Platone
PLATFORM FOR OPERATION
OF DISTRIBUTION NETWORKS

Platone
PLATform for Operation of distribution NEtworks

D2.11

Platone Blockchain Customer Access Layer (v1)

Project Name	Platone
Contractual Delivery Date:	28.02.2021
Actual Delivery Date:	28.02.2021
Main responsible	Ferdinando Bosco (ENG)
Workpackage:	WP2 – Platform Implementation and Data Handling
Security:	P
Nature:	DEM
Version:	V1.0
Total number of pages:	24

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 software delivery of the Platone Blockchain Layer and extends it with an architecture overview and the explanation of a demonstration setup.

The Platone Blockchain Access Layer is part of the Platone Open Framework and includes two main components: the Platone Blockchain Access Platform and the Platone Shared Customer Database.

The version of the Platone Blockchain Access Layer will be integrated, tested and evaluated in the German and Greek demo sites' architectures.

Keyword list

Platone Blockchain Access Layer, Platone Blockchain Access Platform, Platone Shared Customer Database, Data Integration, Data Certification, Smart Contracts

Disclaimer

All information provided reflects the status of the Platone project at the time of writing and may be subject to change. All information reflects only the author's view and the Innovation and Networks Executive Agency (INEA) is not responsible for any use that may be made of the information contained in this deliverable.

Executive Summary

The energy system is facing an incredible revolution whose end target is the creation of a new energy scenario widely dominated by renewable energy sources and mostly based on distributed energy generation. At the centre of this process is the distribution network where the majority of the new energy sources are and will be connected. Flexibility is a key resource in a scenario in which the grid is more and more changing from being a load-driven system to a generation-driven system, given the partial control on energy intake from renewable energy sources. This process implies also that the changes are not only related to the operational aspects but also to the market element. Digitalization is a key enabler of this process, opening the way to smart and efficient management of data sources in a secure way and making the separation between market and operation less and less meaningful.

The Platone solution consists of a layered set of platforms to meet the needs of system operators, aggregators, and end users, named **Platone Open Framework**.

The **Platone Blockchain Access Layer** is one of the core components of the Platone Open Framework. It is a blockchain-based layer that includes two different components: the **Platone Blockchain Access Platform**, which allows the integration of the data coming from the physical infrastructure, adding a level of security, transparency and trustworthiness thanks to the blockchain technology and smart contracts, and the **Platone Shared Customer Database,** which contains all the energy data (e.g., measurements, set points, etc.), providing access to the data to all the stakeholders involved, implementing data security, data privacy and data access policies mechanisms.

The first prototype of the Platone Blockchain Access Layer focuses on the integration, modelling, securing and certification of different energy data coming from the physical infrastructure.

More in detail, it implements a data integration interface for collecting energy measurements coming from smart meters and PMUs (Phasor Measurement Unit), supporting standard CIM IEC 61968-9 [1] modelling and certifying this data thanks to blockchain technology and smart contracts, **ensuring data integrity, and avoiding data tampering**.

The energy data collected, harmonized, and certified are available for all the energy stakeholders involved as well as external platforms and services (e.g. the Platone DSO Technical Platform), who need to use this data, within the Platone Shared Customer Database.

The Platone Shared Customer Database includes rules and mechanisms for defining a Data Access Policy as well as security mechanisms for **ensuring data protection and data privacy**.

Authors and Reviewers

Main responsible		
Partner	**Name**	**E-mail**
ENG		
	Ferdinando Bosco	ferdinando.bosco@eng.it
Author(s)/contributor(s)		
Partner	**Name**	
ENG		
	Angelo Triveri	
Reviewer(s)		
Partner	**Name**	
ARETI		
	Gabriele Fedele	
	Antonio Vito Mantineo	
RSE		
	Carlo Tornelli	
Approver(s)		
Partner	**Name**	
RWTH		
	Padraic McKeever	

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 acquisitions system based on a two-layer approach (an access layer for customers and a distribution system operator (DSO) observability layer) that will allow greater stakeholder involvement and will enable 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), customers and aggregators. In particular, the DSO will invest in a standard, open, non-discriminating, economic dispute settlement blockchain-based infrastructure, to give to both the customers and to the aggregator the possibility to more easily become flexibility market players. This solution will see the DSO evolve into a new form: a market enabler for end users and a smarter observer of the distribution network. By defining this innovative two-layer architecture, Platone removes technical barriers to the achievement of a carbon-free society by 2050 [2], 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 in 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 blockchain architecture named Platone Open Framework that includes a series of core components, including the Platone Blockchain Access Layer.

The Platone Blockchain Access Layer's main goal is to enable a standard, secure, and easy integration of energy data coming from the physical infrastructure and grant the access to this data to DSOs and other energy stakeholders.

Blockchain Technology and Smart Contracts play a key role in this platform bringing two main features:

- transparent unmodifiable data management and sharing is preserved and guaranteed,
- multi-party data sharing can be seamlessly extended to data collected in the field for operational purposes and not for market reasons.

1.1 Task 2.5

This deliverable is related to the Task 2.5 [3] that aims at the implementation of the Platone Blockchain Customer Access Layer, following the functional and non-functional requirements defined in D2.1 [4].

The Platone Blockchain Customer Access Layer was renamed Platone Blockchain Layer, to avoid misunderstandings on the word "Customer", as customers are not the only stakeholders to which the component is addressed. In fact, the Blockchain Access Layer aims to integrate and to manage all the data coming from the physical infrastructure and offers its features to all the energy stakeholders.

1.2 Objectives of the Work Reported in this Deliverable

The objective of this deliverable is to present the first prototype of the Platone Blockchain Access Layer and its realization following the technical specification and requirements expected. 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 deployment instructions for deploying, testing, and integrating the platform.

1.3 Outline of the Deliverable

The second Chapter of this document describe the first realization of the Platone Blockchain Access Layer according to the specification provided in Deliverable D2.1 and discusses the functionalities implemented in this first version more in detail. Chapter 3 provides a brief overview of Interfaces and Communication Mechanisms. Chapter 4 delivers a compilation of Languages, Technologies and

External Tools used throughout the platform. Chapter 5 is closely linked to the software delivery and provides detailed installation, setup and configuration instructions. Finally, Chapter 6 concludes this deliverable.

1.4 How to Read this Document

The document aims to give an overview to the Platone Blockchain Access Layer first prototype release. A description of the foreseen functional and non-functional requirements expected can be found in D2.1.

2 Platform Architecture

2.1 Overview

The Platone Blockchain Access layer (renamed from Platone Blockchain Customer Access Layer) includes two main components:

- **Platone Blockchain Access platform (BAP)**, that 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)**: it contains all the measurements, set points and other needed data collected from customer physical infrastructure. It allows the other components and stakeholders of the Platone Open Framework to access data in an easy way and without compromising security and privacy.

It also includes:

- **Integration Layer**, that allows the integration of data coming from the physical infrastructure using standard communication protocols for IoT (e.g., MQTT) and REST services
- **Communication Layer**, that enables the communication among the different internal layers of the BAP, the SCD and other components (e.g., DSO Technical Platform). It provides standard communication mechanisms like REST APIs and Message Broker.
- **Blockchain Infrastructure**, which includes a private implementation of Ethereum Blockchain infrastructure including some Ethereum nodes.

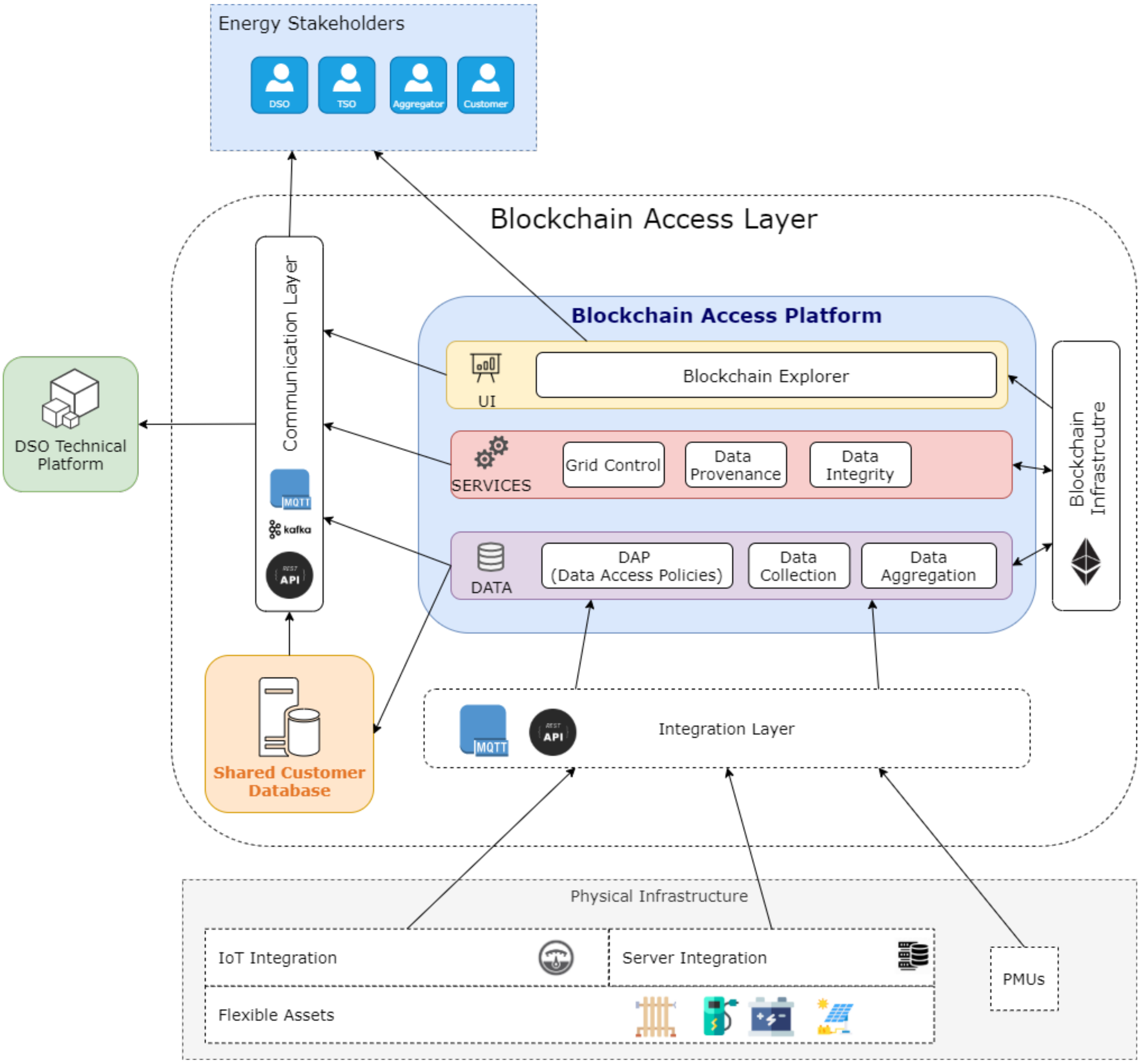

Figure 1: Platone BAL architecture

2.1.1 Platone Blockchain Access Platform

The Platone Blockchain Access Platform consists of a three-layer architecture:

- **UI Layer** includes a web blockchain dashboard that allows to monitoring all the blockchain transactions, including a visual tool for data integrity verification;
- **Services Layer** provides the business logic, including blockchain-based services for data provenance and data integrity certification, grid control mechanisms,
- **Data Layer** provides the management of the energy data, including the collection and aggregation of the data coming from the integration layer and the Data Access Policies for the data management of the Shared Customer Database.

UI Layer

The UI Layer includes the User Interfaces available for the administrator and other energy stakeholders. It includes a Blockchain Explorer for monitoring all the blockchain transactions and smart contracts and a web tool for the energy data verification system.

Services Layer

The services layer includes all the blockchain-based services for ensuring the energy data certification and transparent grid control mechanisms. Each service is implemented in a dedicated Smart Contract and it is connected to the Blockchain Infrastructure using Web3.js [5] API interface.

Data Layer

The Data Layer is the main layer of the BAP architecture. This layer provides all the services necessary for the management of the energy data collected through the integration layer. This layer implements data standard modelling based on CIM IEC 61968-9 [1], data aggregation and storage mechanisms, as well as the DAP (Data Access Policies) for sharing the data to all the relevant stakeholders through the Shared Customer Database and the Communication Layer in a secure and transparent way.

2.1.2 Platone Shared Customer Database

The Platone Shared Customer Database contains all the real-time energy data collected and makes them available to all the energy stakeholders as well as external services and platforms (e.g., DSO Technical Platform).

The main goal of the SCD is to ensure the availability of the data to those who need it and are authorized to access it, exploiting data services provided by the Blockchain Access Platform and the Communication Layer.

Given the large amount of data and high frequency provisioning within a smart grid, we decided to approach the technical implementation using Apache Cassandra [6] database that ensure scalability and high availability without compromising performance.

2.1.3 Integration Layer

The integration layer is probably the main component of the entire Blockchain Access Layer, since it is the entry point for the integration of the energy data and must ensure many aspects related to data security, protection, and privacy, as well as maintaining highly standard of performance, availability and scalability.

The main characteristic for integrating Internet of Things (IoT) devices in an IT architecture is to ensure high performance, minimal network bandwidth and small code footprint. Message Queue Telemetry Transport (MQTT) is an OASIS standard messaging protocol that perfectly matches all these characteristics.

For this reason, we decided to provide MQTT as main communication protocol in the integration layer, using Mosquitto MQTT Message broker [7]. The MQTT broker can receive messages from different sources using a tunnel-encrypted channel, which supports both certificates and authentication mode, in this way it guarantees the security and ownership of the data.

In the next version of the platforms, the integration layer could be extended with other communication mechanisms and protocols (e.g. REST APIs).

2.1.4 Communication Layer

The Blockchain Access Layer architecture includes a Communication Layer, a specific component that provides two different communication mechanisms: synchronous and asynchronous.

A specific architectural component dedicated to communication mechanisms, provides a greater flexibility to the BAL, which can cover different solution and integrate different external systems.

More in detail, the synchronous communication is implemented in the API Gateway via REST APIs. The API gateway is the entry point for every HTTP request that is being launched by the external systems.

The API gateway is developed using open-source framework Express.js [8] and Express Gateway [9]. This central component allows centralization of some middleware functionalities, i.e.:

- Authentication

- Logging
- Caching
- Security
- Load Balancing

The asynchronous communication is implemented in the Message Broker.

A message broker (or queue manager) is a software where queues can be defined. Applications may connect to the queue and transfer a message onto it.

A message can include any kind of information. For example, it could include information about a process/task that should start on another application (that could be on another server), or it could be just a simple text message. The queue-manager software stores the messages until a receiving application connects and takes a message off the queue. The receiving application then processes the message in an appropriate manner.

A message broker can act as a middleware for various services (e.g., different external systems). They can be used to reduce loads and delivery times by web application servers since tasks, which would normally take quite a bit of time to process, can be delegated to a third party whose only job is to perform them.

Message queueing allows web servers to respond to quickly requests instead of being forced to perform resource-heavy procedures on the spot. Message queueing is also good when you want to distribute a message to multiple recipients for consumption or for balancing loads between workers.

The BAL Communication Layer provides two different brokers: Mosquitto MQTT broker [7] for providing high performance data streams and Apache Kafka [10], better suited for integrating external services.

2.1.5 Blockchain Infrastructure

The blockchain infrastructure includes a private implementation of the Ethereum blockchain, to which it is possible to interface through the implementation of the Web3.js [5] services and the related REST APIs.

The blockchain nodes will contain all the transactions registered on the blockchain, as well as the smart contracts deployed for implementing the features offered by the Blockchain Access Platform. The blockchain nodes may or may not coincide with the actors involved in the electrical grid. It is realistic to imagine a scenario in which large producers may afford the host locally their own full node, while small prosumers or consumers may host a Light Node or choose to trust a third-party node.

2.2 Functionalities

The first prototype of the Blockchain Access Layer implements several functionalities, fundamentally based on data collection, data quality and data provisioning.

Data collection - Capturing, modelling and aggregation

The first set of services provided by the BAL is implemented in the Integration Layer, which captures the data and the Data Layer of the BAP, which provides support for standard data modelling and is in charge of harmonizing, aggregating and storing the data.

In the first version of the BAL, two different energy data modelling are supported:

- Smart Meter data in CIM IEC 61968-9 model (i.e. Meter Readings Objects)
- PMU data in a custom data model

All the energy data are collected using the MQTT broker and provided to the BAP for the aggregation and storage functionalities. The BAP is in charge of harmonizing and storing the data in the SCD, providing additional information regarding data ownership and timestamping, which is also used by the blockchain services for implementing data certification and validation.

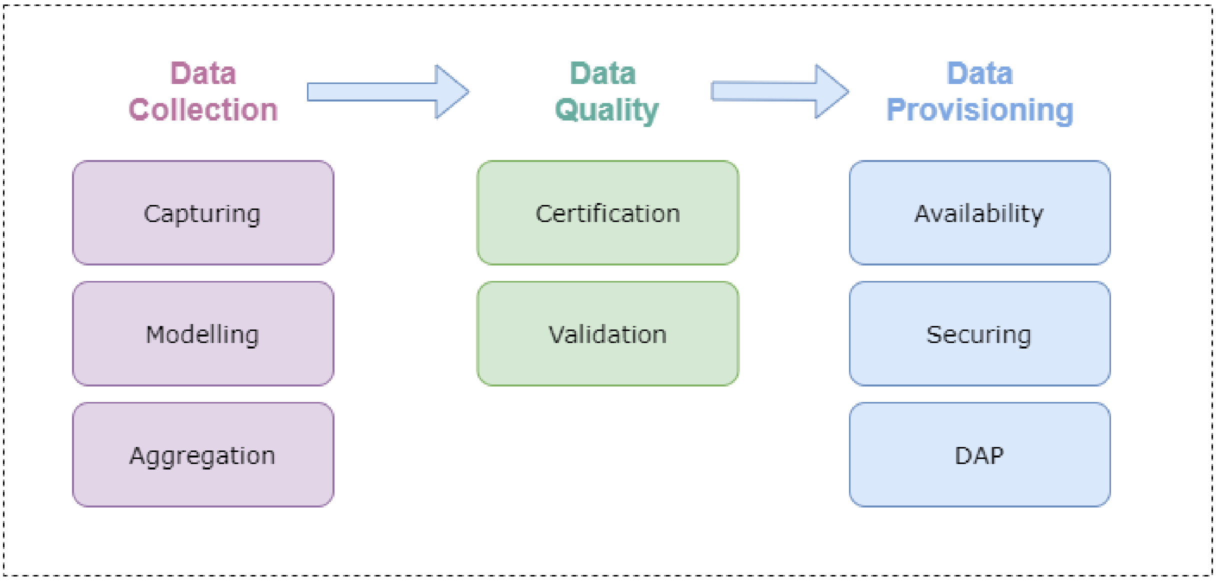

Figure 2: Platone BAL data features

Data Quality – Certification and validation

The second set of services is implemented in the Services Layer of the BAP. This layer includes all the blockchain-based service that leverage the benefits of blockchain technology for ensuring data integrity and provenance and avoiding data tampering.

The energy data certification takes place using three different pieces of information: owner, timestamp and hashed data. The blockchain infrastructure uses this information to create an “immutable” transaction that contains this three important information.

The hashed data is an aggregation of different energy data coming from the same device (the aggregation timing is configurable) and is hashed with a SHA-3 function [11]. This approach allows data encryption to be maintained (all the data registered on the blockchain is by its nature public accessible) and at the same time to easily stay under the transaction limits, with constant and relatively low transaction costs (economic and storage) [12].

The second step is the data validation. Blockchain-based hash validation can be used in several different use cases. In the case of Platone Blockchain Access Layer, the validation aims to ensure data integrity and to avoid data tampering in the energy data shared between different parties (the energy stakeholders).

In fact, at any time, by providing the data set and applying the aggregate hash function used, it will be possible to verify that the data provided are the same certified into the blockchain (data integrity) and that the ownership of the data registered (data ownership).

In the first prototype of the BAL, the data certification and validation will be applied for ensuring that data collected from smart meters and PMUs and stored on SCD are certified, immutable and coming from registered and uniquely identified devices.

Data Provisioning – Availability, Securing and Access Policies

The third set of services is implemented both in the Data Layer of the BAP and in the Communication Layer.

In fact, even if the component that provides the data is the SCD, it is the data layer that provides the data access rules and the communication layer that technically implements the access, through different communication mechanisms and ensuring authenticated access and authorized.

The Data Access Policies module implements a series of dynamic access control policies based on many contextual information (e.g., role models, volume and frequency of the accesses, location, etc.).

The Communication Layer provides an entry point for accessing the data stored in the SCD but also the data coming from the Integration Layer, which, after being aggregated and certified, are provided to other services (e.g., DSO Technical Platform).

In the first version of the BAL, two different approaches are implemented for satisfying the German and Greek use cases. In the first case data are stored in the SCD and are made available through the Communication Layer API Gateway; in the second one, the certified data are available directly on the Apache Kafka Message Broker (or Mosquitto MQTT broker) without any storage feature.

In this first version, an interaction with energy stakeholders is not expected and the only actor involved in the process is the DSO Technical Platform, which uses the energy data for providing services to the DSO.

For this reason, the version does not include the web UI for the energy stakeholders. SCD and data validation will be available through the API Gateway.

2.3 Data Models

The first prototype of the Platone Blockchain Access Layer, implements three core data models:

- Meter Readings (based on CIM IEC 61968-9) for modelling the energy measurements coming from the smart meters;
- PMU Readings, for modelling the energy measurements coming from PMUs;
- Data Certification, for certifying through blockchain technology the energy data.

Meter Readings

```
<mr:MeterReadings xmlns:mr="http://iec.ch/TC57/2011/MeterReadings#">
   <mr:MeterReading>
      <mr:Meter>
         <mr:Names>
            <mr:name>11111111</mr:name>
            <mr:NameType>
               <mr:name>EndpointID 110</mr:name>
               <mr:NameTypeAuthority>
                  <mr:name>NAME</mr:name>
               </mr:NameTypeAuthority>
            </mr:NameType>
         </mr:Names>
      </mr:Meter>
      <mr:Readings>
         <mr:timeStamp>2019-09-13T15:30:00+03:00</mr:timeStamp>
         <mr:value>11.04</mr:value>
         <mr:ReadingType ref="0.26.0.0.1.1.12.0.0.0.0.0.0.0.224.3.72.0"/>
      </mr:Readings>
      <mr:Readings>
         <mr:timeStamp>2019-09-13T15:30:00+03:00</mr:timeStamp>
         <mr:value>3.45</mr:value>
         <mr:ReadingType ref="0.26.0.0.1.1.12.0.0.0.0.0.0.0.224.3.73.0"/>
      </mr:Readings>
      <mr:UsagePoint>
         <mr:mRID>111111</mr:mRID>
      </mr:UsagePoint>
   </mr:MeterReading>
</mr:MeterReadings>
```

Figure 3: Meter Readings Data model

PMU Readings


```
{
    "device": "pmu1",
    "timestamp": "2020-05-20T10:27:57.980802+00:00",
    "readings": [
        {
            "component": "BUS1",
            "measurand": "voltmagnitude",
            "phase": "A",
            "data": 11
        },
        {
            "component": "BUS2",
            "measurand": "voltmagnitude",
            "phase": "A",
            "data": 22
        }
    ]
}
```


Figure 4: PMU Reading Data Model

Data Certification

Table 1: Data Certification model

Field	Type	Description
hashedData	Object	**Required,** the hashed data to be certified
owner	String	**Required,** the identifier of the owner of the data
timestamp	Date	**Required,** the timestamp of the certification
dataset	Object	A reference to the dataset used for the certification (e.g., time query)
transactionHash	String	Hash of the transaction as registered within the blockchain

3 Interfaces and Communication Mechanisms

3.1 MQTT Broker

The MQTT broker, based on Mosquitto, will be deployed, and exposed through a TLS/SSL secure connection. In addition, all the connection to the broker must be authenticated. The authentication mechanisms ensure the possibility to identify the publisher of the data.

For each publisher (device) a topic is assigned and restricted. The expected message data models are described in the data models chapter 2.3. XML or JSON format is supported.

Below are shown some possible communication examples:

Table 2: MQTT communication examples

Publisher	Credentials	Message Format	Message Content	Topic
Device1	Username, password	JSON	**PMU Readings**	Device-1-topic
Device2	Username, Password	XML	**Meter Readings**	Device-2-topic

3.2 API Gateway

All the REST APIs exposed by the Platone Blockchain Access Layer API Gateway implement an authentication mechanism based on Oauth2.0 [13] over HTTPS connection.

Below is shown a table with the list of APIs exposed. In the GIT repository [14] is included the Open API standard documentation.

Table 3: REST APIs list

Name	Url	Method	Parameters	Responses
Get Data	/scd/getData	GET	In body: query: JSON Object with the query	Success (200) requestedData Error (500) Error Message - *String*
Validate Data	/bap/validateData	GET	In body: data: JSON Object with data to be validated	Success (200) Success Message: *String (OK for valid data, KO for invalid data)*

				Error (500) Error Message: *String*
Get Owner	/bap/getOwner	GET	In body: data: JSON Object with data whose owner you want to verify	Success (200) Success Message: *Owner Identifier* Error (500) Error Message: *String*
Get Transactions	/bap/getTransactions	GET	In body: query: JSON Object with the query	Success (200) blockchainTransactions Error (500) Error Message - *String*

4 Languages, Technologies and External Tools

Table 4: Platone BAL languages, technologies and tools

Layer/Component	Languages	Technologies/Framework	External Tools
UI Layer	Javascript HTML5 CSS/SCSS	Docker Vue.js	Nginx
Service Layer	Javascript	Docker NodeJs ExpressJs	
Data Layer	Javascript	Docker NodeJs	Cassandra MongoDB
Integration Layer	Javascript	Docker NodeJs REST APIs	Mosquitto (MQTT) Express Gateway
Communication Layer	Javascript	Docker REST APIs NodeJs	Apache Kafka Mosquitto (MQTT) Express Gateway
Blockchain Infrastructure	Solidity	Docker Truffle Web3	Ethereum Blockchain Nodes

5 Deployment and Availability

5.1 Deployment

The deployment process foresees using Docker containers. The use of Docker ensures not only an easy deployment process and total portability of the solution, but also a high level of scalability of the released applications.

Hardware

Operating System: Linux Host

Ram: > 4GB

Disco: > 100GB

Software

Docker > 18.06.1-ce

DB Container

```
$ docker run -d --name db <your-volume-path>:/data/db -p 27017:27017 mongo:latest
```

SCD Container

```
$ cd app/cassandra #location of docker-compose.yml
$ docker-compose up -d
```

Mosquitto Container

```
$ cd app/mosquitto #location of docker-compose.yml
$ docker-compose up -d
```

BAP Container

```
$ cd app #location of DockerFile
$ docker build -t platone-bap:1.0
$ docker run -p 8082:8082 -e DATABASE_URL=<your-db-url> -e SCD_URL=<your-scd-url> -e
MQTT_URL=<your-mqtt-url> -d platone-bap:1.0
```

API Gateway Container

```
$ cd api #location of DockerFile
$ docker build -t platone-bal-api:1.0
$ docker run -p 3000:3000 -e DATABASE_URL=<your-db-url> -e SCD_URL=<your-scd-url> -d
platone-bal-api:1.0
```

BAP UI Container

```
$ cd client #location of DockerFile
$ export API_URL=<your-api-url> #URL of API Gateway
$ docker build -t platone-bal-ui:1.0
```

```
$ docker run -p 80:80 -p 443:443 -d platone-bal-ui:1.0
```

5.2 Availability

The source code and the DockerFiles necessary for the deployment are available in the RWTH GIT repository. ENG also provides a demo version, hosted in its cloud environment located at Pont-Saint-Martin (Italy).

Software REPO

GitLab-> https://git.rwth-aachen.de/acs/public/deliverables/platone

Demo Version

API Gateway -> http:// platone.eng.it:8082/api

MQTT Broker -> mqtts://platone.eng.it:8883

6 Conclusion

The work done at this stage provides the first prototype of the Platone Blockchain Access Layer that enables a fully data integration chain, including data collection, data quality and data provisioning.

The Integration Layer ensures an easy, standard, and secure data gathering. The Platone Blockchain Access Platform include standard modelling (i.e., CIM IEC 61968-9) and blockchain-based certification and validation. The Platone Shared Customer Database, together with the Communication Layer, offers all the involved energy stakeholders, including external platform and services, a secure and transparent way to access all the data collected and certified.

The first release addresses the integration the Platone Blockchain Access Layer with the physical infrastructure of the German and Greek demo sites and for the integration Platone DSO Technical Platform. More in detail, it implements the complete integration of smart meters using CIM data modelling and the PMUs (including all the features mentioned in Ch. 2.2).

A detailed description for installation and configuration of platform components is provided to ensure the usability. A demonstrative version of the entire Platone Blockchain Access Layer is available within ENG cloud infrastructure.

7 List of Tables

8 List of Figures

9 List of References

[1] "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.

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

[3] *Grant Agreement No. 864300 – Platone.*

[4] *Platone_D2.1_PlatOne Platform requirements and reference architecture (v1).*

[5] "web3.js - Ethereum JavaScript API - Documentation," [Online]. Available: https://web3js.readthedocs.io/en/v1.3.0/.

[6] "Apache Cassandra," [Online]. Available: https://cassandra.apache.org/.

[7] "Mosquitto," [Online]. Available: https://mosquitto.org/.

[8] "Express JS Framework," [Online]. Available: https://expressjs.com/.

[9] "Express Gateway," [Online]. Available: https://www.express-gateway.io/.

[10] "Apache Kafka," [Online]. Available: https://kafka.apache.org/.

[11] "SHA-3 hash function," [Online]. Available: https://csrc.nist.gov/projects/hash-functions/sha-3-project.

[12] Y. K. Peker, X. Rodriguez, J. Ericsson, S. J. Lee and A. J. Perez, "A Cost Analysis of Internet of Things Sensor Data Storage on Blockchain via Smart Contracts," *Electronics,* vol. 9, no. 2, p. 244, 2020.

[13] "Oauth 2.0," [Online]. Available: https://oauth.net/2/.

[14] "Platone GIT Repository," [Online]. Available: https://git.rwth-aachen.de/acs/public/deliverables/platone.

10 List of Abbreviations

Abbreviation	Term
API	Application Programming Interface
BAL	Blockchain Access Layer
BAP	Blockchain Access Platform
CIM	Common Information Model
DB	Database
DSO	Distribution System Operator
ENG	Engineering Ingegneria Informatica S.p.a.
HTTP	Hypertext Transfer Protocol
IEC	International Electrotechnical Commission
JSON	JavaScript Object Notation
MQTT	Message Queuing Telemetry Transport
PMU	Phasor Measurement Unit
REST	REpresentational State Transfer
SCD	Shared Customer Database
SHA	Secure Hash Algorithms
SSL	Secure Sockets Layer
TLS	Transport Layer Security
TSO	Transmission System Operator
UI	User Interface
XML	Extensible Markup Language