Platone
PLATFORM FOR OPERATION
OF DISTRIBUTION NETWORKS

Platone

PLATform for Operation of distribution NEtworks

D2.16

Platone Integrated Framework Prototype (v3)

The project PLATform for Operation of distribution NEtworks (Platone) receives funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement no 864300.

Project name	Platone
Contractual delivery date:	31.08.2023
Actual delivery date:	31.08.2023
Main responsible:	Ferdinando Bosco (ENG)
Work package:	WP2 – Platform Implementation and Data Handling
Security:	P
Nature:	DEM
Version:	V1.0
Total number of pages:	61

Abstract

The Platone Open Framework aims to create an open, flexible, and secure system that enables distribution grid flexibility/congestion management mechanisms, through innovative energy market models involving all the possible actors at many levels (DSOs, TSOs, customers, aggregators). The Platone Framework is an open-source framework based on blockchain technology that enables a secure and shared data management system, allows standard and flexible integration of external solutions (e.g., legacy solutions), and is open to integration of external services through standardized open application programming interfaces (APIs).

This document accompanies the software delivery of the final integrated prototype of the Platone Open Framework. This version of the Platone Open Framework includes the final versions of the Platone Platforms: Platone Market Platform, Platone DSO Technical Platform and Platone Blockchain Access Layer.

In addition, the final list of implemented functionalities and the final integration in the different demos sites are reported. Finally, a technological scalability assessment is performed and the corresponding results are presented.

Keyword list

Platone Open Framework, Platform Integration, Deployment, Testing and Validation

Executive Summary

"Innovation for the customers, innovation for the grid" is the vision of project Platone - Platform for Operation of distribution Networks. Within the H2020 programme "A single, smart European electricity grid", Platone addresses the topic "Flexibility and retail market options for the distribution grid". Modern power grids are moving away from centralised, infrastructure-heavy transmission system operators (TSOs) towards distribution system operators (DSOs) that are flexible and more capable of managing diverse renewable energy sources. DSOs require new ways of managing the increased number of producers, end users and more volatile power distribution systems of the future.

Platone is using blockchain technology to build the Platone Open Framework to meet the needs of modern DSO power systems, including data management. The Platone Open Framework aims to create an open, flexible and secure system that enables distribution grid flexibility/congestion management mechanisms, through innovative energy market models involving all the possible actors at many levels (DSOs, TSOs, customers, aggregators). It is an open-source framework based on blockchain technology that enables a secure and shared data management system, allows standard and flexible integration of external solutions (e.g. legacy solutions), and is open to integration of external services through standardized open application program interfaces (APIs). It is built with existing regulations in mind and will allow small power producers to be easily certified so that they can sell excess energy back to the grid. The Platone Open Framework will also incorporate an open-market system to link with traditional TSOs. The Platone Open Framework will be tested in three European demos.

This solution, based on a two-layer blockchain architecture, and named Platone Open Framework, allows to integrate in easy way both the data coming from the devices installed on the physical infrastructure of distribution grid, as well any other external platform.

The Platone Open Framework offers a configurable and customizable architecture, that can be exploited by the DSOs for empowering their solutions.

In particular, the final prototype of the Platone Open Framework (v3) includes:

- The final version of the Platone Blockchain Access Layer (BAL) that provides an interoperable layer for the integration of Internet of Things devices and external Data Server, ensuring data privacy and security mechanisms.
- The final version of the Platone DSO Technical Platform (DSOTP) that allows the integration of external platforms as Distribution Management System as well as specific DSO services as State Estimation (SE) Tool and Data Visualisation
- The final version of the Platone Market Platform that enables a transparent and shared Flexibility Marketplace based on blockchain technology and open to all the market participants (TSOs, DSOs and aggregators)

The final prototype of the Platone Open Framework described in this deliverable was intended as a consolidation of the results achieved in the first two iterations and integrates the final version of the Platone Platforms, without any final validation or execution in the field test.

Despite this, due to some requirements not consolidated in the previous versions and some tools not fully integrated, the latest version of the Platone Framework was integrated into the Greek and German demo, in order to mainly check the new functionalities offered in the latest version of the DSOTP.

In addition, in collaboration with WP3 and the Italian Demo partners, a scalability assessment of the entire Platone Open Framework has been performed in order to validate the integrated framework functionally and from a technological scalability perspective, too.

Authors and Reviewers

Main responsible		
Partner	**Name**	**E-mail**
ENG		
	Ferdinando Bosco	Ferdinando.bosco@eng.it
Author(s)/contributor(s)		
Partner	**Name**	
ENG		
	Angelo Triveri	Angelo.triveri@eng.it
RWTH		
	Florian Oppermann	florian.oppermann@eonerc.rwth-aachen.de
	Juan Galeano	adolfo.galeano@eonerc.rwth-aachen.de
Reviewer(s)		
Partner	**Name**	
HEDNO		
	Effrosyni Maria Gralista	e.gralista@deddie.gr
Avacon		
	Benjamin Petters Navreet Dult	Benjamin-georg.petters@avacon.de Navreet.dult@avacon.de
Approver(s)		
Partner	**Name**	
RWTH		
	Amir Ahmadifar	amir.ahmadifar@eonerc.rwth-aachen.de

Table of Contents

1 Introduction

The project "PLATform for Operation of distribution Networks – Platone" aims to develop an architecture for testing and implementing a data acquisition system based on a two-layer Blockchain approach: an "Access Layer" to connect customers to the Distribution System Operator (DSO) and a "Service Layer" to link customers and DSO to the Flexibility Market environment (Market Place, Aggregators, …). The two layers are linked by a Shared Customer Database, containing all the data certified by Blockchain and made available to all the relevant stakeholders of the two layers. This Platone Open Framework (POF) architecture allows a greater stakeholder involvement and enables an efficient and smart network management. The tools used for this purpose will be based on platforms able to receive data from different sources, such as weather forecasting systems or distributed smart devices spread all over the urban area. These platforms, namely, the Platone Blockchain Access Layer (BAL), the Platone DSO Technical Platform (DSOTP), and the Platone Market Platform (MP), by talking to each other and exchanging data, will allow collecting and elaborating information useful for DSOs, transmission system operators (TSOs), Market, customers and aggregators. In particular, the DSOs will invest in a standard, open, non-discriminatory, blockchain-based, economic dispute settlement infrastructure, to give to both the customers and to the aggregator the possibility to become flexibility market players. This solution will allow the DSO to acquire a new role as a market enabler for end users and a smarter observer of the distribution network. By defining this innovative two-layer architecture, Platone strongly contributes to aims to removing technical and economic barriers to the achievement of a carbon-free society by 2050 creating the ecosystem for new market mechanisms for a rapid roll out among DSOs and for a large involvement of customers in the active management of grids and in the flexibility markets [1]. The Platone platform is tested in three European demos (Greece, Germany and Italy. The Platone consortium aims to go for a commercial exploitation of the results after the project is finished. Within the H2020 programme "A single, smart European electricity grid" Platone addresses the topic "Flexibility and retail market options for the distribution grid".

All these platforms were integrated in the final prototype of the POF described in this deliverable.

The integration phase was conducted considering the interoperability mechanisms and standards, as well as the system requirements expected for each Platone platform as result of the Use Cases described in the different demos.

In addition, all the feedback and insights collected from the previous iterations were addressed and used for the consolidation of the system to be released in the final version.

1.1 Task 2.6

This deliverable is related to the Task 2.6 that aims at releasing the POF prototype, following an iterative approach. Three versions of the integrated Framework prototype were delivered in an incremental way considering the results of the testing and validation phase performed during the pilot executions.

In addition, this task ensured that Platone Platforms can communicate with each other over the framework, as indicated in the system design and the system requirements defined in D2.1 [2] and then refined and updated in D2.2 [3].

1.2 Objectives of the Work Reported in this Deliverable

This deliverable is the final one in series of three and reports the final prototype release of the POF. This deliverable is intended as a demonstrator. This document accompanies the software release with the final architecture description, the final status of the requirements implemented and the scalability assessment.

1.3 Outline of the Deliverable

The second chapter of this document describes an overview of the POF and some technical details on the integration phase and the release of the final prototype. It also includes the final list of requirements implemented for each Platone Platforms, as expected in Deliverable D2.2 [3].

Chapter 3 provides the description on how the final prototype of the POF was integrated within the demo architectures.

Chapter 4 reports the scalability assessment conducted in collaboration with the Italian Demo partners.

Finally, chapter 5 concludes this deliverable.

1.4 How to Read this Document

The document aims to give an overview to the POF final prototype release. A description of the final list of expected functional and non-functional requirements can be found in D2.2 [3]. A detailed description of the demo use cases can be found in D1.1 [4] while more details on the final implementations of each single platform can be found respectively in D2.5 (MP) [5], D2.8 (DSOTP) [6], and D2.13 (Blockchain Access Layer) [7]. This deliverable consolidates the integration of the three previous releases of the above-mentioned platforms. For more information about the first two prototypes of the Platone integrated framework and their respective integration pleases refer to the D2.14 [8] and D2.15 [9].

2 POF – Third Prototype

POF was implemented following the updated architecture specification and functional requirements described in D2.2 [3] as well as the interoperability mechanisms reported in D2.9 [10] and D2.10 [11].

Following an iterative process, the final prototype of the framework was developed adapting the existing ones (first version described in D2.14 [8] and second version described in D2.15 [9]) according to the feedback collected in the first evaluation phase.

Even if the final version of the POF was intended as a consolidation of the previous ones, some of the missing requirements were added in the final version of the Platone Platforms (see Ch. 2.1) and a final version of the integrated prototype was provided and tested within the Greek and German Demo for the final evaluation of the missing requirements (details on Ch. 3)

All the final versions of the prototypes are public available and under open-source licenses on the RWTH Gitlab repository [12].

In terms of functional architecture, no updates were needed in the final version of the POF. Figure 1 illustrates the final version of the architecture without any updates.

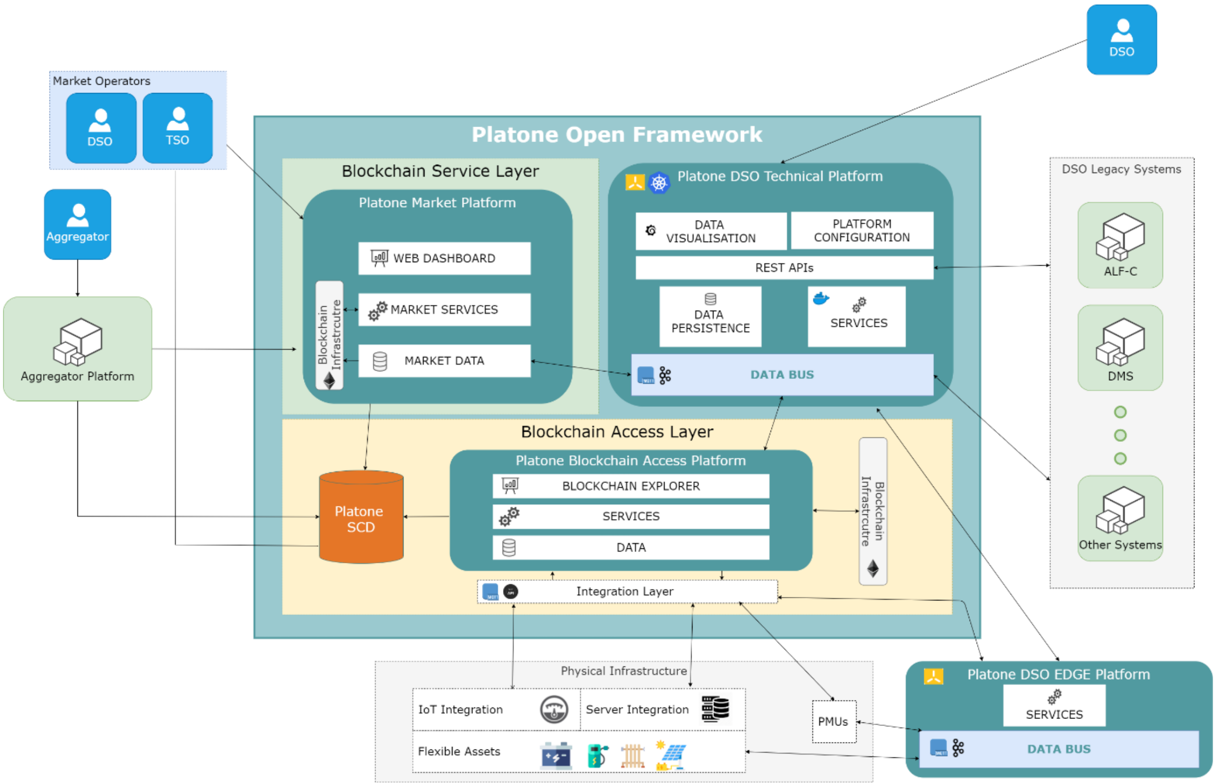

Figure 1: POF Architecture (v3)

2.1 Platone Platforms Integration

Starting from the output of D2.5 [5], D2.8 [6] and D2.13 [7], the activities with respect to the platforms integration were focused on enabling the inter-communication of the different Platone Platforms, ensuring a secure and scalable deployment process of the entire Framework.

In particular, in this final prototype, we focussed on the following activities:

- Integration of the final version of the Platone BAL.
- Integration of final version of the Platone DSOTP.
- Integration of the final version of MP with the Italian demo architectural components (Italian DSOTP, Aggregator Platform and Italian BAL).

2.1.1 BAL (v3) Integration

The Platone BAL is the entry point for the data collected from the network grid. The final prototype of BAL, described in D2.13 [7] implements two specific interoperable layers for the integration: on the one hand, the integration layer enables the integration of the physical infrastructure and allows the collection of the data, and on the other hand, the communication layer enables the integration with other systems and actors who want to use that data.

The final version of BAL did not include any new interfaces or communication mechanism, so did not require any new specific integration activity in the POF prototype. However, it included a new version of the Data Management Tool based on blockchain technology and leveraging on smart contracts for data access management. In particular, the smart contract is able to register any subscription request and verify it during the access request.

This upgrade made it possible to have all the functionalities of the BAL based on blockchain technology and smart contracts, ensuring the benefits of this technology at any level and increasing the level of transparency, security and trustworthiness.

Figure 2 shows the updated version of the Data Management Tool, including now the integration with the blockchain infrastructure.

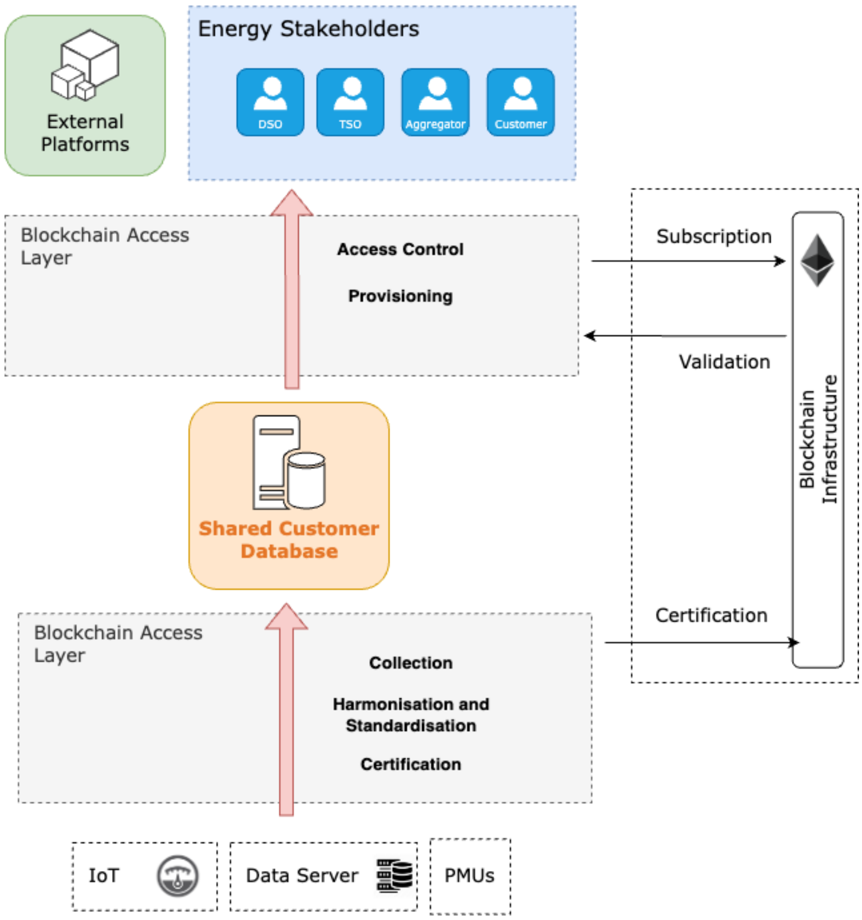

Figure 2: BAL Data Access Management Tool

Since the Blockchain technology and related infrastructure were already part of the second version of BAL and therefore of the POF, the new version of the Data Management tool did not have any impact on the integration activities of BAL itself which was released and automatically upgraded in the final prototype.

2.1.2 DSOTP (v3) Integration

The Platone DSOTP enables distribution system operators to fulfil market requests by evaluating the current grid state and activating local flexibility requests while ensuring the reliability and operational quality of services by enlarged grid observability. The platform design builds on previous work done in the Horizon 2020 project SOGNO [13] and relies massively on a micro-service architecture in which a DSO can easily deploy additional services into the platform. The first and second prototypes of the DSOTP is described in D2.6 [14] and D2.7 [15].

The third release contains a redesign of the load-balancing service. The load-balancing service was renamed Python Microgrid Flexibility Management service (pymfm) to be more recognizable as a concrete software solution. The code for the server part of the service was rewritten completely including a restructuring of the endpoints. The server is now based on FastAPI [16] which automatically generates and hosts an OpenAPI specification [17] and Swagger-UI which allows direct testing of the endpoints as well as use of standardized tools like client code generators and automated API testing. FastAPI also adds validation to the data input improving detection of simple errors.

The structure of the API is now more closely aligned with the REST paradigm. The API is now centred around a single resource resembling a balancing job. All endpoints concerning the resource as a type are available at the endpoint “<hostname>/balancing/” while all endpoints concerning a specific job are directed to “<hostname>/balancing/<ID>”. A summary of the endpoints can be found in Table 1.

Table 1: Endpoints of the pymfm service

Endpoint	Method	Description	Input	HTTP code	Response
/balancing/	GET	List all job IDs.	-	200 OK	List of all job IDs in JSON format.
				400 Bad Request	Error response
	POST	Create a new job for flexibility control.	JSON Entity Payload	200 OK	Job details
				400 Bad Request	Error response.
				422 Unprocessable content	JSON description of unprocessable content.
/balancing/<ID>	GET	Get status of the jobs including available results.	-	200 OK	Job details.
				400 Bad Request	Error response.
	DELETE	Delete a specific job.		200 OK	Job details.

Lastly, a globally available version of the pymfm service in the form of a Docker image was released [18] and a Helm chart for users to easily deploy and manage the service is also available in the SOGNO chart repository [19]. This approach enhances accessibility and simplifies deployment for a wide range of users.

All of these changes aim to improving the usability of the pymfm service, especially in circumstances where there is no direct support by the developers.

2.1.3 MP (v3) Integration

The Platone MP aimed to enable a fully secure and transparent Flexibility Market open to all the market participants and exploiting blockchain technology and smart contracts for handling the management of flexibility services. This approach provides market results to all the stakeholders, validates the flexibility provisioning, and performs the settlement outcome with an innovative incentivisation mechanism for improving customer engagement.

The second version of the Platone MP was already released with all the expected functionalities, so, the final prototype, includes only some minor additional features. These new features are:

- Logging system for scalability evaluation of the platform in a stand-alone and distributed architecture environment.
- Calculation of the KPIs expected for the Italian demo evaluation.
- Dataset export.

These features were mainly used for the evaluation of the results of the Italian Demo (KPIs) and for the scalability assessment.

None of these features impacts the integration with other platforms. Therefore, the final version of the MP was released and automatically upgraded in the final prototype without any integration activity, focusing only on the development and provisioning of the features above mentioned without interfering with other activities.

2.2 Platone Platforms Requirements – Status

This chapter summarizes the list of requirements expected for the three Platone Platforms (BAL, DSOTP and MP) in the different versions, based on the requirements elicited in D2.1 and planned during the whole project duration.

2.2.1 Platone MP

Table 2: Platone MP requirements

Requirement ID	Requirement name	Requirement description	Use Cases	Status	Notes
FR_MP_I_1	Initialisation	MP is able to receive PoDs information and PoM association from SCD in order to initialize a new market session	UC-IT-1 UC-IT-2	**Completed**	New in the second version
FR-MP-FSM-01	Flexibility Services Management	MP allows DSOs and TSOs to create flexibility requests in an automatic way	UC-IT-1 UC-IT-2	**Completed**	
FR-MP-FSM-02	Flexibility Services Management	MP allows DSOs to create flexibility requests through User Interface (UI)	UC-IT-1 UC-IT-2	**Cancelled**	The creation of the market requests and offers is performed automatically from the external platforms (DSOTP and Aggregator Platform). UI is no longer required.
FR-MP-FSM-03	Flexibility Services Management	MP allows Aggregator Platform to create flexibility offers in an automatic way	UC-IT-1 UC-IT-2	**Completed**	
FR-MP-FSM-04	Flexibility Services Management	MP acquires and stores all the flexibility requests and offers	UC-IT-1 UC-IT-2	**Completed**	
FR-MP-MOMV-01	Market Outcomes Matching and Validation	MP is able to match flexibility requests and offers through market clearing algorithms	UC-IT-1 UC-IT-2	**Completed**	
FR-MP-MOMV-02	Market Outcomes	MP is able to provide the Market Outcomes (results of market clearing) to DSOTP for the technical validation	UC-IT-1 UC-IT-2	**Completed**	

	Matching and Validation				
FR-MP-MOMV-03	Market Outcomes Matching and Validation	MP receives the validated market outcomes from DSOTP	UC-IT-1 UC-IT-2	**Completed**	
FR-MP-MOMV-04	Market Outcomes Matching and Validation	DSOs, TSOs and Aggregators receive Market Day Ahead outcomes from the Market Platform	UC-IT-1 UC-IT-2	**Completed**	
FR-MP-SA-01	Services activation	The MP allows DSOs and TSOs to create service activation requests in automatic way	UC-IT-1 UC-IT-2	**Cancelled**	The service activation is not the responsibility of the Market Operator and cannot be performed into the MP
FR-MP-SA-02	Services activation	The MP allows Market participant to create service activation requests through UI	UC-IT-1 UC-IT-2	**Cancelled**	The service activation is not the responsibility of the Market Operator and cannot be performed into the MP
FR-MP-SA-03	Services activation	The MP is able to aggregate the service activation requests (from DSOs and TSOs) and provide them to all the other stakeholders	UC-IT-1 UC-IT-2	**Cancelled**	The service activation is not the responsibility of the Market Operator and cannot be performed automatically into the MP
FR-MP-BC-01	Blockchain certification	MP is able to register on the blockchain all the market data trough Smart Contracts based functionalities	UC-IT-1 UC-IT-2	**Completed**	New in the second version
FR-MP-BC-02	Blockchain certification	MP allows Market participant to verify all the market data registered in the blockchain	UC-IT-1 UC-IT-2	**Completed**	New in the second version

FR-MP-SET-01	Settlement	MP allows Aggregator to create new smart contracts with settlement mechanisms via UI	UC-IT-1 UC-IT-2	**Completed**	New in the second version
FR-MP-SET-02	Settlement	MP provides to Aggregator Platform a list of available Smart Contracts with settlement mechanisms	UC-IT-1 UC-IT-2	**Completed**	New in the second version
FR-MP-SET-03	Settlement	MP is able to read meters measurements from SCD	UC-IT-1 UC-IT-2	**Completed**	**Renamed from** FR-MP-S-01
FR-MP-SET-04	Settlement	MP performs the settlement comparing the metering data and BSP baseline	UC-IT-1 UC-IT-2	**Completed**	Renamed from FR-MP-S-02. BSP replaced BRP.
FR-MP-SET-05	Settlement	BAL is able to provide tokenization system for the settlement through Smart Contracts functionalities	UC-IT-1 UC-IT-2	**Completed**	Renamed from FR-MP-S-03. New in the second version
FR-MP-SET-06	Settlement	MP allows DSO, TSO and Aggregator to read the settlement outcomes	UC-IT-1 UC-IT-2	**Completed**	FR-MP-S-04
MP – Non-Functional Requirements					
P-MP-01	Communication protocols	MP exposes REST APIs for collecting flexibility requests and flexibility offers	UC-IT-1 UC-IT-2	**Completed**	
P-MP-02	Communication protocols	MP provides a message broker for communicating market results	UC-IT-1 UC-IT-2	**Completed**	
T-MP-01	Communication Protocols, Timing	MP is able to receive measurements from SCD Kafka Broker every 15 minutes	UC-IT-1 UC-IT-2	**Completed**	

T-MP-02	Timing	MP is able to schedule day ahead and real time Market sessions at prefixed times and in automatic way	UC-IT-1 UC-IT-2	**Completed**	
S-MP-01	Security	MP must expose all its REST APIs under Oauth2.0 authentication and client credentials	UC-IT-1 UC-IT-2	**Completed**	
S-MP-02	Security	MP must identify all the Kafka clients using two-way authentication and server/client certificates	UC-IT-1 UC-IT-2	**Completed**	
S-MP-03	Security	All the MP interfaces must be exposed using (Transport Layer Security) TLS connections	UC-IT-1 UC-IT-2	**Completed**	

2.2.2 Platone DSOTP

Table 3: Platone DSOTP requirements

Requirement ID	Requirement name	Requirement description	Use Cases	Status	Notes
FR-DSOTP-DA-01	Data Acquisition	DSOTP is able to receive measurements that reflect the network state from DSO Data Server	UC-GR-1 UC-GR-2 UC-GR-3 UC-GR-4 UC-GR-5	**Completed**	
FR-DSOTP-DA-02	Data Acquisition	DSOTP is able to receive data coming from State Estimation Tool	UC-GR-1 UC-GR-2 UC-GR-3 UC-GR-4 UC-GR-5	**Completed**	
FR-DSOTP-DA-03	Data Acquisition	DSOTP is able to receive Phasor Measurement Unit (PMU) measurements that reflect the network state	UC-GR-2 UC-GR-3 UC-GR-4 UC-GR-5	**Completed**	
FR-DSO-TP-DA-04	Data Acquisition	DSOTP is able to receive certified measurement from BAP	UC-GR-1 UC-GR-2 UC-GR-3 UC-GR-4 UC-GR-5 UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4	**Completed**	

FR-DSO-TP-DA-05	Data Acquisition	DSOTP is able to receive setpoints from EMS	UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4	**Cancelled**	After an internal technical evaluation, it was decided to avoid sending the setpoints through the DSOTP since is too complex to be implemented during the project phase.
FR-DSOTP-SE-01	State Estimation	DSOTP is able to trigger the State Estimation Tool via REST API.	UC-GR-1 UC-GR-2 UC-GR-3 UC-GR-4	**Completed**	The State Estimation Tool is deployed inside the DSOTP and can be triggered via REST APIs
FR-DSOTP-SE-02	State Estimation	DSOTP provides DSO with the estimated state vector resulting from the State Estimation Tool.	UC-GR-1 UC-GR-2 UC-GR-3 UC-GR-4 UC-GR-5	Completed	
FR-DSOTP-PMU-01	PMU Data Integration	DSOTP is able to integrate PMU and conventional measurements into a unified measurement set, to be processed by the State Estimation Tool.	UC-GR-2 UC-GR-3 UC-GR-4	**Partially Completed**	For the integration of the SE tool in the DSOTP, in order to process all data coming from existing measurement systems and the PMUs, the development of data converters was required and completed as part of WP2 and WP4. Currently, near real time operation of the tool has been achieved with SCADA and automated meter reading data. Data sets with PMU measurements have only been tested and not yet applied.
FR-DSOTP-T-01	Tariffs retrieval	DSOTP sends to the DSO/Aggregators tariffs that reflect the expected state of the network	UC-GR-3 UC-GR-4	**Completed**	There is a file used now and shows how the data of the tariff should look like. Based on it, the DSOTP can be able to send the data.
FR-DSOTP-T-02	Tariffs retrieval	DSOTP is able to receive data coming from the Algorithm for Distributed Energy Resource (DER) Control and Algorithm for ancillary services	UC-GR-3 UC-GR-4	**Completed**	The DSOTP can receive data via the internal MQTT broker.

FR-DSOTP-AS-01	Data to DER control and Ancillary Services	DSOTP is able to send the output of the state estimation to external tools (DER control and Ancillary Services tools)	UC-GR-3 UC-GR-4	**Completed**	Output though Telegraf output plugins every output supported by Telegraf is available. Specifically, MQTT is available.
FR-DSOTP-DER-01	Optimal DER dispatching	DSOTP is able to trigger the Algorithm for DER Control via REST API	UC-GR-3	**Completed**	The tariffs are precalculated and loaded to the DSOTP via an excel file
P-DSOTP-01	Communication protocols	DSOTP is able to receive data from PMUs via MQTT protocol	UC-GR-2 UC-GR-3 UC-GR-4 UC-GR-5	**Completed**	
P-DSOTP-02	Communication protocols	DSOTP is able to receive data from DSO Data Server via TCP/IP protocol	UC-GR-1 UC-GR-2 UC-GR-3 UC-GR-4 UC-GR-5	**Completed**	
P-DSOTP-03	Communication protocols	DSOTP is able to receive setpoints from Avacon Local Flex Controller via MQTT and/or HTTP protocol	UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4	**Cancelled**	After an internal technical evaluation, it was decided to avoid sending the setpoints through the DSOTP since is too complex to be implemented during the project phase.
T-DSOTP-01	Timing	DSOTP is able to receive measurement every 10 seconds from sensors	UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4	**Completed**	
T-DSOTP-02	Timing	DSOTP is able to receive measurement every 15 minutes from Data Management Backend	UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4	**Completed**	

T-DSOTP-03	Timing	DSOTP is able to receive setpoints every 10 seconds for CBES and every 15 minutes for HBES	UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4	**Cancelled**	After an internal technical evaluation, it was decided to avoid sending the setpoints through the DSOTP since is too complex to be implemented during the project phase.

2.2.3 Platone BAL

Table 4: Platone Blockchain Access Layer requirements

Requirement ID	Requirement name	Requirement description	Use Cases	Status	Notes
P-BAP-01	Communication protocols	The BAP is able to receive data from sensors via MQTT protocol	UC-GR-2 UC-GR-3 UC-GR-4 UC-GR-5 UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4	**Completed**	
FR-BAP-DM-01	Blockchain Data Management	The BAP is able to acquire Measurements from grid.	UC-GR-1 UC-GR-2 UC-GR-3 UC-GR-4 UC-GR-5 UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4	**Completed**	

FR-BAP-DM-02	Blockchain Data Management	The BAP certifies Measurements via Smart Contracts	UC-GR-1 UC-GR-2 UC-GR-3 UC-GR-4 UC-GR-5 UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4	**Completed**	
FR-BAP-DM-03	Blockchain Data Management	The BAP provides certified measurement in a secure way to DSOTP	UC-GR-1 UC-GR-2 UC-GR-3 UC-GR-4 UC-GR-5 UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4	**Completed**	
FR-BAP-NC-01	Network Control	The BAP is able to receive set points from DSOTP	UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4	**Cancelled**	Not Implemented. The device control process is already implemented using Avacon Local Flex Controller and the integration through BAL was deemed unnecessary for the needs

					of the German Demo
FR-BAP-NC-02	Network Control	The BAP certifies set points via Smart Contracts	UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4	**Cancelled**	Not Implemented. The device control process is already implemented using Avacon Local Flex Controller and the integration through BAL was deemed unnecessary for the needs of the German Demo
FR-BAP-NC-03	Network Control	The BAP is able to send certified set points to Data Management Backend	UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4	**Cancelled**	Not Implemented. The device control process is already implemented using Avacon Local Flex Controller and the integration through BAL was deemed unnecessary for the needs

					of the German Demo
P-BAP-02	Communication protocols	The BAP is able to integrate data coming from external server via TCP/IP protocol	UC-GR-1 UC-GR-2 UC-GR-3 UC-GR-4 UC-GR-5 UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4	**Completed**	
T-BAP-01	Timing	BAP is able to receive measurement every 10 seconds from sensors	UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4	**Completed**	
T-BAP-02	Timing	BAP is able to receive measurement every 15 minutes from Data Management Backend	UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4	**Completed**	
S-BAP-01	Security	All the external interfaces of the BAP must be under TLS connection	UC-GR-2 UC-GR-3 UC-GR-4 UC-GR-5 UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4	**Completed**	

3 Platone Demo sites Integration and Deployment

3.1 Italian Demo

Since no additional requirements in terms of integration were made in the Italian Demo version of the POF prototype, no final integration and deployment was performed. The activity of the Italian Demo in the final stage consisted of the evaluation of KPIs, reported in D1.7 [20], the extraction and reporting of specific data sets and the scalability assessment reported in Ch. 4.

3.2 Greek Demo

The main objectives of the Greek demonstration are to economically optimize the use of distributed energy sources to provide ancillary services and balancing market participation to system operators, advanced observability and optimal control of distributed energy sources both in the day-ahead and real-time frames for market participation, mitigation of congestions and voltage limit violations, and minimization of losses.

All these objectives are pursued by POF. The Platform allows managing the integration and certification of data coming from the grid and at the same time, makes them available in real time to the DSOTP. That enables DSOs to exploit several innovative services, such as evaluating the current grid state, activating local flexibility requests, as well as ensuring the reliability and operational quality of services by enlarged grid observability.

Due to the use of different programming languages for the deployment of the State Estimation was necessary to implement a solution to ensure interoperability among different parts of the service in an automated way. The measurement data comes in via an MQTT-broker in the Common Information Model format. It is translated to the internal format used by the SE, which is PSS/E, and saved to a file. This file is then used as input to the core state-estimation algorithm. The SE writes the output to a file in a custom tabular format. In addition, a python script was developed in order to parse this output into a JSON message that conforms to the default JSON format for measurements used by Telegraf. Afterwards, this JSON message is released to an internal MQTT broker from where it is fed into an InfluxDB via Telegraf. The input, SE, and output have been chained by using the Linux CLI tool inotify [21] which waits for the intermediate files to change and then triggers the next step in the pipeline.

The input, SE, and output part of the process has been combined in a single Docker image that can be configured via environment variables. A Docker compose file for the necessary components of SE, MQTT-broker, Telegraf, InfluxDB and Grafana has also been added.

In addition to the Grafana Dashboard showing the data from state-estimation as time series graphs, a custom web app has been developed for presenting the network state as a whole. A detailed structure of the integration is depicted in Figure 3, where green refers to the custom software, and yellow corresponds to pre-made solutions that were configured for our purposes.

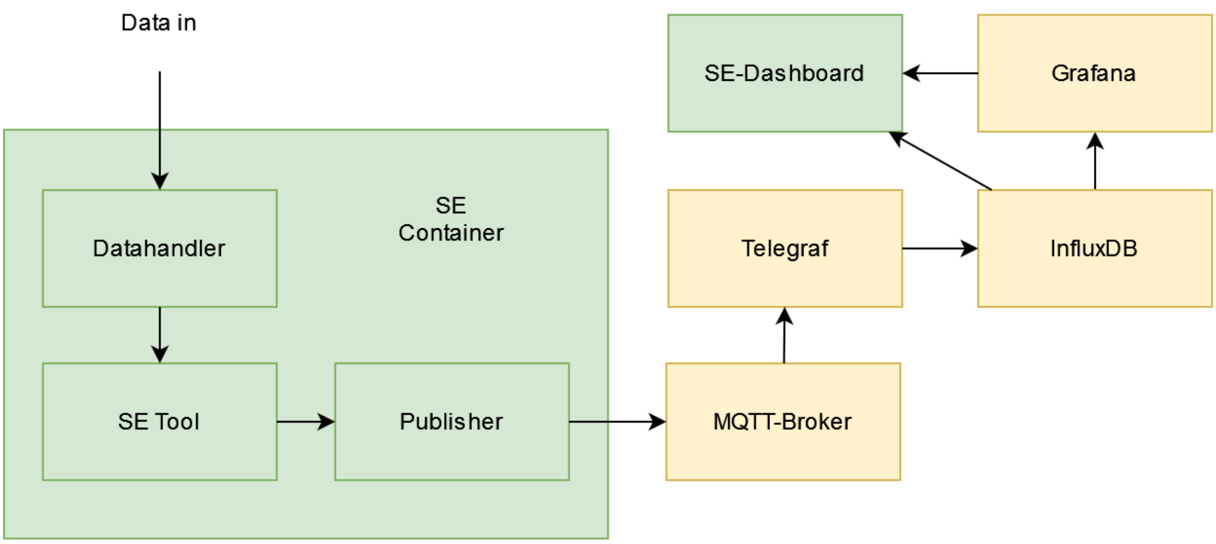

Figure 3: Structure of state-estimation deployment.

3.2.1 Deployment

The updated version of the platforms and the integrated POF were released following the same approach upgrading existing previous versions.

The POF configuration for the Greek Demo was set-up on a virtual machine and thanks to the use of Docker, the new deployment required only a new network configuration.

3.3 German Demo

The main objectives of the German demonstration are the coordination between local balancing mechanism and centralized grid operation and the allocation of flexibility in local networks between the local network and higher-level networks. Additionally, an effective informational and temporal uncoupling of low and medium voltage networks by handling energy supply and export in bulk packages rather than a real time exchange was also targeted for the German demonstrator.

For this third iteration of the Platone framework integration, the integration between the German Demo and the DSOTP is done by using distinct approaches. In one hand, the DSOTP uses the MQTT protocol with Secure Sockets Layer encryption to transmit data that is previously pre-processed to the German Demo Azure endpoint. A script was developed to reformat the data in the Azure platform. On the other hand, the German demo can securely communicate with the flexibility management pymfm service via HTTPS, utilizing JSON payloads for structured exchanges and a defined API as described in Ch. 2.1.2.

There were also slight improvements to the stability of the data processing pipeline of the DSOTP, adding automatic reconnection to the power calculation service.

3.3.1 Deployment

The third version of the POF is integrated in the German Demo cloud infrastructure based on MS Azure.

From the first version of the Framework, the deployment was incrementally done in the German Demo infrastructure using Docker for the packaging of the software and Kubernetes for the deploying and the orchestration of the services and CI/CD pipelines for managing the evolution of the software.

This deployment approach allowed an easy upgrade of the POF from the first version to this last one. For this last one, the newest version of the pymfm service (previously call load-balancing-service) described in Ch.2.1.2 has been deployed to the RWTH Kubernetes and is being tested.

4 Scalability Evaluation of POF

In order to complement the simulations that were performed in the WP7 activities, an assessment of the technological scalability of the POF was also performed in cooperation among WP2 and WP3.

The Italian Demo was selected among the others since its architecture includes all the components of the POF and suits well the scalability evaluation due to its complexity in terms of actors involved (TSO, DSO, Aggregator, Market Operator) and implemented processes.

This technological scalability aims at assessing the performances of the POF when the number of DERs connected to the system increases.

The technological scalability addressed two main aspects:

- Platform scalability
- System scalability

The **Platform scalability** level monitoring checks the state of each single platform during the execution of specific business processes. In particular, it is important to monitor how a single platform is able to scale in terms of computational load when the number of users involved in the test environment increase.

The **System scalability** level monitoring checks the state of the overall system and its workflow and processes through a set of performance metrics. Since POF is a complex architecture, it is important to monitor how the different platforms interact with each other for implementing specific business processes without compromising the performance of the entire system.

4.1 Methodology

The assessment was performed in a large-scale environment, starting from the use cases already implemented in the Italian demo and extending it, increasing the numbers of DER involved in the simulation and testing it in three different stages.

In fact, in order to monitor specific scalability KPIs at different stages, an incremental approach was implemented, in which the number of the involved DER grows at each stage. This approach allowed to track the performances of the different platforms and the entire system.

At each stage, specific KPIs at platform and system level were collected and stored by the platforms themselves. At the final stage, it was possible to compare the different results in the different stages against the expected results.

- **First stage**, the number of DER involved is the same of the Italian Demo evaluation phase (about a dozen). In this iteration, first input data are collected.
- **Second stage**, the number of users is increased x10.
- **Final stage**, the number of users is scaled up to simulate a near real-case environment with approximately the 30% of DER of the Italian demonstrator geographical area. 30% of the DERs represents a good approximation of the potential customers involved in the local flexibility market of the entire Rome area.

4.1.1 KPIs

The following KPIs were collected at Platform and System levels:

Platform level

As described in Ch. 4.1**,** at platform level it was important to monitor how a single platform is able to scale in terms of computational load. For this reason, the following Platform KPIs were collected:

- CPU Load, CPU utilisation during the business processes
- Memory Occupancy, memory utilization of the platform
- Uptime, availability of the platform during the execution

Table 5 shows the details of collected Platform KPIs.

Table 5: Platform Scalability KPIs - Details

KPI	Unit	Description
CPU Load Avg. (core)	Percentage (%)	Monitored during every execution – Week average
CPU Load Max (core)	Percentage (%)	Monitored during every execution – Week Max
RAM Usage Avg.	MegaBytes (MB)	Monitored during every execution – Week average
RAM Usage Max.	MegaBytes (MB)	Monitored during every execution – Week Max
Uptime	Percentage (%)	Monitored during the entire assessment – Unique Value

System Level

As described in Ch. 4.1**,** at system level it was important to monitor how the different platforms interact with each other for implementing specific business processes without compromising the performance of the entire system.

For this reason, the KPIs collected at the system level are:

- Execution time: the time of the execution of the processes for each flow (in milliseconds) .
- Communication time: the time of the communication (networking) of the results for each business process (in milliseconds).

As shown in Figure 4, several communication flows were implemented for the Italian Demo Architecture.

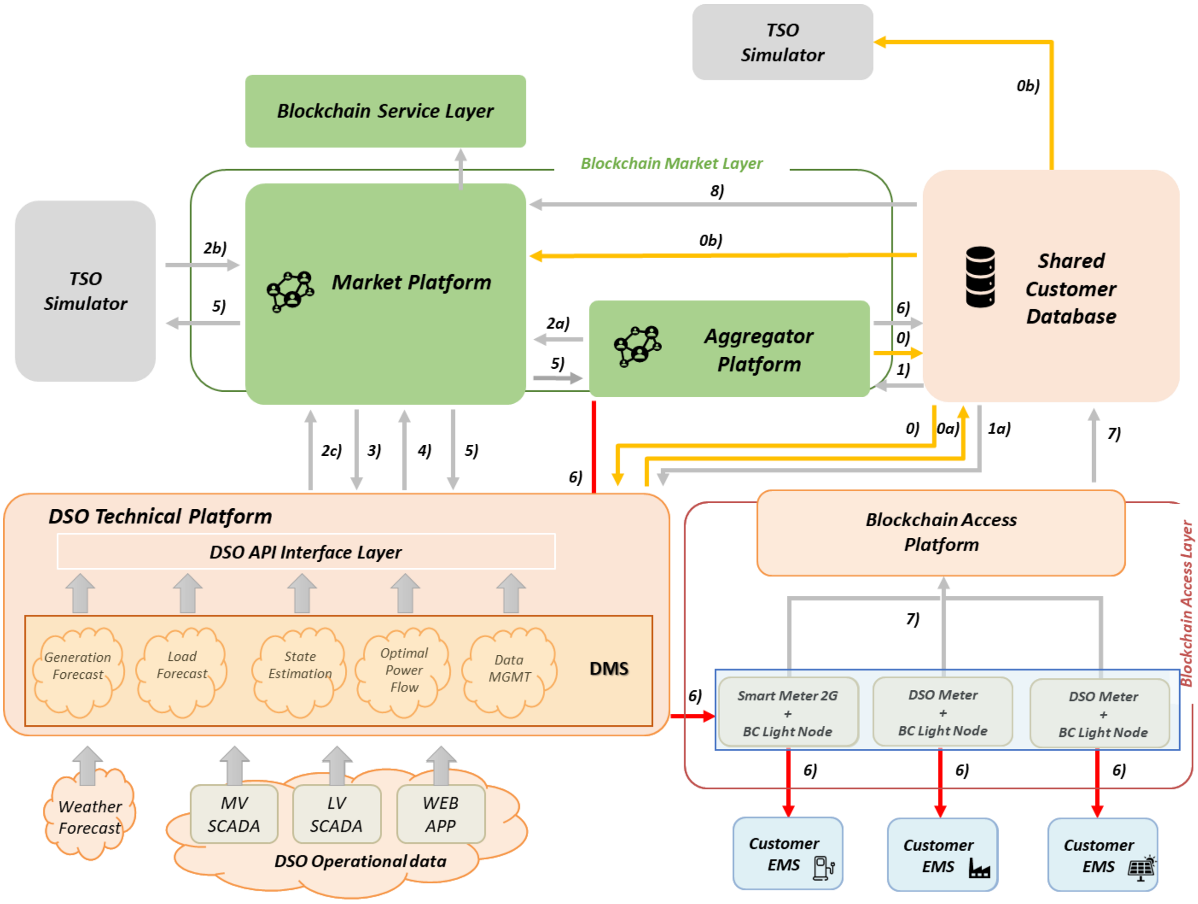

Figure 4: Italian Demo Architecture - Platforms and Processes

.

Table 6 reports the list of the data flows, the involved platform, the description and how each specific flow (including frequency) was tested in the context of scalability assessment.

Table 6: Flows descriptions

Flow	Description	Sender	Receiver	Frequency	Assessment
0	Flexible Point of Delivery (PoD) registration: Aggregator Platform (AP) sends all the data to register the resource in the Shared Customer Database (SCD)	AP	SCD/DSOTP	Once per Week (stage)	**Communication Time.** The large amount of the PoDs can impact on the communication of the data.
1	Flexible PoD data: The SCD sends, in streaming manner, to AP the quartorary measurements.	SCD	AP	Every 15 min	**Communication Time.** The large amount of the PoDs and high frequency can impact on the communication of the data.

2	New flexibility requests/offers: Market Participants sends the flexibility requests/offers to MP	DSOTP/TSO/AP	MP	Once per session (7 times per day)	**Execution Time.** The large amount of pods can impact on the calculation of the flexibility requests.
3	Market outcomes for technical validation: MP sends to DSOTP the more economics offers for the technical validation	MP	DSOTP	Once per session (7 times per day)	**Execution Time**. The large amount of data (request/offers/pods) can impact on the clearing algorithm duration. **Communication Time**. The large number of results can impact on the communication time.
4	Validated market outcomes: DSO TP sends the offers in compliance with the grid limits	DSOTP	MP	Once per session (7 times per day)	**Execution Time**. The large amount of data (request/offers/pods) can impact on the technical validation algorithm duration. **Communication Time**. The large number of results can impact on the communication time.
5	Marker results: MP sends the validated results to all the Market Participants (DSO, TSO, Aggregators)	MP	DSOTP/TSO/AP	Once per session (7 times per day)	**Execution Time**. The large amount of data (request/offers/pods) can impact on the final validation algorithm duration. **Communication Time**. The large number of results can impact on the communication time.
6	Set points: AP send the set points to be activated to DSOTP and SCD. Then the DSOTP sends the setpoints to each Light Node (LN)	AP	DSOTP/SCD/LN	Once per session (7 times per day)	**Communication Time**. The large number of setpoints can impact on the communication time.
7	Measurements and set points: LN sends all the measurements and setpoints to BAL for the certification	LN	BAL	Every 15 min	**Communication Time**. The large number of measurement and setpoints can impact

					on the communication time.
7b	Measurements and set points: BAL sends all the measurements and setpoints to SCD	BAL	SCD	Every 15 min	**Communication Time**. The large number of measurement and setpoints can impact on the communication time.
8	Measurements and set points: SCD sends all the measurements and setpoints to MP for settlement	SCD	MP	Every 15 min	**Communication Time**. The large number of measurement and setpoints can impact on the communication time.

For each of the flows, each platform collected information that would allow the system-level KPIs to be calculated in a specific format.

Table 7 shows the data format collected within each platform during each flow's execution.

Table 7: Log data format

Field	**Description**	**Example**
Flow	The monitored flow	"flow0"
Type	The type of the log	message_received -> the flow start when a message was received from another platform start_execution -> the platform started the flow execution (elaboration) end_execution -> the platform concluded the flow execution (elaboration) message_sent -> the result of the execution is sent to another platform
Platform	The name of the monitored platform	(AP, DSOTP, MP, SCD, BAL, TSO, LN)
DataTime	The date of the log	ISODate("2023-03-24T09:05:57.592Z")

Reference Id	The id that uniquely identify the flow. It depends on the specific flow	“podId”

Following the specifications of the of Italian Demo Architecture, not all platforms are involved in every flow and only for some of these flows the execution time is significant. Details on the data collected for each specific flow are detailed in Ch.4.2

4.1.2 Expected Results

The results collected during the three different stages must be compared in order to evaluate the performance of the platforms. In terms of platform computation load, the expectation is to avoid a greater-than-linear growth [22], ensuring that performance decrease proportionally to the increase of users involved in the different stages, allowing the possibility to apply standard patterns for address the scalability of the platforms such as vertical scaling (adding resources to a single machine) or horizontal scaling (adding new machines to the system)

In addition, the execution and communication timing as well as the availability must comply with the following parameters:

- Execution and communication timing: **must not exceed the timing foreseen for each process** as detailed in D3.4 [23] and reported in Figure 5.
- Availability timing: **the uptime/availability of each platform must be at least the 99.9%** during the assessment stages.

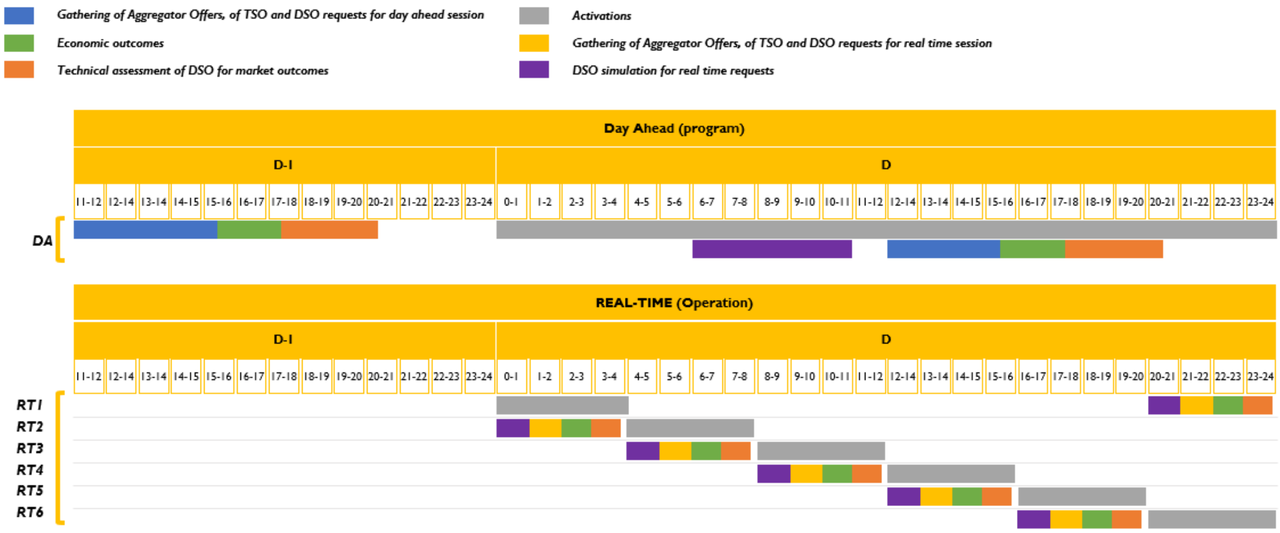

Figure 5: Italian Demo processes timing

4.2 Scalability assessment

4.2.1 Scenario

Methodology

The scalability assessment was implemented in 4 different steps:

1. **Environment Setup** – PoDs are created and each Platform is ready to run and collect KPIs.
2. **Execute the test** – 1 Week of testing the overall workflow with the current setup.
3. **Provide KPIs** – Results are collected and analysed.
4. **Increment and repeat** (3 times)

Scenario

Table 8 represents the scalability assessment scenario with dates of preparation, execution and the number of simulated DER customers.

Table 8: Scalability Assessment - Timeplan

Stage	Setup (Pods Creation)	Start Execution	End Execution	# PoDs
Week 1	03/05/2023	03/05/2023	09/05/2023	12
Week 2	12/05/2023	12/05/2023	18/05/2023	100
Week 3	19/05/023	23/05/2023	29/05/2023	1500

4.2.2 Platform Scalability Data

4.2.2.1 MP

The Platone MP was configured with:

- RAM, 16GB
- CPU, 4 Core

Table 9 and Figure 6 shows the average and max CPU usage of MP, while Table 10 and Figure 7 show the average and max RAM usage.

Table 9: MP CPU Usage

Metric	Week1	Week2	Week3
CPU Load Avg. (core) (%)	70%	71%	76%
CPU Load Max (core) (%)	81%	81%	85%

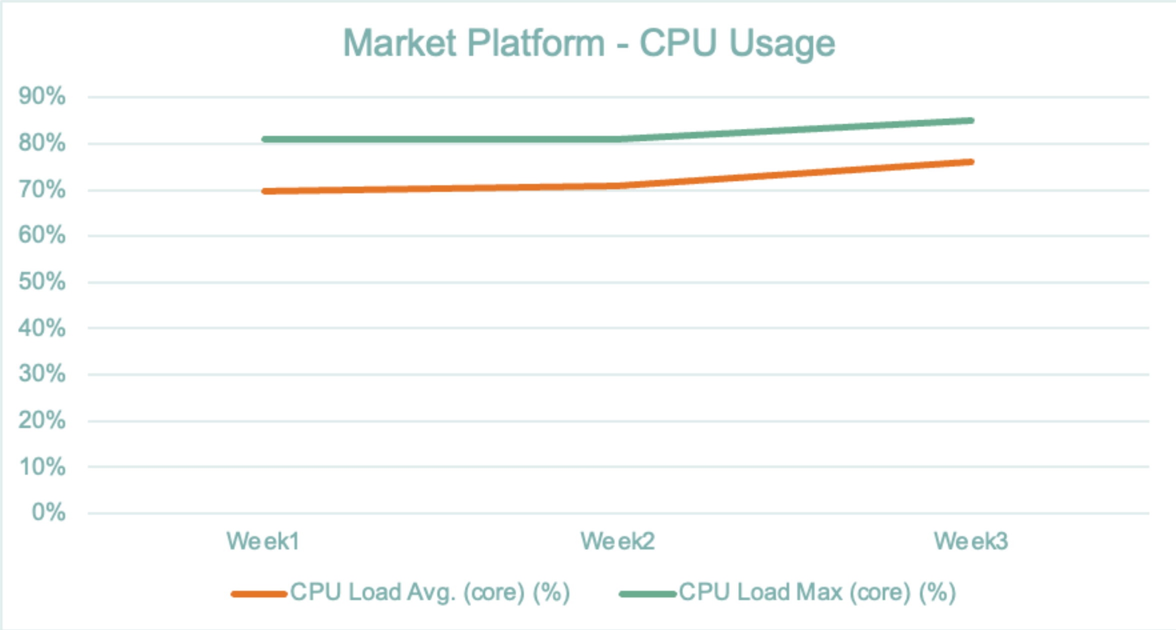

Figure 6: MP CPU Usage

Table 10: MP RAM usage

Metric	Week1	Week2	Week3
RAM Usage Avg. (MB)	5814,28	5842,85	5957,14
RAM Usage Max. (MB)	8242,85	8328,57	8571,42

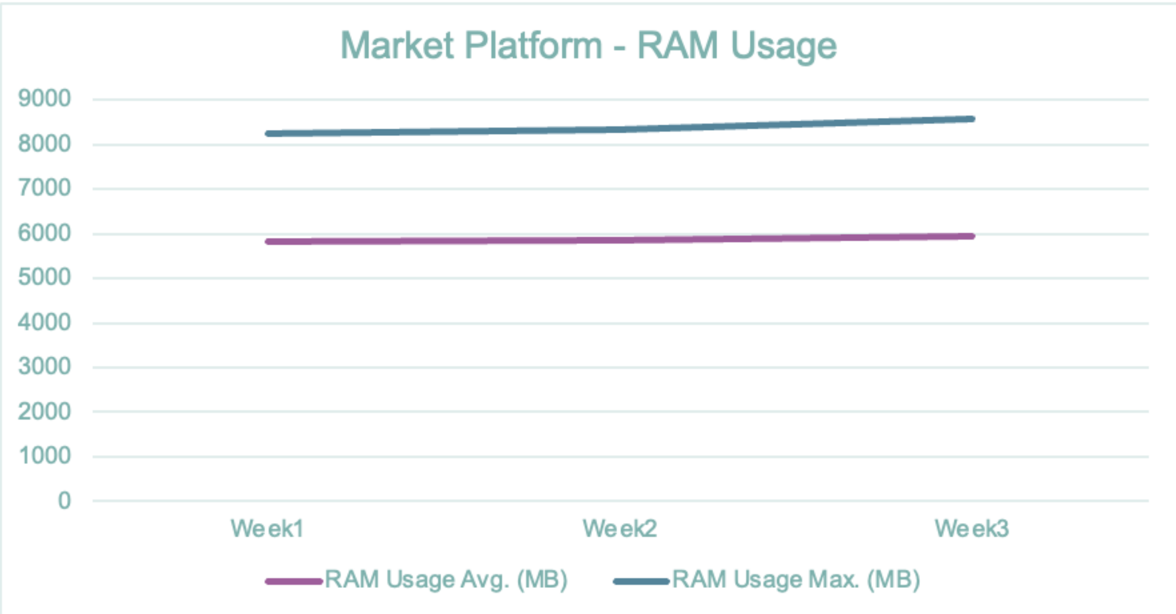

Figure 7: MP RAM Usage

4.2.2.2 Blockchain Access Layer

BAL was configured with:

- RAM, 16GB
- CPU, 4 Core

Table 11 And Figure 8 show the average and max CPU usage of BAL, while Table 12 and Figure 9 show the average and max RAM usage.

Table 11: BAL CPU usage

Metric	Week1	Week2	Week3
CPU Load Avg. (core) (%)	12%	24%	48%
CPU Load Max (core) (%)	15%	29%	61%

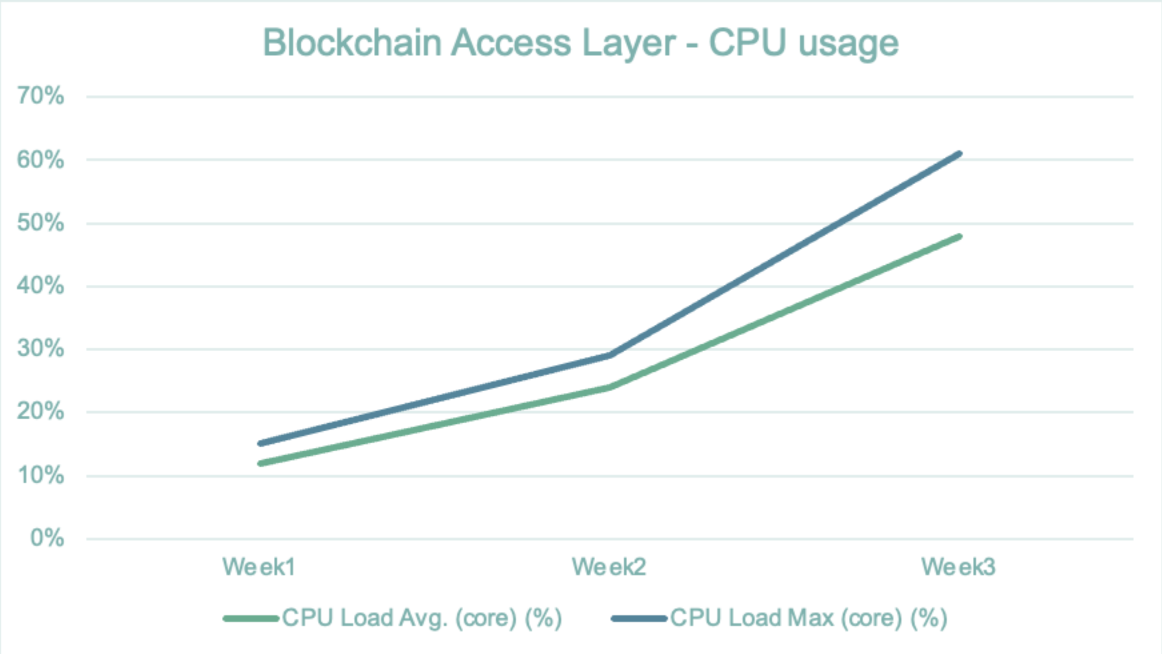

Figure 8: BAL CPU Usage

Table 12: BAL RAM usage

Metric	Week1	Week2	Week3
RAM Usage Avg. (MB)	10950	11540	13830
RAM Usage Max. (MB)	11300	13200	15200

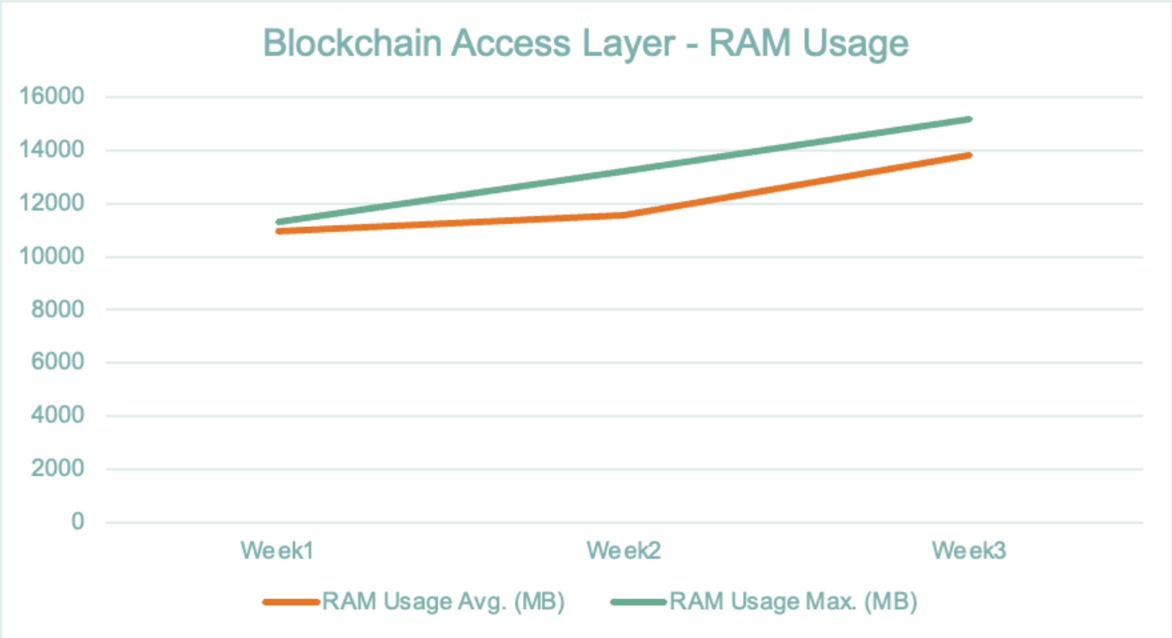

Figure 9: BAL RAM Usage

4.2.2.3 DSOTP

The DSOTP was configured with:

- RAM, 58GB
- CPU, 8 Core

In the following tables and figures are reported the extracted data.

Table 13 and Figure 10 show the average and max CPU usage of BAL, while Table 14 and Figure 11 show the average and max RAM usage.

Table 13: DSOTP CPU Usage

Metric	Week1	Week2	Week3
CPU Load Avg. (core) (%)	2%	3%	9%
CPU Load Max (core) (%)	32%	32%	39%

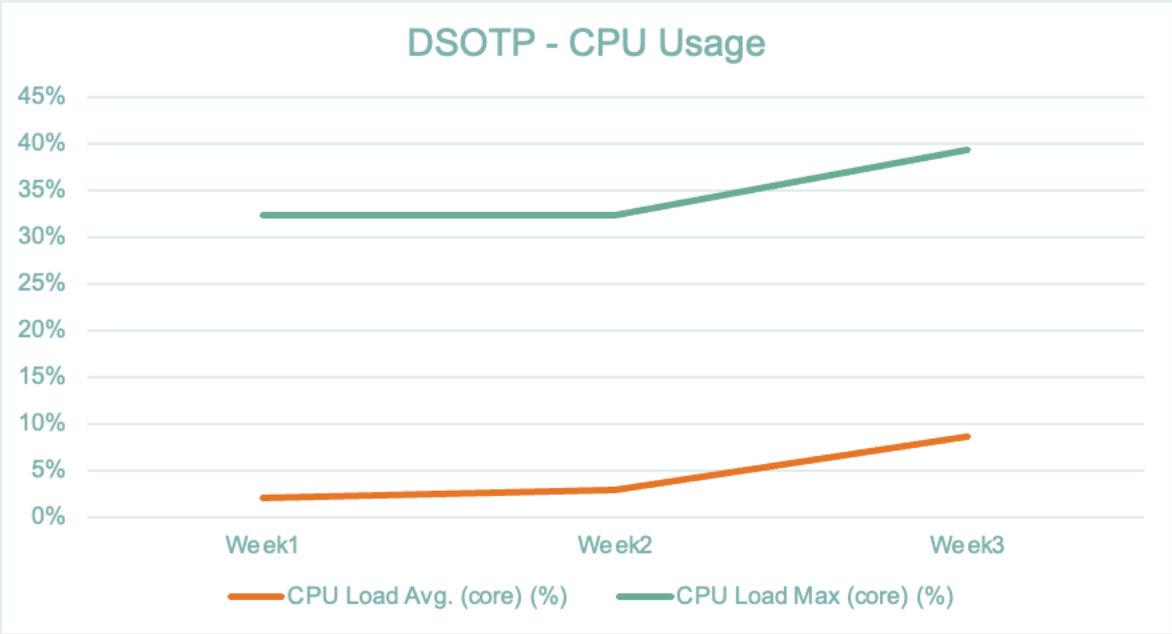

Figure 10: DSOTP CPU Usage

Table 14: DSOTP RAM Usage

Metric	Week1	Week2	Week3
RAM Usage Avg. (MB)	7750	8047	9221
RAM Usage Max. (MB)	14789	14570	21796

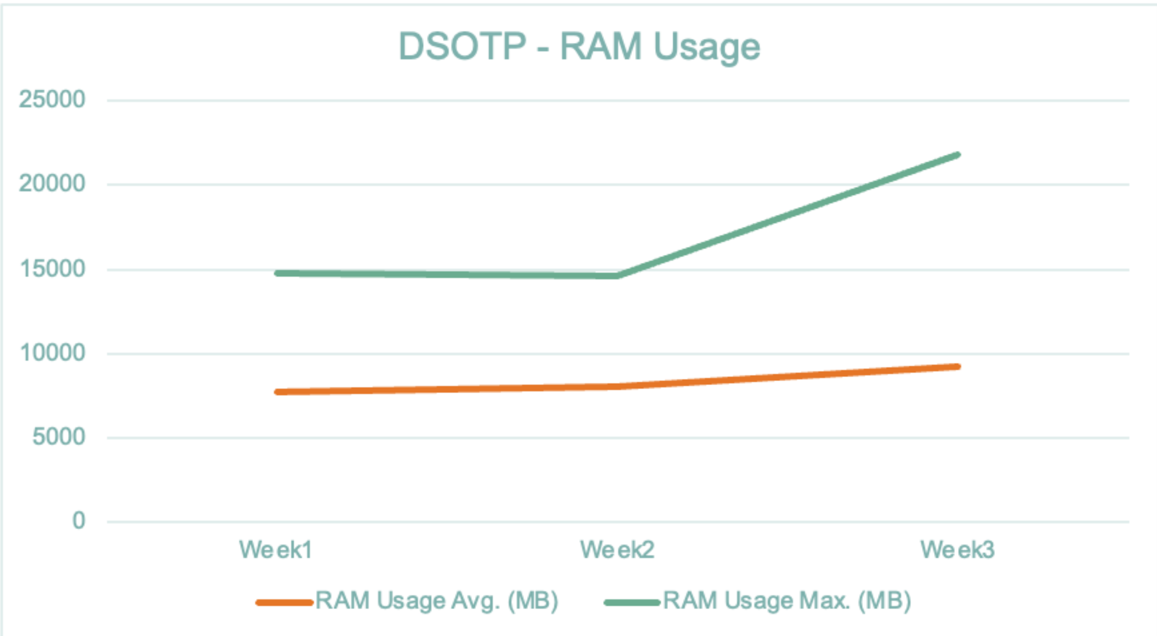

Figure 11: DSOTP RAM Usage

4.2.2.4 Shared Customer Database

The SCD was configured with:

- RAM, 16GB
- CPU, 4 Core

Figure 12 and Figure 13 report the SCD average CPU and RAM usage respectively.

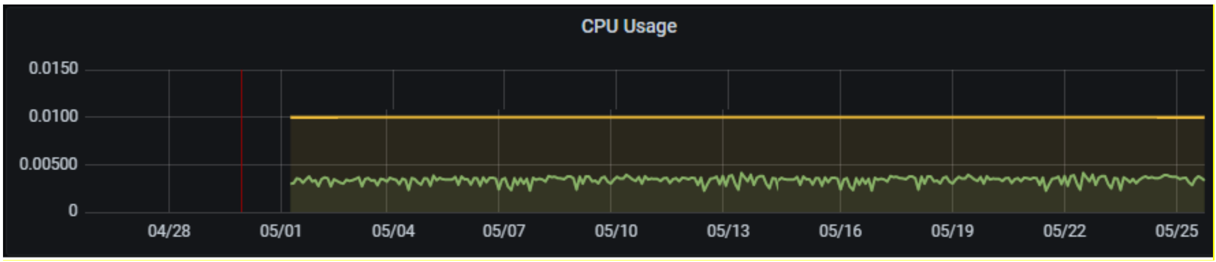

Figure 12: SCD CPU Usage

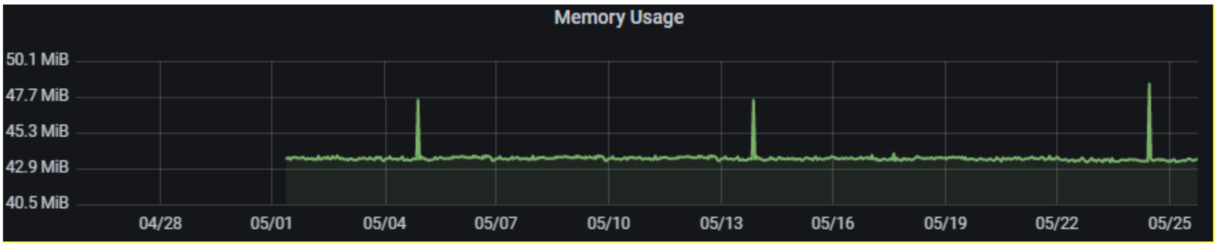

Figure 13: SCD RAM Usage

4.2.3 System Scalability Data

This subchapter reports all the data collected during each flow execution. For each flow daily aggregated data was reported and the daily average communication time was measured in milliseconds.

For the Flow 0, which is in charge to register the new resources (PoDs), only three days were monitored, while for all the other flows the three entire weeks were monitored, and the data are ported for each day of the week.

In addition, for the flows 2,3,4 and 5 also the execution time was monitored, since these flow implements complex algorithms for the market and technical validation of the results.

4.2.3.1 Flow 0

Flow 0 oversees the flexible PoDs registration. Aggregator Platform sends all the data to register the resource in the SCD. It was monitored once per stage (at the start of each stage). Since the the large amount of the PoDs can impact on the communication of the data, the communication time for the three days was monitored.

The Table 15 and the Figure 14 below report the logged data.

Table 15: Flow 0 communication time

Date	Communication Time (ms)
03/05/2023	57

12/05/2023	1733
19/05/2023	1984

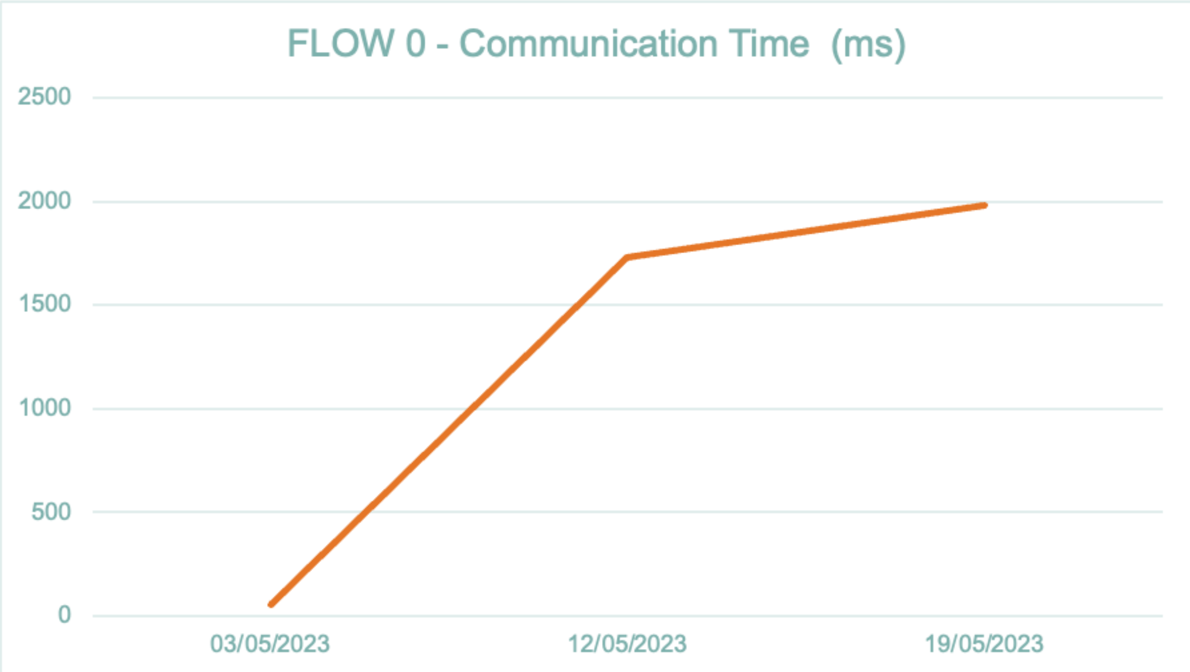

Figure 14: Flow 0 communication time

4.2.3.2 Flow 1

Within Flow 1 SCD sends, in streaming manner, to AP the measurements. It was monitored every 15 min and since the large amount of the PoDs and high frequency can impact on the communication of the data, the communication time was monitored during the entire assessment.

The Table 16 and the Figure 15 below report the logged data.

Table 16: Flow 1 communication time

Date	Average seconds
02/05/23	0,987
03/05/23	0,988
04/05/23	0,987
05/05/23	0,988
06/05/23	0,987
07/05/23	0,988

08/05/23	0,987
09/05/23	0,988
12/05/23	0,987
13/05/23	0,988
14/05/23	0,987
15/05/23	0,988
16/05/23	0,987
17/05/23	0,988
18/05/23	0,987
23/05/23	0,988
24/05/23	0,987
25/05/23	0,988
26/05/23	0,987
27/05/23	0,988
28/05/23	0,987
29/05/23	0,987

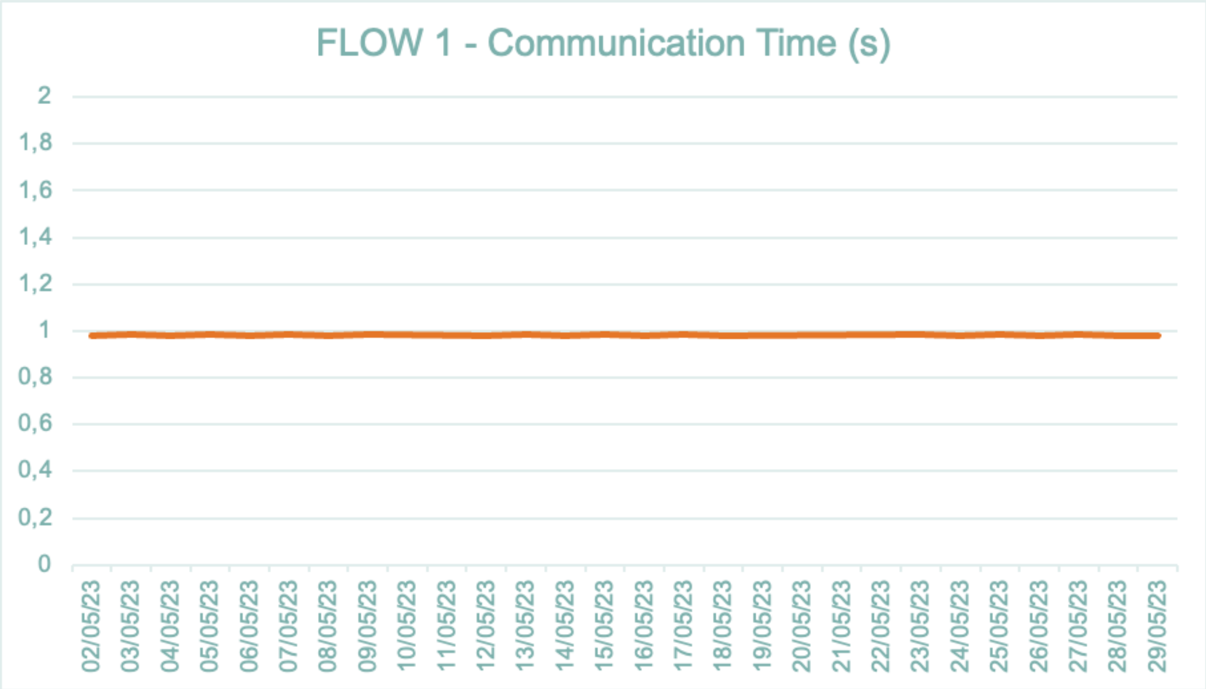

Figure 15: Flow 1 communication time

4.2.3.3 Flow 2

Within Flow 2, DSOTP, TSO simulator and Aggregator Platform send flexibility requests or offers to the MP. It was monitored at each market session (1 day-ahead and 6 real-time sessions). Since, the DSOTP implements a complex algorithm for calculating the congestion and provides the flexibility requests, the execution time of the DSOTP was reported.

The Table 17 and the Figure 16 report the logged data.

Table 17: Flow 2 communication time

Date	Average ms
03/05/23	330,56
04/05/23	344,20
05/05/23	269,80
06/05/23	440,50
07/05/23	392,00
08/05/23	418,00
09/05/23	342,60
12/05/23	493,00

13/05/23	253,40
14/05/23	217,00
15/05/23	276,71
16/05/23	496,71
17/05/23	547,14
18/05/23	699,71
23/05/23	890,57
24/05/23	1051,50
25/05/23	883,67
26/05/23	2837,50
27/05/23	2465,00
28/05/23	2701,67
29/05/23	7650,00

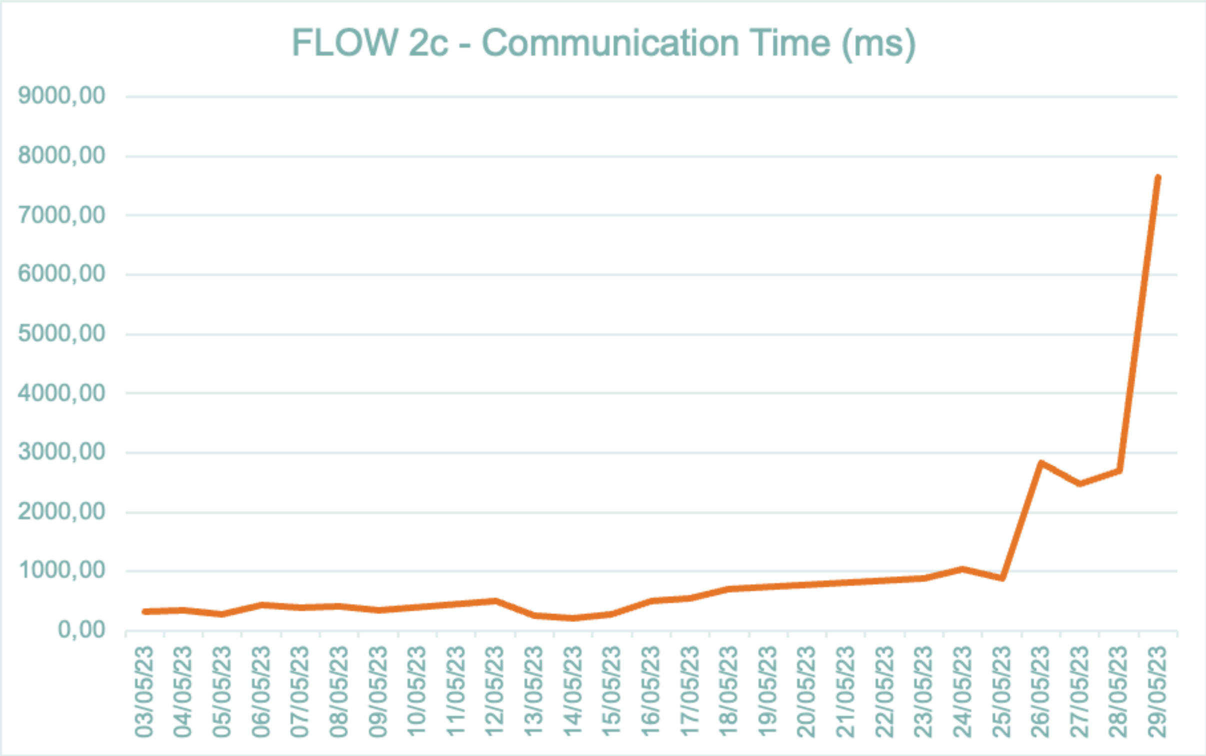

Figure 16: Flow 2 communication time

4.2.3.4 Flow3

Flow 3 aims at calculating the market clearing and provides a preliminary data to the DSOTP for the technical validation. For the flow 3 both execution time and communication time were monitored since this flow implements the market clearing algorithms and need to communicate a large amount to results to the DSOTP.

Table 18 and Figure 17 report the execution time logged data, while Table 19 and Figure 18 report the communication time logged data.

Table 18: Flow 3 execution time

Date	Average ms
03/05/23	1276,00
04/05/23	2557,20
05/05/23	988,50
06/05/23	322,00
07/05/23	1304,00
08/05/23	2543,00
09/05/23	2555,20
12/05/23	2709,40
13/05/23	2455,00
14/05/23	2221,20
15/05/23	3777,17
16/05/23	15646,43
17/05/23	15577,57
18/05/23	15810,43
23/05/23	58316,57
24/05/23	63573,71
25/05/23	79729,14
26/05/23	144668,57
27/05/23	134957,57
28/05/23	125768,57

29/05/23	102464,14

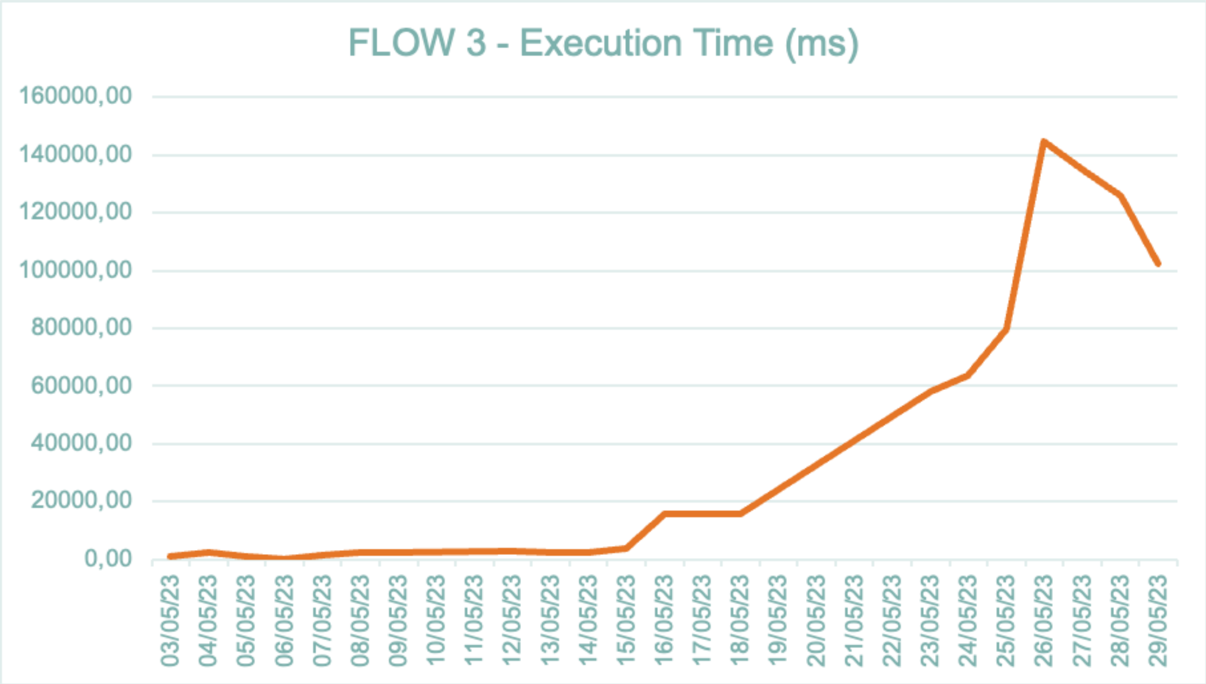

Figure 17: Flow 3 execution time

Table 19: Flow 3 communication time

Date	Average ms
03/05/23	26,50
04/05/23	31,60
05/05/23	40,60
06/05/23	40,00
07/05/23	47,50
08/05/23	65,50
09/05/23	38,80
12/05/23	42,40
13/05/23	43,00
14/05/23	45,40

15/05/23	42,50
16/05/23	48,00
17/05/23	48,33
18/05/23	56,50
23/05/23	37,86
24/05/23	33,17
25/05/23	140,00
26/05/23	101,25
27/05/23	59,20
28/05/23	89,60
29/05/23	166,67

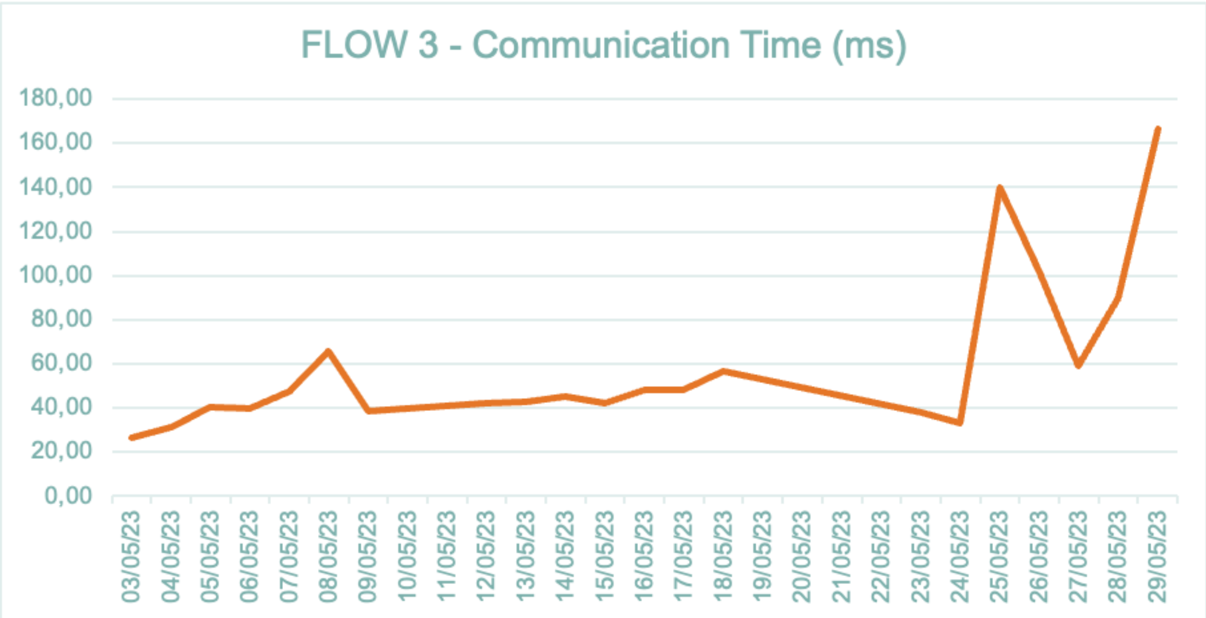

Figure 18: Flow 3 communication time

4.2.3.5 Flow 4

Flow 4 oversees the technical validation of the preliminary market data, performed by the DSOTP. The technical validation is sent to MP for final validation. For Flow 4 both execution time and communication time were monitored, since the DSOTP implements the algorithm for the technical validation and returns a large amount of data to the MP.

Table 20 and Figure 19 report the execution time logged data, while Table 21 and Figure 20 report the communication time logged data.

Table 20: Flow 4 execution time

Date	Average Ms
03/05/23	224,00
04/05/23	182,40
05/05/23	180,40
06/05/23	238,00
07/05/23	214,50
08/05/23	185,50
09/05/23	177,40
12/05/23	196,40
13/05/23	214,60
14/05/23	189,60
15/05/23	151,67
16/05/23	709,33
17/05/23	743,67
18/05/23	733,40
23/05/23	966,50
24/05/23	509,50
25/05/23	4367,50
26/05/23	4867,00
27/05/23	7228,50
28/05/23	6890,00
29/05/23	9090,00

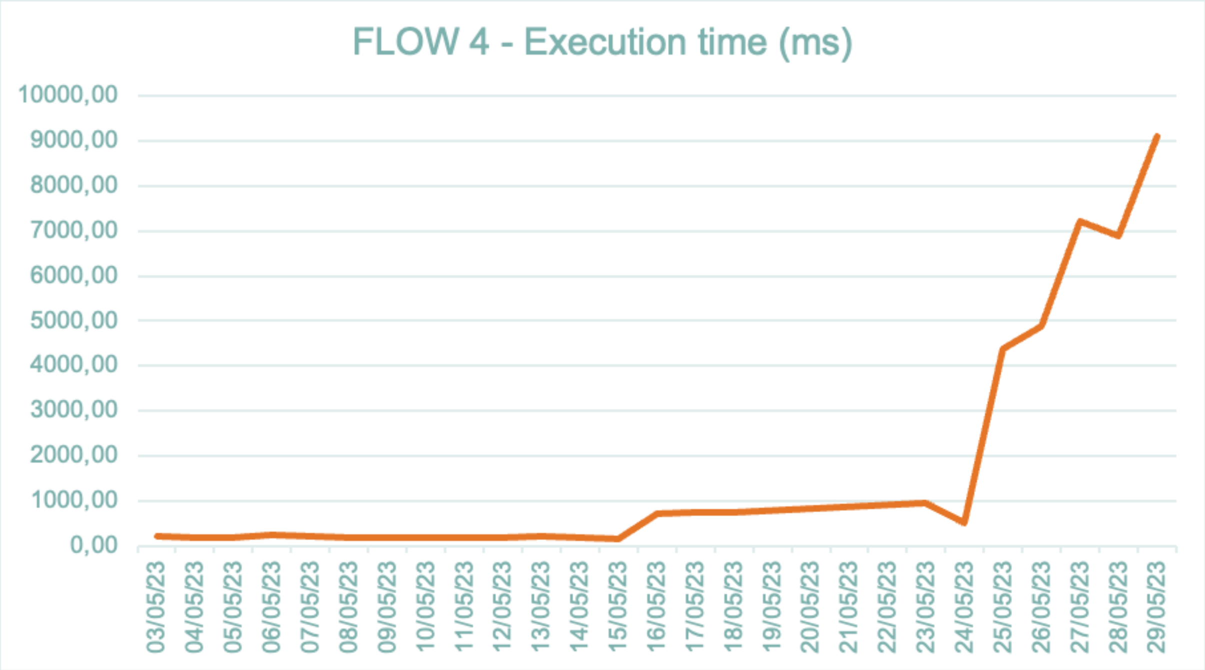

Figure 19: Flow 4 execution time

Table 21: Flow 4 communication time

Date	Average Ms
03/05/23	18,55
04/05/23	21,43
05/05/23	22,34
06/05/23	41,12
07/05/23	45,66
08/05/23	55,56
09/05/23	28,88
12/05/23	66,56
13/05/23	61,12
14/05/23	78,88
15/05/23	71,42
16/05/23	89,98
17/05/23	82,34

18/05/23	72,34
23/05/23	112,22
24/05/23	109,87
25/05/23	134,56
26/05/23	176,54
27/05/23	144,56
28/05/23	189,45
29/05/23	154,76

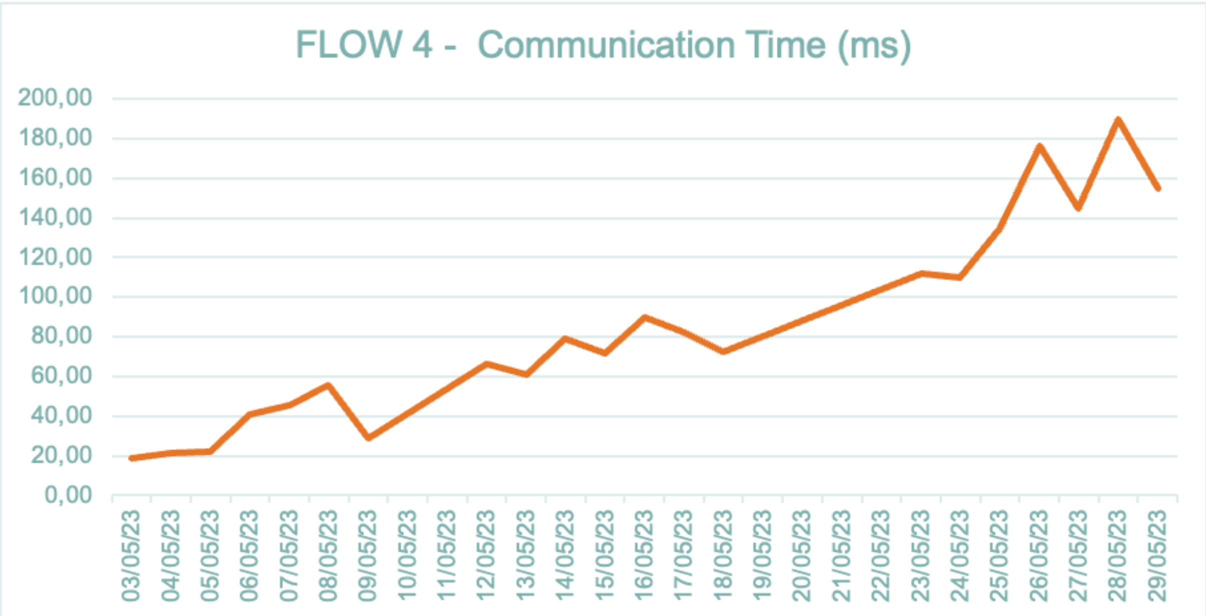

Figure 20: Flow 4 communication time

4.2.3.6 Flow 5

Within Flow 5, MP performs the final validation of the market results after the technical validation. The final validation is sent all the Market Participant using Apache Kafka. Since MP needs to aggregate market and technical data, to perform the final validation and to send a large amount of results to many actors, both the execution time and communication time were monitored.

Table 22and Figure 21 report the execution time logged data, while Table 23 and Figure 22 report the communication time logged data.

Table 22: Flow 5 execution time

Date	Average ms
03/05/23	884,00
04/05/23	908,00
05/05/23	906,00
06/05/23	756,00
07/05/23	912,00
08/05/23	1033,00
09/05/23	988,00
12/05/23	1450,00
13/05/23	1780,00
14/05/23	1944,00
15/05/23	1920,00
16/05/23	2103,00
17/05/23	2245,00
18/05/23	2278,00
23/05/23	36456,00
24/05/23	45678,00
25/05/23	44567,00
26/05/23	78543,00
27/05/23	31246,00
28/05/23	33478,00
29/05/23	43621,00

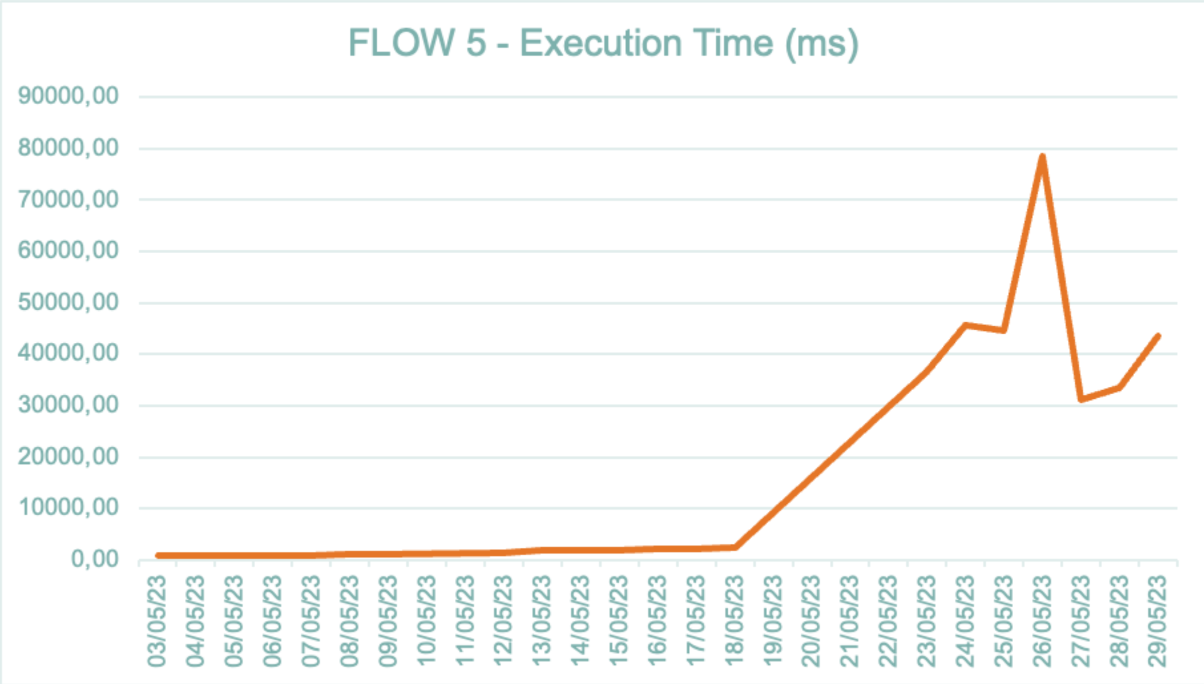

Figure 21: Flow 5 execution time

Table 23: Flow 5 communication time

Date	**Average ms**
03/05/23	23,00
04/05/23	31,00
05/05/23	33,00
06/05/23	43,00
07/05/23	34,00
08/05/23	56,00
09/05/23	41,00
12/05/23	76,00
13/05/23	88,00
14/05/23	78,00
15/05/23	90,00
16/05/23	101,00
17/05/23	87,00

18/05/23	99,00
23/05/23	122,00
24/05/23	110,00
25/05/23	145,00
26/05/23	154,00
27/05/23	143,00
28/05/23	141,00
29/05/23	135,00

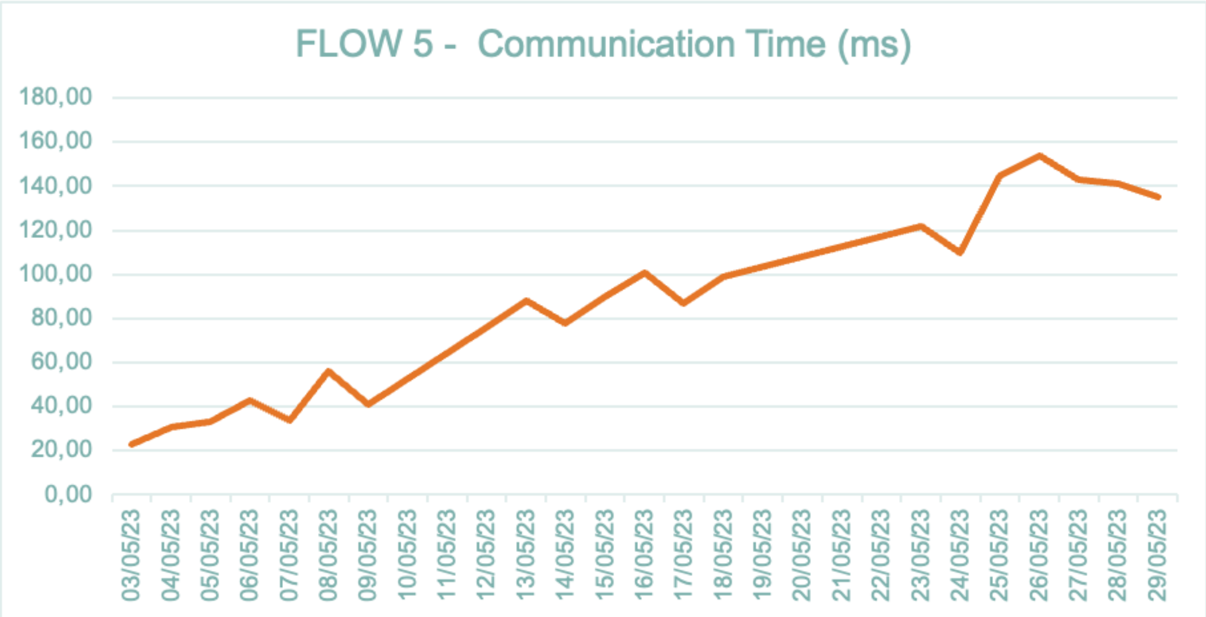

Figure 22: Flow 5 communication time

4.2.3.7 Flow 6

Within Flow 6 the setpoints of the flexible assets, extracted from the final market validation are sent from the Aggregator Platform to the DSOTP and then to each Light Node for the flexibility activation. Due to the large number of setpoints to be sent, the communication time was monitored.

Table 24 and the Figure 23 report the logged data.

Table 24: Flow 6 communication time

Date	Average Ms
03/05/23	219,67

04/05/23	189,75
05/05/23	167,00
06/05/23	263,00
07/05/23	248,33
08/05/23	234,00
09/05/23	212,50
12/05/23	197,00
13/05/23	160,75
14/05/23	348,00
15/05/23	275,40
16/05/23	139,67
17/05/23	192,00
18/05/23	94,00
23/05/23	152,00
24/05/23	64,00
25/05/23	73,00
26/05/23	196,00
27/05/23	70,00
28/05/23	256,00
29/05/23	742,00

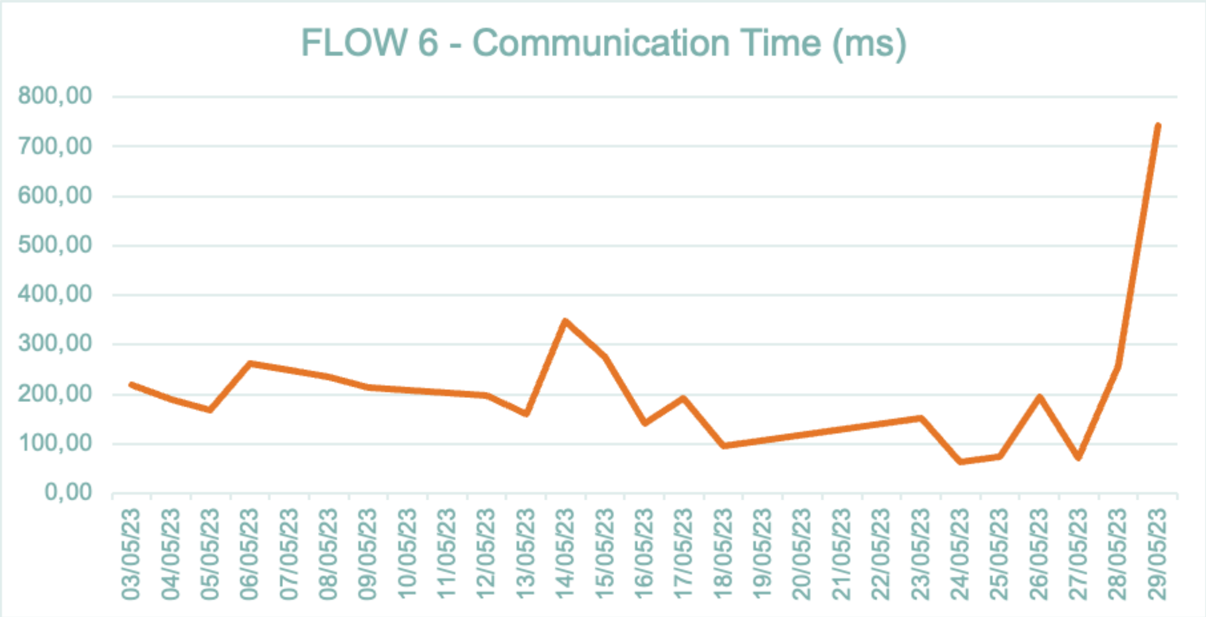

Figure 23: Flow 6 communication time

4.2.3.8 Flow 7

Within Flow 7 Light node sends, in streaming manner, to BAL the measurements including setpoints. Then (Flow 7b) the BAL certifies the measurement and sends it to the SCD.

Table 25 and Figure 24 report the communication data from LN to BAL, while Table 26 and Figure 25 report the communication time from BAL to SCD.

Table 25: Flow 7 (LN to BAL) communication time

Date	Average ms
03/05/23	6,92
04/05/23	6,87
05/05/23	6,85
06/05/23	6,96
07/05/23	6,95
08/05/23	7,05
09/05/23	6,92
12/05/23	6,64
13/05/23	6,95
14/05/23	7,09
15/05/23	7,12
16/05/23	6,85
17/05/23	7,16
18/05/23	7,38
23/05/23	7,09
24/05/23	7,06

25/05/23	7,35
26/05/23	7,22
27/05/23	7,36
28/05/23	7,07
29/05/23	6,77

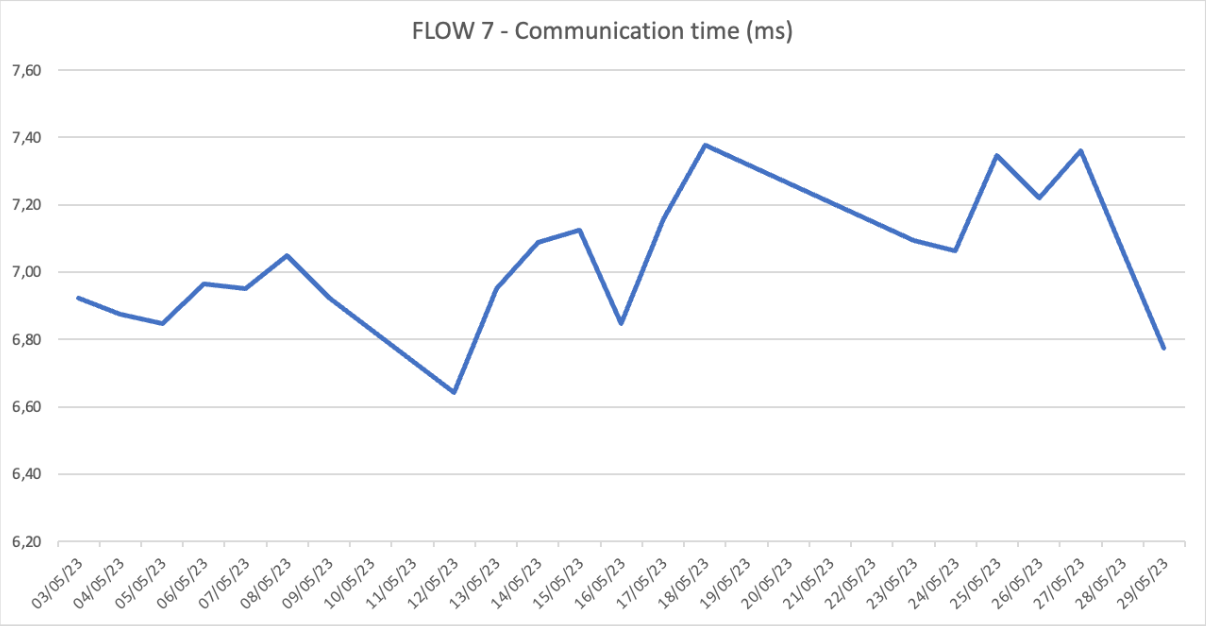

Figure 24: Flow 7 (LN to BAL) communication time

Table 26: Flow 7b (BAL to SCD) communication time

Date	Average ms
03/05/23	10,12
04/05/23	17,24
05/05/23	21,83
06/05/23	24,59
07/05/23	23,26
08/05/23	22,47
09/05/23	22,89
12/05/23	27,17
13/05/23	23,12
14/05/23	24,35
15/05/23	25,65
16/05/23	26,97
17/05/23	55,25
18/05/23	19,92
23/05/23	24,55
24/05/23	25,79
25/05/23	26,74

26/05/23	29,62
27/05/23	104,22
28/05/23	97,32
29/05/23	56,07

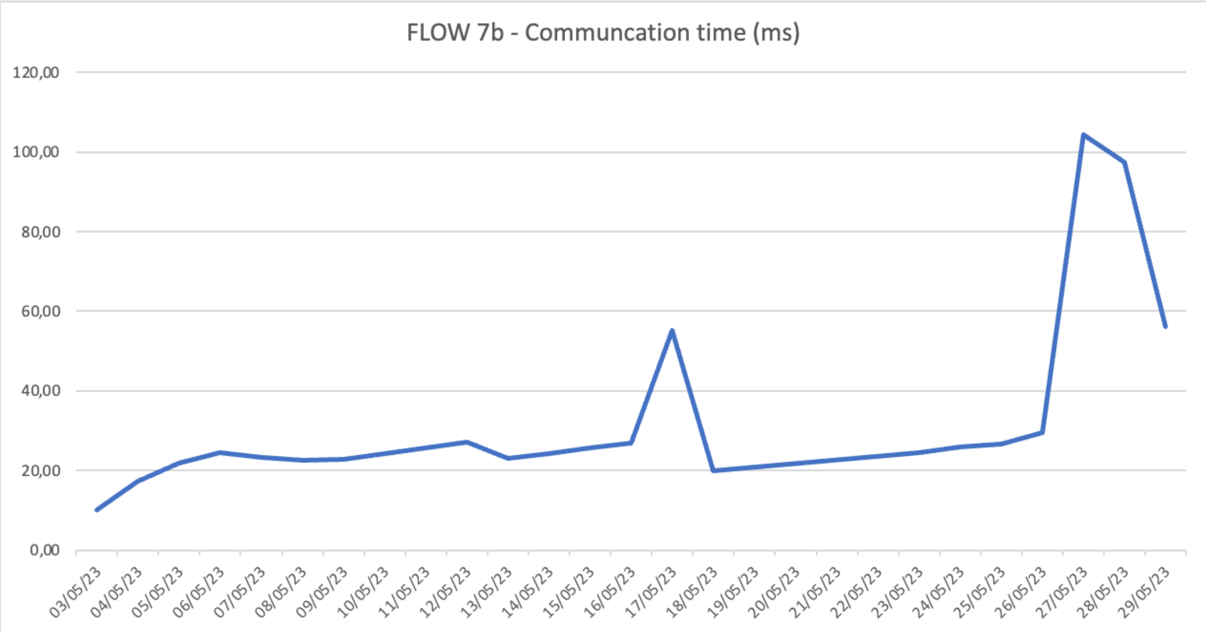

Figure 25: Flow 7b (BAL to SCD) communication time

4.2.3.9 Flow 8

Within Flow 8, SCD sends the measurements including setpoints in a streaming manner to the MP.

Table 27 and Figure 26 report the logged data.

Table 27: Flow 8 communication time

Date	Average seconds
02/05/23	0,988
03/05/23	0,987
04/05/23	0,996
05/05/23	0,987
06/05/23	0,988
07/05/23	0,988
08/05/23	0,987

09/05/23	0,988
12/05/23	0,987
13/05/23	0,996
14/05/23	0,987
15/05/23	0,988
16/05/23	0,987
17/05/23	0,988
18/05/23	0,987
23/05/23	0,988
24/05/23	0,996
25/05/23	0,988
26/05/23	0,996
27/05/23	0,988
28/05/23	0,987
29/05/23	0,987

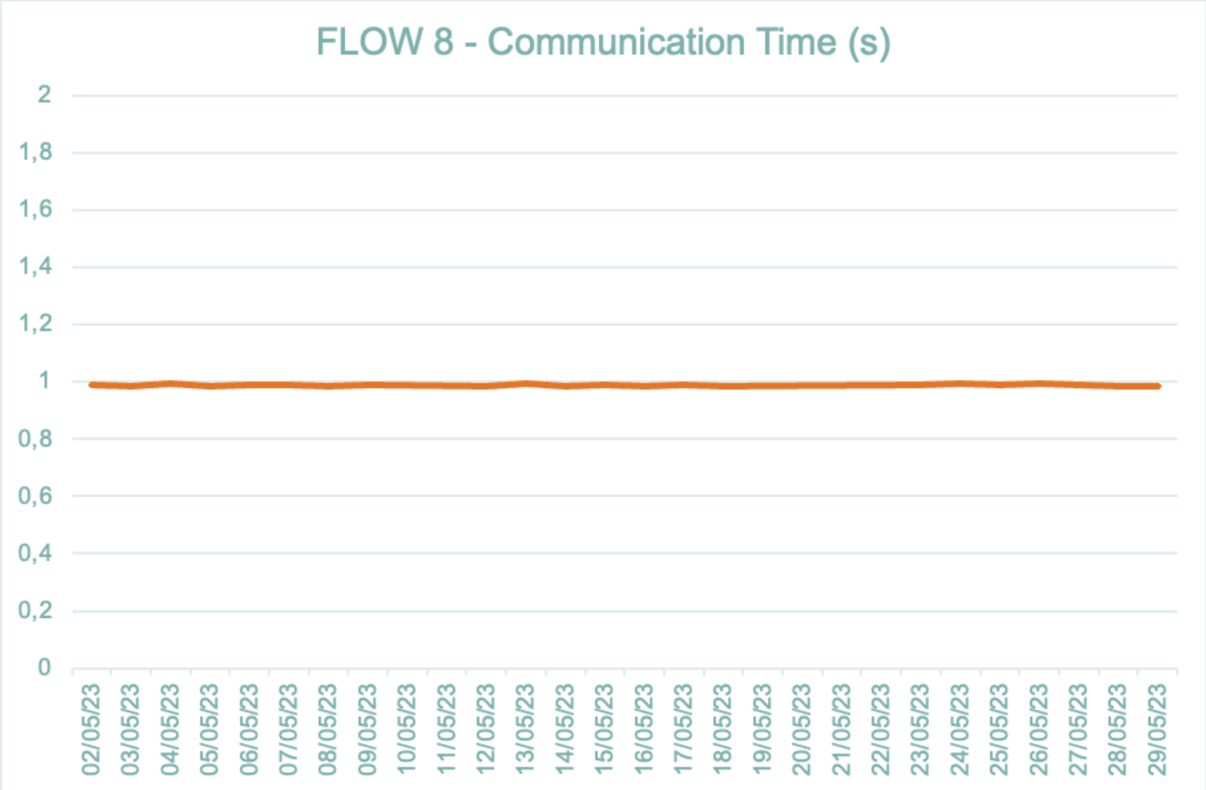

Figure 26: Flow 8 communication time

4.2.4 Evaluation of results

Platform Scalability

In terms of platform scalability, it is evident that no platform has collapsed in terms of CPU performance, with MP which reached high (around 80%, Figure 6) but constant levels of CPU usage between iterations and DSOTP and BAL growing 400% average CPU usage (Figure 10 and Figure 8) between weeks 1 and 3 while maintaining the same hardware configuration.

Also, for the usage of the RAM there were no particular sufferings, with the DSOTP reaching around 21GB of RAM used during the last iteration, with an increase of 47% (Figure 11) compared to the first one.

No particular considerations were made for SCD, which maintained very low levels of both RAM and CPU usage, being a platform devoted more to data provisioning and storage.

With the analysed results it is correct to state that a standard scalability approach (horizontal or vertical) is sufficient to ensure the correct function of the individual platforms without incurring overperformance risks

System Scalability

In terms of system scalability, it is interesting to analyse that flows 2, 3, 4 and 5 can represent a bottleneck to the scalability of the system, since these flows implement complex algorithms, that could slow down the entire system, reducing its efficiency. Below are highlighted the execution times of these flow.

- **Flow 2 maxed out at 7,5 seconds**, growing about **30x** between the third and first stages (Figure 16).
- **Flow 3 maxed out at 144 seconds**, growing about **450x** between the third and first stages (Figure 17).
- **Flow 4 maxed out at 9 seconds**, growing about **50x** between the third and first stages (Figure 19).

- **Flow 5 maxed out at 78 seconds**, growing about **100x** between the third and first stages (Figure 21).

Following the linear growth given with the increase of DERs involved in the assessment, particular attention must be paid to the execution and communication times of the Flow3, which may require further verification as they have not grown enough.

Having said that, the 144 seconds of execution and sending for Flow 3 (Market Clearing), are well within the times set for the execution of this flow, which amount to 2h in the case of day-ahead market sessions and 1h in the case of real time market sessions (see Figure 5).

Being the tested setup corresponding to about 30% of the DERs of the entire Rome area and being this a good approximation of the potential customers involved in the local flexibility market in that area, we can consider this result as sufficient to ensure the correct execution of the process.

No particular considerations were done for the other flows, since they are mostly related to communication flows, and they never exceed times in the order of the second.

In terms of availability of the Platforms, no criticalities were identified, and all the platforms were operational during the entire test execution phase.

5 Conclusion

Starting from the second integrated prototype of the POF, all the Platone Platforms were improved and enhanced in the final version. The integration of these platforms concluded to the release of the final integrated prototype of POF.

This final prototype implements all the expected functionalities and even if a final evaluation of the prototype was not initially expected, an integration and evaluation test was conducted in the Greek and German Demo field tests in order to test the missing requirements.

The iterative process used during the whole project duration allowed to adopt an incremental development and integration approach, facilitating the integration and the deployment on the different demo architectures (Italian, German and Greek) with different configuration.

In addition, during the final phase of the project, in collaboration with the Italian Demo partners, a technological scalability test was performed for the entire POF. The Italian Demo represented the best environment for testing each platform and the entire system scalability, since it was configured for using the entire POF and implemented a lot of complex business processes.

The results provided by the scalability assessment reported in this deliverable highlight how the system responded well in all the stages of the tests without criticalities. As the final stage of the tested setup corresponds to a good approximation of the potential customers involved in the local flexibility market in the area of Rome, we can consider this result as sufficient to ensure a good scalability of POF.

6 List of Tables

7 List of Figures

8 List of References

[1] European Commission, “2050 long-term strategy”, [Online]. Available: https://ec.europa.eu/clima/policies/strategies/2050_en

[2] “D2.1 Platone platform requirements and reference architecture (v1) “, [Online]. Available: https://www.platone-h2020.eu/

[3] “D2.2 Platone platform requirements and reference architecture (v2) “, [Online]. Available: https://www.platone-h2020.eu/

[4] “D1.1 General functional requirements and specifications of joint activities in the demonstrators”, [Online]. Available: https://www.platone-h2020.eu/

[5] “D2.5 Platone Market Platform (v3)”, [Online]. Available: https://www.platone-h2020.eu/

[6] “D2.8 PlatOne DSO Technical Platform (v3)”, [Online]. Available: https://www.platone-h2020.eu/

[7] “D2.13 PlatOne BlockChain Customer Access Layer (v3)”, [Online]. Available: https://www.platone-h2020.eu/

[8] “D2.14 PlatOne Integrated Framework Prototype (v1)”, [Online]. Available: https://www.platone-h2020.eu

[9] “D2.15 PlatOne Integrated Framework Prototype (v2)”, [Online]. Available: https://www.platone-h2020.eu

[10] “D2.9 Specification of the interoperability and standard communication protocols (v1)”, [Online]. Available: https://www.platone-h2020.eu/

[11] “D2.10 Specification of the interoperability and standard communication protocols (v2)”. [Online]. Available: https://www.platone-h2020.eu/

[12] Platone source code repository, [Online]. Available : https://git.rwth-aachen.de/acs/public/deliverables/platone

[13] “SOGNO H2020 Project – Service Oriented Grid for the Network Of the future”, [Online]. Available: https://www.sogno-energy.eu/About/H2020

[14] “D2.6 PlatOne DSO Technical Platform (v1)”, [Online]. Available: https://www.platone-h2020.eu/

[15] “D2.7 PlatOne DSO Technical Platform (v2)”, [Online]. Available: https://www.platone-h2020.eu/

[16] “FastAPI” [Online]. Available: https://fastapi.tiangolo.com/

[17] “OpenAPI Initiative” [Online]. Available: https://www.openapis.org

[18] “PyMFM Docker Image” [Online]. Available: https://hub.docker.com/r/sogno/pymfm

[19] “Sogno Helm-Chart reference” [Online]. Available: http://sogno.energy/helm-charts/

[20] “D1.7 Update of project KPIs”. [Online]. Available.: https://www.platone-h2020.eu/

[21] Linux CLI inotify command [Online] Available : https://linuxhint.com/linux-inotify-command/

[22] “What Does IT Scalability Actually Mean”, [Online]. Available: https://www.forbes.com/sites/adrianbridgwater/2022/07/04/what-does-it-scalability-actually-mean/?sh=4efe09973429

[23] “D3.4 Delivering of technology (v2)”, [Online]. Available: https://www.platone-h2020.eu

9 List of Abbreviations

Abbreviation	Term
AP	Aggregator Platform
API	Application Programming Interface
BAL	Blockchain Access Layer
DER	Distributed Energy Resources
DSO	Distribution System Operator
DSOTP	DSO Technical Platform
MP	Market Platform
MQTT	Message Queue Telemetry Transport
PMU	Phasor Measurement Unit
PoD	Point of Delivery
POF	Platone Open Framework
REST	REpresentational State Transfer
SCD	Shared Customer Database
SE	State Estimation
TLS	Transport Layer Security
TSO	Transmission System Operator
UI	User Interface