data, and it returns information about the most suitable resources where the device can offload functions.
- The **Edge Cluster Frontend** provides a REST API for offloading functions from the Device Client in the Edge Cluster. It acts as a load balancer for the Serverless Runtimes that are responsible for executing the functions offloaded by the devices.
- The **Cloud-Edge Manager** has been enhanced with the Plan Executor that executes the plans produced by the AI-Enabled Orchestrator for the workload optimization of the Serverless Runtimes within an Edge Cluster.
- The **AI-Enabled Orchestrator** allows the optimal placement, scaling and load balancing of Serverless Runtimes across the resources provisioned and managed by the **Cloud-Edge Manager** in order to optimise the energy consumption, according to the new architecture defined in Deliverable 2.4.

These features have been developed in a coordinated way between WP3 and WP4. The new software components and extensions to meet the software requirements have been specified, developed, and tested within the work packages WP3 and WP4. The integration of new functionalities has been verified and demonstrated within WP5 and reported in Deliverable D5.5.

The following section summarises, per component, the work that has been done as part of the Fourth Research & Innovation Cycle (M22-M27), including the completed and pending tasks associated with each of the software requirements that have been active during the cycle.

2.1. Device Client

SR1.1 Interface with COGNIT Frontend

Status: COMPLETED

Completed Tasks:
The current implementation of the Device Client enables the device to communicate with the COGNIT Frontend for authentication, for uploading application requirements, for uploading functions, and for requesting the Edge Cluster where to offload the execution of functions. The device client can upload data to a MinIO service available in the COGNIT Framework.

SR1.2 Interface with Edge Cluster

Status: COMPLETED

Completed Tasks:
The current Device Client implementation supports executing functions on the Edge Cluster through the client Python module methods. The device client can upload data to a MinIO service available in the Edge Cluster.

SR1.3 Programming languages

Status: COMPLETED

Completed Tasks:
The Device Client supports Python based function offloading. The Device Client is implemented both as a Python module and as a C library.

SR1.4 Low memory footprint for constrained devices

Status: COMPLETED

Completed Tasks:
The C version of the Device Client is able to run under the 500kB threshold of memory footprint.

SR1.5 Security

Status: COMPLETED

Completed Tasks:
The device client authenticates with the COGNIT Frontend and then it uses the token returned by the COGNIT Frontend for subsequent requests to the COGNIT Frontend and to the Edge Cluster Frontend. It uses secure channel communication with the COGNIT Frontend and Edge Cluster Frontend.

SR1.6 Collecting Latency Measurements

Status: IN PROGRESS

Completed Tasks:
The thread to enable the latency measurement mechanism is already in place in the State machine within the Device Client.

Pending Tasks:
The measurement of the actual latency to each Edge Cluster Frontend, and the action of submitting it, so COGNIT is aware of the latencies of the Device Client to each of the Edge Clusters present within the system is still in progress.

2.2. COGNIT Frontend

SR2.1 COGNIT Frontend

Status: COMPLETED

Completed Tasks:
The COGNIT Frontend allows devices to authenticate and it returns a Biscuit token once the device is authenticated. It allows the device to upload application requirements, to upload functions, and to query for the optimal Edge Cluster to offload the execution of functions. A MinIO service has been integrated into the COGNIT framework to provide a global endpoint for uploading data from the devices.

2.3. Edge Cluster

SR3.1 Edge Cluster Frontend

Status: COMPLETED

Completed Tasks:
The Edge Cluster Frontend allows devices to offload the execution of the function in one of the Serverless Runtimes available in the Edge Cluster. A MinIO service has been deployed and integrated in the Edge Cluster to upload data from the devices.

SR3.2 Secure and Trusted Serverless Runtimes
Status: COMPLETED
Completed Tasks: The Serverless Runtime exposes a REST API interface for executing Python functions uploaded to the COGNIT Frontend by the Device Client. An image has been built in the Cloud-Manager for the deployment of the Serverless Runtime. The images (from the base one) with specific libraries (e.g. Python libraries for image segmentation) have been built according to the needs of the different Use Cases. The images are ready to incorporate the Trusted Computing Execution Environment once it is available in the COGNIT Framework.

2.4. Cloud-Edge Manager

SR4.1 Provider Catalogue
Status: NOT STARTED YET

SR4.2 Edge Cluster Provisioning
Status: NOT STARTED YET

SR4.3 Serverless Runtime Deployment
Status: COMPLETED
Completed Tasks: A template has been defined for the deployment of the Serverless Runtime. The Cloud-Edge Manager API to manage the lifecycle of Serverless Runtimes is based on OpenNebula Oneflow.

SR4.4 Metrics, Monitoring, Auditing
Status: IN PROGRESS
Completed Tasks: The OpenNebula monitoring system has been enhanced with a component for

collecting energy metrics from Hosts and Serverless Runtimes based on the Scaphandre tool. Metrics are pushed to the Prometheus server integrated in OpenNebula. Hosts and Device geographic locations are collected using geoip techniques.

Pending Tasks:
Gather latency metrics from the devices.
Implementation of intrusion and anomaly detection of the different Edge Cluster entities and Serverless Runtimes.

SR4.5 Authentication & Authorization

Status: COMPLETED

Completed Tasks:
Implementation of delegation mechanisms in the COGNIT Frontend for authentication and authorization. The COGNIT Frontend delegates the credentials provided by the Device Client to the Cloud Edge Manager. Implementation of an authentication/authorization mechanism based on Biscuit tokens for secure communication between the Device Client and the COGNIT Frontend and the Edge Cluster Frontend.

SR4.6 Plan Executor

Status: IN PROGRESS

Completed Tasks:
Integration of the Plan Executor in the Cloud Edge Manager for executing plans related to the placement and migration of Serverless Runtimes within an Edge Cluster.

Pending Tasks:
Extension of the Plan Executor for executing plans related to the management and scaling of Edge Clusters.

2.5. AI-Enabled Orchestrator

SR5.1 Building Learning Models

Status: COMPLETED

Completed Tasks:
An analysis of the state-of-the-art of ML/AI approaches for modelling the Cloud-Edge

Continuum has been carried out. Different approaches have been selected and considered as potential candidates for providing the capability to automate the “optimal” placement of Serverless Runtimes on the distributed Cloud-Edge Continuum according to application and resource availability. Several models for the AI-Enabled Orchestrator have been trained and validated under different categories, such as regression, classification, and prediction, as well as evaluated for diverse orchestration tasks.

SR5.2 Smart Management of Cloud-Edge Resources

Status: IN PROGRESS

Completed Tasks:
Algorithms for interference and energy-aware optimization. Integration with the Cloud-Edge Manager and Plan executor of orchestration tasks for: smart scheduling, scaling, migration of Serverless Runtimes within an Edge Cluster.

Pending Tasks:
Features such as smart management and orchestration of serverless runtimes and Edge Clusters in order to meet workload requirements.

2.6. Secure and Trusted Execution of Computing Environments

SR6.1 Advanced Access Control

Status: IN PROGRESS

Completed Tasks:
Threat analysis has been done on the Cloud Edge Manager. Methods to exploit, detect and remediate overly permissive namespace access defaults in a multi-tenant context have been identified.

Pending Tasks:
Implementation of detection and remediation controls, and subsequent validation.

SR6.2 Confidential Computing

Status: IN PROGRESS

Completed Tasks:
A complete threat analysis has been done on the Serverless Runtime, highlighting the threat of an attacker inspecting the device memory for confidential information.

Confidential computing methods to protect the device against such attacks have been identified, as well as methods and tools to perform the exploit.
Pending Tasks: Implementation of the confidential computing control, validated by the deployment of the exploit.

SR6.3 Federated Learning
Status: NOT STARTED YET

3. Priorities for Fifth Research & Innovation Cycle (M28-M33)

During the Fourth Research & Innovation Cycle (M22-M27), and as part of the research and development tasks undertaken by the Use Cases, we have proceeded with the implementation of an improved version of the COGNIT Framework, focusing mainly on the smart management of Serverless Runtimes within an Edge Cluster in order to optimize the function workloads and their scalability taking into account infrastructure and virtual resource usage, energy consumption and application response time.

During the Fifth Research & Innovation Cycle (M28-M33), the Project will focus on continuing the development of the software requirements that have been identified in the previous cycles. In particular, the Fifth Research & Innovation Cycle will be mainly devoted to the enhancement of the Cloud-Edge Manager and the AI-Enabled Orchestrator for the smart management of Edge Clusters to optimize the application workloads and latency requirements.

Regarding the **Cloud-Edge Manager** component, in the Fifth Research & Innovation Cycle, most of the work will be focused on the ability to dynamically create new edge locations automatically to better satisfy scheduling policies (e.g. energy-aware), and application requirements. On the monitoring side, metrics related to the latency with respect to the Device Client will be stored in order to be used by the AI-Enabled Orchestrator to anticipate the creation of Serverless Runtime and/or Edge Clusters in order to minimise cold start delays and to satisfy latency requirements. Finally, the Plan Executor will be extended to execute plans produced by the AI-Enabled Orchestrator to perform operations at the infrastructure level (i.e., creating, scaling and deleting Edge Clusters resources).

In the Fifth Research & Innovation Cycle, the **AI-Enabled Orchestrator** will focus on the development of multiple features that enable smart and optimised downstream orchestration tasks related to the cloud-edge continuum infrastructure elasticity. In particular, interference-aware, energy-aware, and latency-aware algorithms will be

integrated in order to perform dynamic workload and resource optimization in response to system-changing conditions of cloud-edge infrastructure.

4. Risk analysis update for Secure and Trusted Execution of Computing Environments on the Multi-Provider Cloud-Edge Continuum

The purpose of this section is to provide an update of the security risk analysis of COGNIT based on the initial risk analysis carried out in D2.1. This was an initial high-level analysis done prior to the development of the framework based on the architectural blueprint and usage scenarios available at that time. At this point of the project, we have a number of new elements enabling us to better quantify the risks, we have measures already in place, and we have formulated security recommendations in order to take them into account for the last development iterations of the project. Those elements are:

- Refinement of COGNIT the architecture documented in D2.4 deliverable especially for the major revision of the architecture V2.
- Global security threat analysis documented in D2.4.
- Specialised threat analysis focusing on FaaS documented in D3.4.
- More precise use case scenarios documented in D5.4.

We give here a summary of the risk analysis highlighting the methodology and key analysis steps. In order to present the recommendations in a comprehensive way, they are formulated at the level of the global security architecture with the aim to emphasise on defense in depth and to make assumptions explicit, e.g. about components tightly related to a use case like the device client for which it is hard to identify precise measures and to expect them enforced.

4.1. Methodology

This iteration of the risk analysis was performed using ITSRM, which is developed by the European Commission. The method is composed of seven processes that map to the ISO27005 and is illustrated in the figure hereafter.

- System Security Characterization (P1): gathers initial information concerning the system and its context which is required (or helpful) to proceed further with the risk assessment (scope, objectives, assumptions…), ensuring alignment with business needs.
- Primary Assets Analysis (P2): identifies critical assets that need protection, such as data, services, and functionalities essential to the product or organization.
- Supporting Assets (P3): maps infrastructure, software, and dependencies that support primary assets, helping to identify security dependencies.
- System Modeling (P4): documents the architecture using diagrams (e.g., UML, SysML) to show how assets interact and where security risks might arise.

- Risk Identification (P5): uses threat catalogues (e.g., ITSRM annexes, MITRE ATT&CK) to identify potential security threats targeting the system. **At this step, threat modelling is useful and this is where we use the input of those activities documented in D2.4 and D3.4.**
- Risk Analysis & Evaluation (P6): assesses risks by evaluating the likelihood and impact of each threat exploiting vulnerabilities in the system.
- Risk Treatment (P7): defines how risks are treated using relevant strategies (avoidance, reduction, transfer, acceptance) and selects security controls to address identified risks.

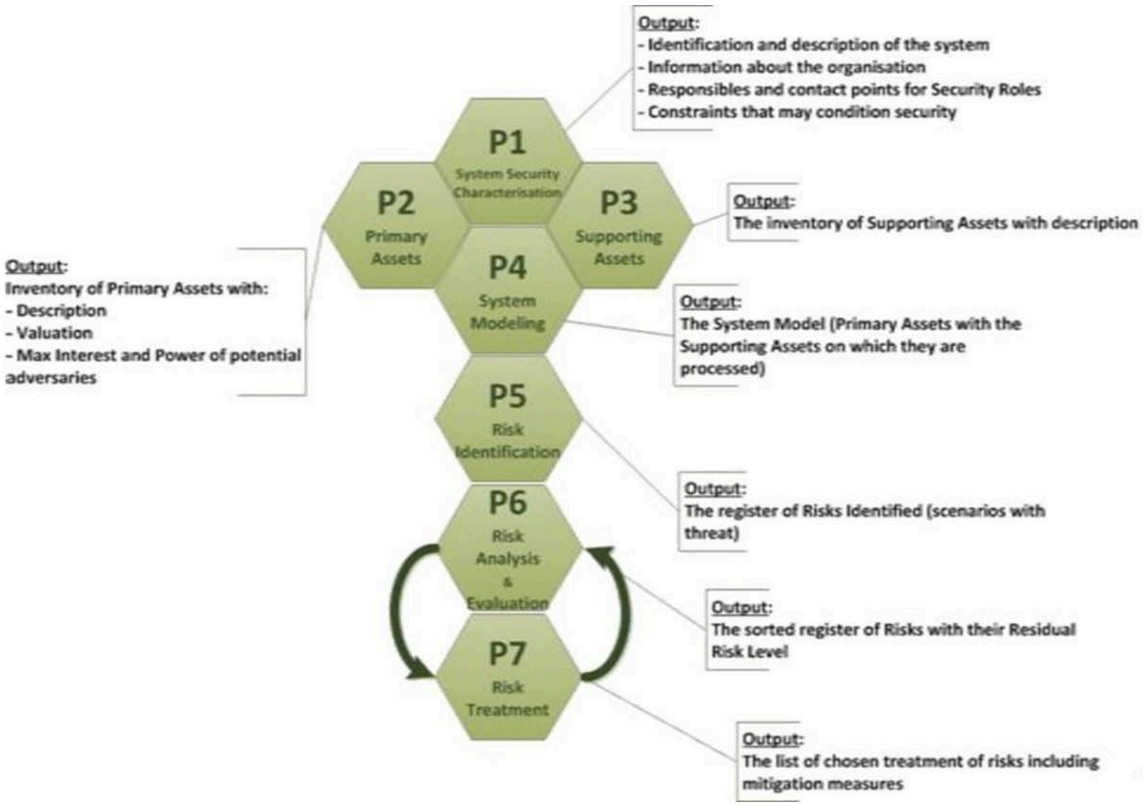

Figure 4.1. Overview of the ITSRM Method.

The methodology covers both unintentional and intentional threats. This document covers the same scope as the initial risk analysis of D2.1, i.e. intentional threats due to an attacker or unintentional threats due to human errors or negligence (e.g. misconfiguration). This document covers the same scope as the initial risk analysis of D2.1, i.e. intentional threats due to an attacker or unintentional threats due to human errors or negligence (e.g. misconfiguration). Some practices from another ISO 27005 methodology called MONARC were also used for the estimation phase.

4.2. Context and assets (P1-P4)

Steps P1 to P4 focus on capturing the context of the system and the protected assets both at primary (at business level, typically data or processes) and secondary. In our serverless context we consider two primary assets involved in the processing.

Primary Asset	Type	Description	Valuation (power+interest)
Function Data (input or result)	Information	Adversaries may be interested in data. We focus here only on data provided as input or as the result of the offloaded function.	medium-high
Offloaded function	Process	Adversaries may want to intercept the function to learn about its processing or to modify it for various purposes (stealing algorithms, altering results to impact client system, conducting attacks on COGNIT platform,...)	medium-high

Table 4.1. COGNIT primary (business) assets.

Note that these assets are quite abstract, which enable us to stay independent of the use case. COGNIT is a platform managing requests from various customers simultaneously. However the related security properties (i.e. Confidentiality, Integrity and Availability properties – CIA) that need to be preserved may vary depending on the customer case. **Risk combines both the impact and the likelihood of breaking such properties.**

- For the **impact**, to be as generic as possible, we decided to deal with the worst case scenario, which means some risk may be overestimated for specific use cases (e.g. confidentiality may not be important when dealing with public data). We assessed the possible impact in the following way, using a scale from 1 (very low impact) to 10 (critical impact).
- For the **likelihood**, ITSRM in formal assessment relied on an average of power+interest+ease. The valuation (power+interest) also depends on the use case, we consider it medium to high and will not rely on it for estimating the attack probability, instead we will focus on the ease of exploiting security vulnerabilities for estimating the likelihood on a scale from 1 (very rare) to 5 (almost certain).

Function Data		
Confidentiality	Depending on the context, data may be public to confidential, including with privacy concerns (e.g. a picture containing people, the location of people). We consider this the worst case.	9
Integrity	Data (input and result) integrity should be preserved but on the other hand, tampering should be detectable	7
Availability	Data availability can vary depending on the application domain but it is more of a prerequisite for making function calls, hence we do not consider it.	N/A
Off-Loaded function		
Confidentiality	The confidentiality of the function can vary, but usually the processing nature is kept internal, even if it could be revealed. Some processing algorithms might need to be kept secret and thus well protected against leaks. However, in such cases, function offloading might not be the best design. We consider it medium-high but not critical.	7
Integrity	The integrity of the function is critical: it may not be altered between the device and its execution on the serverless runtime.	10
Availability	The availability depends on the domain including high availability and short response time which exploit edge location. We consider it high but not critical as there is a trade-off to offload wrt local processing.	8

Table 4.2. Impact analysis per asset security properties.

For capturing the support assets, we relied on the latest version of the architecture depicted in Figure 4.2, modelling all the system parts at an adequate granularity level (avoiding being too fine-grained in the modelling of subparts of the system and also identifying specific trust boundaries within the system. The main components are also recapped in Table 4.3.

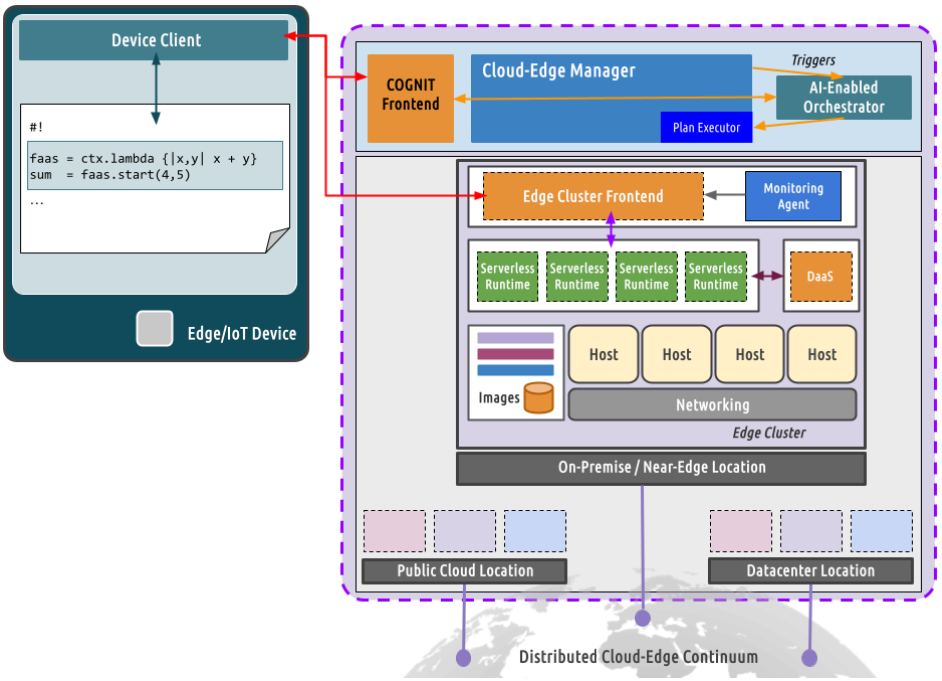

Figure 4.2. Reminder about the COGNIT architecture

Support Assets	Description
Device Client (with communication)	Client device on the field with limited processing power and relying on COGNIT for serverless processing.
COGNIT Frontend	Main entry point for the COGNIT service. It provides device authentication based on token and different endpoints for uploading application requirements, storing functions and data. It informs about the most suitable resources where the device can offload the execution of the functions.
Cloud-Edge Manager	Responsible for managing the highly distributed and heterogeneous resources of the cloud-edge continuum and the lifecycle of Serverless Runtimes within each Edge Cluster.
AI-Enabled Orchestrator	It manages and optimizes the lifecycle of Edge Clusters and their Serverless Runtimes based on the application requirements, infrastructure virtual resource usage, and policies.
Edge Cluster	A semi-autonomous environment with a dedicated software-defined resource set (compute, network and storage, including the Serverless Runtime image repository), which provides the abstraction layer enabling interoperability across the multi-provider cloud-edge infrastructure.
Edge Cluster Frontend	Frontend of the Edge cluster that routes the calls to functions that are offloaded by the devices inside Serverless Runtime and provides access to the results.
Serverless Runtime	Virtual Machine of the Edge Cluster where an offloaded function is executed.
DaaS	Specific component of the Edge Cluster to persist data and pass them across function calls.

Table 4.3. Description of COGNIT support (technical) assets

To model the whole system, we used the Threagile threat modelling notation (https://threagile.io). It produces the formalised architecture depicted in Figure 4.3 capturing trust zones, components, communication, manipulated data, security requirements and existing measures. The Figure shows the client device is clearly in an external trust zone with many potential threats along with a wide range of offloaded functions from customers spanning domains with specific security requirements. The COGNIT processing platform has a more clear trust boundary with two distinct zones: a first one dealing with the global COGNIT management and a second one for (each) edge cluster. Another output is a list of threats and risks used in the next step.

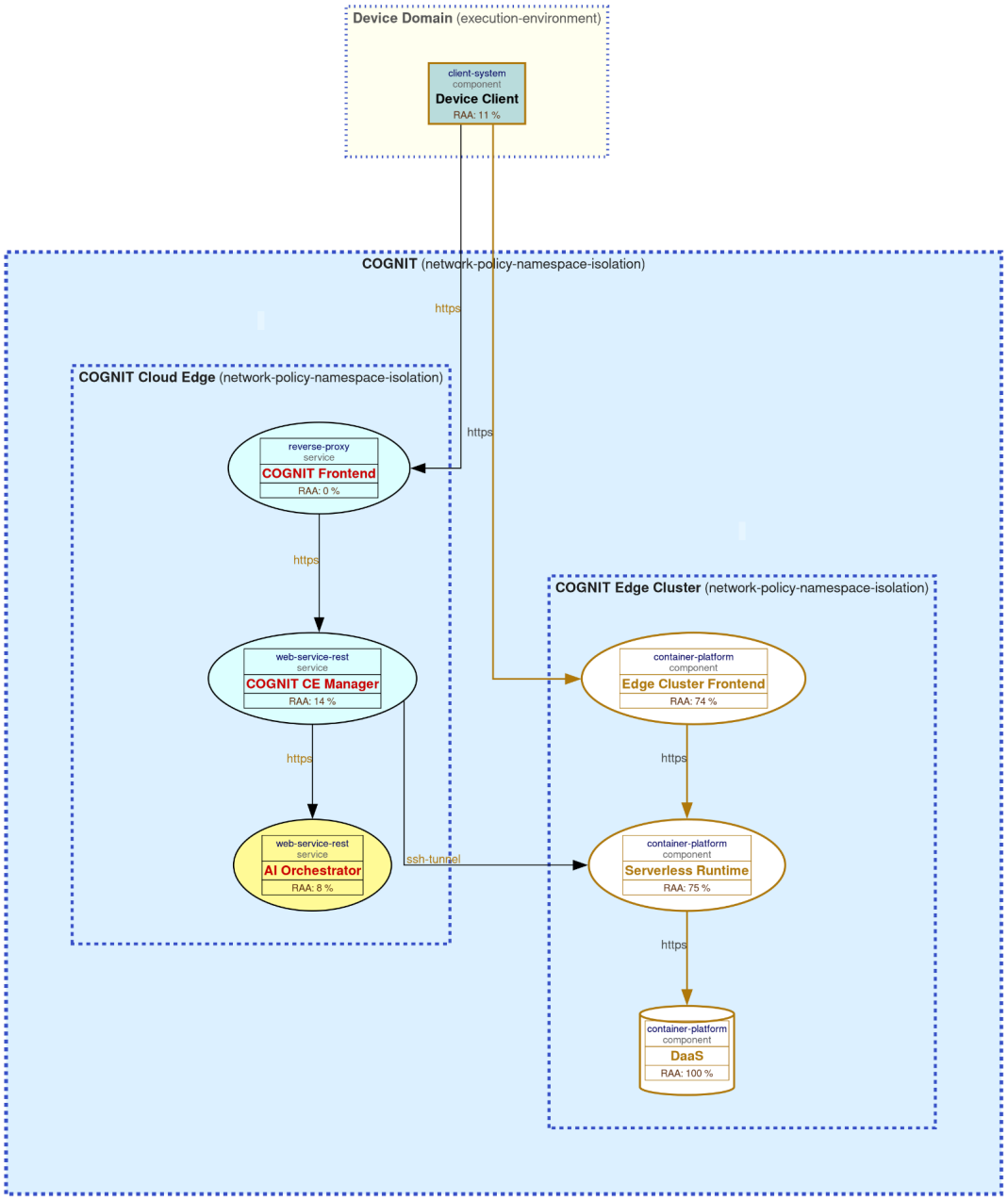

Figure 4.3. Formalised COGNIT architecture with Threagile

4.3. Risk identification and estimation (P5-P6)

ITSRM risk identification relies on reviewing triplets of (primary asset x property y support asset). In our case, all primary assets are exposed to similar threats with only a difference in impact so we focus on threats and identify when a threat covers one or more CIA properties. This way of processing is borrowed from MONARC. The identification process was also fed by threat modelling activities from D2.4 (global) and D3.4 (focus on FaaS).

	Threat	**CIA**	**Lik.**	**Justification of likelihood (including existing security measures, e.g. from Cloud-Edge Manager)**
T1	Client Device threats impacting function and data CIA	CIA	5	We do not perform a detailed analysis of the device client given its variability and the limited control the platform can enforce as it is managed by the customer. We will assume here a maximum likelihood of compromise to trigger a recommendation to minimize the attack surface in the next section.
T2	Compromise of Cloud-Edge Manager leading to disclosure or alteration of data/function in COGNIT component	CI	3	We rank it as moderately likely given that Cloud-Edge Manager provides support for securing services but the configuration could lack hardening and does not enforce end-to-end encryption so some assets could be unnecessarily exposed in clear inside some services.
T3	Credentials theft at the COGNIT frontend/weak authentication	CIA	3	Credentials (API key, tokens) could leak due to a vulnerability or a malevolent user gaining access to the management interface in some way. Token duration could be inadequate (no policy specified).
T4	Re-routing of request to rogue cluster	C	2	Rerouting requires elaborate attacks, either by poisoning the AI enabled orchestrator or forging specific requests at the device level. An example is a location attack where a fake location could be provided for such a redirection.
T5	Isolation failure at the Serverless Runtime and code injection	CI	3	We tag the risk as moderate as the serverless are not run under end-user identity but managed through namespace allocation which are more exposed to configuration or manipulation.
T6	Insecure deletion of function and processed data deletion at the Serverless Runtime	I	3	We also tag this risk as moderate as it depends on the reuse/deletion policy applied in operation.
T7	Tampering of VM images	CI	1	We estimate it unlikely given the protection ensured by Cloud-Edge Manager (see however assumption about management).
T8	Disclosure or tampering of information at DaaS	C	4	DaaS component (based on MiniO) is under early design and no information is available on security isolation of device-related information.
T9	Eavesdropping/manipulation at hardware level.	C	1	We consider them unlikely because these attacks are very technical and require physical access.
T10	Maintenance/update issues on COGNIT components	CIA	3	We consider this moderate, not measurable at this point: it needs to be implemented in the maintenance policy in operation.
T11	DoS on frontend components	A	4	We consider this likely given that the frontend is exposed and no protection is present.
T12	System failure due to exhaustion of resources in Cloud-Edge Manager components	A	3	We consider this risk as moderate as Cloud-Edge Manager can limit resource usage but no information or policy is defined for COGNIT.
T13	System failure due to exhaustion of resources, e.g. DoS (storage flooding) in DaaS	A	4	We consider this risk as likely given that the DaaS component is still under development. The MiniO based implementation provides support per tenant quota but the configuration needs to be checked.

Table 4.4. Threat identification and likelihood evaluation

Note that this estimation takes into account existing measures, especially provided as part of the Cloud-Edge Manager standard architecture and known to be currently implemented:

- authentication and IAM (for client devices and edge clusters).
- token access to API (not used).
- https/VPN support for external communication.
- RBAC access control.
- serverless isolation through namespace (VDC).
- encrypted volumes and secure internal transfer of information (TLS).
- immutable VM images.
- use of non-root services.

Among possible mechanisms currently discarded but available for extra measures:

- confidential computing provided hardware protected enclaves.
- IP/MAC spoofing prevention.
- zone registration/access policies.

Those threats when combined with the relevant CIA impacts and our two primary assets result in more than 40 risks, which can be computed based on the impact for each asset mentioned earlier. The following figures give the resulting heat map with impact in the horizontal axis and likelihood as the vertical axis.

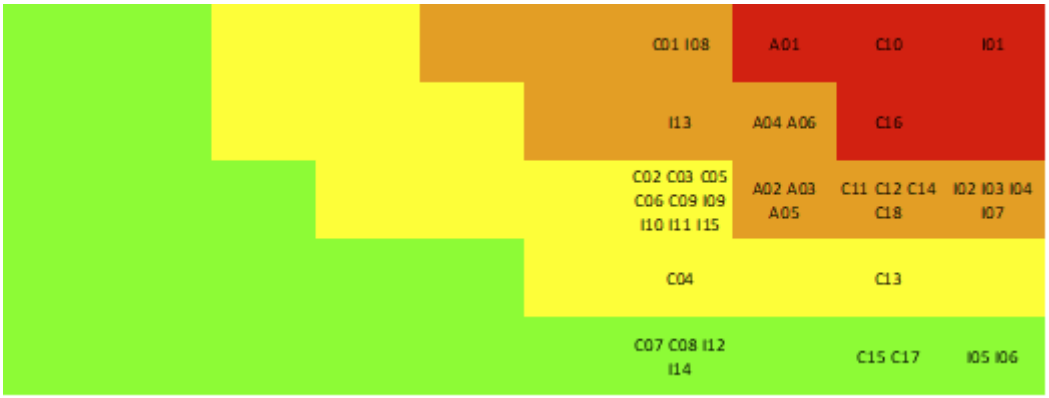

Figure 4.4. Heatmap of initial risks

The important and critical risks (above or equal to 24) are listed by decreasing order in the following table.

ID	Risk	Asset	CIA	Impact.	Likelihood	Risk
I01	Client Device threats impacting function and data CIA	Offloaded function	I	10	5	50
C10	Client Device threats impacting function and data CIA	Function data	C	9	5	45
A01	Client Device threats impacting function and data CIA	Offloaded function	A	8	5	40
C16	Disclosure or tampering of information at DaaS	Function data	C	9	4	36
C01	Client Device threats impacting function and data CIA	Offloaded function	C	7	5	35
I08	Client Device threats impacting function and data CIA	Function data	I	7	5	35
A04	DoS on frontend components	Offloaded function	A	8	4	32
A06	System failure due to exhaustion of resources in DaaS	Offloaded function	A	8	4	32
I02	Compromise of Cloud-Edge Manager leading to disclosure or alteration of data/function in COGNIT component	Offloaded function	I	10	3	30

I03	Credentials theft at the COGNIT frontend/weak authentication	Offloaded function	I	10	3	30
I04	Isolation failure at the Serverless Runtime	Offloaded function	I	10	3	30
I07	Maintenance/update issues on COGNIT components	Offloaded function	I	10	3	30
I13	Disclosure or tampering of information at DaaS	Function data	I	7	4	28
C11	Compromise of service engine leading to disclosure or alteration of data/function in COGNIT component	Function data	C	9	3	27
C12	Credentials theft at the COGNIT frontend	Function data	C	9	3	27
C14	Isolation failure at the Serverless Runtime	Function data	C	9	3	27
C18	Maintenance/update issues on COGNIT components	Function data	C	9	3	27
A02	Credentials theft at the COGNIT frontend	Offloaded function	A	8	3	24
A03	Maintenance/update issues on COGNIT components	Offloaded function	A	8	3	24
A05	System failure due to exhaustion of resources in Cloud-Edge Manager components	Offloaded function	A	8	3	24

Table 4.5. Summary of main risks.

Those threats and resulting risks were also cross-checked with inputs from other analyses and the following checks could be done about relevant threats:

- coverage of initial high-level risks of D2.1.
- coverage of specific risks reported by threat modelling (Microsoft Threat Modelling Tool in D2.4, Threagile in complement here): weak authentication, Denial of Service at frontend, unencrypted assets (at memory level) in services, limited hardening, remote code execution, missing identity propagation, unspecified vault isolation.

4.4. Risk treatment (P7)

To address the risks, we tackle the related threats which allows us to recompute all the risks. The strategy used targets only the likelihood as the impact is domain dependent (the worst level of impact is identified in connection reminded together with the related property). We rely only on risk reduction tactics with a few accepted risks (some are however commented for longer-term measures). We do not apply cumulative reduction factors but reassess the level for the whole set of measures (this does not enable us to consider the impact of partial implementation however).

Threat		**Likelihood**	**Impact (worse)**
T1 Client Device threats impacting function and data CIA		5	10
REDUCE – out of scope but suggested guidelines for a device client to reduce the attack surface could be			
	Encryption of data and functions inside the device		
	Device enrollment, identity tracking in requests, performing remote attestation		
	Secure management of tokens and device identity		
	Secure boot and firmware integrity		
	Sandbox environment for edge cluster interaction		
	Event logging and telemetry		
	Re-estimation	1-2	10
T2 Compromise of the Cloud-Edge Manager leading to disclosure or alteration of data/function in COGNIT component		3	10
REDUCE			
	Configure/check security groups for global and cluster management		
	Enforce end-to-end security between device and serverless function (i.e device signature/encryption + platform verification and decryption at serverless level)		

Threat		Likelihood	Impact (worse)
	Re-estimation	1	10
T3 Credentials theft at the COGNIT frontend		3	10
REDUCE			
	Force duration and scope limits on API keys and tokens		
	Limit privileged accounts		
	Enforce strong multifactor authentication		
	Re-estimation	1-2	10
T4 Re-routing of request to rogue cluster		2	9 (C)
REDUCE			
	Protect the request information used by the AI-Enabled Orchestrator		
	Monitor cluster for compromise indicators (e.g. unusual redirection to a specific edge cluster)		
	Re-estimation	1	9
T5 Isolation failure at the serverless runtime and injections		3	10
REDUCE			
	Check namespace configuration and sharing		
	Scan/sanitize function and rule out insecure commands allowing devices to offload and run code injected through data		
	Re-estimation	1	10
T6 Deletion issue at the Serverless Runtime		3	9 (C)
REDUCE			
	Define and implement a deletion/reuse policy		
	Secure cleaning/destruction of used VMs		
	Re-estimation	1	9
T7 Tampering VM images inside COGNIT		1	9
ACCEPT – the integrity control provided by Cloud-Edge Manager is enough			
T8 Disclosure or tampering of information at DaaS		4	9 (C)
REDUCE			
	DaaS design review and testing		
	DaaS authentication and access control		
	DaaS volume encryption		
	Re-estimation	2	
T9 Eavesdropping/manipulation at hardware level.		1	10
ACCEPT is fine – a possible extra measure for highly sensitive applications is to use confidential computing			
T10 Maintenance/update issues on COGNIT components		4	10
REDUCE			
	Regularly audit the system for security vulnerabilities and apply patches as soon as they become available		
	Monitor the health status of edge clusters and redirect requests to isolate any cluster experiencing problems		
	Redundancy mechanism for the COGNIT frontend		
	Re-estimation	1	
T11 DoS on frontend components		4	8
REDUCE (or TRANSFER if external)			
	Use external DDoS protection services like EfficientIP DNS Blast and Gcore to filter malicious traffic.		
	Monitor edge cluster node load status and redirect requests to isolate clusters under DOS attack		
	Distribute cluster nodes on a diversity of providers		
	Re-estimation	1-2	8
T12 System failure due to exhaustion of resources in Cloud-Edge Manager components		3	8
REDUCE			
	Check that maximum CPU, memory, and storage allocations per user or group are set to prevent any single entity from consuming excessive resources		

Threat		Likelihood	Impact (worse)
	Use anomaly detection to track resource abuse patterns		
	Re-estimation	1	8
T13 System failure due to exhaustion of resources in DaaS		4	8
REDUCE			
	Provision storage according to expected demand		
	Integrate scalability mechanisms to adapt storage to demand		
	Use anomaly detection to track resource abuse patterns		
	Re-estimation	1	8

Table 4.6. Risk treatment strategy and security measures.

The resulting heatmap (see Figure 4.5) considering the recommended measures are correctly implemented is the following. It could remove all important and critical risks, leaving 13 risks in a zone requiring specific monitoring while all the others are in the acceptance zone.

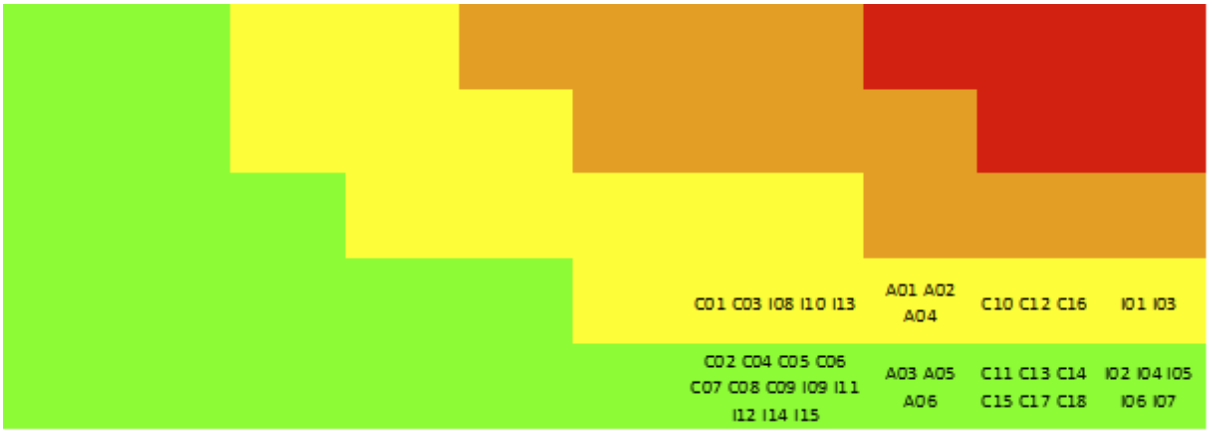

Figure 4.5. Heatmap of residual risks.

4.5. Summary of recommendations on the security Architecture

The current protection provided for COGNIT is depicted in green in Figure 4.6 and the first column of Table 4.7. It is essentially based on Cloud-Edge Manager security features identified in section 4.3 and of limited perimetric nature, i.e. not including the device client.

Taking a step back from the risk analysis, it appears that COGNIT needs to deal with threats that cannot be completely mitigated using perimeter-based security techniques that rely on a strong external boundary (firewalls, VPNs, etc.) to protect internal resources. Indeed the COGNIT cannot control all entry points, for example:

- The Device Client is highly variable across applications and is granted the right to execute functions. As the device design is outside the scope of COGNIT, it could be compromised and needs extra internal protections. The framework already provides good static isolation, but more dynamic checks could be enforced, e.g. through anomaly detection.

- The Platform itself is distributed and composed of a federation of Edge Clusters. The level of trust of cluster nodes can vary based on the security mechanisms available.
- Specific components have been developed internally (AI-Enabled Orchestrator) and could contain vulnerabilities that need to be managed.
- The AI-Enabled Orchestrator's behaviour results from a learning phase based on data which could be poisoned.

Moreover, COGNIT is a mutualised platform managing requests coming from a variety of customers, which may trigger different levels of interest in attacks. This increases the platform's exposure because some attacks may jeopardise the whole platform. **This reinforced the need for extra non-perimetric mechanisms assuming zero-trust and providing defense in depth inside the platform using internal mechanisms to detect any abnormal behaviours related to a client and isolate it before it can result in a larger impact at the platform level.**

Such mechanisms have been identified as the risk treatment and can now be summarised at (security) architecture level in the table below, showing:

- **zero-trust measures** covering protection and defense layers through end-to-end. This covers two lines of defense:
    - **better protection** by achieving end-to-end confidentiality/integrity of function and data through the use of signatures and encryption using device and platform keys. Better trust in the device can be achieved with formal device enrollment.
    - **active detection** through continuous monitoring based on scanning the function and blocking it, but also a more general anomaly detection. This can cover extensions such as the capability for the orchestrator to use location information. In some use cases, anomalies can be detected on a location basis (e.g. incoherent position/speed) possibly in correlation with other devices in the same area. Better trust in the device can be achieved by using identity tracking and performing regular remote attestation.
- some **additional and more advanced perimetric protections** like
    - the use of sandboxing mechanisms inside the device client
    - DDoS protection by using standard filtering services
    - the possibility to use **secured enclaves** (confidential computing) which are being implemented in the last iterations of the project. Such enclaves provide another level of guarantee about execution isolation compared to standard VM isolation provided by COGNIT-based Serverless Runtimes. It comes with a cost but can be dedicated to secure processing in the case of highly critical security requirements.
- **a new recovery line** using the distributed and elastic nature of an edge cluster to cope with the unavailability of some node resulting from a detected compromise.

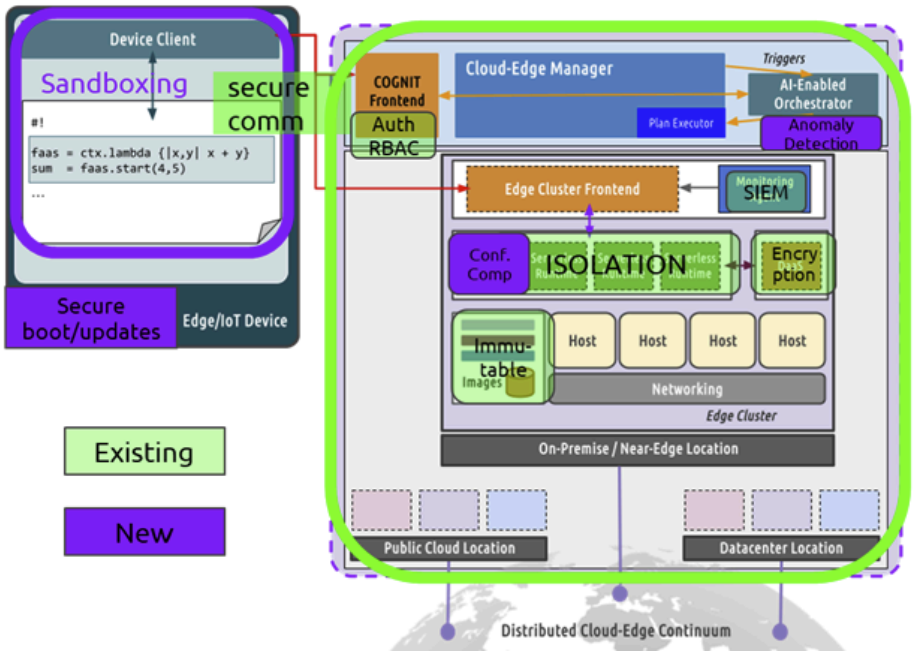

Figure 4.6. Existing and improved security architecture.

	Current	**Recommendations**
Protect	Perimeter-based - Cloud-Edge Manager auth/RBAC - Serverless isolation - Image protection Capability limits: - resource quota (CPU, data) Security updates	Perimeter reinforcement: - DDoS protection - secured enclaves (conf. computing) Zero-trust: end-to-end PKI - keep signed/encrypted payloads for all components not requiring access (i.e. almost end-to-end from device to serverless) - device enrollment and identity tracking in requests - deletion management of VM/reuse policy
Detect/Respond	Logging/SIEM available but not used	Zero-trust: continuous monitoring - function scan - anomaly detection, including for advanced attacks like location-based
Recover	N/A	Rebalance requests on available nodes based on a continuous monitoring of availability/capabilities (AI-Enabled Orchestrator)

Table 4.7. Existing and improved COGNIT defense lines.

5. Conclusions and Next Steps

The initial version of the COGNIT Framework Architecture (Deliverable D2.1), released in M3, identified and analysed the main sovereignty, sustainability, interoperability, and

security requirements, as well as the user requirements, derived from the European context and from the Project’s specific Use Cases. They are expected to guide the research and development of the project, having played a central role in the initial definition of the original COGNIT Architecture. From those global and user requirements, a list of software requirements and functional gaps to be implemented by the components of the COGNIT Framework were identified, followed by a definition of the methodology and scenarios required to verify their fulfilment and applicability in the Project’s Use Cases.

In the first and second incremental versions of D2.1 (Deliverable D2.2, Deliverable 2.3) the open source software components and extensions needed to meet the Software Requirements were specified and developed within the Work Packages WP3 and WP4, with these new functionalities being tested, verified, and demonstrated as part of the Use Cases in their respective associated tasks in WP5.

The Deliverable D2.4 provided a comprehensive overview of the modified COGNIT Architecture with the Project's global development status, offering a summary of the work done in the Third Research & Innovation Cycle (M16-M21), and identified the priorities for the Fourth Research & Innovation Cycle (M22-M27).

This report is the first incremental version of the Deliverable D2.4 that provides an overview of the Project's overall development status, offers a summary of the work done in the Fourth Research & Innovation Cycle (M22-M27), and identifies the priorities for the Fifth Research & Innovation Cycle (M28-M33).