Platone

PLATFORM FOR OPERATION OF DISTRIBUTION NETWORKS

Platone

PLATform for Operation of distribution NEtworks

D2.2 v1.0

Platone Platform requirements and reference architecture (v2)

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:	28.02.2022
Actual delivery date:	28.02.2022
Main responsible:	Ferdinando Bosco (ENG)
Work package:	WP2 – Platform Implementation and Data Handling
Security:	P
Nature:	R
Version:	V1.0
Total number of pages:	42

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 program interfaces (APIs).

This document describes the updated version of the Platone Reference Architecture (v2) and the Platone Platforms Requirements.

The Platone Reference Architecture represents the software architecture of the Platone project, including the list of components with their interfaces, and the list of functions.

The Platone Platform Requirements includes all the functional and non-functional requirements for the design and development of the Platone Platforms.

The Platone Reference Architecture and Platone Platforms requirements are the basis for the implementation of the Platone Open Framework that will be integrated, tested and evaluated in three different demo sites: Greece, Germany and Italy. Each of these demo sites will integrate different parts of the framework.

Keyword list

Platone Reference Architecture, Platone Platforms Requirements, Platone Open Framework, Open Source, Blockchain, Energy Market

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 and within the Canadian Distributed Energy Management Initiative (DEMI).

The implementation of the Platone Open Framework and the Platone Platforms is based on an iterative incremental approach in which the feedback and the findings which have come out of the first validation phase are used to refine and improve the design and the functionalities expected.

The first version of the Platone Reference Architecture and Platone Platforms requirements defined in the first phase of the project was used to develop the first integrated prototype of the Platone Open Framework that was integrated in three real demo sites (Italy, Greece and Germany) and a laboratory test.

The results of this integration and validation phase lays the foundations for the second phase of design and implementation of the Platone project that will conclude with the release of the second version of the Platone Open Framework and Platone Platforms.

More in detail, the findings of the validation phase led to an update of the functional and non-functional requirements of the Platone Platforms and therefore to the redefinition of some functionalities and features.

The overall architecture and the implementation of the Platone Open Framework were positively evaluated and the design characteristics were identified as adequate to ensure a high interoperability, adaptability and reusability of the framework itself.

Authors and Reviewers

Main responsible		
Partner	**Name**	**E-mail**
ENG		
	Ferdinando Bosco	ferdinando.bosco@eng.it
Author(s)/contributor(s)		
Partner	**Name**	
ENG		
	Ferdinando Bosco	ferdinando.bosco@eng.it
	Vincenzo Croce	Vincenzo.croce@eng.it
	Angelo Triveri	Angelo.triveri@eng.it
RWTH		
	Jonas Baude	jonas.baude@eonerc.rwth-aachen.de
Avacon		
	Benjamin Petters	benjamin-georg.petters@avacon.de
Areti		
	Gabriele Fedele	Gabriele.Fedele@areti.it
NTUA		
	Themistoklis Xygkis	txygkis@power.ece.ntua.gr
HEDNO		
	Eleni Daridou	E.Daridou@deddie.gr
Reviewer(s)		
Partner	**Name**	
EDSO		
	Kirsten Glennung	
APIO		
	Mattia Alfieri	
Approver(s)		
Partner	**Name**	
RWTH		
	Padraic McKeever	

Table of Contents

Executive Summary ... 3
Authors and Reviewers ... 4
1 Introduction ... 6
1.1 Task 2.1 ... 7
1.2 Objectives of the Work Reported in this Deliverable ... 7
1.3 Outline of the Deliverable ... 7
1.4 How to Read this Document ... 7
2 Methodology and implementation approach ... 8
2.1 Design methodology ... 8
2.2 Iterative implementation approach ... 8
2.3 Iteration of the design ... 9
3 First Integration and validation – Findings ... 11
3.1.1 RWTH Laboratory ... 11
3.1.2 Italian Demo ... 13
3.1.3 Greek Demo ... 14
3.1.4 German Demo ... 16
4 Updates and evolution of the Platone Reference Architecture ... 18
4.1 Platone Reference Architecture v2 ... 18
4.2 Architectural components ... 19
4.3 Platone Platforms ... 21
4.3.1 Platone Market Platform ... 21
4.3.2 Platone DSO Technical Platform ... 23
4.3.3 Platone Blockchain Access Layer ... 25
5 Requirements of the Platone Platforms - Updates ... 27
6 Conclusion ... 37
7 List of Tables ... 38
8 List of Figures ... 39
9 List of References ... 40
10 List of Abbreviations ... 41
Annex A Platone Household Battery Energy System - Description ... 42

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 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, 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 more easily 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 [1],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. The Platone platform will be tested in three European trials (Greece, Germany and Italy) and within the Distributed Energy Management Initiative (DEMI) in Canada. 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".

The Platone solution consists of a two-layer architecture named **Platone Open Framework**. The Platone Open Framework includes the following components:

Blockchain Service Layer: this layer enables the deployment of different blockchain-based components, providing a blockchain infrastructure and Smart Contracts services. In the context of Platone, the Platone Market platform is an example of blockchain-based platform deployed on it.

Platone Market Platform (MP): allows the support of wide geographical area flexibility requests from TSOs and local flexibility requests from DSOs. These are matched with offers coming from aggregators, resolving conflicts according to pre-defined rules of dispatching priorities. All the market operations are registered and certified within the blockchain service layer, ensuring a transparency, security and trustworthiness among all the market participants.

Blockchain Access Layer (BAL): this layer adds a further level of security and trustworthiness to the framework. It is an extension of the physical infrastructure and performs multiple tasks, among which are data certification and automated flexibility execution through Smart Contracts. It includes the Blockchain Access Platform and the Shared Customer Database.

Platone Blockchain Access Platform (BAP): implements all the functionalities offered by the blockchain technology through smart contracts and provides an interface for the integration of the data coming from the physical infrastructure.

Platone Shared Customer Database (SCD): contains all the measurements, set points and other needed data collected from customer physical infrastructure. It allows the other components of the Platone Open Framework to access data in an easy way and without compromising security and privacy.

Platone DSO Technical Platform (DSOTP): allows DSOs to manage their grid in a secure, efficient and stable manner. It is based on an open-source extensible microservices platform and allows to deploy, as Docker containers, specific services for the DSOs and execute them on Kubernetes. The Data Bus layer, included on the DSO Technical Platform, allows for the integration of other components of the Platone Open Framework as well as of external components (e.g. DSO Management System) with a direct connection to the classical supervisory control and data acquisition (SCADA) system adopted by the DSO and served by standard communication protocols.

The first design and implementation phase of the Platone Open Framework has been conducted starting from use cases, scenarios and requirements defined together with demo participants.

In this second design phase, the results and findings of the integration and first validation phase are taken into account in order to improve the Platone Reference Architecture and Platone Platforms' Requirements.

1.1 Task 2.1

Starting from overall requirements and use cases provided by T1.1, Task 2.1 focuses on the definition of the Platone Reference Architecture and platforms requirements. The Task T2.1 foresees two main deliverables. The D2.1 [2] that includes the first version of the Platone Reference Architecture and Platone Platforms Requirements (delivered at M12, August 2020) and the D2.2 (this document) that includes the updated version of the Platone Reference Architecture and Platone Platform Requirements based on the results and feedback of the first integration and validation phase.

The Platone Reference Architecture is the base for the implementation of the Platone Open Framework including the development of the components (T2.2, T2.3 and T2.5), the definition of the interoperability mechanisms and the communication protocols (T2.4) and the delivery of the Platone integrated framework prototype (T2.6). Functional requirements are represented as a list of functional properties that need to be implemented and supported by the Platone Open Framework. Non-functional requirements will concern security, performance, interoperability and scalability aspects.

1.2 Objectives of the Work Reported in this Deliverable

This deliverable provides the second version of the Platone Reference Architecture together with the updated functional and non-functional requirements for each of the Platone Platforms (Market Platform, DSO Technical Platform and Blockchain Access Layer).

The updated version of the reference architecture, the functional and non-functional requirements is the starting point for the design, development and release of the second prototype of the Platone Platforms and the overall Platone Open Framework.

1.3 Outline of the Deliverable

Chapter 2 describes the iterative methodology applied for the design of Platone architecture and the collection of the requirements for the evolution of the Platone Reference Architecture and the updates of the Platone Platforms Requirements.

Chapter 3 provides an overview of the results of the integration and validation phase and describes all the relevant findings for the evolution of the Platone Reference Architecture and the updates of Platone Platform Requirements.

Chapter 4 describes the second version of the Platone Reference Architecture and the updated list of the architectural components, including a detailed description of the Platone Platforms.

Chapter 5 reports the updated list of the Platone platform requirements (functional and non-functional).

Finally, Chapter 6 presents the conclusions of the work performed.

1.4 How to Read this Document

This document reports the updates on the Platone Reference Architecture with focus on Platone Platforms developed within WP2. Other external systems, implemented within other WPs (WP3, WP4 and WP5) are listed and briefly described. A greater level of detail regarding these systems will be included in the deliverables of the respective WPs.

Use cases and scenarios, used for defining the functional and non-functional requirements, are publicly available in Smart Grid Use Case Repository [3].

For the first version of the Platone Reference Architecture and Platone Platforms Requirements please refer to D2.1 [2].

2 Methodology and implementation approach

2.1 Design methodology

During the first phase of the project, the main goal of the task T2.1 was to design and develop the Platone Reference Architecture, as well as to define the functional and non-functional requirements for the Platone Platforms.

D2.1 described the methodology that led to the results of the first version of the Platone Reference Architecture, the Platone Platforms, and the overall Platone Open Framework. This methodology is described below and represented in the Figure 1.

1. Use cases definition and General requirements collection (conducted in WP1)
2. Concept architecture definition and collection of the architectural and hardware components description and technical specifications
3. Design and representation of the architecture in a standard model
4. Definition of functional and non-functional requirements for the architectural components

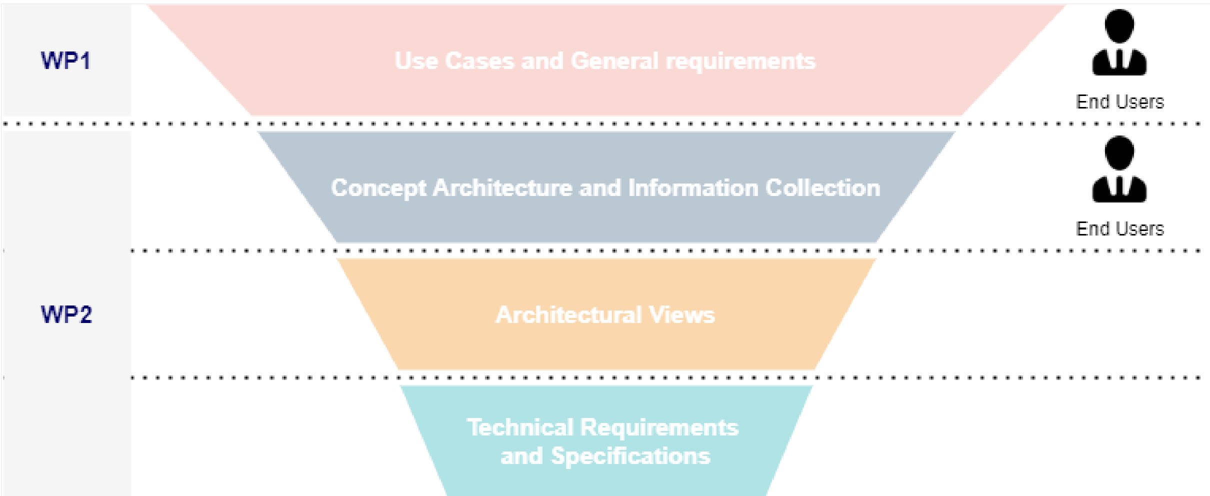

Figure 1: Methodology for the design of the Platone Reference Architecture and definition of the Platone Platforms requirements

2.2 Iterative implementation approach

The Platone project applies an iterative incremental approach with a SCRUM-like [4] software engineering methodology.

From a holistic point of view, the project is structured according two macro streams (see Figure 2): the first, the development stream, collects all the technologies, integrates, and tests them for the delivery of the three versions of the Platone Open Framework. The second stream is the validation stream. It is devoted to the execution of the framework evaluation; it exploits the first and second framework versions, and involves users in the different trials. Moreover, the validation stream provides feedback to the development stream, leading to improvements from both the functional and non-functional point of view.

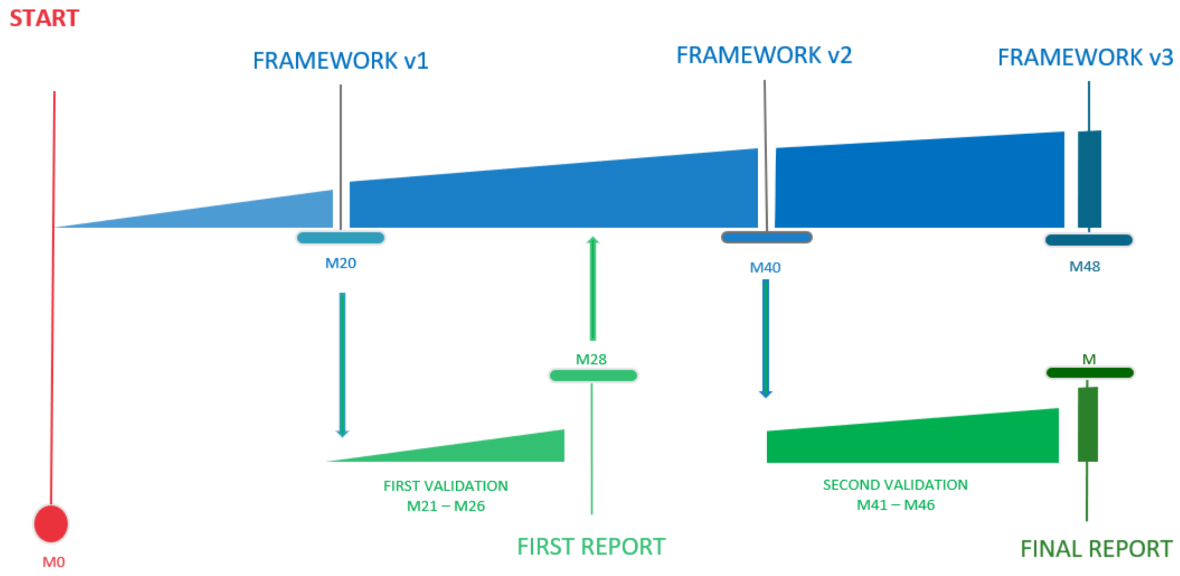

Figure 2: Platone Implementation approach

From a methodological point of view, the implementation approach is based on three main phases, in which each phase feeds the next. This means that an initial, intermediate and final version of the project framework and related platforms will be derived each time building on the previous version.

In total three phases are foreseen each will cover a full development cycle:

- Phase 1 (M1-M20), the phase has already been concluded. It included the definition of the scenarios, use cases, the elicitation of user requirements and system requirements, as well as the dynamic assessment modelling, definition of system architecture and data models and finally the release of the integrated prototype.
- Phase 2 (M21-M40), the ongoing phase, includes the first validation of the Platone Open Framework and the Platone Platforms integrated in the three demo trials, as well as the testing in the RWTH lab. After this validation phase, the use cases, the architecture and both user and technical requirements may be refined depending on the outcomes. This will lead to a new version of the platforms which is expected to be integrated in an intermediate version of the framework which is functionally complete. The framework will be evaluated at both usability and performance level. The testing will be increased, including a user evaluation with widest group of users.
- Phase 3 (M41-M48), the final phase, will foresee the evaluation of the intermediate version leading to the refinement of component and system level technology consolidation and enhancing the final prototype.

2.3 Iteration of the design

As described in the previous sub-chapter, the second phase of the project foresees an evolution of the prototype into its intermediate version based on the results of the first phase and the validation of the integration of the first prototype of the Platone Open Framework in the various trials. In addition, if necessary, the phase will also bring a refinement, of both the Platone Reference Architecture and the functional and non-functional requirements of the Platone Platforms.

Figure 3 below, shows how the methodology implemented in the first phase of the design and described in Chapter 2.1, can be applied iteratively in the second phase of refinement of the architecture and requirements, taking as input also the results and feedback of the validation in the various field trials.

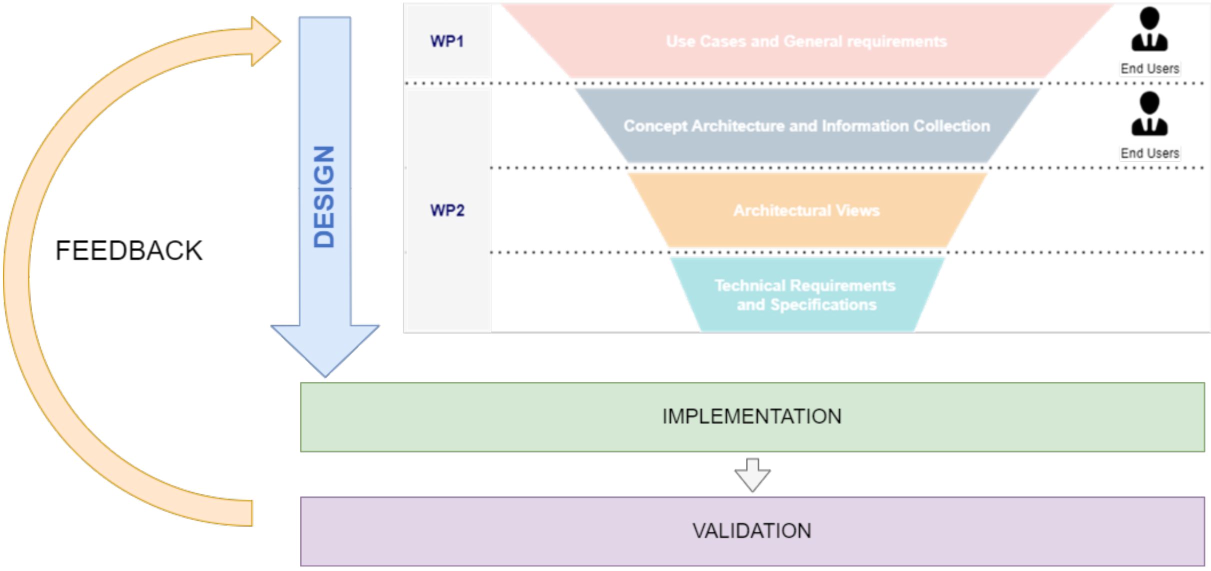

Figure 3: Design methodology applied to an iterative approach

The outcome of this activity led to a definition of a Platone Reference Architecture v2 (Chapter 4) and an update of the functional and non-functional requirements which is reported in Chapter 5 and which will be implemented in the intermediate version (v2) of the Platone Platforms (M38, October 2022) and Platone Open Framework (M40, December 2022).

This second version of the prototypes will be fully functional and will therefore include all the functionalities reported in this document. The final version of the prototypes[1] will be considered as the consolidated version based on the feedback received in the second validation phase.

[1] Expected at the end of the project, M48 (August 2023).

3 First Integration and validation – Findings

This chapter will describe the findings first from the testing in the laboratory of RWTH Aachen and then from the three demo trials in Italy, Greece and Germany.

3.1.1 RWTH Laboratory

After the release of the Platone Open Framework, a first integration test was conducted in the experimental RWTH laboratory. The scope of the test was to implement the standard configuration that would also be integrated into the Greek and German demos.

This integration, initially described in D2.14 [5], included two of the core components of the Platone Open Framework, the **Blockchain Access Layer (including the SCD) and the DSOTP**, as well as one of the devices to be used in both the demos and developed within the Platone project, **the Phasor Measurement Unit (PMU)**.

The RWTH Lab has made available a PMU for the simulation of measurements, as well as a cloud environment for the deployment and integration of the DSOTP. The BAL was instead deployed in the ENG cloud infrastructure , as described in D2.11 [6].The main goal of this integration test was to demonstrate the interoperability between the two components of the Platone Open Framework, to be reused in the Greek and German demos. In addition, the test was used to investigate the possibility of collecting measurements from this type of devices (the PMU) through an architectural component that includes a technological layer that exploits the blockchain infrastructure for the certification of the measurements themselves.

The findings from this integration led to the decision to separate the two data acquisition and certification processes. In fact, due to the huge amount of measurement data possibly received from the PMU (one per second in the experimentation lab but ideally up to 50 or 100 per second) and the latency of the blockchain infrastructure for the certification of the measurements, this choose allowed a high level of efficiency and scalability of the data acquisition and historicization process to be maintained while ensuring the certification of the data within the blockchain, as a temporal aggregation of the data itself.

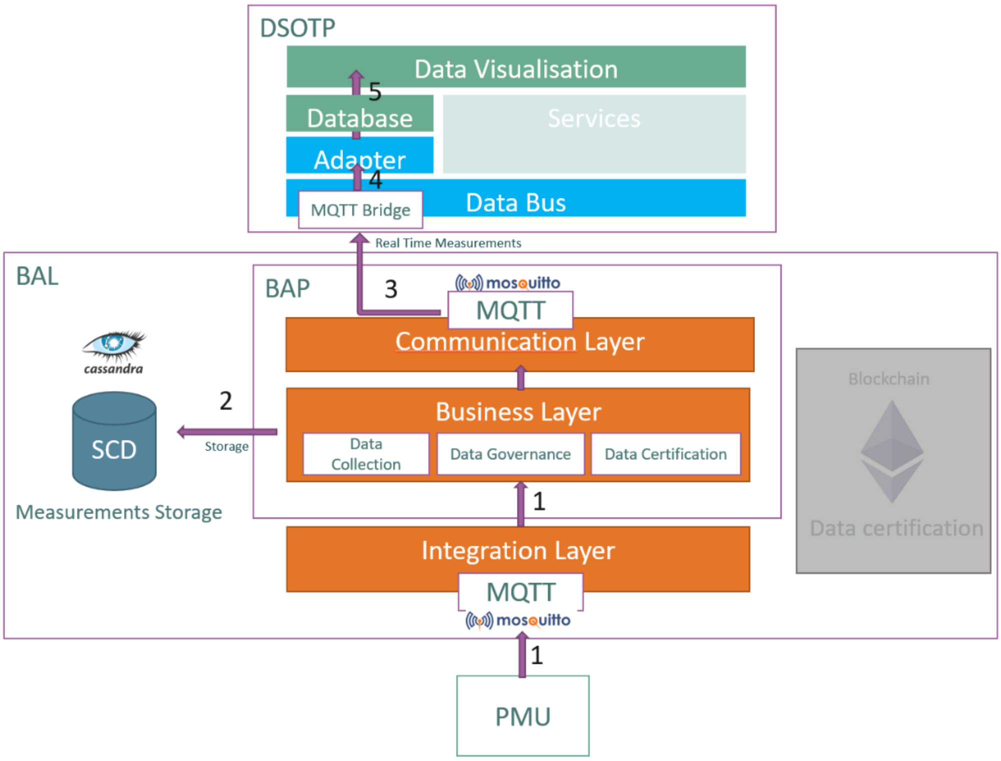

Figure 4: BAL and DSOTP integration test - Process 1

During the tests measurement data from the **PMU** was collected **every second** and certified into the blockchain infrastructure every 3 minutes (aggregation of 180 measurements).

The two main processes are described in detail below and represented in Figure 4 and Figure 5 .

Process 1 – PMU Data collection, storage and provisioning

1. MQTT Broker of the BAL receives data from PMU, in authenticated way and under TLS connection every seconds. Every device can only write in its dedicated topic;

2. BAL stores data into SCD

3. BAL provides data to DSOTP via MQTT bridging. DSOTP has its credentials for authenticating on MQTT. It has to request the access to the specific dataset.

4. DSOTP makes selected topics available on its internal message broker for processing by services;

5. Within DSOTP, PMU data are stored in a timeseries database and visualized in customizable dashboards;

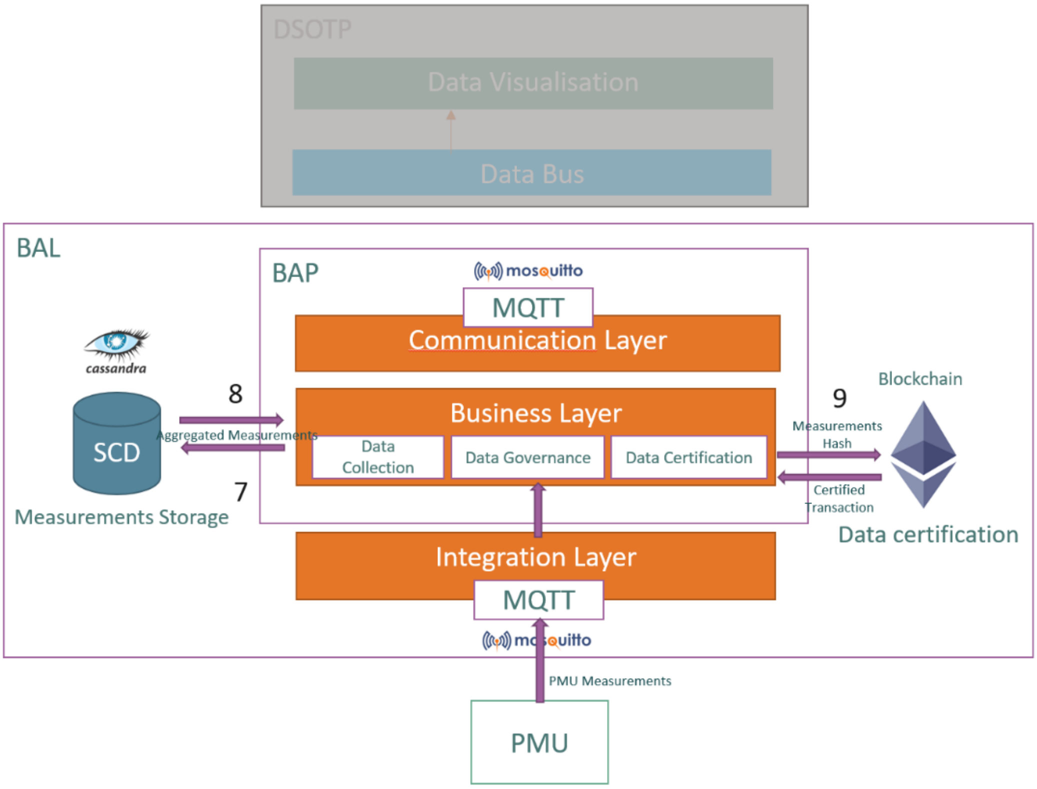

Figure 5: BAL and DSOTP integration test - Process 2

Process 2 – Periodic PMU Data certification

7. BAP periodically request aggregated Data to SCD;

8. SCD send aggregated data to BAP for data certification;

9. BAP certifies the data, creating the hash of the aggregated data and registering it on the blockchain infrastructure;

3.1.2 Italian Demo

The main purposes of the Italian Use Cases are to prevent congestion issues and avoid voltage violations in transmission and distribution systems by exploiting flexibility resources, considering all the phases concerned (procurement, activation, and settlement) in the day-ahead and real time flexibility market.

The Italian Demo Architecture implemented and integrated the entire Platone Open Framework, following the Platone Reference Architecture and requirements, developing its own Blockchain Access Layer, DSO Technical Platform and Shared Customer Database (see Table 1) as well as integrating the Platone Market Platform developed within WP2.

In addition, also the Aggregator Platform was already implemented and integrated, while the Aggregator-Customer App development is ongoing and will be integrated in the second version.

The Italian Demo Architecture, shown in Figure 6 represents the architecture that was deployed in the Italian demo site and was designed following the Use Cases UC-IT-1 regarding congestion management and UC-IT-2 concerning voltage control, described in D1.1 [7], and the related scenarios.

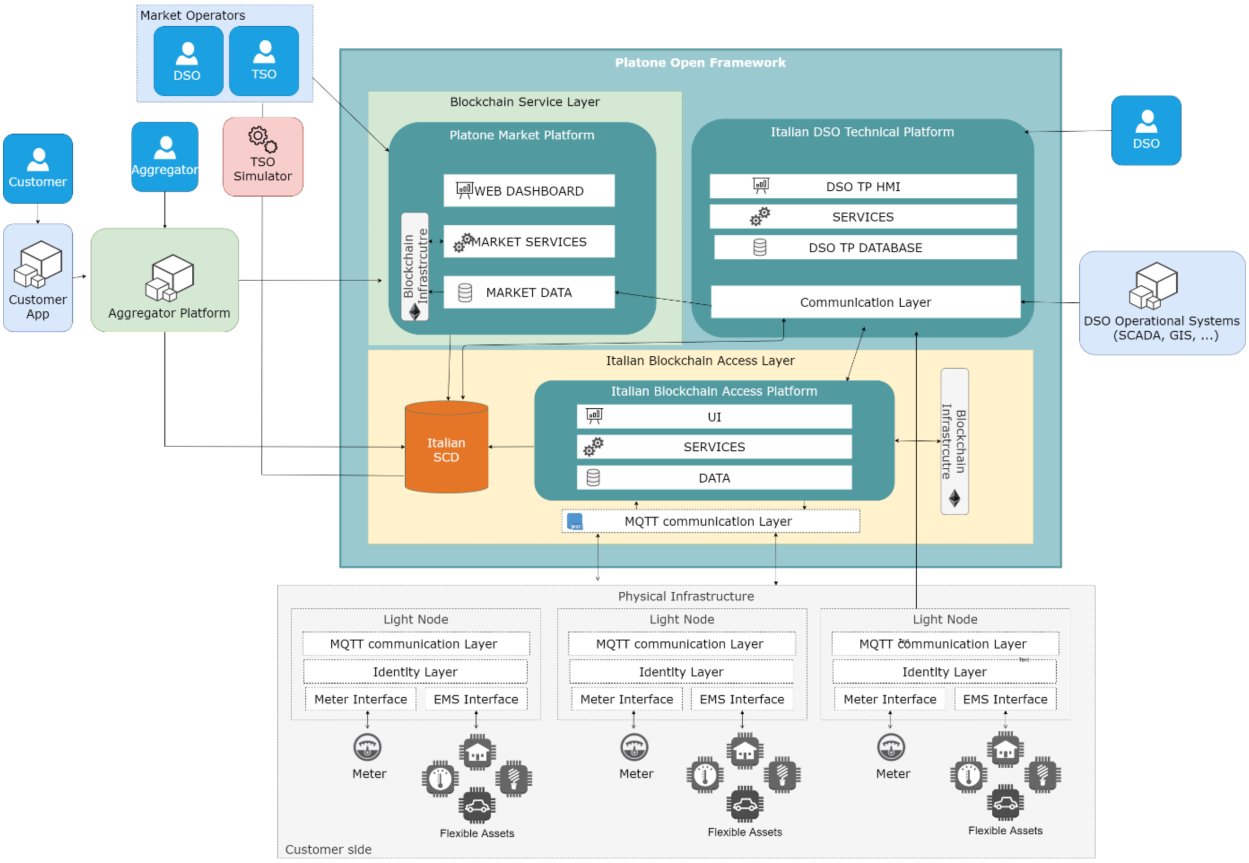

Figure 6: Italian demo architecture

The integration on the ground of the Italian demo was carried out in different distributed cloud environments and this prompted us to focus on the authentication and security systems of the different platforms.

In fact, while from a functional point of view, it was already clear what had to be tested and evaluated, the integration made it possible to define some additional non-functional requirements (among other things already implemented and tested during the first validation phase) essential for the success of the demo.

In the context of the Italian trial day-ahead market session were simulated. The UC-IT-1 was tested and implemented using the Platone Open Framework that enabled DSOs, using the Italian DSO Technical

Platform, and TSOs, using the TSO simulator, to solve network congestions exploiting the flexibility market and the consumer flexibility offers published on the market by the Aggregator.

In addition, all the measurements collected from the network through the Light Node, the innovative device based on blockchain technologies, which allows to receive and provide data and certify them via blockchain, has been integrated with the SCD. This makes the reading measures available to all, in safe and certified way and at the same time receive from the DSOTP the setpoints for the activation of the flexibility.

As mentioned before, some non-functional requirements have been envisaged and implemented (see Table 2) to allow the distributed deployment of the entire Platone architecture which involved various cloud environments such as those of Engineering, Siemens, and Areti.

More in detail, all the platforms of the Italian Demo architecture implemented a set of REST APIs that leverage on an authentication mechanism based on Oauth2.0 [8] over a HTTPS connection, and all the connections using Message Broker are secured through a mutual authentication based on TLS [9].

The second implementation in the Italian demo will focus more on the second use case (UC-IT-2). This use case addresses the management of voltage violations within the real-time market in parallel with the day-ahead market and will focus on the implementation of the remaining functions not yet implemented. Additional details on the second version of the prototype to be released for the Italian demo will be provided in D3.4 expected in M38 (October 2022).

3.1.3 Greek Demo

Figure 7 represents the architecture planned to be deployed and integrated in the first release of the Greek demo site, designed following the Use Cases UC-GR-1, UC-GR-2, UC-GR-3, UC-GR-4 and UC-GR-5, described in D1.1 [7], and D4.1 [10], and the related scenarios.

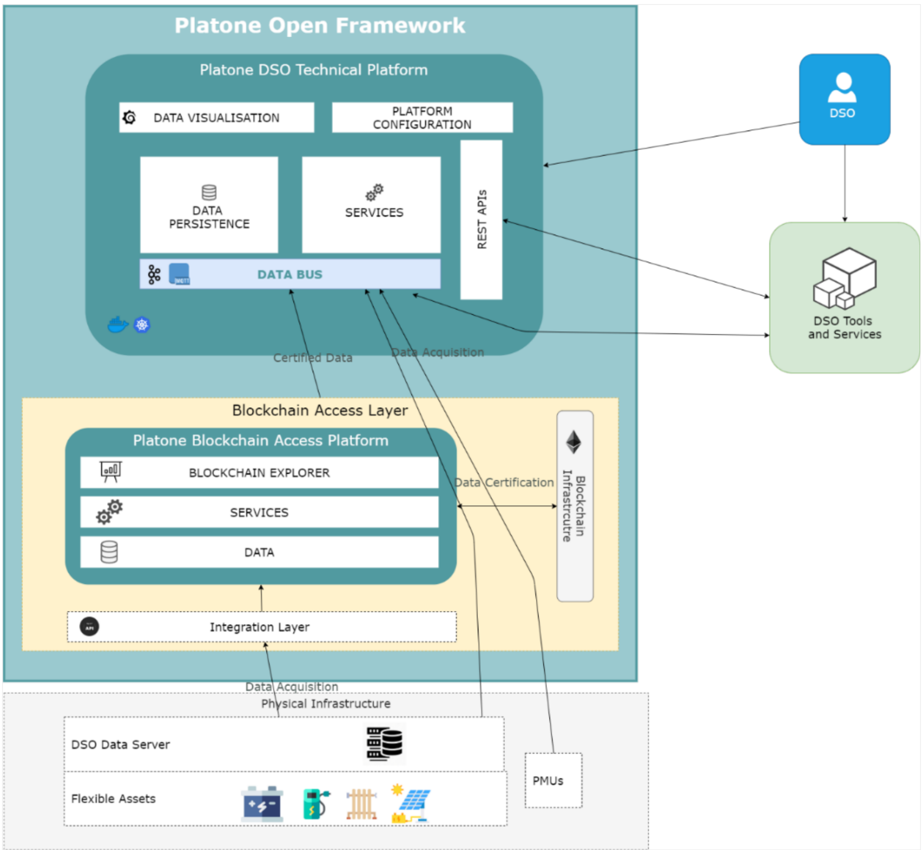

Figure 7: Greek Demo Architecture - Initial Version (v1)

The main goal of Greek demo is to enhance the DSO's observability of its network via an advanced state estimation (SE) tool. In addition, the demo will investigate whether adopting variable network tariffs, enables a more efficient operation of the distribution network or even allows for provision of ancillary services to the TSO by the end users of the distribution network.

The first version of the Platone Blockchain Access Layer was successfully deployed and integrated into the Greek Demo environment. Some examples of measurement data coming from the DSO Data Server was acquired, registered within the Shared Customer Database, and certified within the blockchain infrastructure.

Although not originally foreseen in the Greek environment, it was decided to also include the Shared Customer Database to add the possibility to store all the measurement data of the DSO Data Server in a secure and certified way and make them accessible and available for other actors and stakeholder.

Following this, it was also decided to acquire the measurement data only from the Blockchain Access Layer (including PMU data expected for the second version) and not directly for the DSO Technical Platform that will be integrated to the BAL, as already tested and evaluated in other demos (RWTH Lab and German Demo).

The updated Greek integrated architecture are shown in Figure 8.

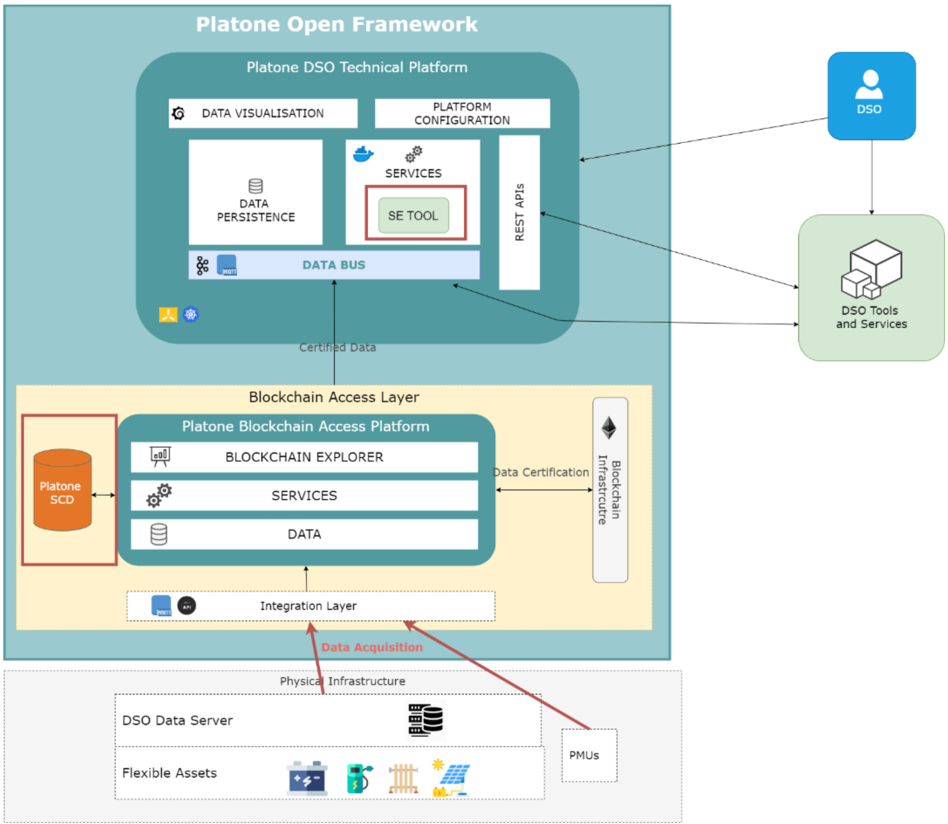

Figure 8: Greek Demo Architecture - Updated Version (v2)

The DSOTP is not yet fully deployed and integrated in the Greek Demo environment since after the initial integration tests it was decided to make another change. The SE Tool was originally planned as an external service, to be integrated using REST APIs. It is instead being integrated and released directly as a service of the DSOTP. This is done on the one hand to improve its performance and on the other hand to enrich the DSOTP with core functions developed within the project.

The integration of these decisions will be concluded in the second version of the DSOTP, expected in October 2022 and will be tested and validated, together with all the other requirements, in the second validation phase.

All the details about the tools and services developed in the Greek Demo are available in the D4.2 [11] and D4.3 [12].

3.1.4 German Demo

Figure 9, represents the architecture that was deployed in the German demo site and was designed following the Use Cases UC-GE-1, UC-GE-2, UC-GE-3 and UC-GE-4 and the related scenarios, described in D1.1 [7], and D5.2 [13].

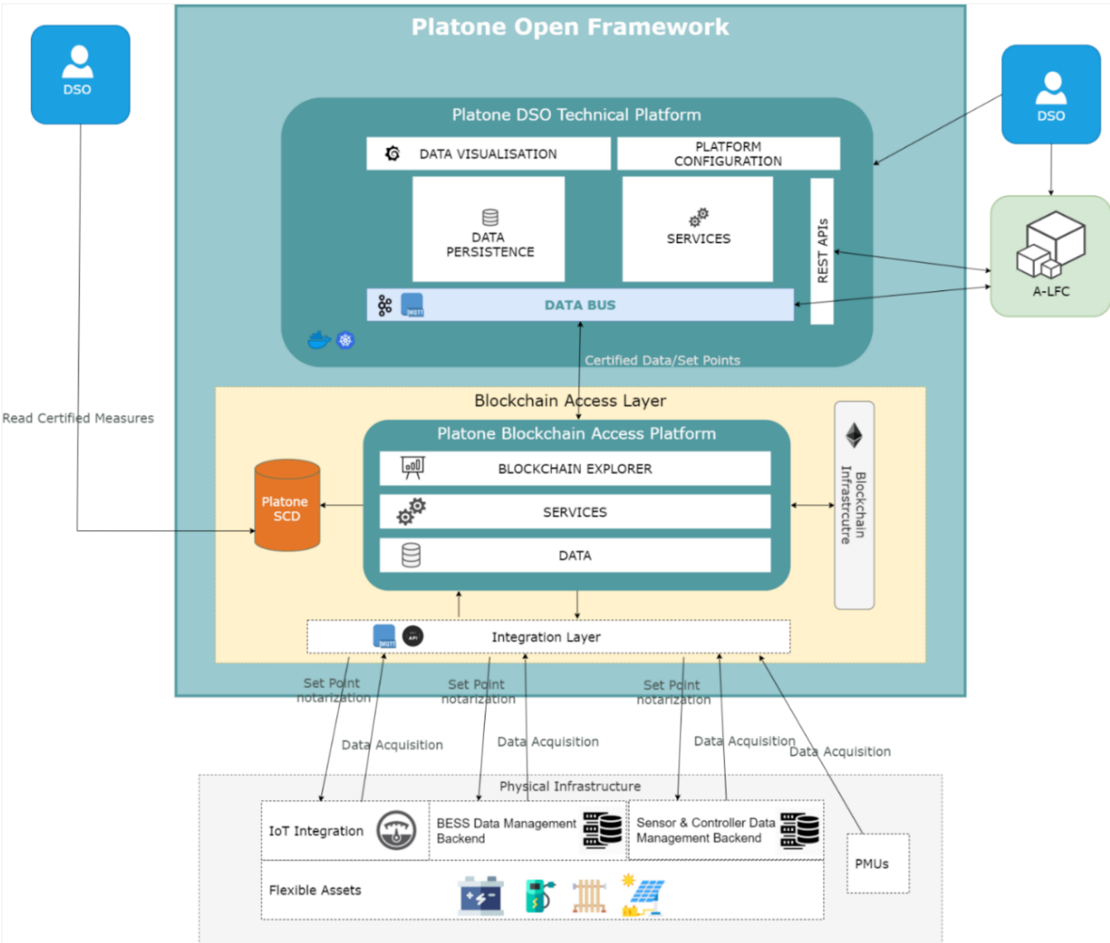

Figure 9: German Demo architecture – Initial version (v1)

The German Demo architecture foresees the integration of Avacon Local Flex-Controller (ALF-C) with the Platone framework, with the main goal of monitoring and balancing a local network and implementing new strategies of energy supply. More details on German Demo architecture can be found on D5.2 [13].

In this first integrated prototype released for the German Demo, the expected architecture was fully deployed in the Avacon Cloud environment based on Kubernetes platform [14].

The entire architectural stack expected, that includes the Blockchain Access Layer for collecting and certifying data from the sensors and the DSOTP for providing DSO services and interfaces for the Avacon Local Flex-Controller was successfully integrated.

The ALF-C was integrated with the Platone DSOTP and connected to systems located in the field, such as sensors in customer households and the secondary substation providing the measurement data.

The Blockchain Access Layer has been revised from a deployment point of view, since it was not originally ready to be released using Kubernetes, but only Docker [15] and Docker-compose [16].

Therefore, the deployment approach was changed and configured for being more reusable and scalable [17]. Finally, it was integrated with the local PMU installed in the German demo site and the features expected for this first execution phase was tested (data acquisition, data certification and data provisioning to the DSOTP).

After a technical evaluation, the possibility of using the DSOTP and the BAL revised for the activation and sending of setpoints was revised and it was therefore considered appropriate to modify the expectations from this point of view, using ALF-C directly for sending setpoints to individual devices. The architecture for the German Demo (Figure 10) and the list of requirements have been revised accordingly.

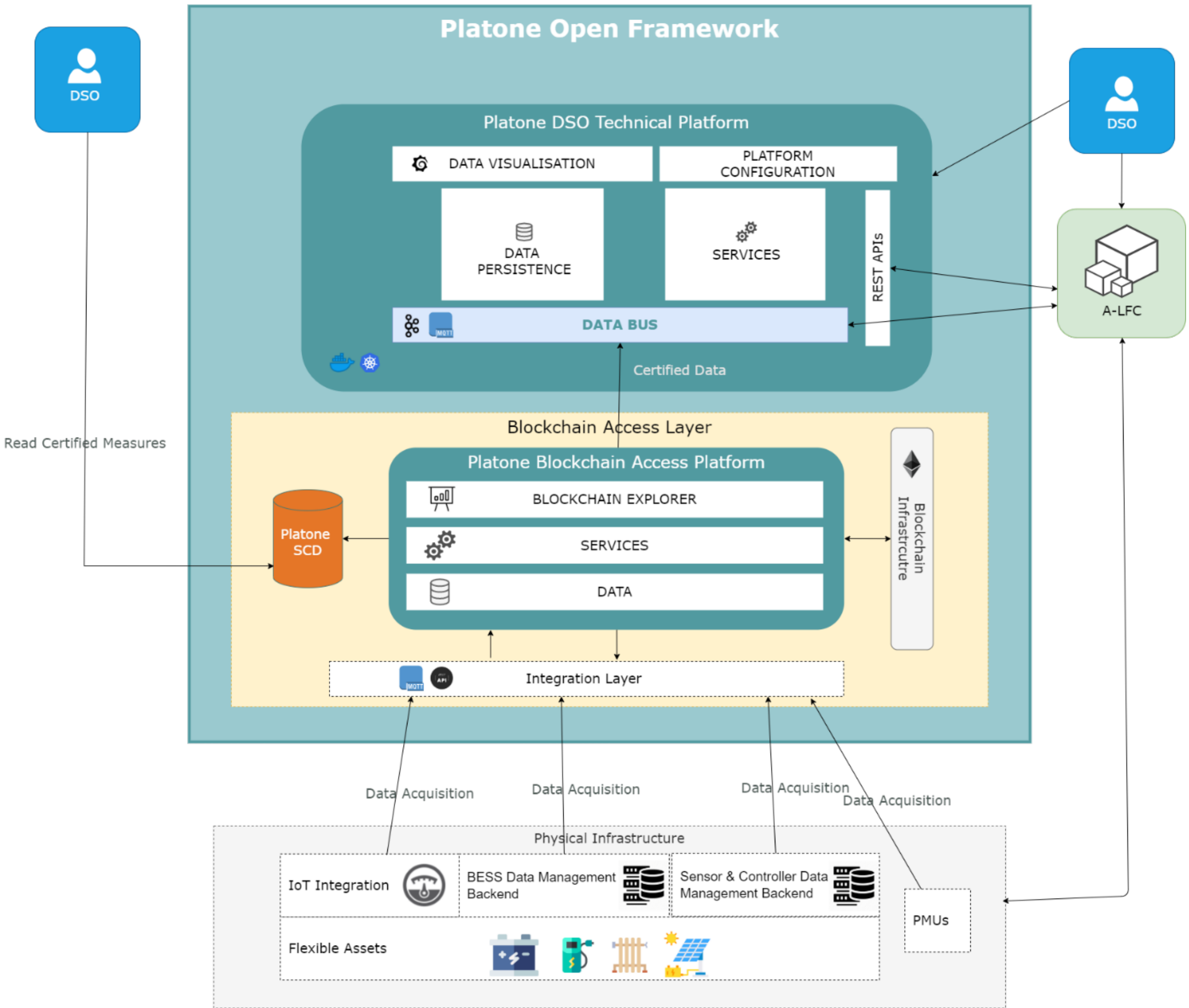

Figure 10: German Demo Architecture - Updated Version (v2)

The next version of the prototype will focus on the completion of the expected activities to meet the 4 different German Use Cases and the list of requirements not yet completed.

4 Updates and evolution of the Platone Reference Architecture

4.1 Platone Reference Architecture v2

As described in the D2.1 [2], the Platone Reference Architecture leverages on the definition of the Platone Open Framework. The main characteristics of the Platone Open Framework are

- **Open**, to be extended and integrated with external services providing standard interfaces
- **Flexible**, in terms of integration with already existing external solutions (e.g. legacy platforms)
- **Secure**, for data handling, using blockchain technology for data certification and integrity and for data interoperability, using standard communication protocols and following the data security and data privacy best practices

The Platone Open Framework's main components are the Platone Platforms which are designed to pursue these objectives and to have these characteristics.

The Platone Reference Architecture v1 paved the road for the implementation of the first prototype of the Platone Platforms and the overall Platone Open Framework.

The Market Platform enables the creation of an open energy marketplace creating the ecosystem 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. The first prototype of the Market Platform was integrated, tested and evaluated within the Italian Demo architecture with the main goal to implement an overall flexibility market for solving congestion issue coming from DSO grid.

The DSO Technical Platform ensures an easy integration of external systems both at data level and service level. The first prototype of the DSOTP was integrated with the Blockchain Access Layer and tested and evaluated within the RWTH Lab, the German Demo and the Greek Demo. The main goal of this validation focused on the possibility to integrated secure and certified data coming from the physical infrastructure trough the Blockchain Access Layer and at the same time offer specific services exploiting those data to the DSO. In the Greek Demo, the DSOTP is also used as testbed for the deployment of the additional services (e.g. the State Estimation Tool)

The Blockchain Access Layer allows the integration of data coming from the physical infrastructure adding a further level of security and trustworthiness to the framework, exploiting the blockchain and smart contracts technologies. The first prototype of the Blockchain Access Layer was fully integrated with the devices installed in the German and Greek demo site, as well as the PMU tested in the RWTH lab. All the data collected in the BAL was certified and stored in a secured manner in the SCD and then provided to the DSOTP for further applications.

The feedback received during the integration and validation phase was more than satisfactory and the main characteristics of the architecture that have been reflected in the implementation have satisfied the prerequisites and needs of the subset of the use cases and requirements implemented and evaluated so far.

For these reasons, the second version of the Platone Reference Architecture does not make substantial changes to the previous one.

The only change came from experimenting with the possibility of using the DSOTP as a decentralised platform with a series of multiple instances at edge level, named DSO Edge Platforms.

This concept foresees additional instances of the DSOTP that can be deployed on the edge level[2] in order to aggregate data and potentially conduct reprocessing steps in order to reduce the amount of data that needs to be transferred to the main DSOTP.

This additional component will not be tested and integrated in the trial sites but will be tested as a proof-of-concept in the RWTH lab. More details on this component are provided in the Chapter 4.3.2.1.

[2] Computation at edge level (or Edge Computing) is performed at or near the source of the data, as opposed to cloud computing, which is performed remotely.

Figure 11 represents the evolution of the Platone Reference Architecture, including the decentralised DSO Edge Platform.

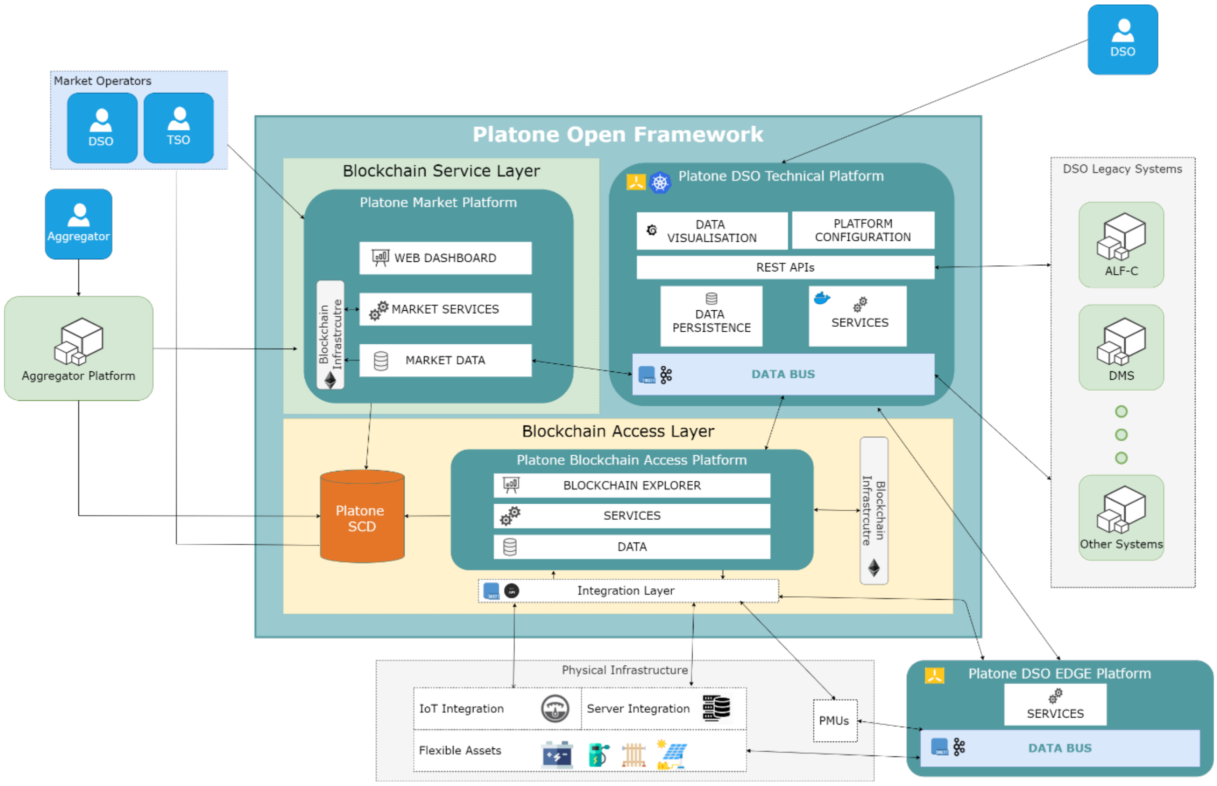

Figure 11: Platone Reference Architecture v2

4.2 Architectural components

As already mentioned in the previous subchapter, the Platone Reference Architecture does not foresee substantial changes except for the new conceptual decentralised component, the DSO Edge Platform which is an EDGE-level extension of the DSOTP.

Some architectural components of the Italian Demo (i.e., Operation System and Energy Management System) are no longer foreseen, as they will no longer be implemented within the project but components already existing in the Areti IT environment will be used.

Finally, within the German demo, a new component for the management of storage and household devices is planned.

Table 1 below shows the updated list of the Architectural Components expected within the Platone project.

Table 1: Platone Architectural Components

Component	WP	Responsible Partner	Contributing Partners	Notes
Platone Market Platform	WP2	ENG		
Platone DSO Technical Platform	WP2	RWTH	ENG, SIEM	
Platone DSO EDGE Platform	WP2	RWTH		**New.** Conceptual component. Edge level instance of the DSOTP.

Platone Blockchain Access Platform	WP2	ENG		
Platone Shared Customer Database	WP2	ENG		
Aggregator Platform	WP3	ACEA	SIEM	
Italian DSO Technical Platform	WP3	ARETI	SIEM	
Italian Blockchain Access Layer (including Light Node)	WP3	APIO		Renamed from Italian Blockchain Access Platform
Italian Shared Customer Database	WP3	ARETI		
Aggregator-Customer App	WP3	ACEA	APIO	
Energy Management System	WP3	APIO		Not implemented in WP3. Already in place in the demo site.
Operational Systems (including Supervisory Control and Data Acquisition - SCADA)	WP3	ARETI		Not implemented in WP3. Already in place in the demo site.
TSO Simulator	WP3	ENG		
DSO Data Server	WP4	HEDNO	NTUA	
Algorithm for DER control	WP4	NTUA		
State Estimation Tool	WP4	NTUA		
Algorithm for ancillary services	WP4	NTUA		
EMS - Avacon Local Flex Controller (A-LFC)	WP5	Avacon		
CBES Data Management Backend	WP5	Avacon		Renamed from BESS
Sensor & Controller Data Management Backend	WP5	Avacon		
HBES Data Management Backend	WP5	Avacon		**New** (Description included in Annex A**)**

4.3 Platone Platforms

4.3.1 Platone Market Platform

The Platone Market Platform is one of core components of the Platone Open Framework and its main goal is to enable a secure and transparent flexibility market, exploiting blockchain technology and smart contracts.

The blockchain technology and the implementation of specific smart contracts allow to handle the management of flexibility services, providing market results to all the stakeholders, validating the flexibility provisioning, and performing the settlement outcome with an innovative incentivisation mechanism for improving customer engagement.

The first version of the Platone Market Platform (shown in Figure 12) was released and integrated in the Italian Demo.

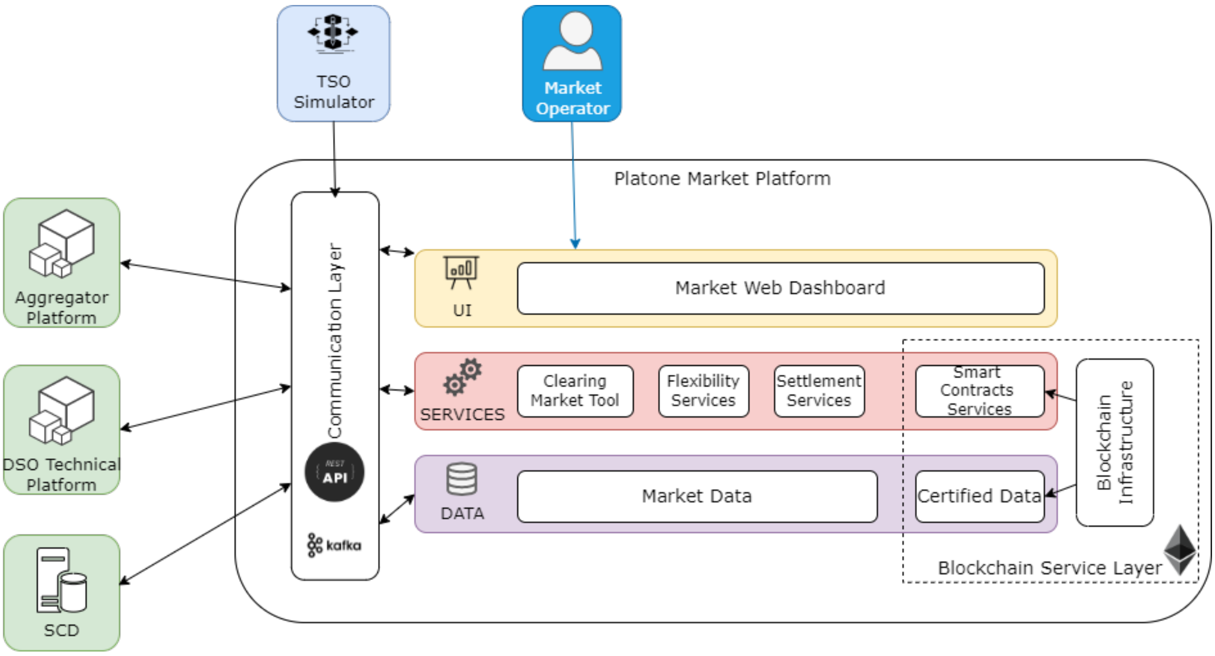

Figure 12: Platone Market Platform Architecture v1

All the architectural layers have been implemented, and all services have been included with the exception of the Service Activation which is no longer expected as the responsibility of the Market Platform. It has therefore removed and moved to the Aggregator Platform.

The integration was focused, as described in Chapter 3.1.2, above all on the implementation of the flexibility market in day ahead sessions with the following features:

- receive requests and offers from Market Participants (DSOs, TSOs and Aggregators),
- produce Market Results (Clearing mechanisms),
- provide the Market Results to the DSOTP for traffic light mechanism (technical validation),
- aggregating Market and Technical Results and provide them to all Market Participants for activation,
- verify the measurement from the SCD and perform the settlement phase.

A web dashboard was also made available for the Administration User (Market Operator).

The deployment of the Market Platform, shown in Figure 13, was done in the ENG cloud environment and the integration took place through components distributed in other cloud systems. For this reason greater emphasis was given to the communication and security aspects between the various environments and the different platforms.

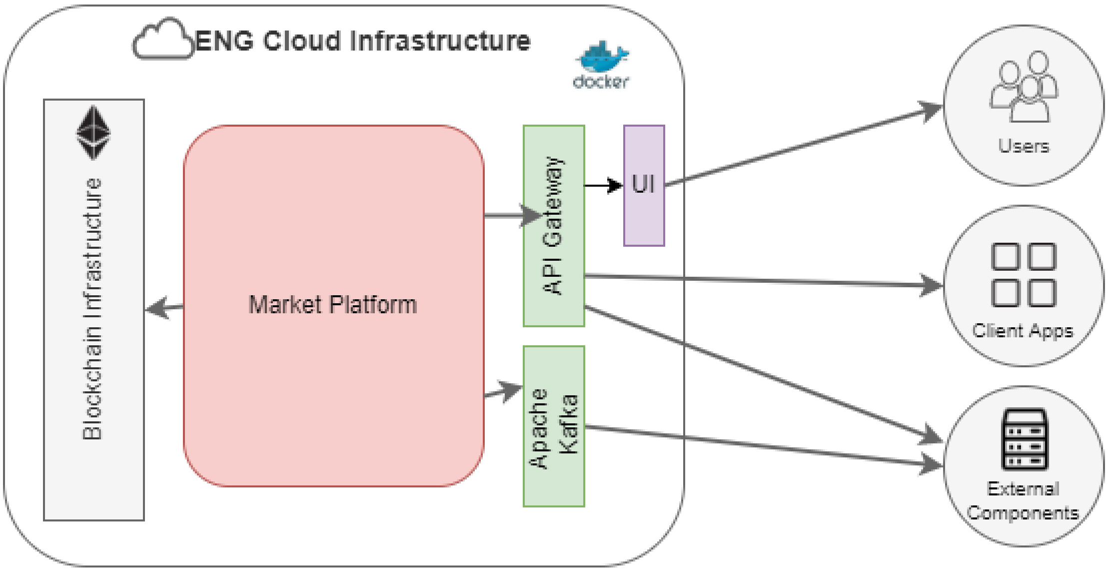

Figure 13; Market Platform cloud deployment

As already described, the API Gateway of the Market Platform implements an authentication mechanism based on Oauth2.0 over HTTPS. The Kafka Broker is secured through a mutual authentication based on TLS and client-side certificates. The web dashboard is also under HTTPS connection, and it is accessible by the Administrator User (Market Operator) using basic credentials (username and password).

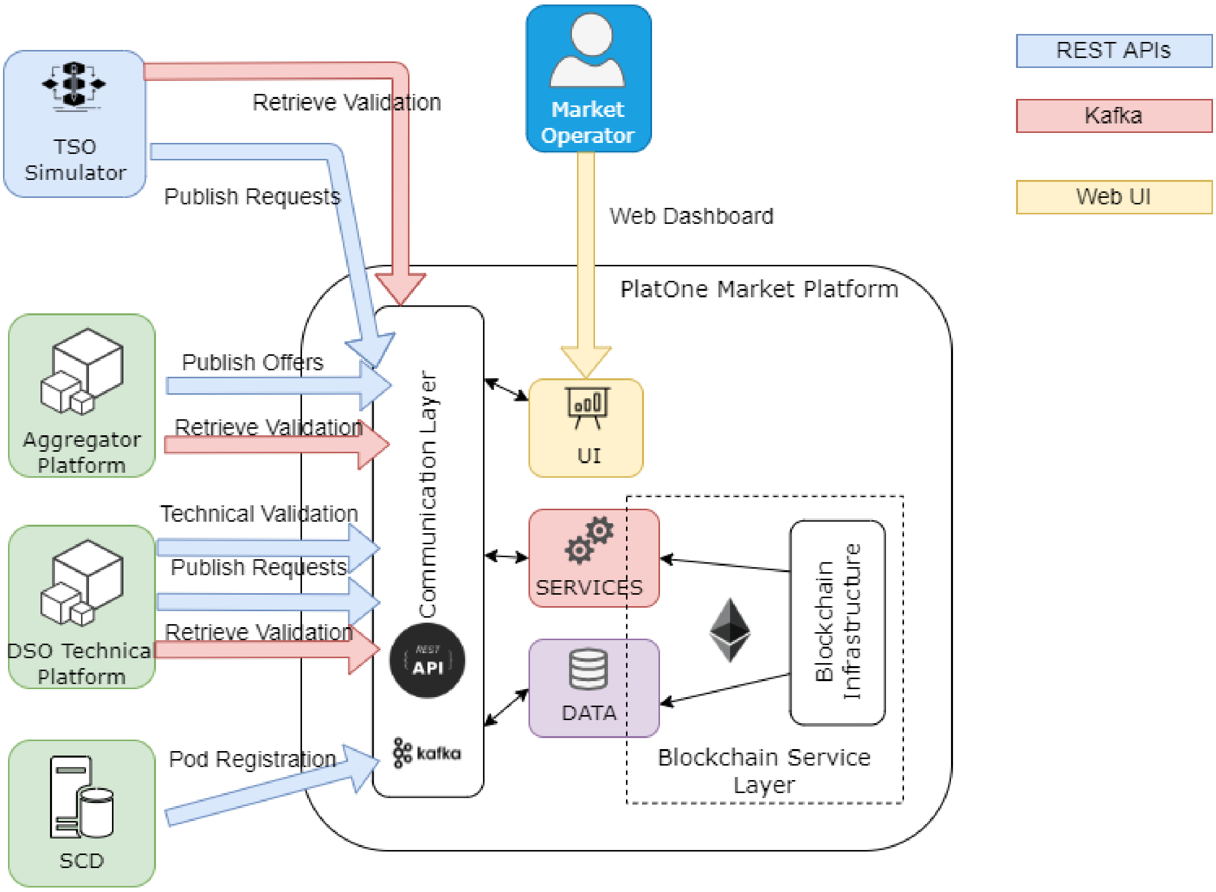

Figure 14: Platone Market Platform integration and communication mechanisms

Figure 14 shows in detail the different communication and integration mechanisms.

The second version of the Platone Market Platform will focus on the implementation of the functional and non-functional requirements not yet completed and, on the integration, and execution in the Italian Demo for completing the use cases expected (voltage violation management and real time market sessions).

4.3.2 Platone DSO Technical Platform

The Platone DSOTP (shown in Figure 15) is another of the core components of the Platone Open Framework. It is designed to host auxiliary grid services for the grid operation of a DSO such as grid monitoring and observability, market interactions, or flexibility optimization. As described in D2.1 [2], a key design aspect lies in the decoupling of different functionalities into individual micro services to ensure modularity of the different services and core infrastructure components of the platform. Furthermore, it facilitates the availability and scalability of the platform by the means of replicating and scaling individual micro services. All services running on the DSO Technical Platform are deployed in individual containers that are orchestrated by Kubernetes [14], an open-source container orchestration system and de-facto standard for automating deployment, scaling, and management of containerized workloads.

With respect to the previous description of the DSOTP, certain technological aspects evolved from the integrations with the other platform components of the Platone Open Framework and the deployments and integrations in the different demo sites. These aspects include the integrated services and their exposed interfaces, the platform deployment, and the required core infrastructure components of the platform.

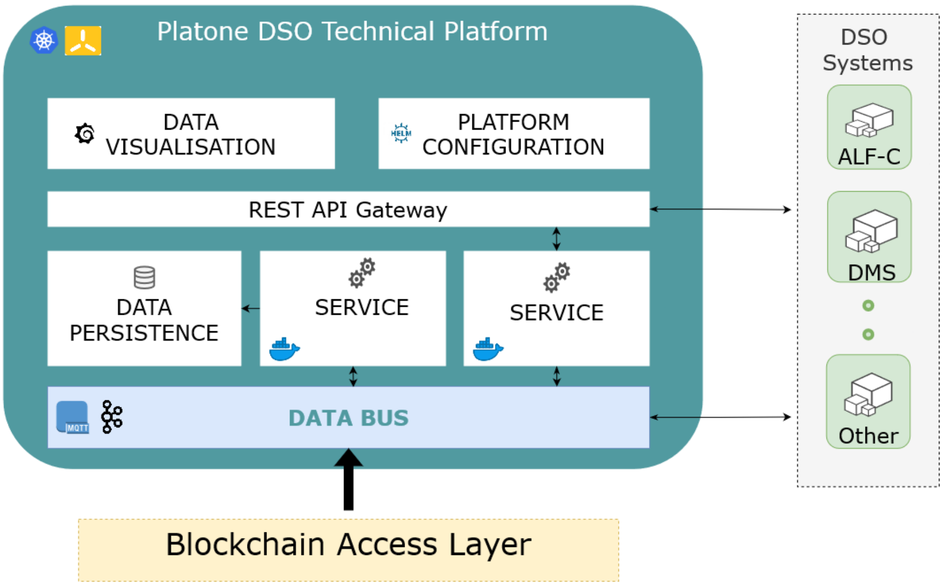

Figure 15: Platone DSOTP Architecture

During an early stage of the demo preparation, it became apparent that the demos had different requirements for the deployment of the platform components. For the Greek demo (WP4), the deployment of the platform on self-operated servers on premises was preferred, whereas for the German Demo (WP5), the deployment of the platform on a public cloud infrastructure was required. The choice of Kubernetes as the underlying orchestration layer for the DSO Technical Platform simplified meeting these contrasting requirements as it is available as managed Kubernetes clusters from all large cloud providers and can also be self-hosted on one's own premises. For development and quick setup, a light-weight Kubernetes distribution such as k3s [18] can be used to remove the entry barrier and reduce prototyping time. The deployment phase of the platform for the German Demo also included a step of hardening the Kubernetes deployment configurations to ensure that the most stringent security requirements are met. A detailed description of these adjustments will be included in the second release of the DSO Technical Platform.

The integration of services developed in other work packages into the DSO Technical Platform revealed additional requirements to the ones reported in D2.1 [2]. As described previously, services on the DSOTP are encapsulated in individual Docker containers that contain the core algorithm or logic of the service and the required interfaces towards the Databus layer and/or the data persistence layer (cf. Figure 15). In addition, a service can expose a REST API for configuration or event-based triggering. The integration of the Balancer Service developed in WP5 revealed the need to also expose service REST APIs to other DSO Systems such as the ALF-C. Furthermore, integrating the State-Estimation tool developed in WP4 as a service into the DSOTP also requires a data store for static grid data such as gird topology models. To fulfil these requirements, the implementation of an additional data store for grid model data and a REST API Gateway is foreseen for the next release of the DSOTP.

The DSO Technical Platform is based on previous work done in the Horizon 2020 project SOGNO [19] which is continued as an open-source project under the umbrella of the Linux Foundation Energy [20]. Platone WP2 is actively contributing to the community of the SOGNO LFE Project [21] and aims at merging new features upstream to the code base.

4.3.2.1 Platone DSO Edge Platform Concept

In addition to the original concept of the DSO Technical Platform as a central instance for data processing of DSO auxiliary services, the concept of the Platone DSO Edge Platform (see Figure 16) evolved in WP2. In this concept, not only a single instance of the DSO Technical Platform is used to host grid services, but additional instances can be deployed on the Edge in order to aggregate data and potentially conduct reprocessing steps in order to reduce the amount of data that needs to be transferred to the main DSOTP. However, such use cases are beyond the scope of Platone and not foreseen to be tested in the project field-trails. A proof-of-concept could be conducted in the scope of the Platform Lab Tests.

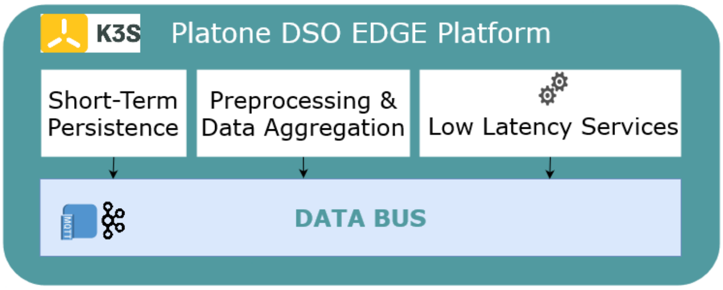

Figure 16: Platone DSO Edge Platform

Bringing the DSO Technical Platform closer to the edge and the physical grid assets, would also allow for the implementation of services with higher real time requirements for the acquired data such as control, fault detection or reconfiguration. In this context, the Edge can refer to a public (mobile) cloud instance that is located geographically close to the affected grid section or it can be realised by deploying the DSO Edge Platform on operator-owned hardware, e.g. on substation level. Again, the Kubernetes orchestration layer provides the required flexibility for deploying the DSO Edge platform either on public infrastructure or on custom infrastructure. In particular, the k3s reference setup of the DSO Technical Platform is reusable for the deployment, as k3s is designed for production workloads in unattended, resource-constrained, remote locations or inside IoT appliances [18].

As illustrated in Figure 16, the Platone DSO Edge Platform follows the architectural design of the DSO Technical platform of using a central data bus. On top of the data bus, pre-processing or data aggregation algorithms can be executed in order to reduce the amount of data to be transferred to the main DSO Technical Platform. Besides the data processing stage, additional services that require a low latency link to the affected grid region can also be deployed on the DSO Edge Platform as illustrated on the left side of the service layer. Depending on the requirements of such low latency services, also a data persistence layer can be added to the DSO Edge platform. The data persistence layer of the DSO Technical Platform can be reused and adopted for the DSO Edge Platform.

4.3.3 Platone Blockchain Access Layer

The Blockchain access layer, is an architectural layer included in the Platone Open Framework that adds a further level of security and trustworthiness to the framework. It is an extension of physical infrastructure and performs multiple tasks, among which are the data certification and automated flexibility execution through Smart Contracts.

The Figure 17 represents the first version of the BAL released at M18 (February 2021).

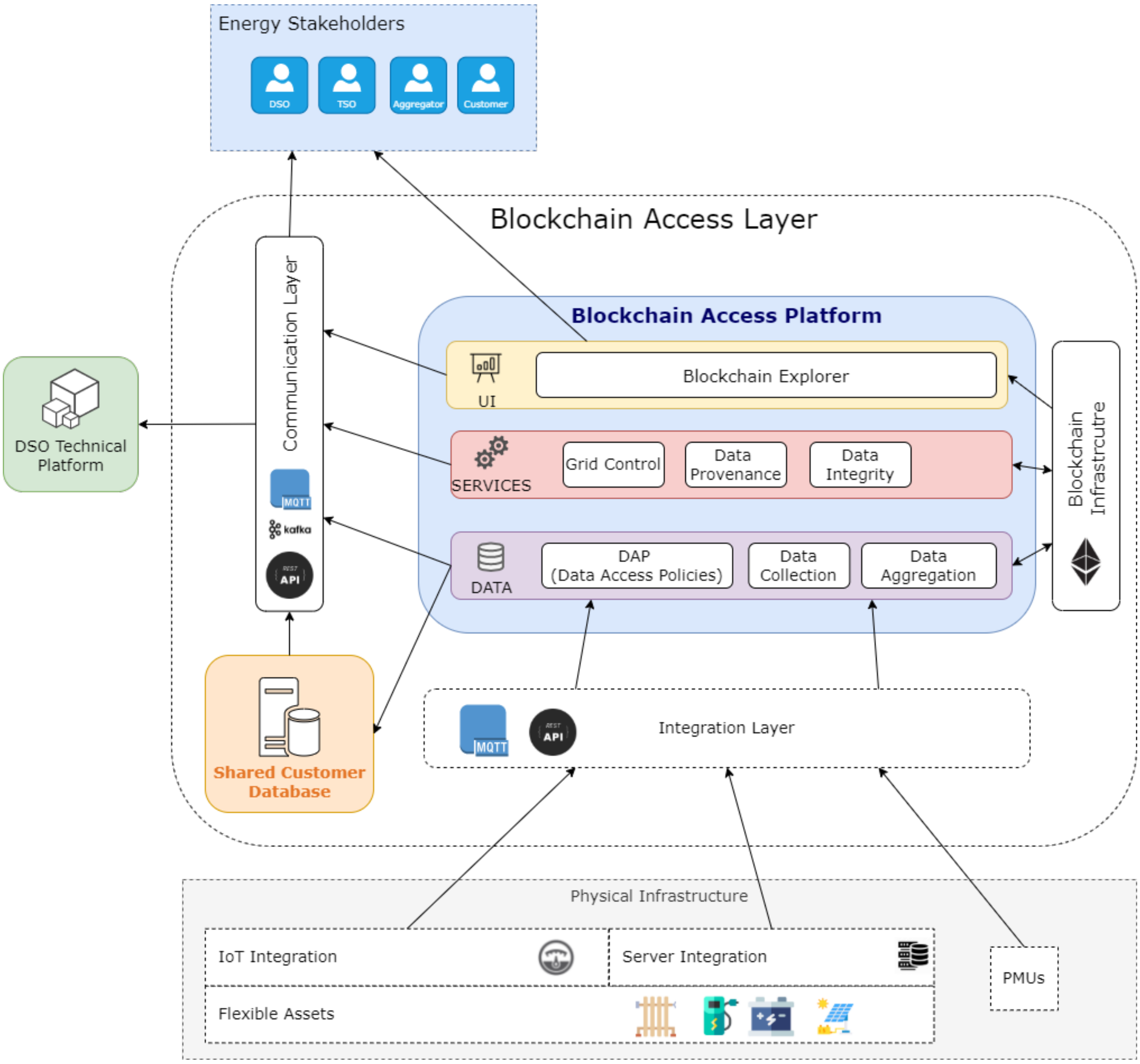

Figure 17: Platone Blockchain Access Layer Architecture v1

The first prototype of the BAL was integrated into the overall Platone Open Framework and with the devices coming from the field and the DSOTP and supports two different data models: standard CIM IEC 61968-9 [22] for measurement data of smart meters and customised data model for the PMU measurement.

It provides also a basic set of functionalities for:

- Data collection and aggregation,
- Certification of data ownership and timestamping based on blockchain technology,
- Data provisioning to external actors (e.g., DSOTP).

The integrated version was initially tested in the RWTH Lab as described in Chapter 3.1.1 and included the integration of a PMU using MQTT under TLS protocol with server-side certification, data acquisition (simulated with one measurement per second), data certification (aggregated data, every 3 minutes) and data provisioning (instantly to the DSOTP).

The complete same configuration was replicated in the German Demo, even if, as described in Chapter 3.1.4, the deployment approach was different as it introduced Kubernetes solutions for the deployment in the German Cloud environment.

The BAL was also deployed the Greek Demo environment exploiting Docker containerisation and integrating measurement data coming from the DSO Data Server in the CIM format every 15 minutes. The integration with the DSOTP in the Greek Demo environment is expected for the second integration phase.

All the functionalities expected for the BAL was already implemented in the first version. In fact, the only missing functional requirement that originally foresaw the integration, certification, and communication of the setpoints to the network in the German demo, have been cancelled after a technical validation that has assessed the implementation of this function as too complex. It will therefore instead be carried out directly by the Energy Management System ALF-C.

The second version of the BAL will focus on an additional tool, the Data Access Policies service, which even if not foreseen in any demo use case, could be a relevant feature as it allows to enrich the possibilities offered by BAL in terms of data management. In addition, scalability, security and performance test will be conducted and the integration in the Greek Demo (as described above) will be completed.

5 Requirements of the Platone Platforms - Updates

This list of requirements represents the baseline for the implementation of the three versions of the Platone Platforms that will be respectively released in M18 (February 2021, already released), M38 (October 2022, the next one) and M46 (June 2023, the final one).

The integrated prototype of the Platone Open Framework includes all the Platone Platforms and is evaluated with in the three different trials in Italy, Greece, and Germany, as well as in the experimentation phase in the RWTH lab.

As already mentioned in the Chapter 2 Use Cases, Scenarios, and Information flows have been used as the input for defining the first version (see D2.1) of functional and non-functional requirements for Platone Platforms. The validation phase allowed to update the status of these requirements, defining new ones, and cancelling others. Table 2 below reports the updated version of the Platone Platform Requirements (v2), that includes all the functional and non-functional requirements expected for the Platone platforms developed within WP2. The information regarding the other components and services expected in WP3, WP4 and WP5 will be described in the respective deliverable by the responsible WPs.

Table 2: Platone Platforms Requirements (v2)

Requirement ID	Requirement name	Requirement description	Use Cases	Status	Notes
Market Platform – Functional Requirements					
FR_MP_I_1	Initialisation	The Market Platform 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	**New**	
FR-MP-FSM-01	Flexibility Services Management	The Market Platform allows DSOs and TSOs to create flexibility requests in automatic way	UC-IT-1 UC-IT-2	**Completed**	
FR-MP-FSM-02	Flexibility Services Management	The Market Platform allows DSOs to create flexibility requests through 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	The Market Platform allows Aggregator Platform to create flexibility offers in automatic way	UC-IT-1 UC-IT-2	**Completed**	
FR-MP-FSM-04	Flexibility Services Management	The Market Platform 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	The Market Platform is able to match flexibility requests and offers through clearing market algorithms	UC-IT-1 UC-IT-2	**Completed**	
FR-MP-MOMV-02	Market Outcomes Matching and Validation	The Market Platform is able to provide the Market Outcomes (results of market clearing) to the DSO Technical Platform for the technical validation	UC-IT-1 UC-IT-2	**Completed**	
FR-MP-MOMV-03	Market Outcomes Matching and Validation	The Market Platform receives the validated market outcomes from DSO Technical Platform	UC-IT-1 UC-IT-2	**Completed**	
FR-MP-MOMV-04	Market Outcomes Matching and Validation	DSOs, TSOs and Aggregators receives Market Day Ahead outcomes from the Market Platform	UC-IT-1 UC-IT-2	**Completed**	
FR-MP-SA-01	Services activation	The Market Platform allows to DSOs and TSOs to create service activation requests in automatic way	UC-IT-1 UC-IT-2	**Cancelled**	The service activation is not responsibility of the Market Operator and cannot be performed into the Market Platform
FR-MP-SA-02	Services activation	The Market Platform allows to Market participant to create	UC-IT-1 UC-IT-2	**Cancelled**	The service activation is not

		service activation requests through UI			responsibility of the Market Operator and cannot be performed into the Market Platform
FR-MP-SA-03	Services activation	The Market Platform 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 responsibility of the Market Operator and cannot be performed automatically into the Market Platform
FR-MP-BC-01	Blockchain certification	The Market Platform is able to register on the blockchain all the market data trough Smart Contracts based functionalities	UC-IT-1 UC-IT-2	**Expected in V2 of Platone Platforms (M38)**	
FR-MP-BC-02	Blockchain certification	The Market Platform allows to Market participant to verify all the market data registered in the blockchain	UC-IT-1 UC-IT-2	**Expected in V2 of Platone Platforms (M38)**	
FR-MP-SET-01		The Market Platform allows to Aggregator to create new smart contracts with settlement mechanisms via UI	UC-IT-1 UC-IT-2	**New**	
FR-MP-SET-02		The Market Platform provides to Aggregator Platform a list of available Smart Contracts with settlement mechanisms	UC-IT-1 UC-IT-2	**New**	

FR-MP-SET-03	Settlement	The Market Platform 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	The Market Platform 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	The Blockchain Service Layer is able to provide tokenization system for the settlement through Smart Contracts functionalities	UC-IT-1 UC-IT-2	**Expected in V2 of Platone Platforms (M38)**	Renamed from FR-MP-S-03
FR-MP-SET-06	Settlement	The Market Platform allows to DSO, TSO and Aggregator to read the settlement outcomes	UC-IT-1 UC-IT-2	**Completed**	FR-MP-S-04
Market Platform – Non-Functional Requirements					
P-MP-01	Communication protocols	The Market Platform exposes REST APIs for collecting flexibility requests and flexibility offers	UC-IT-1 UC-IT-2	**Completed**	
P-MP-02	Communication protocols	The Market Platform provides a message broker for communicating market results	UC-IT-1 UC-IT-2	**Completed**	
T-MP-01	Communication Protocols, Timing	The Market Platform is able to receive measurements from SCD Kafka Broker every 15 minutes	UC-IT-1 UC-IT-2	**New/Completed**	
T-MP-02	Timing	The Market Platform is able to schedule day ahead and real time Market sessions at prefixed times and in automatic way	UC-IT-1 UC-IT-2	**New/Completed**	

S-MP-01	Security	Market Platform must expose all its REST APIs under Oauth2.0 authentication and client credentials	UC-IT-1 UC-IT-2	**New/Completed**	
S-MP-02	Security	Market Platform must identify all the Kafka clients using two-way authentication and server/client certificates	UC-IT-1 UC-IT-2	**New/Completed**	
S-MP-03	Security	All the Market Platform interfaces must be exposed using TLS connections	UC-IT-1 UC-IT-2	**New/Completed**	
DSOTP - Functional Requirements					
FR-DSOTP-DA-01	Data Acquisition	The 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	The 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	**Expected in V2 of Platone Platforms (M38)**	
FR-DSOTP-DA-03	Data Acquisition	The DSOTP is able to receive PMU measurements that reflect the network state	UC-GR-2 UC-GR-3 UC-GR-4 UC-GR-5	**Expected in V2 of Platone Platforms (M38)**	
FR-DSO-TP-DA-04	Data Acquisition	The 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	The 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 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	The DSOTP is able to trigger the State Estimation Tool via REST API.	UC-GR-1 UC-GR-2 UC-GR-3 UC-GR-4	**Updated**	
FR-DSOTP-SE-02	State Estimation	The 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	**Expected in V2 of Platone Platforms (M38)**	
FR-DSOTP-PMU-01	PMU Data Integration	The 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	**Expected in V2 of Platone Platforms (M38)**	
FR-DSOTP-T-01	Tariffs retrieval	The DSOTP sends to the DSO/Aggregators tariffs that reflect the expected state of the network	UC-GR-3 UC-GR-4	**Expected in V2 of Platone Platforms (M38)**	
FR-DSOTP-T-02	Tariffs retrieval	The DSOTP is able to receive data coming from the Algorithm for DER Control and Algorithm for ancillary services	UC-GR-3 UC-GR-4	**Expected in V2 of Platone Platforms (M38)**	
FR-DSOTP-AS-01	Data to DER control and Ancillary Services	The 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	**New**	
FR-DSOTP-DER-01	Optimal DER dispatching	DSOTP is able to trigger the	UC-GR-3	**Expected in V2 of**	

		Algorithm for DER Control via REST API		Platone Platforms (M38)	
DSOTP – Non-Functional Requirements					
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 A-LFC via MQTT and/or HTTP protocol	UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4	**Expected in V2 of Platone Platforms (M38)**	Specified correct protocols. Changed from generic TCP/IP
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	**Expected in V2 of Platone Platforms (M38)**	
BAP – Functional Requirements					
FR-BAP-DM-01	Blockchain Data Management	The BAP is able to acquire Measurements from network	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**	After an internal technical evaluation was decided to avoid sending the Setpoints through the BAL since is too complex to be implemented during the project phase.
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**	After an internal technical evaluation was decided to avoid sending the Setpoints through the BAL since is too complex to be implemented during

					the project phase.
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**	After an internal technical evaluation was decided to avoid sending the Setpoints through the BAL since is too complex to be implemente d during the project phase.
BAP – Non-Functional Requirements					
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**	
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	UC-GR-2 UC-GR-3	**New/Completed**	

		must be under TLS connection	UC-GR-4 UC-GR-5 UC-GE-1 UC-GE-2 UC-GE-3 UC-GE-4		

6 Conclusion

The work done at this stage means that the design phase of the Platone Open Framework is concluded. This included a first round of important feedback coming from the integration and validation phase in the three different field tests (Italy, Greece, and Germany) plus the experimentation in the RWTH Lab.

The findings of the validation phase confirmed that the characteristics of the Platone Open Framework identified in the first design phase are relevant for the needs and requirements of the different use cases.

In particular, it has been demonstrated that a single framework supports the desired feature of adaptability and reusability. This was demonstrated as the same architectural stack was for example used in three different IT environments (RWTH Lab, Greek Demo and German Demo) for covering different use cases with different business goals but with similar technical needs.

In addition, the overall framework characteristic was fully replicated in the Italian Demo, thus demonstrating the goodness of the architectural choices for the interaction and cooperation of the different platforms to meet the necessary requirements and needs of the DSO and the other energy stakeholders.

Some necessary adjustment has already been made to the architecture during the integration phase, based on the requests and needs of the different demos. Other adjustments will be necessary to include in the second version of the prototypes, together with the requirements not yet completed in this run.

In addition, the DSO Technical platform will be extended adding a decentralized component, the DSO Edge Platform for allowing the integration of data sources and computation at edge level. This new configuration could (later in the project) be tested in the RWTH lab and then proposed to a real field test.

The results obtained so far will allow us to further evolve the Platone solutions, consolidating and improving both the individual Platone Platforms as well as the overall Platone Open Framework.

The intermediate prototype, that will be realized and released at the end of the second phase (M40) will include all the functionalities expected and can be tested not only from the functional point of view but above all in terms of scalability, replicability, and security being able to leverage not only in field tests involving a wider and more complex group of users but also on the results of the activities of the WP7.

7 List of Tables

Table 1: Platone Architectural Components 19
Table 2: Platone Platforms Requirements (v2) 27
Table 3: Platone Household Battery Energy System 42

8 List of Figures

Figure 1: Methodology for the design of the Platone Reference Architecture and definition of the Platone Platforms requirements 8
Figure 2: Platone Implementation approach 9
Figure 3: Design methodology applied to an iterative approach 10
Figure 4: BAL and DSOTP integration test - Process 1 11
Figure 5: BAL and DSOTP integration test - Process 2 12
Figure 6: Italian demo architecture 13
Figure 7: Greek Demo Architecture - Initial Version (v1) 14
Figure 8: Greek Demo Architecture - Updated Version (v2) 15
Figure 9: German Demo architecture – Initial version (v1) 16
Figure 10: German Demo Architecture - Updated Version (v2) 17
Figure 11: Platone Reference Architecture v2 19
Figure 12: Platone Market Platform Architecture v1 21
Figure 13; Market Platform cloud deployment 22
Figure 14: Platone Market Platform integration and communication mechanisms 22
Figure 15: Platone DSOTP Architecture 23
Figure 16: Platone DSO Edge Platform 24
Figure 17: Platone Blockchain Access Layer Architecture v1 25

9 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)”, Platone EU Horizon 2020 Project, 2020. [Online]. Available: https://platone-h2020.eu/data/deliverables/864300_M12_D2.1.pdf.

[3] “Smart Grid Use Cases Repository,” [Online]. Available: https://smart-grid-use-cases.github.io/.

[4] A. Srivastava, S. Bhardwaj and S. Saraswat, “SCRUM model for agile methodology,” International Conference on Computing, Communication and Automation (ICCCA), pp. 864-869, 2017.

[5] “D2.14 - PlatOne Integrated Framework Prototype (v1)”, Platone EU Horizon 2020 Project, 2021. [Online]. Available: https://www.platone-h2020.eu/media/Deliverables.

[6] “D2.11 - PlatOne BlockChain Customer Access Layer (v1)”, Platone EU Horizon 2020 Project, 2021 [Online]. Available: https://www.platone-h2020.eu/media/Deliverables.

[7] “D1.1 General Functional Requirements and specifications of joint activities in the Demonstrators”, Platone EU Horizon 2020 Project, 2020. [Online]. Available: https://platone-h2020.eu/data/deliverables/864300_M12_D1.1.pdf.

[8] “OAuth 2.0,” [Online]. Available: https://oauth.net/2/.

[9] B. Pournader, “What is two-way TLS?” 23 May 2018. [Online]. Available: https://benpournader.medium.com/what-is-two-way-tls-d90600e2fc06.

[10] “D4.1 Report on the definitions of KPIs and UCs”, Platone EU Horizon 2020 Project, 2020. [Online]. Available: https://platone-h2020.eu/data/deliverables/864300_M12_D4.1.pdf.

[11] “D4.2 - State estimation tool”, Platone EU Horizon 2020 Project, 2020 [Online]. Available: https://platone-h2020.eu/data/deliverables/864300_M15_D4.21.pdf.

[12] “D4.3 - Algorithm for ancillary services”, Platone EU Horizon 2020 Project, 2021. [Online]. Available: https://www.platone-h2020.eu/media/Deliverables.

[13] “D5.2 Detailed Use Case Descriptions”, Platone EU Horizon 2020 Project, 2020. [Online]. Available: https://platone-h2020.eu/data/deliverables/864300_M12_D5.2.pdf.

[14] “Kubernetes (K8s),” [Online]. Available: https://kubernetes.io/.

[15] “Docker,” [Online]. Available: https://www.docker.com/.

[16] “Docker Compose,” [Online]. Available: https://docs.docker.com/compose/.

[17] D. Fontani, “Five good reasons for using Kubernetes,” 15 November 2020. [Online]. Available: https://towardsdatascience.com/5-reason-for-using-kubernetes-b7ade82eda90.

[18] “Lightweight Kubernetes,” [Online]. Available: https://k3s.io/.

[19] “SOGNO H2020 Project,” [Online]. Available: https://www.sogno-energy.eu.

[20] “Linux Foundation Energy,” [Online]. Available: https://www.lfenergy.org/.

[21] “Linux Foundation Energy - SOGNO,” [Online]. Available: https://www.lfenergy.org/projects/sogno/.

[22] “IEC 61968-9:2013 - Application integration at electric utilities - System interfaces for distribution management - Part 9: Interfaces for meter reading and control,” [Online]. Available: https://webstore.iec.ch/publication/6204.

10 List of Abbreviations

Abbreviation	Term
ALF-C	Avacon Local Flex-Controller
API	Application programming interface
BAL	Blockchain Access Layer
BAP	Blockchain Access Platform
CBES	Control Battery Energy Storage System
CIM	Common Information Model
DER	Distributed Energy Resource
DSO	Distribution System Operator
DSOTP	DSO Technical Platform
EMS	Energy Management System
ENG	Engineering Ingegneria Informatica S.p.a.
HBES	Household Battery Energy Storage System
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over SSL
IEC	International Electrotechnical Commission
IoT	Internet of Things
IT	Information Technology
JSON	JavaScript Object Notation
MP	Market Platform
MQTT	Message Queue Telemetry Transport
PoD	Point of Delivery
PoM	Point of Measurement
PMU	Phasor Measurement Unit
REST	Representational State Transfer
RWTH	Rheinisch-Westfälische Technische Hochschule Aachen
SCD	Shared Customer Database
SE	State Estimation
TCP/IP	Transmission Control Protocol / Internet Protocol
TLS	Transport Layer Security
TSO	Transmission System Operator
UC	Use Case
UI	User Interface
WP	Work Package

Annex A Platone Household Battery Energy System - Description

Table 3: Platone Household Battery Energy System

Name of Device	**Household Battery Energy Storage System**
Description	Prototype for domestic battery storage system, operated with rooftop photovoltaic system for PV self-consumption (Battery, Inverter, Smart Energy Meter incl. sensor, Router)
Functionality	Household: PV self-consumption, peak-shaving of battery charging EMS: PV generation data, battery status data, control charging/discharging
Measurement	PV-Generation: P (t), E (t); Battery status: SOC (%), SOE) (kWh), Battery control (P);
Measurement Range	Details listed in the document: https://www.kostal-solar-electric.com/de-de/produkte/hybrid-wechselrichter/plenticore-plus/-/media/document-library-folder---kse/2020/12/15/13/38/ba_kostal-interface-description-modbus-tcp_sunspec_hybrid.pdf
Measurement Resolution	15 minutes
Accuracy	NA
Data Connections	Modbus TCP/IP via RJ45 (Ethernet)
Data Output Format	U16 An unsigned word (16-bit). U32 An unsigned double word (32-bit). S16 A signed word (16-bit). S32 A signed double word (32-bit). MBD Multiple bytes data **Detailed Description:** https://www.kostal-solar-electric.com/de-de/produkte/hybrid-wechselrichter/plenticore-plus/-/media/document-library-folder---kse/2020/12/15/13/38/ba_kostal-interface-description-modbus-tcp_sunspec_hybrid.pdf
Data Rate	15 minutes
Data Availability	NA