Platone

PLATFORM FOR OPERATION OF DISTRIBUTION NETWORKS

Platone

PLATform for Operation of distribution NEtworks

D2.7

Platone DSO Technical Platform (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:	31/10/2022
Actual delivery date:	28/10/2022
Main responsible:	Siemens SPA
Work package:	WP2 – Platform Implementation and Data Handling
Security:	P
Nature:	DEM
Version:	V1.0
Total number of pages:	18

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 accompanies the second software delivery of the Platone DSO Technical Platform and extends it with an architecture overview and a description of the additional services developed to enable the use cases from the demo sites.

Keyword list

Platone DSO Technical Platform, Platone Framework, Open Source, micro-service, control centre, kubernetes

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).

This Deliverable addresses the Platone DSO Technical Platform that is part of that Open Framework and aims at enabling DSOs to fulfil market requests by evaluating the current grid state and activating local flexibility requests while ensuring the reliability and operational quality of service. This document accompanies the second software delivery of the Platone DSO Technical Platform confirming the adopted approach and building on the solid base delivered with the first version of the platform more interfaces, communication mechanisms and internal services; this phase of the work involved actively the teams of the demo sites enabling their specific use case.

Authors and Reviewers

Main responsible		
Partner	**Name**	**E-mail**
SIEMENS		
	Brunella Conte	brunella.conte@siemens,com
	Carlo Adriano Arrigoni	carlo.arrigoni@siemens.com
Author(s)/contributor(s)		
Partner	**Name**	
RWTH		
	Jonas Baude	jonas.baude@eonerc.rwth-aachen.de
ENG		
	Ferdinando Bosco	ferdinando.bosco@eng.it
Reviewer(s)		
Partner	**Name**	
HEDNO		
	Effrosyni Maria Gralista	e.gralista@deddie.gr
	Theodoros Panagiotis Stathakopoulos	t.stathakopoulos@deddie.gr
NTUA		
	Panagiotis Pediaditis	panped@mail.ntua.gr
Approver(s)		
Partner	**Name**	
RWTH		
	Amir Ahmadifar	aahmadifar@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 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 demos (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 overall solution named Platone Open Framework (cf. Figure 1) includes the following three layers:

- Blockchain Service Layer
- Blockchain Access Layer
- Platone DSO Technical Platform

For more detail please refer to the Introduction of Deliverable D2.6 [2].

The second version of the Platone DSO Technical Platform described in this Deliverable adds on the existing platform architecture the following:

- alternative Databus implementation based on the Apache Kafka
- DSOTP integration with the Blockchain Access Layer by means of an MQTT-Bridge
- API Gateway improvements
- new services development and integration (Balancer Optimizer, Power Calculation, State Estimation)
- adoption of a central identity management reference building-block

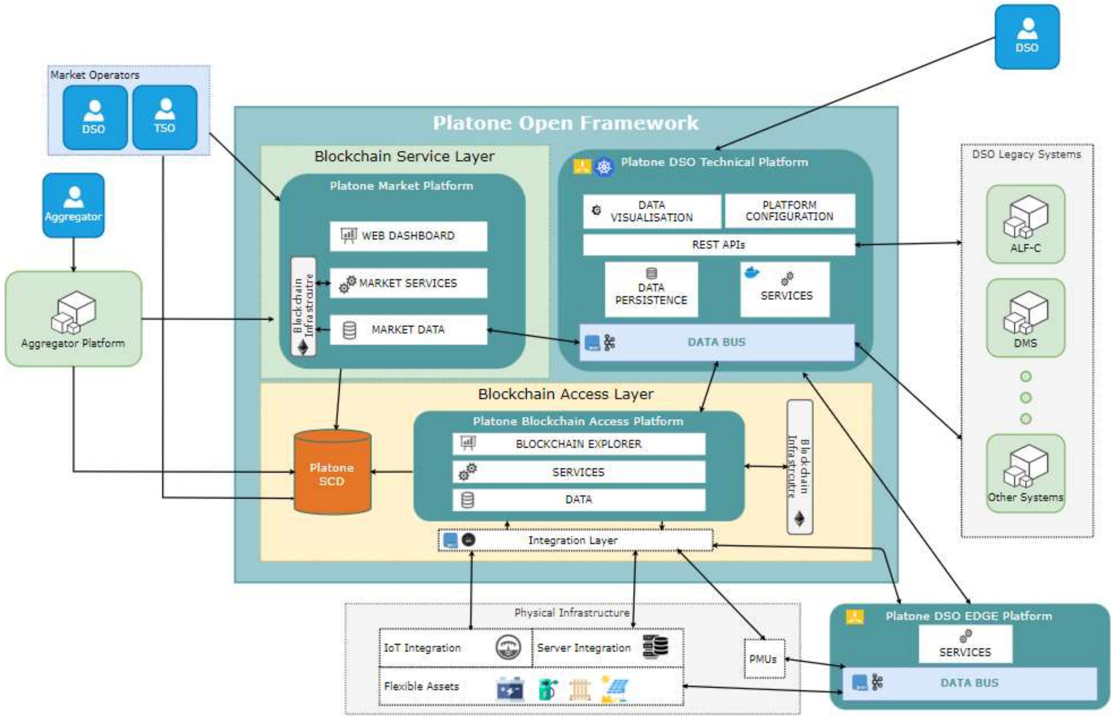

Figure 1 Platone Open Framework

1.1 Task 2.3

This deliverable is related to the Task 2.3 that aims at the implementation of a DSO Technical Platform, which allows a DSO to fulfil market requests by evaluating the current grid state and activating local flexibility requests while ensuring the reliability and operational quality of service. Therefore, a micro-service based platform architecture is presented that allows the deployment of services like state-estimation and load prediction. Furthermore, the platform aims at an enlarged grid observability by providing a visualization of measured and predicted data.

1.2 Objectives of the Work Reported in this Deliverable

The objective of this deliverable is to present the architecture of the Platone DSO Technical Platform and its realization by standard components that allows the integration of custom DSO system services. The Platone description of action defines this deliverable as a demonstrator. This document accompanies the code repository with a more detailed architecture description as well as some extended instructions for deploying, testing and integrating the platform.

1.3 Outline of the Deliverable

The second Chapter of this document describes the second realization of the Platone DSO Technical Platform evolving the first implementation (refer to Deliverable D2.6 [2]) and discusses the new interfaces, communication mechanisms and functionalities. Chapter 3 delivers a compilation of Languages, Technologies and External Tools used throughout the platform. Chapter 4 is closely linked to the software delivery and provides the guidelines about installation, setup and configuration. Finally, Chapter 5 concludes this Deliverable.

1.4 How to Read this Document

This document reports the software delivery of the second release of the Platone DSO Technical Platform, which is part of the Platone Open Framework that is implemented within WP2 of Platone. Further information on the first release version is available in the Platone Deliverable D2.6 [2]. For a better understanding of the open framework of platform and requirements for the platform, we recommend reading D2.1 [3] and D2.2 [4]. Besides that, this document provides a condensed version of the platform documentation and relates it to the formal design of the platform.

2 Platform Architecture

This chapter recalls briefly the architecture of the Platone DSO Technical Platform (for more detail please refer to Deliverable D2.6 [2]). The second and third sections provide a detailed description of the improvements introduced in the second Platone DSO Technical Platform release.

2.1 Architecture

The Platone DSO Technical Platform design builds on previous work done in the Horizon 2020 project *SOGNO* [5] and relies massively on a micro-service architecture.

The presented platform architecture aims at facilitating the transition to modular, micro-services based control center software solution for distribution system operators. This allows for faster adjustment and independent development of components.

To address requirements such as high availability, scalability and modularity from the very beginning, the DSO Technical Platform is designed for deployment on kubernetes [6] clusters. As all mirco-services of the platform are per requirement containerized in Docker [7] containers, they can easily be deployed on a kubernetes cluster. Kubernetes also simplifies different deployment approaches: from edge- and public-cloud to on-premise installation.

Figure 2 illustrates the architecture of the DSO Technical Platform. The Databus is one of its core components and is implemented by means of a message broker to which all services can publish and / or subscribe in order to exchange data with other services, with field devices, or with external systems.

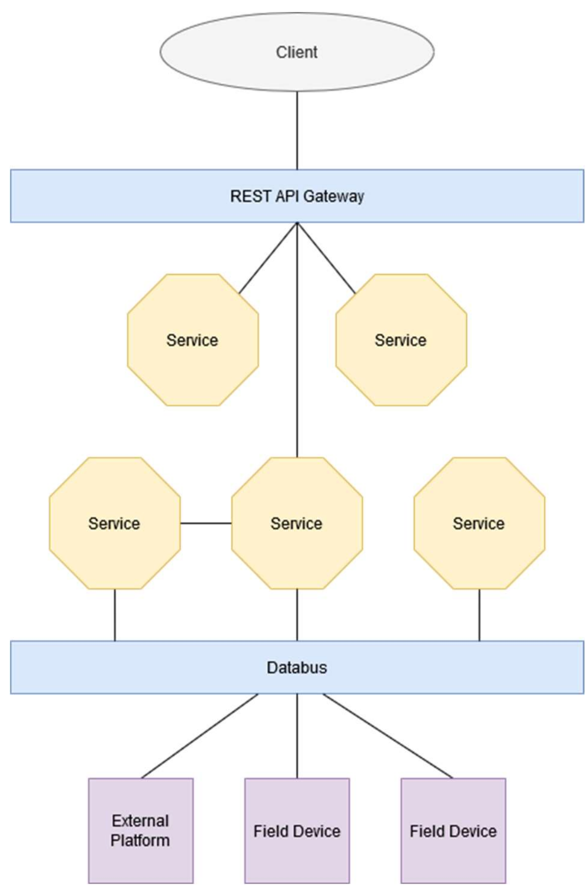

Figure 2 Platone DSO Technical Platform Architecture

2.2 Interfaces and Communication Mechanisms

This subsection briefly presents the functionalities of the different platform reference components.

Databus

For the second release, an alternative Databus implementation based on the Apache Kafka broker was developed. Apache Kafka is a distributed stream-processing platform that is commonly used in various industry domains due to its high-throughput and low-latency data streaming capabilities. During the development phase, initial tests with PMU data streaming and a time series database connector were successfully conducted over the Kafka Databus. As a next step, the integration with the Kafka interface of the Blockchain Access Layer of the Platone Open Framework should be conducted (this task will be further discussed during the next integration step expected for December 2022). Furthermore, the increasing spread of Kafka can facilitate the integration of other legacy or modern external systems.

MQTT Bridge

In the second release of Platone DSO Technical Platform, the DSOTP was integrated with the Blockchain Access Layer or other external systems by means of an MQTT-Bridge. The bridge connects to the MQTT interface of the BAL and forwards certain MQTT topics from the BAL into the message broker of the DSOTP. The set of topics to bridge between the platforms is configurable. Furthermore, the bridge allows for a topic-remapping. For example, all MQTT subtopics of "platone/dsotp/#" from the BAL can be mapped to topics with the prefix "/bal/" within the message broker of the DSOTP order to avoid any naming conflicts between the MQTT topic namespaces of the two platforms.

In this integration approach the MQTT-bridge of the DSOTP acts as a client that connects to the BAL. Therefore, the BAL has to provide credentials for the DSOTP and can apply its internal authentication and authorization mechanisms. During the integration phase, the bridge was successfully tested to forward data from the BAL to the DSOTP while remapping the MQTT topics.

Technically, the MQTT-Bridge is based on the Mosquitto MQTT Broker using a bridge mode configuration to connect to the internal MQTT Brokers of the BAL and the DSOTP. It is based on the official Mosquitto docker image. In addition, the Platone DSOTP git repository contains a template for the custom bridge configuration file and a Kubernetes deployment allowing to deploy the bridge alongside the DSOTP.

API Gateway

The first release of the DSO Technical Platform used the basic functionality of the kubernetes ingress reverse proxy.

While the message broker or event bus implementations are based on well-established open-source implementations such as RabbitMQ or Apache Kafka, they usually bring native support for TLS and sufficient authentication mechanisms. In contrast, the REST API interfaces are custom implementations e.g. for platform components or services. To achieve the same level of security without replicating the effort for each service, the TLS support and the authentication mechanisms are offloaded to an API gateway component. That serves as entry point for external REST API requests via HTTPS and handles the TLS termination and the authentication of the request. Within the respective platform, successfully authenticated requests are then forwarded to the responsible services via plain HTTP as illustrated in Figure 3.

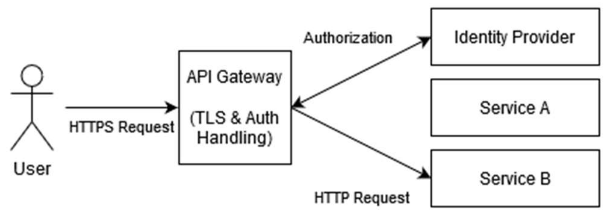

Figure 3 TLS offload and Authentication in API Gateway

2.3 Functionalities

Services

All services are able to connect to the Databus and may provide individual REST APIs as an interface for other services or to be exposed to external clients (users or systems). All services for the platform are deployed in Docker containers and restricted to the usage of the specified communication protocols. The programming languages are not specified to allow a wide flexibility for service developers. Based on the needs of the different Platone Work Package field trial activities, the following tranche of services was developed, or integration activities were initialized to support the services on the DSOTP:

- Balancer Optimizer (Platone WP5)
  - The balancer is a rule-based and optimization-based scheduling service for BESS for a virtual islanding in a local energy community with the objective of demand-generation balance. It is implemented in python and the optimization service is available via a REST API.
- Power Calculation (Platone WP5)
  - The Power Calculation Service is capable of enriching measurement data from Phasor Measurement Unit (PMU) field devices with additional power quantities. Therefore, the streaming data from the field device is consumed via MQTT and the resulting quantities are published back to the Databus to serve additional services. The power calculation service is implemented in Golang.
- State Estimation (Platone WP4)
  - State Estimation plays an important role in monitoring and control of power systems. In the context of WP4 activities related to the Greek field trial, an existing state estimation toolbox is about to be integrated as a service into the DSOTP. First integration activities included the alignment of the respective interfaces and used data models as well as a reference for deploying Matlab code (or its open-source derivative Octave) on the DSOTP.

Identity Management

For the second release of the DSOTP, a central identity management reference building-block was added to ensure a uniform user and access management for all integrated components and interfaces. The DSOTP Identity Management is based on Keycloak [10], an open-source implementation of an identity management solution that allows single sign-on functionalities by default with many existing open-source tools, e.g. the Grafana based visualization layer, via a variety of implemented standard authentication protocols. The Keycloak instance also provides the oAuth2 endpoints for the API request authorization in the API gateway proxy for the services.

3 Languages, Technologies and External Tools

The architecture of the DSO Technical Platform consists of different open-source tools and new components developed in the context of the different Platone demo use case requirements. The following table provides an overview over the core components of the platform and the integrated services including an insight into the used technologies, deployment methods and programming languages. The orange rows originate from the first release of the platform while the additional services and components of the second release are highlighted in green.

Table 1 Languages, Technologies and External Tools

Layer/Component	Technologies/Framework	Deployment	Languages
Infrastructure	Kubernetes (K8s, K3s) Helm Docker	bare-metal	
Databus	RabbitMQ, Kafka Broker	Helm Chart, Operator	
Databus Bridge	Mosquitto	K8S Deployment	
Timeseries Database	InfluxDB, TimescaleDB	Helm Chart	
Database Adapter	Telegraf	Docker image Kubernetes deployment	Golang
Visualization Service	Grafana	Helm Chart	
Identity Management	Keycloak	Helm Chart	
Authentication Proxy	Nginx	K8S deployment	
BESS Optimizer Service	Microservice	K8S deployment	Python
PMU Power Calculation Service	Microservice	K8S deployment	Golang
State Esimation Service	Microservice	K8S deployment	Matlab, Octave

4 Packaging and Deployment

Users that are already experienced with kubernetes and have access to a full-fledged kubernetes cluster (either on premise, hybrid or on public cloud infrastructure) can deploy the DSO Technical Platform there. For other users, for development setups, or for edge cloud deployments we provided in Deliverable D2.6 [2] instructions for setting up a minimal kubernetes cluster based on k3s [8], a lightweight kubernetes distribution. All mentioned configuration files as well as the entire documentation are available at [12]. In addition to the initial reference deployment manual of the first version of the DSOTP platform, further automation of the initial system provisioning was provided with an installation script and helm charts for a simplified deployment and configuration of services.

5 Conclusion

The work already done on the first release of the Platone DSO Technical Platform enabled the implementation of flexibility control and grid observability tools based on a micro-service oriented architecture.

This second release extends the architecture components from the minimal architecture description provided in the first released version with additional components and services required for the demo use cases. The services were defined and developed collaborating with the single Work Package teams while the integration into the DSOTP and the additional components were conducted by WP2 partners to ensure a reliable and secure operation of the services in the field trials environments of the demos.

While the additional services are key results of the individual demo work packages, the extensions to the core platform components were partially already merged into the upstream open-source project SOGNO within the Linux Foundation Energy [13]. The open source approach will ensure a better outreach and re-use of the results as well as an increment of the impact of the Platone project on the scientific community and in particular on the energy stakeholders.

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] Platone “D2.6 Platone DSO Technical Platform (v1)” [Online].

Available: https://www.platone-h2020.eu/data/deliverables/864300_M18_D2.6.pdf

[3] Platone “D2.1 Platone Platform requirements and reference architecture (v1)” [Online]. Available: https://www.platone-h2020.eu/data/deliverables/864300_M12_D2.1.pdf

[4] Platone “D2.2 Platone Platform requirements and reference architecture (v2)” [Online]. Available: https://www.platone-h2020.eu/data/deliverables/864300_M30_D2.2.pdf

[5] “SOGNO energy: Service Oriented Grid for the Network of the Future” [Online]. Available: https://www.sogno-energy.eu/

[6] “Kubernetes open source” [Online]. Available: https://kubernetes.io/

[7] “Docker: Empowering App Development for Developers” [Online]. Available: https://www.docker.com/

[8] “K3s” [Online]. Available: https://k3s.io/

[9] “RabbitMQ” [Online]. Available: https://www.rabbitmq.com/

[10] “Keycloak” [Online]. Available https://www.keycloak.org/

[11] “Sogno documentation” [Online]. Available: https://sogno-platform.github.io/docs/

[12] “GitLab of the RWTH Aachen University” [Online]. Available: https://git.rwth-aachen.de/acs/public/deliverables/platone

[13] “LF Energy – Sogno” [Online]. Available: https://www.lfenergy.org/projects/sogno/

9 List of Abbreviations

Abbreviation	Term
API	Application Programming Interface
BAL	Blockchain Access Layer
BESS	Battery Energy Storage System
DSO	Distribution System Operator
DSOTP	DSO Technical Platform
MQTT	Message Queuing Telemetry Transport
OAuth2	Open standard for Authorization
PMU	Phasor Measurement Unit
REST	REpresentational State Transfer
TLS	Transport Layer Security
TSO	Transmission System Operator