SovereignEDGE.EU COGNIT

A Cognitive Serverless Framework for the Cloud-Edge Continuum

D5.2 Use Cases - Scientific Report - b

Version 1.0

31 October 2023

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. This document provides an overall status of the contribution of the Project's software requirements towards meeting the user requirements that guide the development of the COGNIT Framework, offers additional information about the domains targeted by the Use Cases and the Partners involved in them, lists new user requirements identified during the First Research & Innovation Cycle (M4-M9), and provides an update on the Project's software integration process and infrastructure, on its testbed environment, and on the progress of the software requirement verification tasks.

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:	WP5. Adaptive Serverless Framework Integration and Validation
Nature:	R: Report
Dissemination Level:	PU: Public
Version:	1.0
Contractual Date of Delivery:	30/09/2023
Actual Date of Delivery:	31/10/2023
Lead Author:	Thomas Ohlson Timoudas (RISE)
Authors:	Monowar Bhuyan (UMU), Marek Białowąs (Phoenix), Dominik Bocheński (Atende), Malik Bouhou (CETIC), Aritz Brosa (Ikerlan), Idoia de la Iglesia (Ikerlan), Sébastien Dupont (CETIC), Torsten Hallmann (SUSE), Joan Iglesias (ACISA), Rafał Jurkiewicz (Phoenix), Tomasz Korniluk (Phoenix), Johan Kristiansson (RISE), Antonio Lalaguna (ACISA), Martxel Lasa (Ikerlan), Marco Mancini (OpenNebula), Alberto P. Martí (OpenNebula), Philippe Massonet (CETIC), Nikolaos Matskanis (CETIC), Behnam Ojaghi (ACISA), Daniel Olsson (RISE), Goiuri Peralta (Ikerlan), Samuel Pérez (Ikerlan), Holger Pfister (SUSE), Tomasz Piasecki (Atende), Francesco Renzi (Nature4.0), Bruno Rodríguez (OpenNebula), Juan José Ruiz (ACISA), Kaja Swat (Phoenix), Gerard Świderski (Phoenix), Paul Townend (UMU), Iván Valdés (Ikerlan), Riccardo Valentini (Nature4.0), Constantino Vázquez (OpenNebula).
Status:	Submitted

Document History

Version	Issue Date	Status[1]	Content and changes
0.1	20/10/2023	Draft	Initial Draft
0.2	27/10/2023	Peer-Reviewed	Reviewed Draft
1.0	31/10/2023	Submitted	Final Version

Peer Review History

Version	Peer Review Date	Reviewed By
0.1	27/10/2023	Marco Mancini (OpenNebula)
0.1	27/10/2023	Paul Townend (UMU)

Summary of Changes from Previous Versions

First Version of Deliverable D5.2

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

Executive Summary

Deliverable D5.2, released at the end of the First Research & Innovation Cycle (M9), is the first incremental version of the Use Cases Scientific Report in WP5 ”Adaptive Serverless Framework Integration and Validation”. This report provides an overview of the overall status of the contribution of the Project’s software requirements towards meeting the user requirements that guide the development of the COGNIT Framework, offers additional information about the domains targeted by the Use Cases and the Partners involved in them, lists new user requirements identified during the First Research & Innovation Cycle (M4-M9), and provides an update on the Project’s software integration process and infrastructure, on its testbed environment, and on the progress of the software requirement verification tasks per component.

In connection with the main components of the COGNIT Architecture (i.e. Device Client, Serverless Runtime, Provisioning Engine, Cloud-Edge Manager, and AI-Enabled Orchestrator), the Project has delivered progress specifically in those software requirements needed to achieve Milestone 2 in M15, and to provide a basic set of functionalities for the Use Cases to be able to launch their own research and development activities in the next cycle (M10-M15).

Apart from this document (D5.2) and the Project's global overview provided by Deliverable D2.2, specific research and development activities performed in WP3 “Distributed FaaS Model for Edge Application Development” (related to the Device Client, the Serverless Runtime, the Provisioning Engine, and the Secure and Trusted Execution of Computing Environments) are described in detail in reports D3.1 “COGNIT FaaS Model - Scientific Report” and D3.6 “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 D4.1 “COGNIT Serverless Platform - Scientific Report” and D4.6 “COGNIT Serverless Platform - Software Source”.

This deliverable has been released at the end of the First Research & Innovation Cycle (M9), and will be updated with incremental releases at the end of each research and innovation cycle (i.e. M15, M21, M27, M33).

Table of Contents

Abbreviations and Acronyms 5
1. Introduction 7
PART I. Validation Use Cases 8
2. Overall Status 8
3. Use Case 1: Smart Cities 10
4. Use Case 2: Wildfire Detection 13
5. Use Case 3: Energy 15
6. Use Case 4: Cybersecurity 18
7. New User Requirements 21
PART II. Software Integration and Verification 22
8. Software Integration Process and Infrastructure 22
9. Testbed Environment 29
10. Software Requirements Verification 30
    10.1 Device Client 30
    10.2 Serverless Runtime 32
    10.3 Provisioning Engine 32
    10.4 Cloud-Edge Manager 34
    10.5 AI-Enabled Orchestrator 36
    10.6 Secure and Trusted Execution of Computing Environments 37
11. Conclusions and Next Steps 39

Abbreviations and Acronyms

3G	Third Generation Mobile Network
4G	Fourth Generation Mobile Network
5G	Fifth Generation Mobile Network
AI	Artificial Intelligence
API	Application Programming Interface
AWS	Amazon Web Services
CAM	Cooperative Awareness Message
CC	Control Center
C-V2X	Cellular Vehicle to Everything communication
DaaS	Data as a Service
FaaS	Function as a Service
GNSS	Global Navigation Satellite System
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure as a Service
IP	Internet Protocol
ITS	Intelligent Transport System
LAN	Local Area Network
LTE	Long-Term Evolution
MCU	MicroController Unit
MEC	Multi-Access Edge Computing
ML	Machine Learning
M-Hub	Mobility Hub (advanced TLC)
OBU	On Board Unit
OS	Operating System
PaaS	Platform as a Service
QoS	Quality of Service
RES	Renewable Energy Source
REST	Representational State Transfer
RSU	Road Side Unit
SaaS	Software as a Service
SLA	Service Level Agreement
SOAR	Security Orchestration, Automation, and Response
SREM	Signal Request Extended Message

SSEM	Signal request Status Extended Message
TCP	Transmission Control Protocol
TLC	Traffic Light Controller
V2X	Vehicle to Everything communication technology
VLAN	Virtual Local Area Network
VM	Virtual Machine
VPN	Virtual Private Network
WLAN	Wireless Local Area Network

1. Introduction

The initial version of the Use Cases Scientific Report (Deliverable D5.1), released in M3, provided an initial collection of user requirements, a description of each of the Use Cases—including an initial architecture design and a plan for the demonstration and validation to take place in Tasks T5.3, T5.4, T5.5, T5.6—and an update of the COGNIT Testbed environment. The aim of this incremental version (Deliverable D5.2) is to provide an update at the end of the First Research & Innovation Cycle (M9), offering an overview of the overall status of the contribution of the Project's software requirements towards meeting the user requirements that guide the development of the COGNIT Framework, providing additional information about the domains targeted by the Use Cases and the Partners involved in them, listing new user requirements identified during the First Research & Innovation Cycle (M4-M9), and offering an update on the Project's software integration process and infrastructure, on its testbed environment, and on the progress of the software requirement verification tasks per component. An incremental version of this report will be released at the end of each research and innovation cycle (i.e. M15, M21, M27, M33).

D5.2 is a living document that is composed of an introductory section and nine additional sections organised in two main blocks of content:

- **Part I** focuses on tracking the progress of the Project's software requirements towards meeting the user requirements (Section 2) and on each of the Use Cases, which have a dedicated section (Sections 3 to 6). New user requirements identified during the reporting period are listed in Section 7.
- **Part II** focuses on the Project's software integration process and infrastructure (Section 8), on the evolution of the Testbed environment (Section 9), and on the software requirements verification tasks carried out per component (Section 10).

The document ends with a conclusion section.

PART I. Validation Use Cases

2. Overall Status

The table below shows the current status of each Software Requirement towards meeting its associated global and user requirements, following a simple colour code: 🟥 for activities that have not started yet, 🟨 for activities in progress, and 🟩 for completed activities:

	ID	DESCRIPTION	Device Client					Serverless Runtime		Prov. Eng.	Cloud-Edge Manager					AI-Enabled Orchestrator			Secure & Trusted Exec of Comp.Envs.		
			SR1.1	SR1.2	SR1.3	SR1.4	SR1.5	SR2.1	SR2.2	SR3.1	SR4.1	SR4.2	SR4.3	SR4.4	SR4.5	SR5.1	SR5.2	SR5.3	SR6.1	SR6.2	SR6.3
Sovereignty	SOR0.1	The COGNIT Framework shall be able to leverage public, private, and self-hosted cloud and edge infrastructures hosted in the European Union.									🟥										
	SOR0.2	The implementation of the COGNIT Architecture shall maximise the use of European open source technologies and frameworks.									🟥										
	SOR0.3	The COGNIT Framework shall provide an abstraction layer that ensures workload portability seamlessly across different infrastructure providers.										🟥									
	SOR0.4	Data handling by the COGNIT Framework shall be compliant with the GDPR.					🟥		🟥												🟥
Sustainability	SUR0.1	Sustainability performance needs to be measurable (e.g. energy profiles should be queryable and updatable for every feature/component within the framework), including energy sources (e.g. renewable, non-renewable) and energy consumption profiles (e.g. estimated power consumption).						🟥			🟥			🟨							
	SUR0.2	Sustainability needs to be maximised to reduce environmental footprint by leveraging edge characteristics (e.g. by increasing the share of renewables, minimising battery use/size, using energy otherwise wasted, or scaling down active Runtimes).														🟨	🟨	🟥			
	SUR0.3	The whole energy lifecycle should be taken into account in order to implement a circular economy, including e.g. energy availability and cost and hardware degradation.														🟥					
Interoperability	IR0.1	Deployment of the COGNIT Framework and of its components should be as portable as possible across heterogeneous infrastructures or cloud/edge service providers (e.g. by using broadly-adopted virtualisation and container technologies).						🟨	🟥			🟥	🟨								
	IR0.2	Preference should be given to expanding existing frameworks, tools, and open standards.					🟥				🟥	🟥	🟨	🟨	🟨	🟥					🟥
	IR0.3	The interfaces of the COGNIT Framework shall be documented in order to facilitate discovery of its features by third-parties.	🟨	🟨	🟨					🟨	🟥	🟥	🟨	🟨	🟨		🟨	🟨			
Security	SER0.1	Communications inside COGNIT, and between the COGNIT environment and the outside (e.g. IoT devices) should be encrypted and signed using security mechanisms such as SSLv3.	🟥			🟥	🟥	🟥	🟥	🟥		🟥	🟥							🟨	
	SER0.2	The COGNIT Framework should be built following security-by-design and Zero Trust practices.					🟥	🟥	🟥	🟨		🟥	🟨						🟨	🟨	🟥
	SER0.3	The implementation of the COGNIT Framework should be aligned with the latest legislative frameworks, such as the NIS2 Directive, the GDPR, and the future Cyber Resilience Act (CRA).	🟥				🟥		🟥												🟥
	SER0.4	Runtimes should be protected against threats by the enforcement of security controls such as secure defaults, vulnerability scans, intrusion and anomaly detection and continuous security assessment (the specific controls to be implemented will be determined by a risk analysis).						🟥	🟥				🟥							🟨	🟥
	SER0.5	Resources should be protected by an Identity and Access Management (IAM) system, implementing role based access control (RBAC), security zones, and support for a multi-tenant security model.				🟥				🟨									🟨		
	SER0.6	Integrity of the offloaded functions needs to be guaranteed, including the function inputs and outputs (also during the live migration of FaaS Runtimes).				🟥		🟥													

	ID	DESCRIPTION	Device Client					Serverless Runtime		Prov. Eng.	Cloud-Edge Manager					AI-Enabled Orchestrator			Secure & Trusted Exec of Comp.Envs.		
			SR1.1	SR1.2	SR1.3	SR1.4	SR1.5	SR2.1	SR2.2	SR3.1	SR4.1	SR4.2	SR4.3	SR4.4	SR4.5	SR5.1	SR5.2	SR5.3	SR6.1	SR6.2	SR6.3
Common Requirements	UR0.1	Device applications should be able to offload any function written in C or Python languages.		🟨	🟨			🟨													
	UR0.2	Device applications should be able to upload data from the device ensuring data locality with respect to where the offloaded function is executed.		🟥					🟥												
	UR0.3	Device applications should be able to upload data from external backend storages ensuring data locality with respect to where the offloaded function is executed.		🟥					🟥												
	UR0.4	Execution of functions such as ML inference engines should be able to load machine learning models stored ensuring data locality with respect to where the function is executed.		🟥					🟥												
	UR0.5	Function execution can be executed in different tiers of the Cloud-Edge continuum according to network latency requirements.						🟥		🟨	🟥	🟥	🟨				🟥	🟨			
	UR0.6	Device application shall have the ability to define maximum execution time of the offloaded function upon offloading.	🟥														🟥				
	UR0.7	Device application shall have the ability to specify and enforce runtime maximum provisioning time and runtime shall be provisioned within the previously specified time.	🟥							🟥							🟥	🟨			
	UR0.8	Device applications must be able to request and obtain an authorization prior to establishing any further interaction with COGNIT.	🟥							🟥					🟥				🟨		
	UR0.9	IAM system integration for high granularity authentication and user management for device clients, provisioning engine and serverless runtime.								🟥			🟥		🟥						
	UR0.10	Push mechanism to inform about status or events from Provisioning Engine and Serverless runtime back to the requestor device client.	🟥	🟥				🟥		🟥											
UC1	UR1.1	Function execution shall be supported in shared, multi-provider environments (with different access and authorization procedures), and the execution must be isolated from other processes on the host system.											🟥		🟥				🟥	🟥	🟥
	UR1.2	Device application shall have the ability to dynamically scale resources for offloading function execution to maximise exploitation of resources in shared environments, while avoid saturation or resources kidnapping.	🟥									🟥				🟥	🟥	🟥			
	UR1.3	Function execution should exploit data locality and prioritise edge nodes where the required data is already stored.		🟥													🟥	🟥			
	UR1.4	The whole life cycle of either function execution or code offloading should be auditable and non repudiable.	🟥					🟥						🟥							
	UR1.5	Device applications should be able to request execution over GPUs.	🟥					🟥										🟥			
UC2	UR2.1	It shall be possible to obtain both a-priori estimates of expected, and actual measurements of, energy consumption of the execution of function.						🟥						🟥		🟥					
	UR2.2	COGNIT Framework should be able to adapt to rare events with sudden peaks of FaaS requests, in which the offloaded function requires much heavier computations and more frequent execution than usual.														🟥	🟥	🟥			
	UR2.3	Possibility for devices to request access to GPUs, when available, during high-alert mode.	🟥					🟥										🟥			
UC3	UR3.1	Device Client and user applications shall share a maximum of 500 kB of available RAM in total.	🟥			🟥															
	UR3.2	It shall be possible for the user application to dynamically scale up/down resources for function execution due to changes in the user preferences.	🟥							🟥			🟥				🟥				
	UR3.3	The SDK for the Device Client shall have support for the C programming language.	🟥		🟥	🟥															
UC4	UR4.1	The Device Client should have the ability to dynamically set the permissible edge nodes for executing the function based on policy (e.g. geographic security zones, distance to edge node).	🟥							🟥	🟥						🟥	🟥	🟥		
	UR4.2	The COGNIT Framework should have the ability to live migrate of data/runtime to different edge locations based on policy and location of function execution (e.g. geographic security zones, distance to edge node).	🟥										🟥				🟥	🟥			
	UR4.3	The Device should be able to request the execution of a function as close as possible (in terms of latency) to the Device's location.	🟥														🟥	🟥			

Table 2.1. Current status of each Software Requirement towards meeting its associated global and user requirements (new SRs/URs in red).

3. Use Case 1: Smart Cities

Connected vehicles and autonomous driving are expected to revolutionise transportation systems, improving road safety and traffic efficiency while reducing accidents. This Use Case will demonstrate the capabilities of edge computing for future smart city platforms and smart mobility services—the main challenge will be managing dense networks of edge infrastructure resources, as well as a variety of services with different QoS requirements. The transportation systems need to be interoperable, intelligent, secure and support the development of multi-tier edge applications that can be deployed across the cloud-edge continuum, leveraging data locality to reduce overhead, and be managed securely using cloud-native practices.

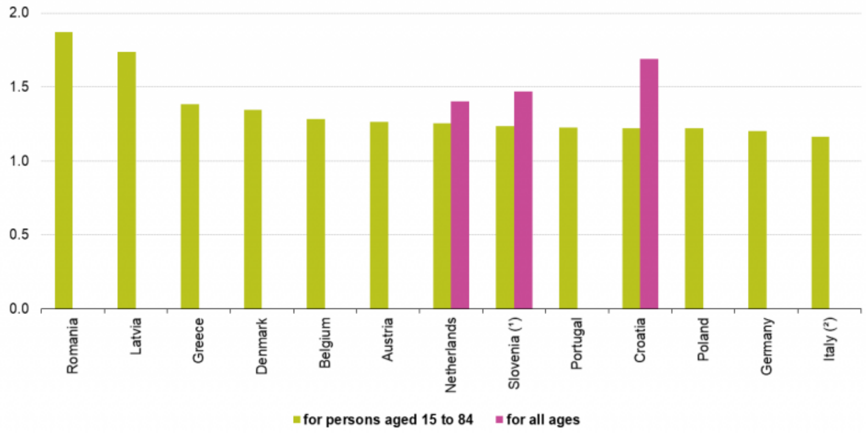
Densified cities face serious traffic and pollution problems every day, and it is necessary for people to move in a more optimal and safe way to be able to maintain acceptable levels of time and quality in transport while preserving low levels of pollution.

With reference to the organisation of transport in a city, there are still certain differences in the approach according to the cities, but there are structural lines in which all the cities converge thanks to the European directives. Reducing atmospheric pollution, optimising travel time in collective public transport, promoting the use of bicycles or walking through safe and universally accessible infrastructures, these are the common values that will lead citizens to have a better quality of life.

The key aspect for better mobility in a city is public transport, and all the measures oriented to improve it will have a great impact on the city and on the life of the citizens.

According to Eurostat[2], the average occupancy of vehicles in daily urban journeys does not reach 1.5 people in most of the study countries, and this makes us think of the inefficiency of private vehicles in terms of the space occupied on the streets and roads in relation to the number of passengers:

Average passenger car occupancy for urban mobility on all days
(number of persons per vehicle)



Note: data calculated according to Eurostat Guidelines;
(¹) SURS (Statistical Office of the Republic of Slovenia) data- slightly different methodology for persons aged 15 to 84.
(²) Italian data are based on the national survey that covers the population aged 15 to 80
Source: Data from thirteen Member States (nine pilot surveys and four national surveys on passenger mobility)
eurostat

[2] https://ec.europa.eu/eurostat/statistics-explained/index.php?title=File:Average_passenger_car_occupancy_for_urban_mobility_on_all_days_v2.png

On the other hand, road transport is one of the main causes of air pollution and greenhouse gas emissions in urban areas, and the costs of congestion to society amount to approximately €270 billion per year.

Poor air quality, especially in urban areas, continues to affect the health of the European population. According to the latest EEA estimates[3], at least 238,000 people died prematurely in the EU in 2020 due to exposure to PM2.5 pollution above the OMS guidance level of 5 μg/m³. Nitrogen dioxide pollution caused 49,000 premature deaths in the EU and 24,000 from ozone exposure.

ACISA has been designing and manufacturing ITS Software and Hardware solutions for cities, roads and tunnels for more than 50 years and in the last decade all city solutions have been aimed at optimising traffic systems, providing significant improvements in the quality of life of citizens and focusing on reducing pollution and protecting VRUs (Vulnerable Road Users).

To date, the level of investment in V2X digital infrastructures is still incipient, and although most communication standards and technologies have been on the market for years, the technology adoption is being slow. There are many variables and actors that must join forces to bring forward complete and functional solutions. Infrastructure managers, vehicle manufacturers, logistics companies, fleet managers... they are all collaborating to find use cases that are viable, safe and that offer benefits to all the users. But there are strong dependencies between the investments in different markets that slows the implementation of these use cases.

Despite the slow rollout, in part due to the transformation of the automotive sector, the future market that the autonomous and connected vehicle will generate is promising and companies that have the product ready when demand takes off will have a competitive advantage that they can take advantage of.

Several companies share this vision with ACISA and have emerged as key players in the competitive V2X automotive market, including Continental AG, Autotalks, ETrans, Qualcomm, Delphi (Aptiv), Denso, General Motors, HARMAN, Arada, Cohda Wireless, Savari, and Kapsch and are investing not only in the development of new devices and technologies but also in the adoption of these technologies in serial cars and trucks. ACISA is testing the technology that most of these players are offering in multiple R&D projects to bring to our clients better solutions.

Some good news outside Europe in the horizon in the last year are good reasons to stay hopeful about the massive adoption of the technology:

“Hyundai Mobis Co., the world’s sixth-largest auto component maker, is slated to partner with Israeli chipset maker Autotalks Ltd. to develop 5G network-based vehicle-to-everything (V2X) integrated control technology to bolster the safety of autonomous vehicles”.[4]

“At the end of 2022 ITS America together with 5GAA and other transportation organizations reaffirmed their continued support for the "rapid, widespread deployment of Vehicle to Everything (V2X) technologies in order to further improve safety on American roads". [5]

[3] https://www.eea.europa.eu/publications/air-quality-in-europe-2022/air-quality-in-europe-2022
[4] https://auto-talks.com/hyundai-mobis-israels-autotalks-to-develop-v2x-for-self-driving-cars/
[5] https://www.itsinternational.com/its4/its6/its8/news/2023-pivotal-year-us-v2x

ACISA

ACISA is developing its new generation of Cooperative Intelligent Transport System (C-ITS) for urban and interurban environments, aspiring to leverage open technologies in the cloud and edge computing fields to achieve a fastest pace of innovation, more resilient systems, digital sovereignty and less siloed proprietary technologies. This will benefit both their customers, by avoiding vendor lock-in scenarios, and other stakeholders by attracting the vibrant open source community to the ITS domain.

For the last 2 years, ACISA's R&D team has been working on a new concept of Traffic Light Controller (TLC) that aims to leverage the privileged location held by each TLC in the layout of a city, to provision an enhanced distributed computing capacity, which enables real low-latency, high-capacity infrastructure to deploy advanced services for traffic management, video analytics, and mobility.

As a member of Aldesa Group, one of the largest construction groups in Spain, ACISA is a technological company that focuses on innovative engineering solutions offering comprehensive services within four industries:

Mobility **Security & control** **Energy efficiency** **Telecomms**

With 30 years of experience and a remarkably diversified portfolio of high-profile projects worldwide, we are constantly adapting our technological solutions to the needs of the global market on all levels: social, technological, financial, and environmental. Characterised by personalisation, scalability and adaptability, our software intelligence serves a variety of applications within the business areas of urban and interurban transit, public transport, fleet management and connected vehicle automation.

Believing in positive change, our products are designed to contribute towards a greener, safer and smarter future based on IoT, ITS & C-ITS, ATMS and MaaS – the areas where our R&D efforts are currently focused. By applying our next generation solutions to improve the quality of life and the efficiency of operations, we are very proudly collaborating with local, national and international stakeholders both from the public and private sector.

ACISA's long experience in the fields of urban and interurban infrastructure, energy efficiency, information technologies and telecommunication services give us the competitive advantage of quality that our clients trust and rely on.

- 1000 collaborators around the world
- Management of urban traffic in cities of 200,000+ inhabitants, including: Terrassa, Córdoba, Granada, Alicante, Barcelona and Madrid.
- Maintenance of 100,000+ public light points
- 600+ km of ITS installed on roads to reduce accident rates.

4. Use Case 2: Wildfire Detection

Across Earth's ecosystems, wildfires are growing in intensity and spreading in range. From Australia to Canada, the United States to China, across Europe and the Amazon, wildfires are wreaking havoc on the environment, wildlife, human health, and infrastructure. The costs in human lives and livelihoods can be counted in the number who perish in the flames, or contract respiratory diseases from the toxic smoke, or in the number of towns, homes, businesses, and communities affected by fire.

Not only can wildfires reduce biodiversity, but they contribute to a climate change feedback loop by emitting huge quantities of greenhouse gases into the atmosphere, spurring more warming, more drying, and more burning. When it comes to fighting wildfires, technology has very clear limitations. This is because controlling wildfire behaviour is highly dependent on the prevailing weather and fuel conditions, and accessibility. It is often only a change in weather that can help bring a wildfire under control. Therefore, the limits and appropriateness of suppression strategies and tactics and the associated suppression resource types should be overcome by new IT and AI applications to 1) reduce latency time between fire detection and fire extinction chain of operations 2) Improve fire behaviour predictions and associated smokes and particulate dispersion 3) improve safety of firefighters during field operations.

Nature 4.0 has developed a new device TT-fire which is able to detect real time flame events by using a novel sensor based on UV light emission, with a distance ranging of 200 m and an angle of view of 120°. At the same time gaseous compounds (CO2, O3, CH49 and particulate PM2.5-5-10) are measured. The devices send the data by NB-IoT or 5G connection given the possibly large amount of information that should be passed during a relatively short amount of time. At the fire occurrence also real time images are collected to evaluate the fire intensity and behaviour. The real time collected data streams are activated in a cascade of events that could involve all the wireless sensors of the network with an increased scalability of required computation resources. Such particular features require a new approach to cloud computing services and edge integration which is the core of the Cognit-EU project.

Nature 4.0

Nature 4.0 was born in May 2018 as a Benefit Company and innovative start-up following a series of innovations and technologies developed in university laboratories. Currently the company has created various IoT devices for monitoring environmental parameters with applications in the agricultural, forestry and marine fields, exploiting the most modern technologies. The company's flagship product is the TreeTalker® (patent no. 102019000013362 released on 07.21.2021), capable of measuring in real time the water consumption of trees, the growth of biomass (diameter) and the health status of the leaves through spectral indices and innovative processes. The latest version is the TT Cyber, which is available as a LoRa/LoRaWan node and also as a standalone device with NB-IoT communication.

Nature 4.0 is now recognized on the national and international market with its flagship product TreeTalker®, having attracted the attention of several scientific and industrial stakeholders. There is therefore a network of approximately 4000 TreeTalkers® currently operating in the world: Italy, Germany, France, Switzerland, Poland, Serbia, Spain, Russia, United Kingdom, USA, Canada, China and South Korea.

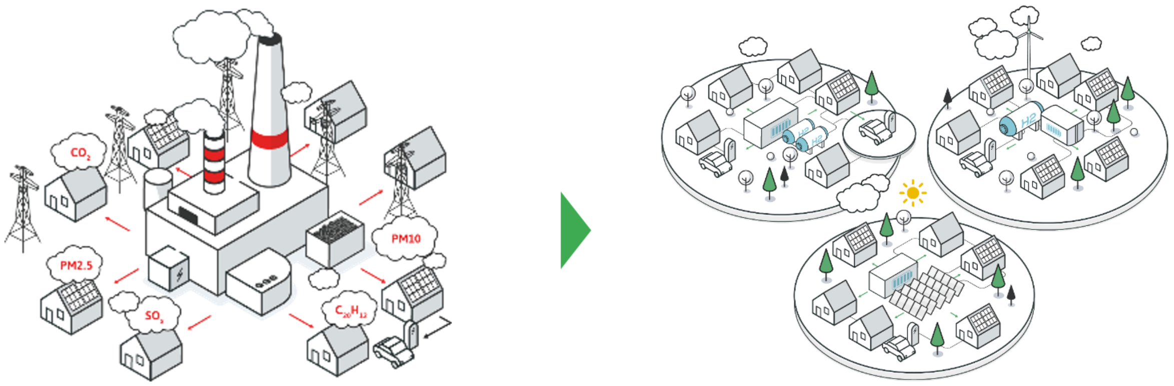
As follow up Nature 4.0 developed the TT-Fire device for real time monitoring of fires occurrence and behaviour predictions. The device has been successfully used by the Civil Protection agency of Puglia Region in Italy for their operations of fire prevention and suppression.
Nature4.0 is also producing several other IoT devices with application in agriculture, forestry and environmental monitoring:

- AnimalTalker®: for the real time animal tracking and health conditions (heart rate, body temperature, animal movements).
- TT-Air®: for air quality monitoring with specific applications with drone surveys (under patent).
- TT-Marine®: for ocean and coastal monitoring water quality parameters as well satellite communication.
- ESASPEC®: miniature- hand held spectrometer VIS-NIR bands for plant and food organic matrices characterization of chemical compounds.

5. Use Case 3: Energy

Energy sector is indisputably one of the most important as with the energy use and production it drives the world's economy. With the continuously increasing need for energy use and growth of renewable sources as well as diminishing fossil fuel resources (i.e. oil, gas, carbon) and the current geopolitical context, it is critical to develop solutions for individual energy independence of a household and/or small energy clusters. Those needs are the main driver of the sector transformation.

Supporting the energy sector transition in Europe requires allowing the wide and open accessibility of energy data. In order to provide that it is necessary to develop methods for monitoring, predicting, and managing both energy production and consumption in a given environment. For this, the implementation of smart edge applications that make use of advanced AI/ML algorithms is a clear need. The following figure summarises the current tendencies in the European energy sector:



- Centralized model - multiple recipients and one source
- Extensive transmission and distribution network - large energy losses
- Rising system maintenance costs = higher bills
- Every failure affects a significant part of the system

- Distributed model - set of energy communities
- Energy consumers can also be energy producers (prosumers)
- Energy produced locally is consumed locally
- Failure in a single community does not affect the system

Phoenix Systems

Phoenix Systems is the developer of the open source, real-time operating system Phoenix-RTOS, designed specifically for resource-constrained platforms, such as IoT devices. Phoenix Systems is developing next-generation electricity meters, leveraging the Phoenix-RTOS platform, with the ambition to turn it into an Energy Assistant capable of running user applications directly on the electricity meters and managing appliances relevant to household energy balancing. Phoenix-RTOS stands out for its reliability, flexibility and scalability of software projects for low resources platforms. The transformation of the European energy sector brings new challenges and requirements that electricity meters equipped with Phoenix-RTOS can meet.

Phoenix Systems currently operates on the Smart Grid market offering the following software-defined Edge-IoT devices based on Phoenix-RTOS and developed jointly with the industrial partners:

- smart gas meters – mass deployment of Apator Metrix iSmart gas meter in Belgium for Fluvius (1 M)
- smart data concentrators – large deployment of PRIME 1.3.6/1.4 DCUs manufactured by Andra, Energa-Operator and Apator gathering data from 1.5 M smart meters in the Energa-Operator grid in Poland (largest smart metering implementation in Poland)
- smart energy meter (Edge-IoT devices) - next-generation smart meter developed jointly with Apator S.A. (largest utility meters manufacturer in Poland) - now in the pre-manufacturing phase

Phoenix-RTOS—the open source scalable real-time operating system for Edge-IoT—is the core product of Phoenix Systems and has been developed for the last 20 years (initially on Warsaw University of Technology). The latest microkernel-based version was created in the years 2017-2020 as the result of the R&D project within the framework of Operational Programme Smart Growth 2014-2020.

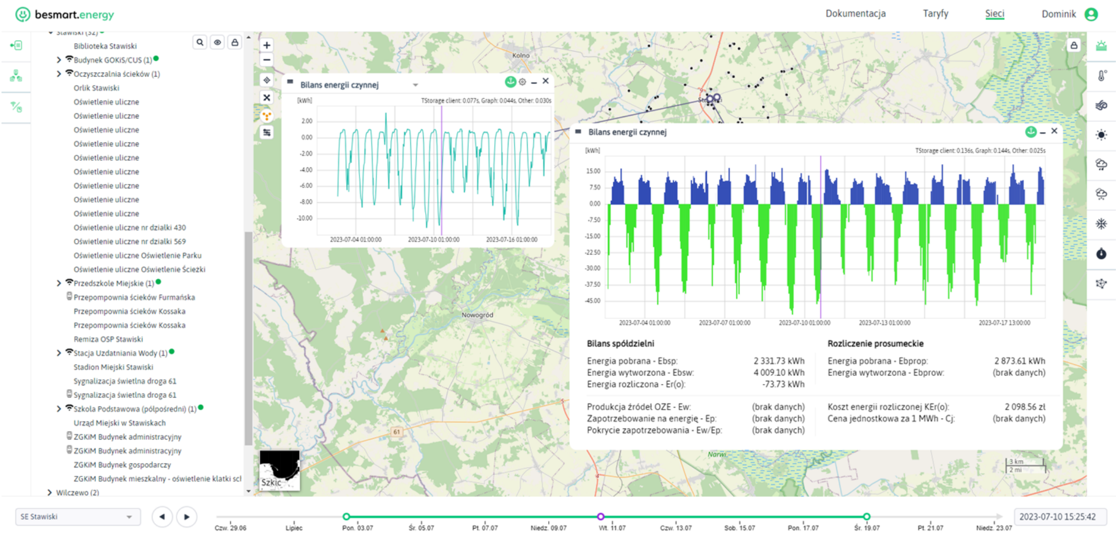
Phoenix-RTOS's goal is to facilitate software development for Edge-IoT devices and far-edge orchestration. This goal is being firstly implemented in the Smart Grid sector where Phoenix-RTOS effectively transforms smart meters into Edge-IoT devices that communicate with the cloud. With the use of Phoenix-RTOS on the electricity meter users will be able to create their own applications to manage power generators, energy storage or household appliances. This highly innovative idea leads to increased energy efficiency in line with EU Green Deal assumptions and will allow future, Phoenix-RTOS based, IoT devices to implement functionality allowing true, green orchestration while moving computing to the far edge.

Atende Industries

Atende Industries develops the *besmart.energy* cloud platform, which allows energy communities, such as energy clusters or energy cooperatives, to balance energy and increase self-consumption rates, thereby reducing electricity costs for their members. Built-in high accuracy weather forecasts allow to predict production from PV or wind farms. Artificial intelligence algorithms can predict the energy demand of energy community members, resulting in active control of the demand side.

In 2011, having the rich experience in massive data processing gained from multimedia industry, company decided to enter into the smart grid sector and fill the technology niche by developing the software for the biggest smart metering implementation in Poland (finally 3M of meters) conducted by Energa-Operator—one of the largest DSOs in Poland. The developed redGrid technology (MDM + HES) supports from the beginning until today Energa-Operator’s smart metering implementation gathering and processing data from about 2.5M of municipal and industry smart meters and sharing it to external systems (e.g. billing) and end-users.

In 2017 the company, based on experience gathered from smart metering and aiming to fulfil the EU energy sector transformation policy, decided to develop the besmart.energy platform for energy management for energy communities utilising renewables and energy storage technologies. The main goal of energy management is the improvement of the local energy balance and the maximisation of ROI from renewable technologies investment. The following figure shows Atende's Besmart.energy platform performing an analysis of energy balancing:



Bilans spółdzielni

Energia pobrana - Ebsp:	2 331.73 kWh
Energia wytworzona - Ebsw:	4 009.10 kWh
Energia rozliczona - Er(o):	-73.73 kWh
Produkcja źródeł OZE - Ew:	(brak danych)
Zapotrzebowanie na energię - Ep:	(brak danych)
Pokrycie zapotrzebowania - Ew/Ep:	(brak danych)

Rozliczenie prosumeckie

Energia pobrana - Ebprop:	2 873.61 kWh
Energia wytworzona - Ebprow:	(brak danych)
Koszt energii rozliczonej KEr(o):	2 098.56 zł
Cena jednostkowa za 1 MWh - Cj:	(brak danych)

Understanding the bottlenecks of traditional database technologies and specific requirements for Big Data processing algorithms and having the rich experience gained from object file system developments the company decided to develop the new, proprietary NoSQL database named TStorage. TStorage is devoted for versioned, time-stamped data, searched using key ranges queries. This database acts as the core and foundation technology of besmart.energy platform. After the extensive development enabling the offering of TStorage NoSQL database as the product it can be used as one of the foundation technologies offered by EU cloud service providers for efficient processing of timestamped and versioned data, provided by any kind of IoT devices (e.g. smart meters, robots, UAV's, satellites etc.).

6. Use Case 4: Cybersecurity

The application of innovation and technology to enhance and optimise mobility and transportation systems is known as the "smart mobility sector." It includes a broad range of services, products, and technologies designed to improve the effectiveness, sustainability, and convenience of transportation. This sector is critical for the EU for various reasons. It has an important potential for economic growth and job creation and it improves environmental sustainability with the promotion of electric vehicles and public transportation. Transportation can also be made more efficient with technologies such as intelligent transportation systems (ITS), autonomous vehicles, …

Technological challenges of the smart mobility sector include improving interoperability between the various technologies and systems, the development of a smart infrastructure with for example sensors for autonomous driving, new regulatory frameworks addressing safety, liability and data sharing.

Cybersecurity is a crucial challenge in the smart mobility sector. The safety of passengers and pedestrians is at risk in case of cyber attacks on the smart mobility systems. Hackers can cause harm and economic loss when creating disruptions and accidents by taking control of vehicles, traffic signals, ... Additionally, the data collected, processed and transmitted by smart mobility systems is often sensitive and needs to be protected accordingly.

Edge Computing poses a number of intrinsic challenges and risks that have to be properly addressed through research and innovation. Far from secured datacenters, edge systems face new threats and require a new generation of security controls. In a context where latency is key and a reliable access to the cloud is not guaranteed, security detection and remediation activities need to benefit from a certain level of autonomy and resilience. An edge system needs to be able to defend itself against a denial of service attack, using its sometimes limited resources and capabilities.

This Use Case will demonstrate an integrated cybersecurity solution for edge computing in a smart mobility context. The main challenge will be migration of vehicles between edge nodes, and how to enforce dynamic (geo-dependent) security policies and controls reliably and in an autonomous way.

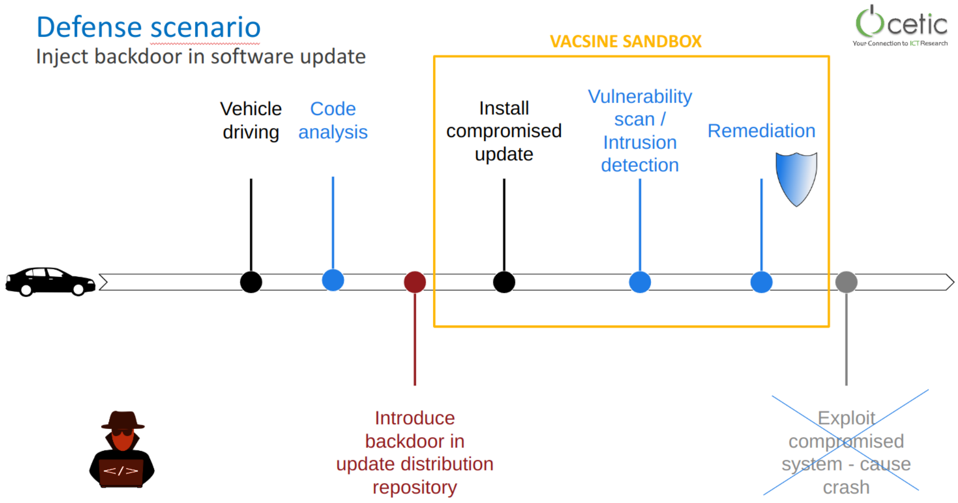
CETIC

As an applied research centre in the field of ICT, CETIC supports economic development by transferring the results of the most innovative research in ICT to companies. CETIC helps companies integrate these technological breakthroughs into their products, processes and services, enabling them to innovate faster, save time and money and develop new markets.

CETIC develops its expertise in key technologies, including trust and security of IT systems, Cloud/Edge/IoT, Big Data, and software quality. These innovations are applied in domains of primary importance to society, such as health, smart mobility, energy and industry.

In the context of the Win4Excellence Belgian programme[6], CETIC is building software factories to facilitate collaboration between Walloon research and industry actors, and the dissemination of research results. The software factories propose a service that integrates development, security and operations so that research results can be made available and demonstrated quickly and with a high level of quality. In 2022 and 2023, the factories have been deployed in various verticals, for example in security or artificial intelligence research and industry communities. In order to propose demonstration environments that are as close as possible to the industry context, the factories include edge capabilities, and integrations with cyber physical systems such as robots.

To manage the deployment of security services for DevSecOps, CETIC has developed the Vacsine tool. Vacsine is a digital immune system and provides continuous security orchestration to secure the whole software life-cycle "by design". It relies on continuous monitoring of systems to define, evaluate and apply automated countermeasures such as firewalls, intrusion detection systems, honeypots or quarantining. The automated response is triggered by changes to security requirements, indicators of compromise, incidents and vulnerabilities. The efficiency and speed of countermeasures deployment is evaluated in automatically provisioned sandbox environments that shadow the target systems. Those sandboxes provide observability and scalability for the training and maintenance of security response strategies.


Defense scenario
Inject backdoor in software update
VACSINE SANDBOX
cetic
Your Connection to ICT Research
Vehicle driving
Code analysis
Install compromised update
Vulnerability scan / Intrusion detection
Remediation
</>
Introduce backdoor in update distribution repository
~~Exploit compromised system - cause crash~~

A Securover demonstrator has been built at CETIC laboratories in order to show how Vacsine can help improve the security of robots and CPS. In this simple experiment, we show how the update process of autonomous robots can be attacked, the possible consequences of an attack and how to protect that system. Development of the Securover case study and Vacsine has started in the context of the Horizon 2020 research and innovation programme SPARTA. In 2022, Securover and Vacsine have been extended to provide a cyber range scenario in the CRS project. In 2023, in the context of the TRAIL and CyberExcellence projects, Vaccine has been included as a security component of the Win4Excellence factories and Securover has been integrated as a testbed for edge and cyber physical systems.

[6] https://recherche.wallonie.be/win4excellence

SUSE

SUSE is a global leader in innovative, open and secure infrastructure software solutions for multi-cloud environments, relied upon by more than 60% of the Fortune 500 to power their mission-critical workloads. We specialise in Business-critical Linux, Enterprise Container Management and Edge solutions, and collaborate with partners and communities to empower our customers to innovate everywhere – from the data center to the cloud, to the edge and beyond. SUSE puts the “open” back in open source, giving customers the agility to tackle innovation challenges today and the freedom to evolve their strategy and solutions tomorrow.

With over 30 years of experience on how to harden open source software to enterprise requirements, we know that infrastructure is a cornerstone towards security of modern applications. As a result we organise our development and build processes accordingly, to fulfil top requirements of regulations for critical environments. Proof Points are security certifications like Common Criteria EAL4+ or FIPS-140-2/3 for many of our solutions. Beside that, SUSE is investing in projects like NeuVector or COCONUT-SVSM to extend and complete the open source ecosystem in terms of Zero Trust and Confidential Computing.

Based on experience with many customers orchestrating their Kubernetes infrastructure with Rancher, SUSE identified multi-cluster and multi-tenant observability as a key challenge to reliably and securely operate such infrastructures. Although observability data is part of every Kubernetes environment, it is hard to gain any focused and structured insights from that huge amount of log data. Main reasons are the complex setups of available open source observability tools and the fact that for integrating AIOps for log anomaly detection common solutions require data scientist capabilities, which are hardly found in IT Operation teams. Based on these grounds SUSE started the development of OPNI, which focuses on Kubernetes for Logging, Monitoring and Tracing to complement Rancher environments. The over time modularised architecture and extension of capabilities now also allows to cover more general logging and monitoring use cases beyond the Kubernetes components, which makes it a candidate for integration into the Security Use Case regarding log anomaly detection.

Kubewarden is a dynamic admission controller for Kubernetes environments that validates incoming requests against WebAssembly policies to keep clusters secure and compliant. The policy engine does not require policy authors to learn a domain-specific language of the policy framework. Instead it allows the use of any programming language that supports WebAssembly as a compilation target. Beside lowering the barriers to write policies it also eases the operation, as developed policies can be distributed via container registries that are common part container infrastructures. Also the portable nature of once compiled policies as WebAssembly modules allows for deployment across architectures and operating systems. Addressing these various challenges, Kubewarden simplifies the adoption of policy-as-code for cloud-native applications.

7. New User Requirements

As an extension to Deliverable D5.1, a number of additional user requirements have been identified during the First Research & Innovation Cycle (M4-M9):

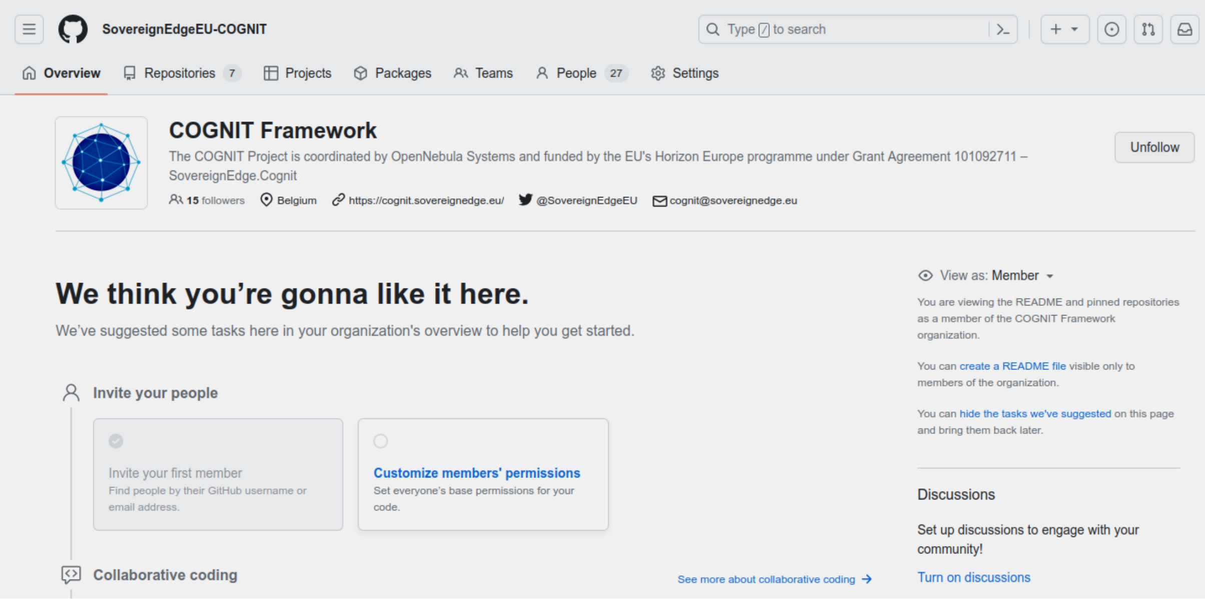
Id	Description	Source
UR0.9	IAM system integration for high granularity authentication and user management for device clients, provisioning engine and serverless runtime.	All
UR0.10	Push mechanism to inform about status or events from Provisioning Engine and Serverless runtime back to the requestor device client.	All

Table 7.1. New common user requirements identified in M4-M9.

PART II. Software Integration and Verification

8. Software Integration Process and Infrastructure

A GitHub organisation, called SovereignEdgeEU-COGNIT[7] (see Figure 8.1), is used as the development hub to aid in the software development and provide a framework for the integration of the different components that form the COGNIT software stack:



COGNIT Framework

The COGNIT Project is coordinated by OpenNebula Systems and funded by the EU's Horizon Europe programme under Grant Agreement 101092711 – SovereignEdge.Cognit

We think you're gonna like it here.

We've suggested some tasks here in your organization's overview to help you get started.

Invite your first member

Find people by their GitHub username or email address.

Customize members' permissions

Set everyone's base permissions for your code.

You are viewing the README and pinned repositories as a member of the COGNIT Framework organization.

You can create a README file visible only to members of the organization.

You can hide the tasks we've suggested on this page and bring them back later.

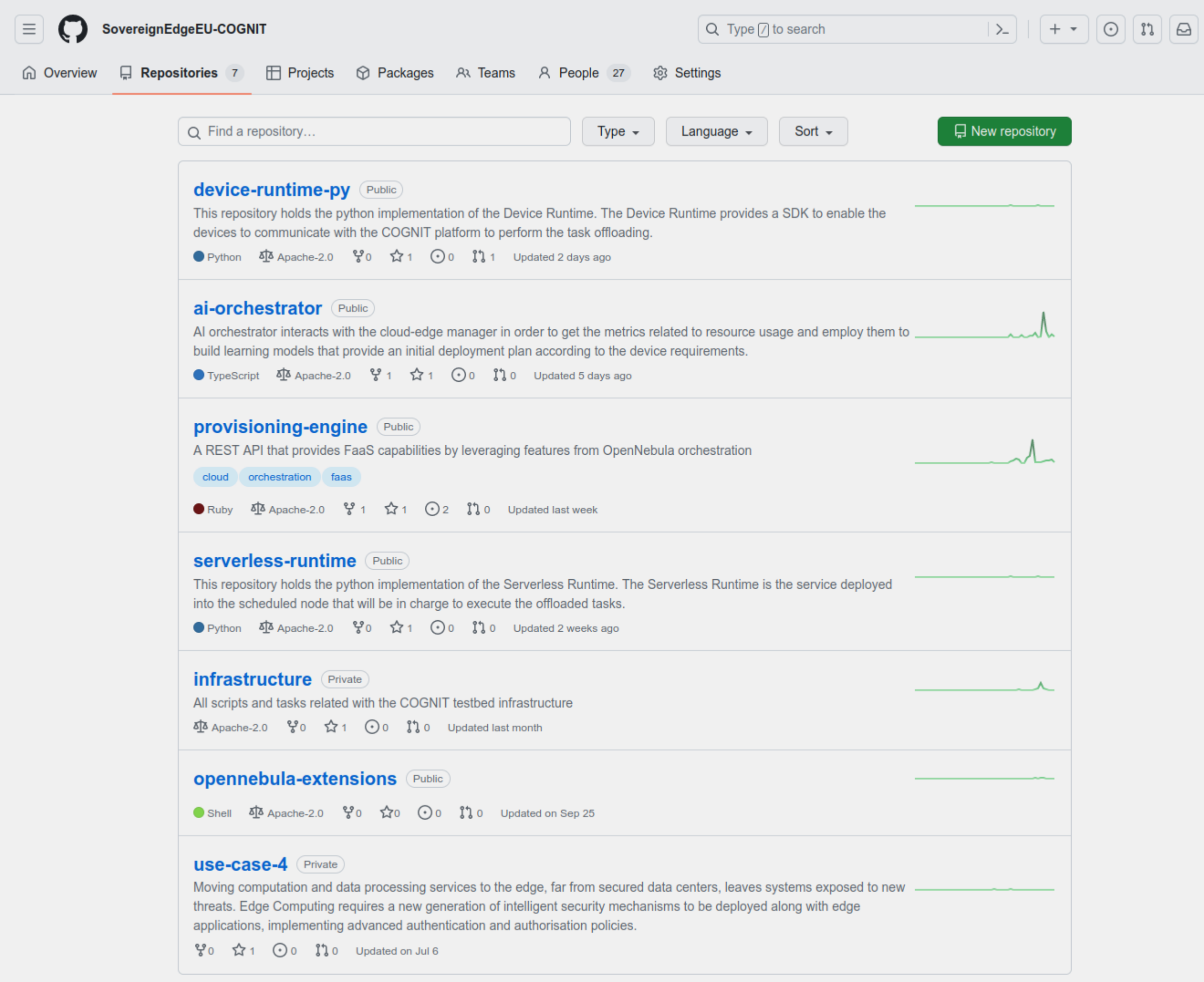
Discussions

Set up discussions to engage with your community!

Figure 8.1. GitHub organisation created by the COGNIT Project

Currently there are several repositories created under this organisation (see Figure 8.2), corresponding to different components of the COGNIT Framework (e.g. Device Client, Serverless Runtime, Provisioning Engine, etc.). Additionally, there are other repositories used for Use Case coordination and to contain the details and recipes for best practices in the shared COGNIT infrastructure.

[7] https://github.com/SovereignEdgeEU-COGNIT



device-runtime-py Public
This repository holds the python implementation of the Device Runtime. The Device Runtime provides a SDK to enable the devices to communicate with the COGNIT platform to perform the task offloading.
Python Apache-2.0 0 1 0 1 Updated 2 days ago

ai-orchestrator Public
AI orchestrator interacts with the cloud-edge manager in order to get the metrics related to resource usage and employ them to build learning models that provide an initial deployment plan according to the device requirements.
TypeScript Apache-2.0 1 1 0 0 Updated 5 days ago

provisioning-engine Public
A REST API that provides FaaS capabilities by leveraging features from OpenNebula orchestration
cloud orchestration faas
Ruby Apache-2.0 1 1 2 0 Updated last week

serverless-runtime Public
This repository holds the python implementation of the Serverless Runtime. The Serverless Runtime is the service deployed into the scheduled node that will be in charge to execute the offloaded tasks.
Python Apache-2.0 0 1 0 0 Updated 2 weeks ago

infrastructure Private
All scripts and tasks related with the COGNIT testbed infrastructure
Apache-2.0 0 1 0 0 Updated last month

opennebula-extensions Public
Shell Apache-2.0 0 0 0 0 Updated on Sep 25

use-case-4 Private
Moving computation and data processing services to the edge, far from secured data centers, leaves systems exposed to new threats. Edge Computing requires a new generation of intelligent security mechanisms to be deployed along with edge applications, implementing advanced authentication and authorisation policies.
0 1 0 0 Updated on Jul 6

Figure 8.2. GitHub repositories of the COGNIT Project

It is important to recognise the three distinct categories of software components that will be part of the COGNIT software stack. Firstly, the COGNIT Framework will integrate **existing components**, some of which may undergo modifications to align with the evolving needs and objectives of the Project. These existing components are the technological foundation upon which the COGNIT Framework is being constructed. An example of this is the Cloud-Edge Manager, based on the existing OpenNebula project, whose scheduler has already been expanded during the First Research & Innovation Cycle (M4-M9) to be able to delegate placement recommendations to the new AI-Enabled Orchestrator being developed by the COGNIT Project.

Furthermore, the COGNIT Framework envisions **improvements to existing open source components** that, while crucial for our operations, *may not necessarily find their way upstream* for broader utilisation. These tailored enhancements are intended to optimise the performance, reliability, and versatility of the COGNIT Framework, meeting our unique requirements and objectives. An example of this is the integration with Scaphandre as an expansion of the OpenNebula native monitoring subsystem. While interesting for the broader OpenNebula community, it is not yet mature enough to be contributed upstream. As such, this is contained in the “opennebula-extensions” GitHub repository.

Finally, the COGNIT Project will integrate **new components**. These contributions from Consortium Partners will bring fresh perspectives and functionalities into the COGNIT Framework, introducing new capabilities and extending the scope of possibilities in our pursuit of creating innovative European open source technologies for the cloud and edge. This approach embodies the adaptive and collaborative nature of this Project, with an example being the Provisioning Engine component of the COGNIT Framework, which is located in a GitHub repository with its own licence, test, and documentation.

Currently, the COGNIT Architecture comprises separate components that perform distinct functions. While this modular approach offers advantages in terms of scalability and flexibility, it also presents certain challenges. Each component is essentially self-contained, which means they function independently and interact at the API level. One notable aspect of our current status is the integration of these components primarily at the infrastructure level. This means that the components are deployed on the same infrastructure but may not be intricately linked or designed to work seamlessly together. Such infrastructure-level integration is often effective for ensuring that individual components are accessible and operational, but it may not maximise their combined potential.

The logical next step is to transition from this component-centric approach towards a scenario in which we are able to deliver a unified solution. This transition involves several key elements:

1. Single Deployment Process: To streamline the deployment of our system, we must consolidate the deployment process. This involves creating a standardised procedure to install, configure, and launch the entire system as a cohesive unit. This simplifies the deployment, reduces the likelihood of configuration errors, and ensures consistency across different environments.
2. Comprehensive Documentation: The move towards a unified solution requires comprehensive documentation. This documentation should encompass every aspect of the system, including architecture, component interactions, installation instructions, and configuration details. Having well-documented guidelines enhances usability and fosters better understanding among developers and users.
3. Script and Ansible Recipes: To further automate the deployment and maintenance processes, creating scripts and Ansible recipes is crucial[8]. Scripts can automate tasks such as setting up infrastructure details (whether on-premises or on public cloud/edge providers) and managing the system's credentials. Ansible recipes provide a higher level of automation, enabling the provisioning, configuration, and management of various system components through a simple, declarative language.
4. Centralised Configuration: Moving to a unified solution also involves centralising configuration management. This allows for consistent and efficient control of settings and parameters across all system components. Centralised configuration simplifies updates, reduces redundancy, and enhances the system's overall manageability.

[8] https://www.ansible.com

The transition from separate components and infrastructure-level integration towards a unified solution represents a significant leap forward. This shift promises enhanced efficiency, better control, and improved user experience. By creating a single deployment process, comprehensive documentation, and embracing automation, we aim to deliver a more robust and user-friendly solution that is adaptable to different environments and user needs.

There are three different aspects that we need to cover in future research and innovation cycles of the Project as part of that transition: 1) a way to manually *build* the whole COGNIT software stack, 2) a way to automatically *deploy* the COGNIT Framework into an infrastructure, and 3) a way to automate *testing* through a Continuous Integration approach. The following subsections elaborate on these key aspects:

Building

In order to deliver a single-solution experience, the first step is to produce an installer that is able to pull all the different components that creates the COGNIT Framework. A combination of Terraform[9] plans and associated Ansible playbooks will be produced to achieve this end.

The goal is to achieve a software artefact containing all the components needed for a user to be able to instantiate a fully operational version of the COGNIT Framework. These components can be pre-existing packages (e.g. OpenNebula core), extensions for these packages (e.g. the refactored OpenNebula Scheduler), specific appliances (e.g. a Serverless Runtime that needs to be accessible as a qcow2 image), and new components (e.g. a tarball containing all the Provisioning Engine source code).

Deployment

The building installer can be leveraged in this phase to deploy an instance of the COGNIT Framework on a pre-existing infrastructure. For this end, the parameters in the following table can be used in this stage:

Parameter	Description
Virtualization Host list	A list of comma separated IPs with root SSH enabled hosts that can be used as virtualization hosts for COGNIT.
Cloud/Edge Provider Credentials	Credentials to request resources from specific Cloud/Edge providers.
Cloud Provider Regions	Locations of a specific cloud/edge infrastructure provider to request resources from.

Table 8.1. COGNIT software stack deployment parameters

[9] https://www.terraform.io

In order to automate the deployment of an instance of the COGNIT Framework, the Project will leverage the OneHosted tool, developed by OpenNebula Systems in the context of the H2020 project **ONEedge** (2019-2022)[10]

OneHosted allows you to define, deploy, and use fully-managed OpenNebula instances. OneHosted offers an OpenNebula management front-end that has been deployed automatically on AWS virtual resources using a specific version of OpenNebula. The following figure shows an example of how a basic OneHosted instance looks like:

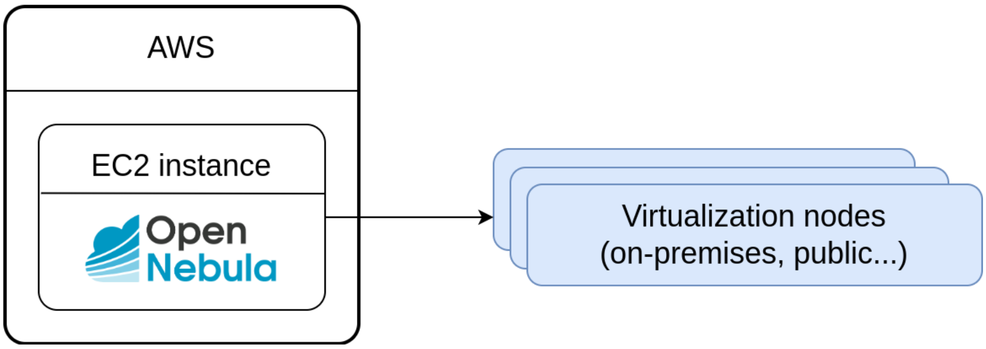


Figure 8.3. OneHosted Deployment

All the interaction with OneHosted is done through GitHub Issues and associated labels, which trigger GitHub Actions. These actions create tailored Terraform and Ansible documents according to the parameters provided in the GitHub Issues, which shape the produced OpenNebula environment. The life-cycle of a OneHosted instance can be seen in the figure below:

[10] https://oneedge.io

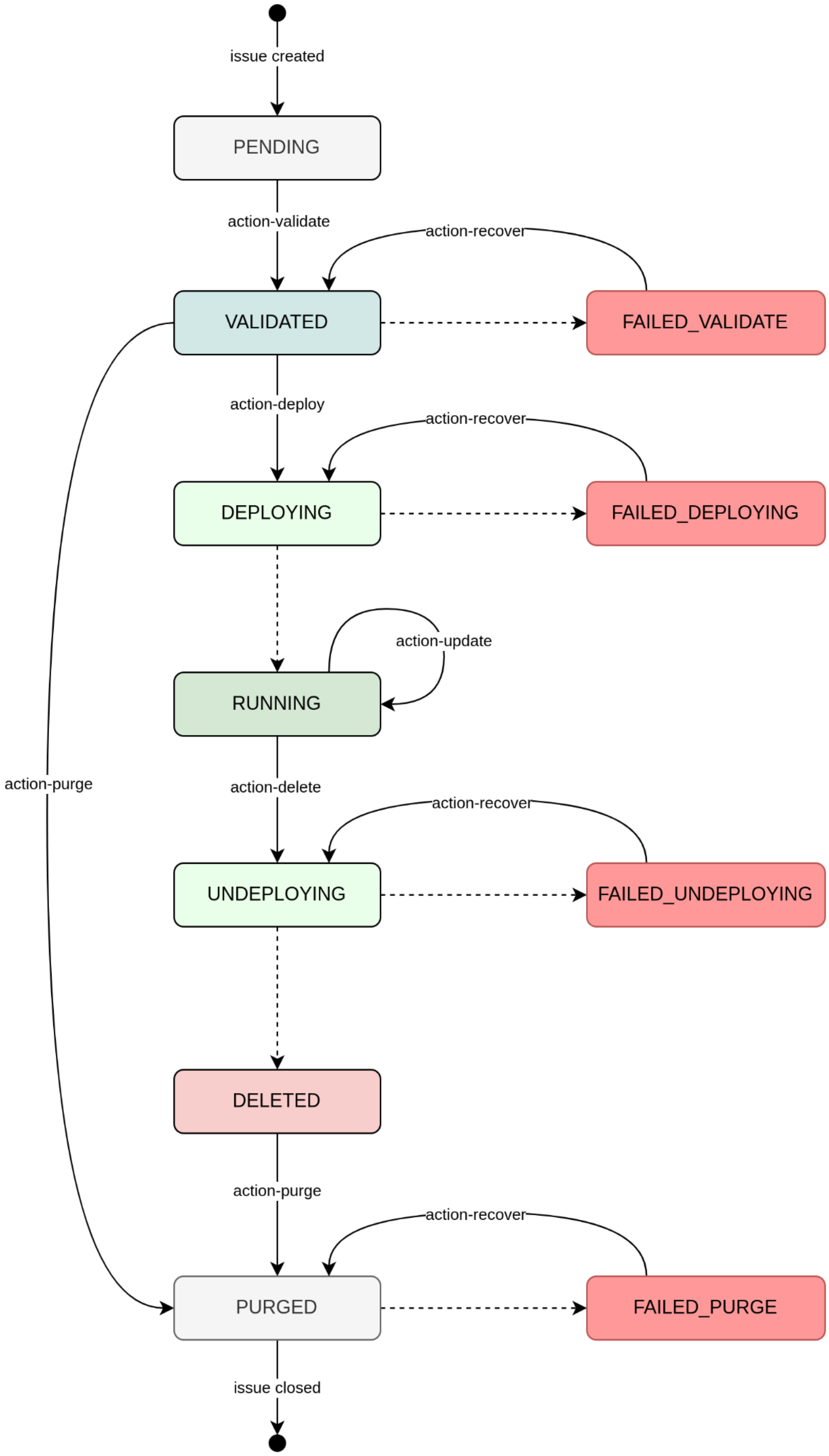


Figure 8.4. Life-cycle of a OneHosted instance.

All the states in Figure 8.4 are managed through the creation or deletion of GitHub issues, as well as to the assignment or removal of specific Labels to those issues.

We will leverage the OneHosted tool as part of the COGNIT Project, building on top of a solution initially developed under a previous research & innovation action funded by the H2020 programme (SME Instrument) and expanding it so that it is eventually able to easily manage the installation of all the different components of the COGNIT Framework.

Testing

As part of the creation of a unified solution, the COGNIT Projects needs to set up a Continuous Integration framework to be able to ensure the correctness of the software developed by the Project.

We are planning to follow a similar mechanism to the one implemented for the Provisioning Engine in WP3 (see Deliverable D3.1 for more details). In that case, each commit to the master branch of the Provisioning Engine repository[11] spawns a new GitHub-backed container where:

1. The Provisioning Engine repository is checked out.
2. All the Provisioning Engine dependencies are installed.
3. The Provisioning Engine is installed.
4. The Provisioning Engine is configured using the appropriate Cloud-Edge Manager endpoint.
5. The Provisioning Engine is launched.
6. A set of smoke tests are run (i.e. to create a new Serverless Runtime, update it, and delete it).
7. The Provisioning Engine is stopped.
8. The Provisioning Engine is uninstalled.

This comprehensive testing process ensures that errors are identified during the development phase as soon as possible. This check is compulsory before a new version of the Provisioning Engine is deployed on the COGNIT infrastructure.

This integration testing needs to be expanded in order to cover not only the Provisioning Engine, but all the other integrated components and functionality of the COGNIT Framework, and also in the future incorporating security tests related to penetration, vulnerability analysis, intrusion detection, and security remediation.

[11] https://github.com/SovereignEdgeEU-COGNIT/provisioning-engine

9. Testbed Environment

Nothing to report at this time in terms of new infrastructure resources or additional capabilities. Please, refer to the original information provided in Deliverable D5.1 about the COGNIT Testbed hosted at the RISE ICE Data Center in Luleå, Sweden.

10. Software Requirements Verification

Possible status are: NOT STARTED | IN PROGRESS | COMPLETED

10.1 Device Client

SR1.1 Interface with Provisioning Engine

Status: IN PROGRESS

Description: Users are able to download and build the Device Runtime Python library. Following the instructions[12] of the Project's GitHub repository, a user can configure the device runtime to point to a valid Provisioning Engine endpoint and communicate with the requested serverless runtime to offload the execution of a python function. All the library functionalities can be tested standalone by executing the unit tests provided on the GitHub repository[13].

Completed Verification Scenarios:

- [VS1.1.1] The Device Client asks the Provisioning Engine for a Serverless Runtime with certain characteristics and it receives the ID of the created Serverless Runtime.
- [VS1.1.2] The Device Client asks the Provisioning Engine for information about the created Serverless Runtime.
- [VS1.1.4] The Device Client requests the Provisioning Engine to delete a Serverless Runtime. The Device Client requests information about the Serverless Runtime to verify that it no longer exists.

Pending Verification Scenarios:

- [VS1.1.3] The Device Client requests the Provisioning Engine to update the features of a Serverless Runtime. The Device Client requests again information about the Serverless Runtime to verify that the Serverless Runtime has been modified.

SR1.2 Interface with Serverless Runtime

Status: IN PROGRESS

Description: Currently, the Device Client can communicate with the implemented Serverless Runtime component to upload a function, execute it, and retrieve the result.

Completed Verification Scenarios:

- [VS1.2.2] The Device Client uploads a function to the Serverless Runtime and

[12] https://github.com/SovereignEdgeEU-COGNIT/device-runtime-py/blob/main/README.md
[13] https://github.com/SovereignEdgeEU-COGNIT/device-runtime-py/tree/main/cognit/test

receives an acknowledgement.

- [VS1.2.3] The Device Client requests the execution of a function to the Serverless Runtime and receives the result of the execution.

Pending Verification Scenarios:

- [VS1.2.1] The Device Client uploads data from the device to the Serverless Runtime and receives an acknowledgement.
- [VS1.2.4] The Device Client requests transfer data from external resources to the Serverless Runtime and receives an acknowledgement.

SR1.3 Programming languages

Status: IN PROGRESS

Description: The current implementation of the Device Client only supports Python. The tested verification scenarios for the Python language are described above, at the SR1.2 section.

Completed Verification Scenarios:

- [VS1.3.2] Test previously described validation scenarios implemented in Python language.

Pending Verification Scenarios:

- [VS1.3.1] Test previously described validation scenarios implemented in C language.

SR1.4 Low memory footprint for constrained devices

Status: NOT STARTED

Pending Verification Scenarios:

- [VS1.4.1] Test validation scenarios described above on a device with less than 500kB of RAM.

SR1.5 Security

Status: NOT STARTED

Pending Verification Scenarios:

- [VS1.5.1] The Device Client asks the Provisioning Engine for a Serverless Runtime with the data encrypted and signed and the request is accepted.

- [VS1.5.2] The Device Client asks the Provisioning Engine for a Serverless Runtime without the data encrypted or signed and the request is refused.

10.2 Serverless Runtime

SR2.1 Secure and Trusted FaaS Runtimes

Status: IN PROGRESS

Description: The current COGNIT Framework is able to orchestrate on-demand VMs containing Serverless Runtimes through the Provisioning Engine. Currently the access to this API is not secured. In addition, the execution of the FaaS functions can be tested standalone. This component has been developed with a set of unit tests[14] that can be executed to validate that the component is able to execute and return the result of Python functions.

Completed Verification Scenarios:

- [VS2.1.1] Build and instantiate a FaaS Runtime image for Python language and test the execution of a function using a secure communication channel.

Pending Verification Scenarios:

- [VS2.1.2] Build and instantiate a FaaS Runtime image for C language and test the execution of a function using a secure communication channel.

SR2.2 Secure and Trusted DaaS Runtimes

Status: NOT STARTED

Pending Verification Scenarios:

- [VS2.2.1] Build and instantiate a DaaS Runtime image for SQL DB (e.g. MariaDB) and test uploading and copying data using a secure communication channel.
- [VS2.2.2] Build and instantiate a DaaS Runtime image for Object Storage (e.g. MinIO) and test uploading and copying data using a secure communication channel.

10.3 Provisioning Engine

SR3.1 Provisioning Interface for the Device to manage Serverless Runtimes

Status: IN PROGRESS

[14] https://github.com/SovereignEdgeEU-COGNIT/serverless-runtime/tree/main/app/test

Description: A new set of integration tests were developed in this cycle as GitHub Actions[15], that in each commit (or by request) performs the following steps:

- creates a new fresh vanilla container
- installs the Provisioning Engine dependencies
- checks out the whole Provisioning Engine repository
- installs the latest Provisioning Engine from master
- configures the Cloud-Edge Manager endpoint. We are using the COGNIT infrastructure OpenNebula endpoint for this purpose
- starts the Provisioning Engine service
- runs a predefined set of tests

Since we are using the COGNIT OpenNebula deployment, the whole stack below the Provisioning Engine is exercised. This means that on each test, a new VM containing a Serverless Runtime is created and subsequently deleted, which also indirectly exercises the AI-Enabled Orchestrator.

This workflow[16] and associated tests[17] can be inspected in the Provisioning Engine GitHub's repository. The tests cover all the Verification Scenarios stated below, and other corner cases to ensure no regressions are introduced in subsequent cycles: missing authorization, invalid Serverless Runtime flavours, VMs with different characteristics (multiple NICs, DISKs), invalid Serverless Runtime representations, etc. The full checklist implemented so far can be consulted in the corresponding ticket[18] in the repository. We will be extending these tests as new functionality is added to the Provisioning Engine.

Completed Verification Scenarios:

- [VS3.1.1] A YAML file with the device requirements is provided to the Provisioning Engine and it returns the Serverless Runtime ID.
- [VS3.1.2] Query the Provisioning Engine to return the status of a Serverless Runtime identified by its ID.
- [VS3.1.4] Delete a Serverless Runtime providing its ID.

Pending Verification Scenarios:

- [VS3.1.3] A YAML file with the updated device requirements is provided to the Provisioning Engine that updates the associated Serverless Runtime.

[15] https://github.com/features/actions
[16] https://github.com/SovereignEdgeEU-COGNIT/provisioning-engine/blob/main/.github/workflows/rspec.yaml
[17] https://github.com/SovereignEdgeEU-COGNIT/provisioning-engine/tree/main/tests
[18] https://github.com/SovereignEdgeEU-COGNIT/provisioning-engine/issues/19

10.4 Cloud-Edge Manager

SR4.1 Provider Catalog

Status: NOT STARTED

Pending Verification Scenarios:

- [VS4.1.1] Listing the providers belonging to the Provider Catalog.
- [VS4.1.2] Filtering the providers according to a desired latency threshold on a geographic area.
- [VS4.1.3] Filtering the providers according to a cost per hour threshold.
- [VS4.1.4] Filtering the providers according to energy consumption per hour threshold.
- [VS4.1.5] Filtering the providers according to some specific hardware characteristics (e.g. GPUs, Trusted Execution Environments).

SR4.2 Edge Cluster Provisioning

Status: NOT STARTED

Pending Verification Scenarios:

- [VS4.2.1] A YAML file containing the information about the provision is provided to the Cloud-Edge Manager that creates a new Edge Cluster.
- [VS4.2.2] Query the Cloud-Edge Manager to return the status of an Edge Cluster identified by its ID.
- [VS4.2.3] Query the Cloud-Edge Manager to scale up/down the number of hosts of an Edge Cluster identified by its ID.
- [VS4.2.4] Query the Cloud-Edge Manager to delete an Edge Cluster identified by its ID.

SR4.3 Serverless Runtime Deployment

Status: IN PROGRESS

Description: The deployment of a Serverless Runtime by the Cloud-Edge Manager is currently being tested in two different Q&A processes. The first one is indirectly in the certification process carried out by upstream OpenNebula, which stresses the OneFlow component using most of the functionality required by COGNIT. The other one is directly linked with the Provisioning Engine tests, which act as an end to end integration test where an actual Serverless Runtime is deployed, retrieved, and deleted.

Completed Verification Scenarios:

- [VS4.3.1] A YAML file containing the information about the deployment is provided to the Cloud-Edge Manager that creates a new Serverless Runtime.
- [VS4.3.2] Query the Cloud-Edge Manager to return the status of a Serverless Runtime identified by its ID.
- [VS4.3.4] Query the Cloud-Edge Manager to update the deployment of the Serverless Runtime identified by its ID.
- [VS4.3.5] Query the Cloud-Edge Manager to delete a Serverless Runtime identified by its ID.

Pending Verification Scenarios:

- [VS4.3.3] Query the Cloud-Edge Manager to scale up/down the resources (CPU, memory and disks) of a Serverless Runtime identified by its ID.

SR4.4 Metrics, Monitoring, Auditing

Status: IN PROGRESS

Description: Development is still in process, so there are currently no integration tests that stress the Verification Scenarios.

Pending Verification Scenarios:

- [VS4.4.1] Create an Edge Cluster and deploy a Serverless Runtime and check the metrics collected for a certain period of time.

SR4.5 Authentication & Authorization

Status: IN PROGRESS

Description: Development is still in process, so there are currently no integration tests that stress the Verification Scenarios.

Pending Verification Scenarios:

- [VS4.5.1] Test the creation of new users and groups.
- [VS4.5.2] Assign ACLs to designated users and test the creation of new Edge Clusters and Serverless Runtimes.
- [VS4.5.3] Communicate with Provisioning Engine using a secure channel.

10.5 AI-Enabled Orchestrator

SR5.1 Building Learning Model

Status: IN PROGRESS

Description: Development is still in process, so there are currently no integration tests that stress the Verification Scenarios.

Pending Verification Scenarios:

- [VS5.1.1] List instances from Devices to Applications to System for metrics to be collected.
- [VS5.1.2] Correlate and represent features that ready to take as input to the Model.
- [VS5.1.3] Feedback-aware performance check when train the model on represented features.
- [VS5.1.4] Assess the ability in terms of AUROC score for each task (e.g. scheduling).

SR5.2 Smart Deployment of Serverless Runtimes

Status: IN PROGRESS

Description: Development is still in process, so there are currently no integration tests that stress the Verification Scenarios.

Pending Verification Scenarios:

- [VS5.2.1] Users Quality of Service (QoS) / Quality of Experience (QoE) will check for each AI-Enabled Orchestrator decision for deployment of Serverless Runtimes.

[NEW] SR5.3 Scheduling Mechanisms

Status: IN PROGRESS

Description: The refactor of the OpenNebula Scheduler to request placement suggestions from an external module is currently tested in the context of the upstream OpenNebula Q&A process. These tests need an external scheduler implementing a REST API conforming to the requirements of the AI-Enabled Orchestrator; for this end a mockup AI-Enabled Orchestrator has been implemented in the Q&A process of OpenNebula, implementing a simple round-robin scheduling algorithm.

Completed Verification Scenarios:

- [VS5.3.1] The Cloud-Edge Manager must allocate Virtual Machines to hosts when an external scheduler is configured.
- [VS5.3.2] The external scheduler must receive a request whenever there are Virtual Machines in pending states.

Pending Verification Scenarios:

- [VS5.3.3] The external scheduler must receive a request for replacement whenever there are Virtual Machines in running states.

10.6 Secure and Trusted Execution of Computing Environments

SR6.1 Advanced Access Control

Status: IN PROGRESS

Description: Development is still in process, so there are currently no integration tests that stress the Verification Scenarios.

Pending Verification Scenarios:

- [VS6.1.1] Define a security policy that based on geographic zone attribute.
- [VS6.1.2] Check enforcement of new security policy when edge device moves closer from one edge node than another.

SR6.2 Confidential Computing

Status: IN PROGRESS

Description: Development is still in process, so there are currently no integration tests that stress the Verification Scenarios.

Pending Verification Scenarios:

- [VS6.2.1] Deploy a function on a host that provides confidential computing capability.
- [VS 6.2.2] Check that the function is executed inside the host trusted execution environment (TEE).

SR6.3 Federated Learning

Status: NOT STARTED

Pending Verification Scenarios:

- [VS6.3.1] Perform training of the ML algorithm without exchanging local data.
- [VS6.3.2] Check that the redistributed models for inference do not contain private data.

11. Conclusions and Next Steps

On the basis of the initial version of the Use Cases Scientific Report (Deliverable D5.1), this first incremental version provides an overview of the overall status of the contribution of the Project’s software requirements towards meeting the user requirements that guide the development of the COGNIT Framework, offering additional information about the domains targeted by the Use Cases and the Partners involved in them, listing new user requirements identified during the First Research & Innovation Cycle (M4-M9), and providing an update on the Project’s software integration process and infrastructure, on its testbed environment, and on the progress of the software requirement verification tasks per component.

This document (D5.2) complements the Project's global overview provided by Deliverable D2.2, as well as the component-specific research and development activities reported in Deliverables D3.1, D3.6, D4.1, and D4.6.

This first incremental version of the Use Cases Scientific Report (Deliverable D5.2) offers a summary of the work done in the First Research & Innovation Cycle (M4-M9). Additional incremental versions of this report will be released at the end of each research and innovation cycle (i.e. M15, M21, M27, M33).