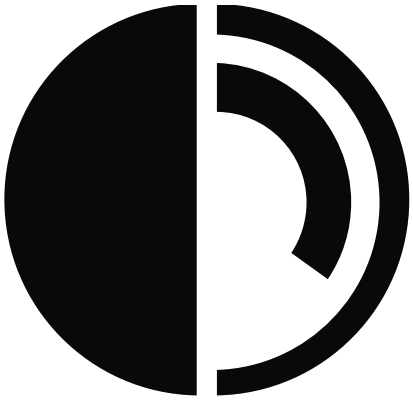


BD4OPEM

Big Data for Open innovation Energy Marketplace

Deliverable 2.3

Open innovation Project architecture description

Document information

Deliverable nr	D2.3
Deliverable Name	Open innovation Project architecture description
Version	1.0
Release date	31/12/2020
Dissemination level	Public
Status	Submitted
Author	Lourdes Gallego - ATOS

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under **Grant Agreement No 872525**.

Document history:

Version	Date of issue	Content and changes	Edited by
0.1	26/10/2020	Table of Contents	Lourdes Gallego
0.12	31/10/2020	First draft	Lourdes Gallego
0.13	14/11/202	Inputs in process and implementation views	Lourdes Gallego
0.2	18/11/2020	Second draft	Lourdes Gallego, Javier Valiño, Nicolas Figueroa Viteri
0.21	27/11/2020	Draft for 1st Review	Lourdes Gallego, Javier Valiño, Nicolas Figueroa Viteri
0.3	04/12/2020	1st Peer Review	Amit Eytan, Valerio Gentile, Dušan Gabrijelčič
0.4	07/12/2020	Inputs in implementation views	Valerio Gentile, Pau Lloret, Luc Richaud, Albert Gonzalez Navarro
0.41	10/12/2020	Draft for 2nd Review	Lourdes Gallego
0.5	11/12/2020	2nd Peer Review	Amit Eytan, Valerio Gentile, Dušan Gabrijelčič, Isidoros Kokos, Ioanna Katidioti, , Massimiliano Garella, Ross Little
0.6	15/12/2020	Final quality check and release	Lourdes Gallego
1.0	18/12/2020	Final version	Lourdes Gallego

Peer reviewed by:

Partner	Reviewer
WEP	Amit Eytan, Valerio Gentile

JSI	Dušan Gabrijelčič

Deliverable beneficiaries:

WP / task
WP2 - T2.4, T2.5
WP3 - T3.3, T3.4
WP4 - T4.1, T4.2, T4.3, T4.4, T4.5, T4.6
WP5 - T5.2
WP6
WP7

Table of contents

List of figures

List of tables

Abbreviations and Acronyms

Acronym	Description
API	Application Programming Interface
ASM	Analytics Service Management
B2B2C	Business-to-Business-to-Consumer
BRP	Balance Responsible Party
BUC	Business Use Case
CEC	Citizen Energy Community
CIM	Common Information Model
CRU	Create, Read, Update
CRUD	Create, Read, Update and Delete
DAE	Distributed Analytics Engine
DB	Database
DER	Distributed Energy Resource
DID	Decentralised Identifier
DIDCOMM	DIDComm Messaging Protocol
DoA	Description of Action
DP	Data Provider
DRES	Dynamic Range Encoding Scheme
DSL	Digital Subscriber Line
DSO	Distribution System Operator
DU	Data User
EC	European Commission
ETL	Extract, Transform and Load
EV	Electric Vehicle
HTTPS	Hyper Text Transfer Protocol Secure
HW	Hardware
ICT	Information and Communications Technology
IEC	International Electrotechnical Commission
JWE	JSON Web Encryption
JWS	JSON Web Signature

JWT	JSON Web Token
KPI	Key Performance Indicator
OOP	Object Oriented Programming
P2P	Peer to Peer
PaaS	Platform as a Service
PV	PhotoVoltaic
R	Read
R&D	Research and Development
SC	Smart Contract
SCADA	Supervisory Control And Data Acquisition
SFTP	Secure File Transfer Protocol
SGAM	Smart Grid Architectural Model
SIEM	Security Information and Event Management
SM	Smart Meter
SP	Service Provider
SSI	Self-Sovereign Identity
SU	Service User
SUC	System Use Case
SW	Software
T	Task
TBD	To Be Defined
TLS	Transport Layer Security
ToC	Table of Content
UC	Use Cases
UML	Unified Modelling Language
VCs	Verifiable Credentials
WP	Work Package

1 Executive summary

This deliverable D2.3 presents the first iteration of BD4OPEM platform architecture. It provides the conceptual architecture, which is focused on the coexistence of both ICT and energy domains. The report explores the methodological approaches available, selecting the 4+1 Architectural View Model [1] as the most suitable approach for BD4OPEM. This enables addressing different stakeholders' concerns and establishing the baseline for supporting future tasks, as it is a reusable and versatile model. Moreover, it provides a complete mapping between the 4+1 architectural model and SGAM reference architecture [2].

Based on the selected architectural design, this first release focuses on three views: the *logical view*, corresponding to the logical connection between the different layers; the *implementation view* covering the internal and external specifications and functionalities with their respective UML diagrams; and, finally, the *process view* presenting the dynamic behaviour of the BD4OPEM system. The behaviour is presented through UML sequence diagrams, based on the use cases submitted in D2.1 ("Concept design and use-cases")[3].

The result of this deliverable is a full description of BD4OPEM platform built in five main layers: Physical Layer, Data Layer, Analytics Layer, Marketplace Layer and Security, Privacy and Blockchain Layer.

- **Physical Layer**. It is the lowest layer of BD4OPEM platform designed for any field device that will be tested only in the field devices of the five pilot sites envisioned in the project, as well as Open Data Sources. This layer acts as the basic interface facilitating the data acquisition, for its later use in upper layers.
- **Data Layer**, aiming at serving as an interface for field devices in all five pilot sites. It provides a unified interface for delivering the data to the services, handles the data persistence, and provides a management framework for a big data volume.
- The **Analytics Layer** is the core of BD4OPEM Platform. It contains innovative tools, both analytic and added value services for the energy market. Topology modelling, Fraud detection, Flexibility forecast, P2P trading or Grid disturbance simulations can be cited as service examples. They will be offered as services for the upper layer, the Marketplace.
- **Marketplace Laye**r. It is the interface between stakeholders and BD4OPEM offering of innovative solutions and services; since it enables them to support business decisions and acts as a bridge for transactions.
- Lastly, **Security, Privacy and Blockchain Layer** is the cross-cutting layer of the platform. It provides the security controls to maintain the Confidentiality, Integrity and Availability of the personal and strategic information processed and shared inside BD4OPEM.

Furthermore, the methodology and users of BD4OPEM platform are presented.

The aim of this deliverable is to serve as the reference architecture for BD4OPEM technical tasks, where more detailed descriptions and implementations will be carried out.

2 Introduction

2.1 Scope and Objectives

This document aims to serve as a baseline to provide information needed for the definition of BD4OPEM platform, using as a starting point the first draft of architecture introduced at proposal level in BD4OPEM DoA. Therefore, this report details the logical design of the conceptual architecture, providing the specifications and requirement to be used in future development and implementation tasks, as part of WP3, WP4, WP5, WP6 and WP7 activities.

To this end, a study on available methodological approaches for architecture elicitation is conducted. The result of this study, as outlined in section 3, is the selection of Kruchten '4+1 view' architectural model[1]. Using this reference, these 4+1 views are detailed, covering the functional components and requirements (*logical view)*, the definition, functionality, interoperability aspects, dependencies and relationships (*process view*), the internal and external specifications and functionalities (*implementation view*), the customized architectural perspective for each pilot sites (*physical view*) and the requirements derived from all previous views (*use case view*).

More precisely, this deliverable covers the logical, implementation and process views which concentrate the strategic issues that manage not only to improve and refine the concept design but also to provide the basis for an open, modular and scalable platform. Both the physical (deployment) view and use case view will be covered thoroughly in the next deliverable D2.4 (update on BD4OPEM architecture), due on M42.

2.2 Relation to other Tasks and Deliverables

This deliverable, D2.3, uses the outputs of Task 2.1, addressed in its D2.1 titled "Concept design and use-cases"[3], since it establishes the definition of the business and system use cases, from which *physical view* and *use cases view* will build upon. In particular, this document extends the descriptions of the identified actors involved in the platform, so as to cover their interactions (actor-platform), the way of accessing and the functionalities they will be enabled with inside BD4OPEM platform.

Additionally, this deliverable will serve as a reference guide for further iterations in the architecture which will be submitted as part of T2.4 "Power system architecture adaptation to pilot sites Legacy and new roadmap" and T2.5 "BD4OPEM Architecture review" activities. In addition, this report will serve as input for the following WPs, with the intention to provide functional, technical and communication aspects:

- WP3 "Big Data Integration and Management"
- WP4 "Big Data Analytics and Business Intelligence for Energy System"
- WP5 "Big Data Security and Cybersecurity"
- WP6 "Integrated BD4OPEM Marketplace platform"
- WP7 "Pilot"

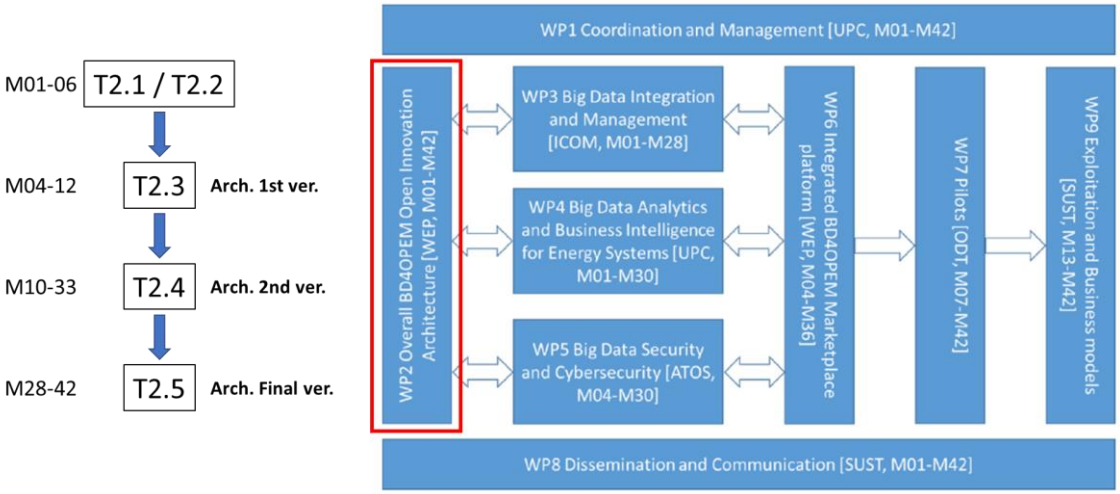


Figure 1. D2.3 relation to other Tasks and Deliverables of BD4OPEM

2.3 Structure

This document is broken down in sections as described below:

- **Section 2** presents the scope and objectives of this deliverable and its relationship with other tasks and deliverables of the project
- **Section 3** is devoted to presenting the selection of "4+1 Architectural View Model" as the best option for BD4OPEM, including the description of every view and, furthermore, describes the methodology adopted to carry out each view description
- **Section 4** covers the logical view, detailing the structure and functionalities of BD4OPEM platform. Furthermore, it includes one subsection identifying actors involved in BD4OPEM platform, detailing their interactions and functionalities
- **Section 5** describes the implementation view, showing how BD4OPEM platform is divided into components, their dependencies and connections.
- **Section 6** introduces the process view, illustrating the processes of the BD4OPEM platform and the way they are communicated, based on the project's use cases.
- **Section 7** offers an overview of the deployment view.
- **Section 8** covers the "+1" or use case view, detailing the needed information.
- **Section 9** presents the relevant conclusions drawn from the different views throughout this deliverable

3 Methodology for BD4OPEM architecture description

This section is devoted to describing the methodology followed by the BD4OPEM in order to define its architectural design approach. Initially, the relevant architectural models are exposed to select the most appropriate model for BD4OPEM. Subsequently, the reasons for choosing "4 + 1" as the best model are presented, and the views of which it consists are described. Finally, the methodology followed to define each view is expounded.

3.1 Architectural model selection

As described on the deliverable D2.1"Concept design and use-cases"[3], the Smart Grid Architectural Model (SGAM)[2] has been used as a reference for the definition of the Use Cases (UC) of the project. The SGAM defines a framework in order to design the use cases of a smart grid development with an architectural approach, meant to be used for current and/or future implementation of the electrical grid. It gives a common modelling methodology and architecture to unify the perception of the smart grid viewpoints, into five layers.

Although BD4OPEM platform is able to map into SGAM; it offers tools, based on Big data techniques as the development of an Analytic Toolbox for enabling efficient business processes in the energy sector, which are complex by nature. These innovative tools, together with the wide variety of external systems it includes (hardware devices, external APIs, interfacing modules, etc. …) require interaction with more than one layer for the development of the platform.

For this reason, it has been decided to follow the ISO/IEC/IEEE/42010[4] standard, as a reference guideline in order to choose the most appropriate architectural model. Some examples of architectural models that follow the standards can be find below:

- **Documenting Software Architectures: views and beyond** [5]: approach with three different views (Module, Component/Connector and Allocation) and their rules to understand the information to be provided for each one of them
- **The Process of Software Architecting** [6]: comprises of 9 viewpoints named as Requirements, Functional, Deployment, Validation, Application, Infrastructure, Systems Management, Availability, Performance and Security
- **Defining execution viewpoints for a large and complex software-intensive system** [7]: oriented to software, this model describes the templates for 4 execution views, named Execution Profiles, Execution Deployment, Execution Concurrency and Resource Usage
- **The '4+1' view model of architecture** [1]: describes 4 views: Logical, Development, Process and Physical. The resulting views are integrated via a fifth view named Scenarios.
- **Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives** [8]: This model encompasses 6 viewpoints (Functional, Information, Concurrency, Development, Deployment and Operational) that can be completed with optional viewpoints (Security, Performance and Scalability).

Considering the need to define an architecture that allows addressing the concerns of the different project participants, managing the complexity of the system, as well as, testing the defined use cases in the pilot sites; the "4+1 view model" is postulated such as the best choice for the BD4OPEM project.

The 4+1 view model architecture is a SW-oriented, framed in the Energy R&D framework that offers the three main advantages of a software system: (1) provides the communication among the stakeholders involved in the project, (2) manifests design decisions early (e.g. define implementation constraints, enable iterations) and (3) establishes a reusable and transferable model. Moreover, it offers a complete architectural definition allocating few resources in its definition and implementation per pilot sites.

3.2 The 4+1 View Model of Architecture

As aforementioned, the 4+1 View Model Architecture [1] counts with four views, and one added, as shown in Figure 2 Philippe Kruchten, the author of this model, describes the concept of 'view' as the representation of the whole software system from a certain perspective; but he does not detail the term 'point of view'. The latter has been included because of the increasing development of software carrying out under the OOP (Object-oriented programming) paradigm, where a set of rules (or norms) are established for making and understanding the views.

Below, the entire architecture of the system is expounded in-depth from each view:

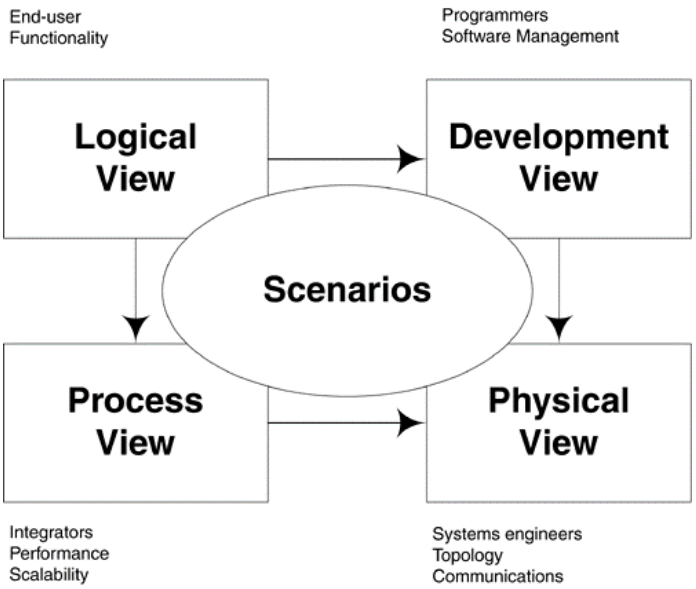


Figure 2. The "4+1" architectural view model

- **Logical View or structural view:** This view represents the functionality that the system will provide to end-users. In other words, it represents the static approach of what the system should do and the functions and services it offers. This view is addressed in depth in section 4.
- **Implementation view or development view:** This view shows the system from the perspective of a programmer and deals with software management. It will show how the software system is divided into components and will define their technical requirements and the dependencies between the components. This view is introduced in section 5.
- **Process view or behavioural view:** This view depicts the processes in the system and the way in which these processes communicate; that is, it deals with the dynamic aspect of the system, represented from the perspective of a systems integrator, the step-by-step business and operational workflow of the components that make up the system. A representation of this view can be seen in section 6.
- **Deployment view or Physical view:** This view introduces from the perspective of a systems engineer all the physical components of the system as well as the physical connections between those components that make up the solution (including services). It maps the hardware devices with the components of the system.
- **"+1" Scenarios view or use case view:** This view will be represented by the software use cases and will have the function of joining and relating the other 4 views. This means that from a use case it can be seen how the other 4 views are linked. The scenario view allows for a traceability of components, classes, equipment, packages, etc., needed to carry out each use case.

3.3 Steps of elaboration

This deliverable uses both, the architecture conceived from DoA and the deliverable D2.1[3] as the main information sources for starting with the architecture development. On the one hand, the D2.1 provides the description of the actors that are involved in the platform, being the first step in order to establish their relationships and functionalities with the platform. This series of connections will enable to answer questions, detailed in section 4.6, which shall affect in the definition of the architecture.

On the other hand, the D2.1, which addresses all the use cases, serves as the frame of reference since it will determine the deployment of the architecture. In fact, the use cases set up, to some extent, the interactions between the layers and modules of the layers, besides define the dynamic behaviour of the platform reflected in the process view.

Beside the previous information, the following actions have been carried out for the consolidated definition of BD4OPEM architecture:

- Arranging conferences calls: one to one with the key partners besides the whole consortium. These calls have the purpose of addressing each view,

explaining its definition and objectives in addition to answering diverse key questions for progressing and improving the proposed view

- Iterative improvements of delivered templates. The methodology for defining the architecture is mostly based on one template per view to fill in by the partner responsible.
- Close collaboration with the pilots since the definition of BD4OPEM platform affect the physical and use cases views which will be tested in each pilot site.

4 BD4OPEM reference Architecture. Logical View

This section presents the logical view, a static approach towards the system performance, including functions and services offered to end-users. The main purpose of this chapter aims at describing the functionalities provided by BD4OPEM architecture by detailing the layers and modules that conform it.

Figure 3 presents a consolidated draft of the architecture, conceived as modular and already presented at the BD4OPEM proposal/DoA. This draft has served as a reference point for the final architecture to be outlined in this chapter.

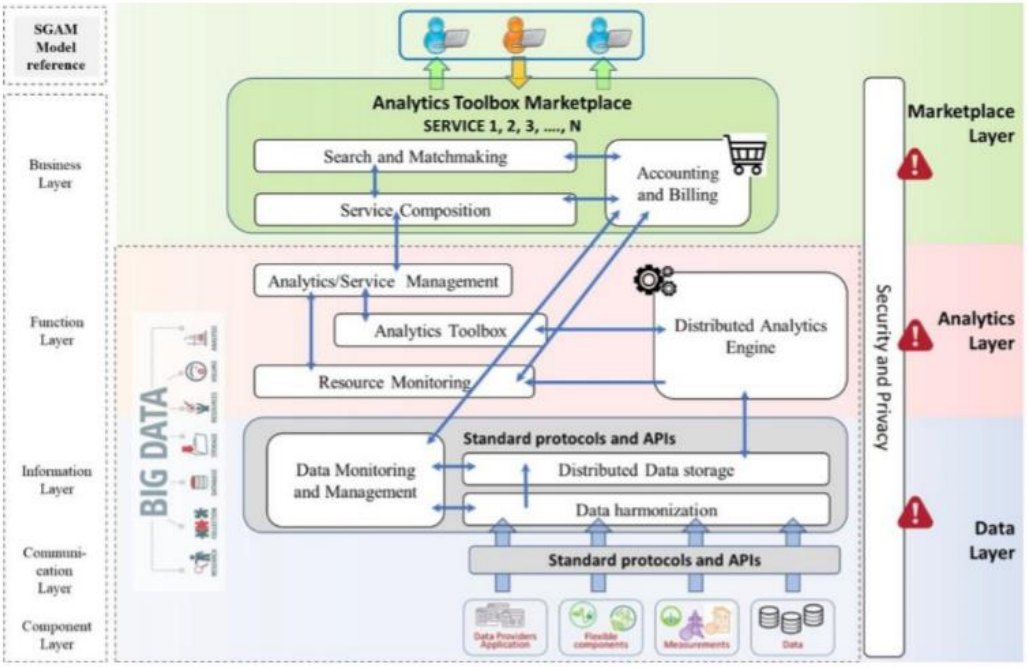


Figure 3. First BD4OPEM architecture presented at proposal

Several iterations were carried out, bringing major changes and finally enabling BD4OPEM to unlock the inherent innovation associated with the use of Big Data techniques, and linking them to a Marketplace. This first design set up included a cross reference to SGAM model, keeping that as a constant. In this second version, the "4+1" model, improved the conception towards an extended ICT-Big Data approach, as presented in Figure 4. BD4OPEM logical architecture and mapping to SGAM Model interoperability layers

The seamless intersection of both approaches allows for a comprehensive mapping and provides a complete technical and interoperability understanding, through the combination of all "views" and "points of views" with the smart grids; as well as, being able to capture the defined use cases and all supported by standards.

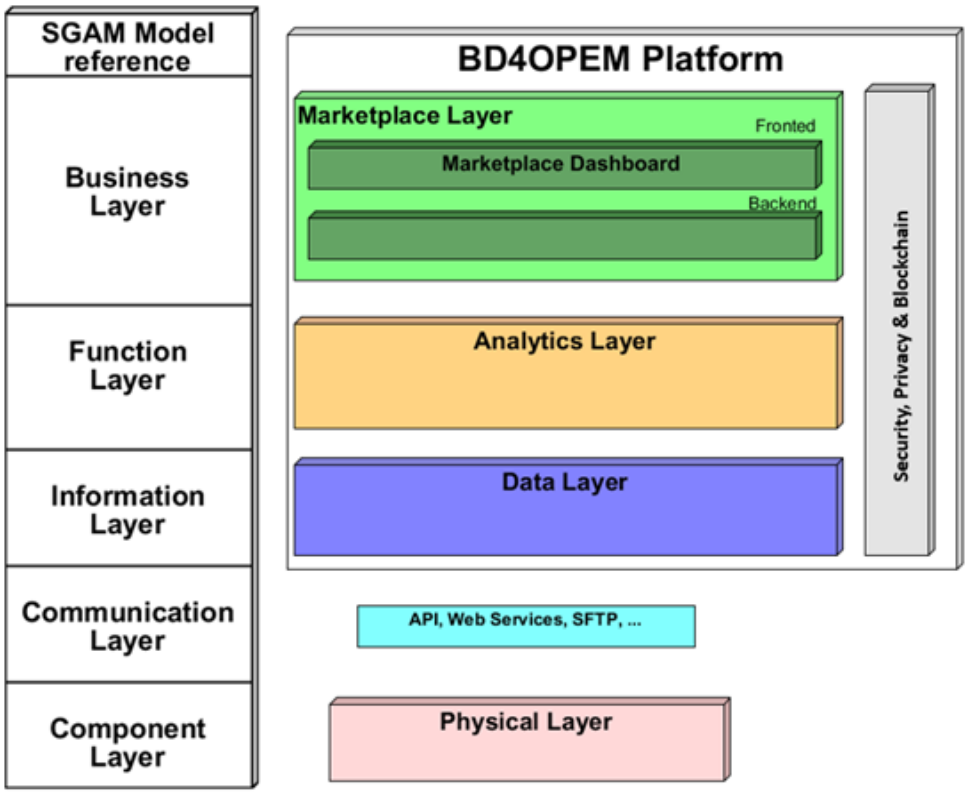


Figure 4. BD4OPEM logical architecture and mapping to SGAM Model interoperability layers

- SGAM *Interoperability layers* match the logical view as follows:
  - *Component layer*. It describes the devices and components and their physical distribution. It is equivalent to BD4OPEM physical layer, made up of devices (Smart Meters, EV, SCADAs, ...)
  - *Communication layer*. This layer presents the communication protocols and mechanisms for information exchange between components. It matches the communication protocols presented in BD4OPEM project (API, Web Services, SFTP, ...);
  - *Information layer*. This is represented in BD4OPEM by the data layer, as it includes the information that is being used and exchanged between components, functions and services.
  - *Functional layer*. It corresponds to BD4OPEM analytics layer given that it provides features and services and their relationships independently to actors, implementations, systems and components (devices).
  - Finally, the *business layer* is mapped with BD4OPEM Marketplace layer. This layer represents the business vision of a SG and offers markets, business models, products and services, etc.
- The zones are described as the different hierarchical levels of management systems for energy, based on the concepts of aggregation and functional separation. BD4OPEM project platform envisions on enabling services in

various zones; from field up to market, so it can be considered as zone independent.

- Domains. BD4OPEM covers the domains of distribution, DER and Customer Premises

To the logical view details, the different parts and modules included inside together with the interaction among them.

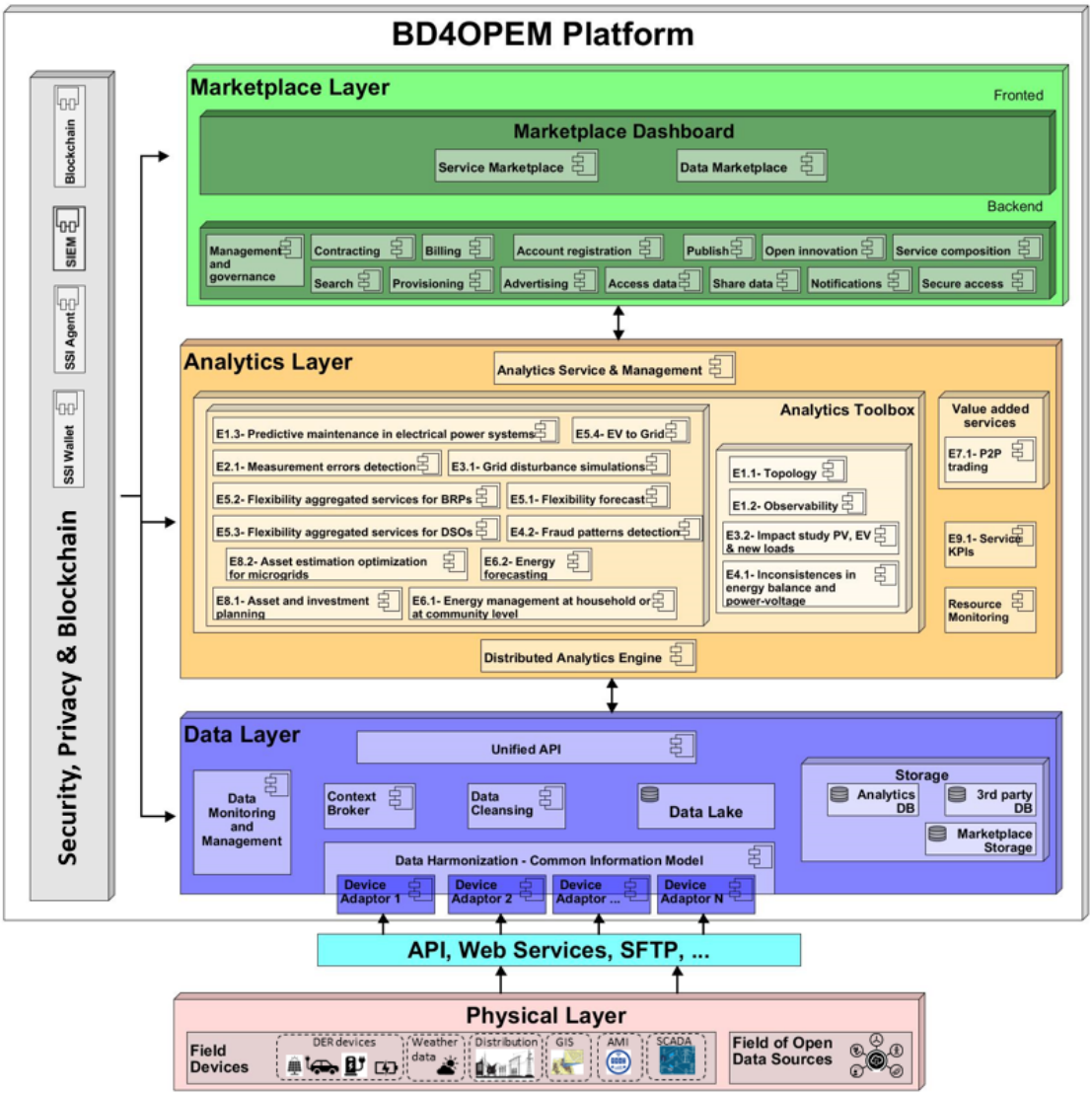


Figure 5. BD4OPEM platform architecture

BD4OPEM platform is divided into five layers, as illustrated in Figure 5, distributed in:

- Four central layers: (bottom-up) Physical Layer, Data Layer, Analytics Layer and Marketplace Layer, with these last three being the most relevant ones of the platform; and
- One holistic layer: Security, Privacy & Blockchain Layer.

Layer	Modules
Physical Layer	Field Devices
	Open Data Sources
Data Layer	Data Management and Monitoring
	Device Adaptors
	Data Harmonization–Common Information Model
	Context Broker
	Data Cleansing
	Storage
	Data Lake
	Unified API
Analytics Layer	Analytics Toolbox
	E1.1-Topology
	E1.2-Observability
	E1.3- Predictive maintenance in electrical power systems
	E2.1 Measurement errors detection
	E3.1- Grid disturbance simulations
	E3.2- Impact study PV, EV & new loads
	E4.1- Inconsistencies in energy balance and power-voltage
	E4.2- Fraud patterns detection
	E5.1- Flexibility forecast
	E5.2- Flexibility aggregated services for BRPs
	E5.3- Flexibility aggregated services for DSOs
	E5.4- EV to Grid
	E6.1- Energy management at household or at community level
	E6.2- Energy forecasting
	E8.1- Asset and investment planning
	E8.2- Asset estimation optimization for microgrids
	Distributed Analytics Engine

	Analytics Service & Management
	Resource monitoring
	E9.1-Service KPIs
	Value added services
	E7.1- P2P trading
Marketplace Layer	Contracting
	Search
	Provisioning
	Billing
	Management and governance
	Advertising
	Publish
	Open innovation
	Secure access
	Share data
	Access data
	Service composition
	Notifications
	Account registration
Security, Privacy and Blockchain Layer	SSI Wallet
	Blockchain
	SSI Agent
	SIEM

Table 1. List of architecture modules

The following template has been used with the purpose of providing a full description, of the different layers:

Layer Name	Name of the layer
Description	General characteristics and technologies used

Main functionalities	Description of how the layer is expected to operate. Identification of potentially needed data to be used and/or stored
Dependencies to another layer	Identification of dependencies with other layer(s), in terms of data, operation or functionalities

Table 2. Logical view template

4.1 Physical Layer – Field devices

Physical Layer is the lowest layer of BD4OPEM platform, composed by all field devices envisioned in the five pilot sites of the project, as well as Open Data Sources. The former field devices might be modular electronic units, SW and HW components, that enable the integration with the real world, since they offer the endpoint information allowing to be accessible and actionable at any time. Whereas the latter, essentially is data that can be freely used, reused and redistributed by anyone, normally though APIs. In other words, this layer acts as the basic interface, facilitating the data acquisition, that measures or controls one physical characteristic of the physical world, for its later use in upper layers.

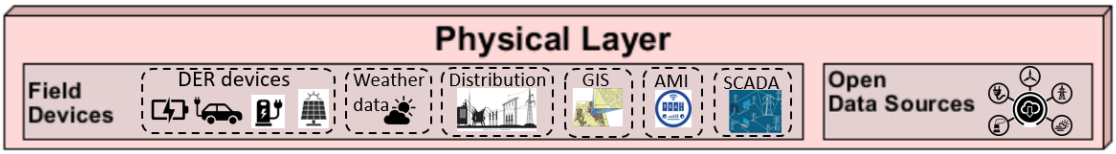


Figure 6. Physical Layer - Field Devices

The two main functionalities of field devices are:

- Provide data to BD4OPEM platform, both in real-time (e.g. SCADA) and in batch/historical data (e.g. active/reactive power of Smart Meters) by converting bits to signals which will be understood by the platform
- Enable a communication protocol with two-fold objectives: control the two-way communication in available devices; and manage the configuration in order to acquire the data in a determinate frequency. These settings will be specified in the context of piloting activities (WP7).

This layer is only related to Data Layer, providing the data needed for BD4OPEM platform.

4.2 Security, Privacy and Blockchain Layer

Security, Privacy and Blockchain Layer is one of the two vertical layers which provides the implementation modules and guidelines to support the cloud deployment of the BD4OPEM platform. It provides the security controls to maintain the Confidentiality, Integrity and Availability of the personal and strategic information processed and shared over BD4OPEM.

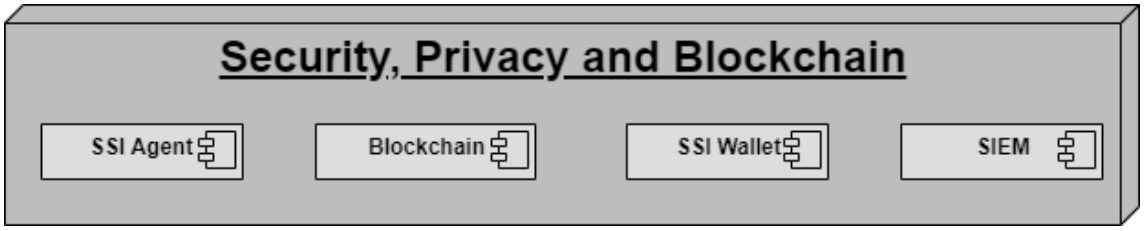


Figure 7. Security, Privacy and Blockchain Layer

This layer supplies four main functionalities:

- Secure access and authorisation of all users and external System accessing the platform as well as service users, SP/DP administrators, hybrid deployed SP Systems, DP data upload Systems etc.
- Layered security architecture cloud solution based on Azure Technology.
- Blockchain solution to support the Smart Contract based Value Added Services and the Self Sovereign Identity for Secure access for all stakeholders in the ecosystem (Service Users, Data Providers, Service Providers, Data Users and Marketplace Operator)
- SIEM solution to capture security events, analyze logs and eventually raise security alerts.

As transversal layer, it is related with Data Layer, Analytics Layer and Marketplace Layer.

4.3 Data Layer

The Data Layer of the BD4OPEM solution will offer a distributed storage solution able to store a variety of data types, handle data ingestion at different velocities, as well as manage significant volume of data. The data layer will monitor and maintain the quality of data through its lifecycle and provide standard means of accessibility for the upper layers of the architecture, as well as manage integration with external data sources through standard protocols and interfaces.

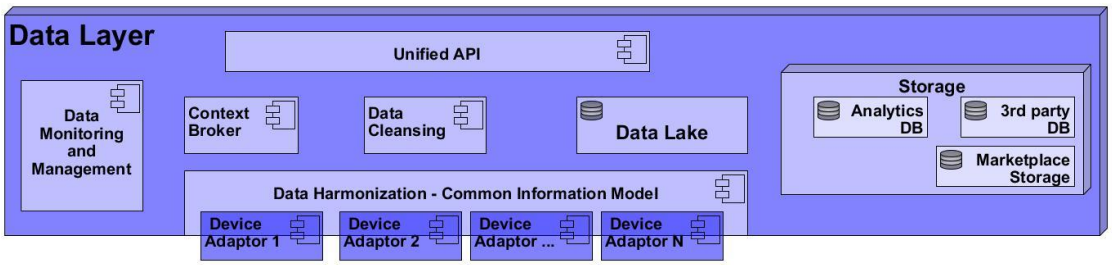


Figure 8. Data Layer

This layer provides the following functionalities, where mainly it stands out by two-fold purposes: on the one hand, it serves such as an interface with the physical layer, enabling its ingestion and storage; and on the other hand, it encapsulates the access to previous harmonized and quality data.

- *Distributed Storage*: A scalable storage solution of data for facilitating data sharing and building analytic services shall be provided. The solution should enable the ingestion and storage of various data types.
- *Data Mediation*: The Data Layer should act as a facilitator of data exchanges with external APIs for the services built on the Analytics layer, offering transparent integration with external data sources through standard interfaces.
- *Data Access*: Accessibility for the upper layers of the solution shall be provided. Data Access could take the form of queries (in some DSL) or APIs. A mechanism for managing accessibility of different requests to the data will be provided, following an established permission policy.
- *Data Harmonization*: The solution will offer data transformation based on a harmonization model enabling the access of data through standard interfaces.
- *Data Governance*: A mechanism should enable the management of the lifecycle of the data entering the platform, storing metadata, monitoring the quality of data, its availability as well as managing the end-of-life of the data.
- *Data Quality Management*: Data quality should be managed through data curation processes. Such processes could be part of the ingestion process of data (as a pre-process of data storage) or ad-hoc operations performed on stored data. Different data curation processes shall be supported and parameterized by the relevant responsible party.

This layer is related to the Analytics and Marketplace layers providing, respectively, the necessary data for the development big data services and data access; as well as acting as the storage of all the modules of the platform.

4.4 Analytics Layer

This layer represents the "core" of BD4OPEM Platform; since, as its name denotes, it contains the development and analytics of all the services that will be offered in the next layer, the Marketplace Layer, to different users.

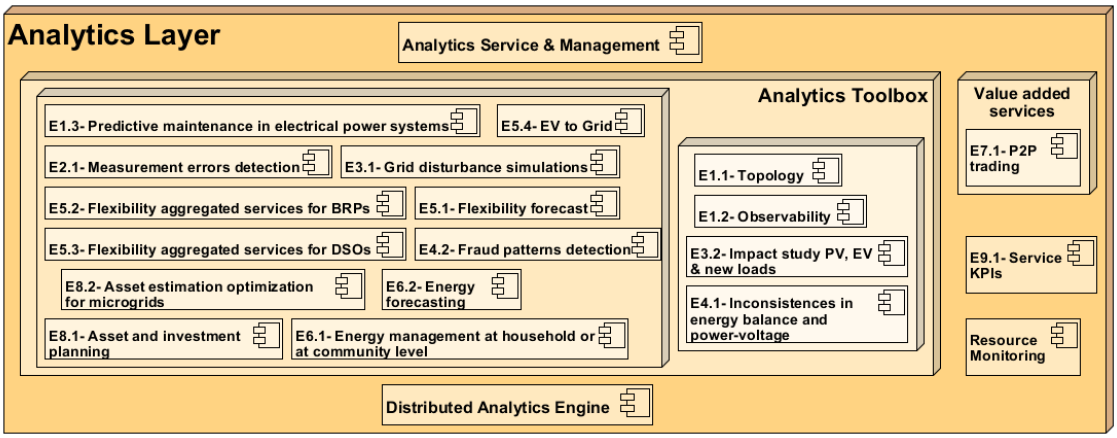


Figure 9. Analytics Layer

In particular, the Analytics layer is the place where analytics tools, such as Energy forecast or Fraud pattern detection, use the harmonized/cleansed data from the Data layer and extract its inherent value. Finally, the Analytics tools expose the results in the Marketplace layer and/or the Data layer, with an aim of storing them for future use or new services that can use them as an input. Furthermore, this layer counts with value added services (VAS), in particular the P2P Trading, and a resource monitoring module to monitor the Analytics Toolbox performance.

Their main functionalities are listed below:

- *Distributed Analytics Engine* is a solution for hosting analytics applications in the BD4OPEM platform, providing a Platform-as-a-service (PaaS) approach.
- *Resource monitoring* is a supervision tool in charge of verifying the analytics tools performance. It includes activity logs to record when resources are used and metrics to tell how the service is performing and the resources that it is consuming.
- *E9.1 Service KPIs* coordinates and manages the calculation of service KPIs to evaluate their performance.
- *Analytics Service/ Management* provides the connection between the Analytics layer and the Marketplace Layer. It can act as a trigger, calling/activating/launching any module of analytics toolbox, as well as, processing and generating the demanded outputs.
- *Value added services* module includes the P2P Trading, as a non-analytic service that facilitates the purchase and sale of energy between two or more parties connected to the grid (homes, communities, EVs, companies, etc.) in a safe and transparent way and without intermediaries or central authority.
- *Analytics toolbox* is the central core of the analytics layer, where analysis tools are developed and included, allowing their use as energy innovation services divided into seven categories, as shown in Table 3. These tools can be integrated into the BD4OPEM platform, making use of the *Distributed Analytics Engine* or run on a third-party platform.

Service category	Service ID	Service in BD4OPEM
Operation and maintenance	E1.1	Topology
	E1.2	Observability
	E1.3	Predictive maintenance in electrical power systems
	E2.1	Measurement errors detection
	E3.1	Grid disturbance simulations
	E3.2	Impact study PV, EV & new loads
Fraud detection	E4.1	Inconsistences in energy balance and power-voltage
	E4.2	Fraud patterns detection

Flexibility and demand response	E5.1	Flexibility forecast
	E5.2	Flexibility aggregated services for BRPs
	E5.3	Flexibility aggregated services for DSOs
	E5.4	EV to Grid
Smart houses, buildings and industries	E6.1	Energy management at household or at community level
	E6.2	Energy forecasting
Trading	E7.1	P2P trading
Planning	E8.1	Asset and investment planning
	E8.2	Asset estimation optimization for microgrids
Monitoring	E9.1	Service KPIs

Table 3. List of services by categories

As it can be inferred from the previous paragraphs, the Analytics layer is directly connected to the Data layer and the Marketplace layer, exchanging data of different nature.

4.5 Marketplace Layer

The Marketplace layer is an open and modular platform in itself, offering innovative solutions and services to the energy market, relying on the data sharing and analytics functionalities of the previously defined layers; enabling stakeholders to connect to BD4OPEM environment. The functionalities offered are twofold:

- It acts as a bridge for transactions of Business to Business and/or Business to Consumer (B2B2C).
- It provides the interface among the offered services and the actors involved.

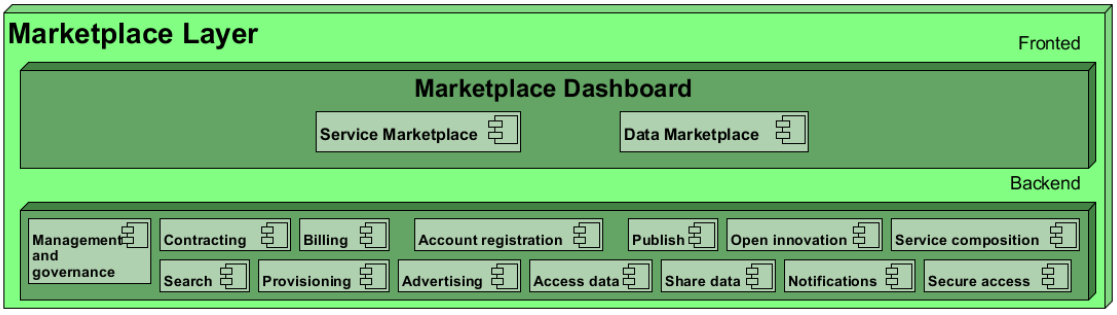


Figure 10. Marketplace Layer

The Marketplace layer contains the "client-side" functionalities (related to the Frontend) and the "server-side" one (corresponding to the Backend). The former represents the BD4OPEM marketplace as a "virtual store" for end users, where two types of services will be offered:

- *Business-energy services (Service Marketplace)*: those are analytic services (e.g. flexibility forecast, predictive maintenance in electrical power systems, fraud patterns detection, …), added value services (P2P trading), and energy grid services (such as flexibility Aggregated BRPs or DSOs, as detailed in section 5.3).
- *Data services (Data Marketplace)*: they are the ones enabling to ingest/store, share and access data, with the needed confidentiality and integrity levels and in compliance with the regulations and directives. Indirectly, these services support the development of the previously described business-energy services.

Therefore, the Marketplace layer integrates all the capabilities and tools that allow the interaction between the preceding services and the stakeholders. These capabilities comprise contracting, searching, provisioning, billing, advertising, publishing, Open innovation (proposal of new services), authentication and service composition, which will be addressed in detail in section 5.4.

The Marketplace layer relates to both the Data Layer, enabling to access, exchange and storage data; and the Analytics Layer from where developed services and tools are used.

4.6 End Users and their high level interaction with BD4OPEM

This section addresses three key objectives: (1) identify the users of BD4OPEM Marketplace, (2) define their supported functionalities and (3) describe the way they interact with the platform. Following that line of thought, below questions were proposed as the methodology in order to cover the allowed functionalities and access procedures.

- Which actors will interact with the platform?
- Do all the actors have to register previously to interact with the platform?
- Which level of registration do the actors have?
- Which functionalities are enabled for each actor?

As outlined before, the Marketplace is the "BD4OPEM virtual store" that acts as a bridge for transactions linking the innovative energy services and data among the actors involved. These marketplace actors were identified in D2.1[3], namely marketplace operator, service provider, service user, data owner, data provider, data user and data consumer.

Once their functionalities have been further analysed using the newly defined marketplace, these seven actors can be represented by the following five core marketplace business actors:

- *Service provider (SP)*: an entity with the objective of developing and selling added value services. The services can be provided using several options, such as licensing or SaaS. As a fine grain distinction, subcategories for service providers can be, for instance, Analytic service providers and Analytic tool/App providers. Both parties analyse data and provide results of interest for other service providers or end users. The former's revenue model is based on monetizing provisioned analytical services. In the case of tool/App providers, they profit from provisioning a tool or app.
- *Service user (SU)*: an entity utilizing the added value services that service providers offer in the marketplace. The actor benefits directly from using those services, for example, it can improve its own operations, get deeper insights into their managed energy system, optimize investments or lower its costs.
- *Data provider (DP)*: an entity creating data and/or executing control over it. The actor makes data available for a Data consumer. It profits out of data provision (direct monetization of data) or provides data to comply with policies (e.g. DSO providing access to Smart Meter data).
- *Data User (DU)*: an entity that has the right to use the data of a Data provider as specified by a contract policy. The actor consumes the data and/or builds added value data services.
- *Marketplace operator*: a party managing the marketplace and facilitating service provisioning process and data sharing. It facilitates service publishing and provisioning process, can charge for its services or utilize other means of generating a revenue.

Table 4**Table 4.** Supported functionalities of BD4OPEM users

below presents the supported functionalities per user and service type:

	Business- energy services	Data Marketplace
Marketplace operator	Manages/Oversees the registration and login process, defines users' functionalities according to their role and maintenance	
Service provider	Leverages the marketplace for provision of its services and for the data layer, (Optional) Utilizes the analytics layer (PaaS)	Accesses/Buys data for training his algorithms
Service user	Utilizes the services of the BD4OPEM platform solution, taking advantage of the variety of offering in the marketplace layer, the scalability of the analytics layer and transparent data sharing mechanisms in data layer	N/A
Data provider	Integrates with the marketplace for sharing data, in some cases identical with the service user (e.g. DSO sharing SCADA data) in others not (e.g. weather service providing forecasting data). In most cases, data providers act as data owners providing consent.	Integrates with the marketplace for sharing data. Additionally, provides consent that may be monetized by selling data.
Data user	Same as Service Provider in the context of the services domain	Integrates with the marketplace for accessing data

Table 4. Supported functionalities of BD4OPEM users

The following paragraphs present, on a high-level approach, how these users will access the platform, how the platform handles each profile and provides illustrative figures on how the user interface will look.

BD4OPEM Marketplace Authorization Process

BD4OPEM Marketplace handles the users according to their profile; meaning that, each user profile will have its customized interface, displaying different user interface elements and, consequently, having different functionalities within the marketplace, as detailed in Table 4**Table 4.** Supported functionalities of BD4OPEM users

.

Profiles considered in this sense are:

- Marketplace operators, which can be split in:
  - Marketplace BD4OPEM administrator
  - Marketplace company administrator
- Marketplace users, broken down also in different groups:
  - Service Provider
  - Service User
  - Data Provider

- Data User

BD4OPEM Marketplace Registration Process

BD4OPEM Marketplace, at this stage of the project, considers that their potential users will be companies/organisations with their respectyive employee representatives or members accessing the platform. The platform should also be able to support private citizens to use the marketplace. However, the focus in this deliverable is to support access by companies & organisations and their respective employees and members.

As the access to the Marketplace is based on Self-Sovereign Identity as per the W3C sepcifications on Decentralised Identifiers [9] and Verifiable Credentials [10], the company or organisation will issue its employeess / members with identity claims in a Verifiable Credential so that the Marketplace can identify the individual and information on the company/organisation that they are a member of and their role in it. This will be facilitated by an SSI Agent (see section 5.5.3) made available to all companies & organisations that will interact with BD4OPEM and who do not yet implement their own SSI Agent.

Figure 11. Welcome Page of BD4OPEM Marketplace

Account registration process

A company employee or orgnaisation member with the role of administrator is able to sign up their company to the Marketplace will face a process composed of these stages:

- *Registration:* request made via dedicated webpage on the Marketplace. Upon requesting to register the user is asked to:
  - Present their identity claims from their SSI Wallet (name, email, role company name, company url, company DID, etc.).
  - Specify which Marketplace role it is applying for (service provider, service user, data provider, data user). One or more roles can be applied for either at registration or subsequently.
  - Accept the privacy requirements and policies
- *Acceptance or rejection* of the request is provided automatically. A specific process takes place in each case:

- in case of rejection, explanation of what went wrong
- in case of success, the Marketplace Account is created, and is issued to the company (as a Verifiable Credential to the company/organisation online wallet) with status pending. The address of the company wallet was obtained by the claims presented by the employee / member.
- The Marketplace requests a reciept of the Account from the company wallet to agree the creation of the Account and upon its receipt the staus of the Account is changed to active and is issued again to the company wallet.

The account will be created based on the information received in the employee's Verifiable Credential, and will be valid as long as the account between marketplace and company is set to active.

Figure 12. Account registration

Company user management process

After a successful registration process, the company employee or organisation member with administrator role can log in into the platform and assume any of the roles assigned to the account in the registration process.

The company or organisation is able to issue identity Verifiable Credentials to any of its employees / members that will allow access to the Marketplace and can limit their specific access in the Marketplace, based on their roles.

A company can have 0 users, but at least 1 administrator is needed to setup the account in the first place.

The different roles forseen so far are:

- Administrator
- Service Provider
- Data Provider
- Service User
- Data User

In other words, when a company employee or organisation member wants to become a BD4OPEM Marketplace user, s/he will not use the Marketplace website to register but will contact his/her internal company / organisation administrator to request the required identity Verifiable Credential.

BD4OPEM Marketplace Management Functionalities

BD4OPEM Marketplace Operator has a *company / organisation management section,* useful to search for company accounts and edit, disabling or monitoring their usage in the platform.

BD4OPEM Marketplace Authorisation matrix

Table 5, illustrates the matrix of BD4OPEM Marketplace capabilities (detailed in section 5.4) and marketplace business actors

Authorisation matrix					
	Providers		Consumers		Administrator
BD4OPEM Marketplace	Data provider	Service Provider	Data User	Service User	Marketplace Operator
Contracting	X	X	X	X	
Search	X	X	X	X	
Provisioning	X	X			
Billing	X	X	X	X	
Management & governance					X
Use service				X	
Advertising		X			
Publish		X			
Open innovation request				X	
Open Innovation service procurement		X			
Authentication	X	X	X	X	X
Access data			X		
Share data	X				
Service composition request				X	

Service composition procurement		X			

Table 5. BD4OPEM Marketplace Authorisation matrix

Figure 13 depicts one proposal of principal menu for a service or data user, where different user interface elements will be accessible or not depending on the role.

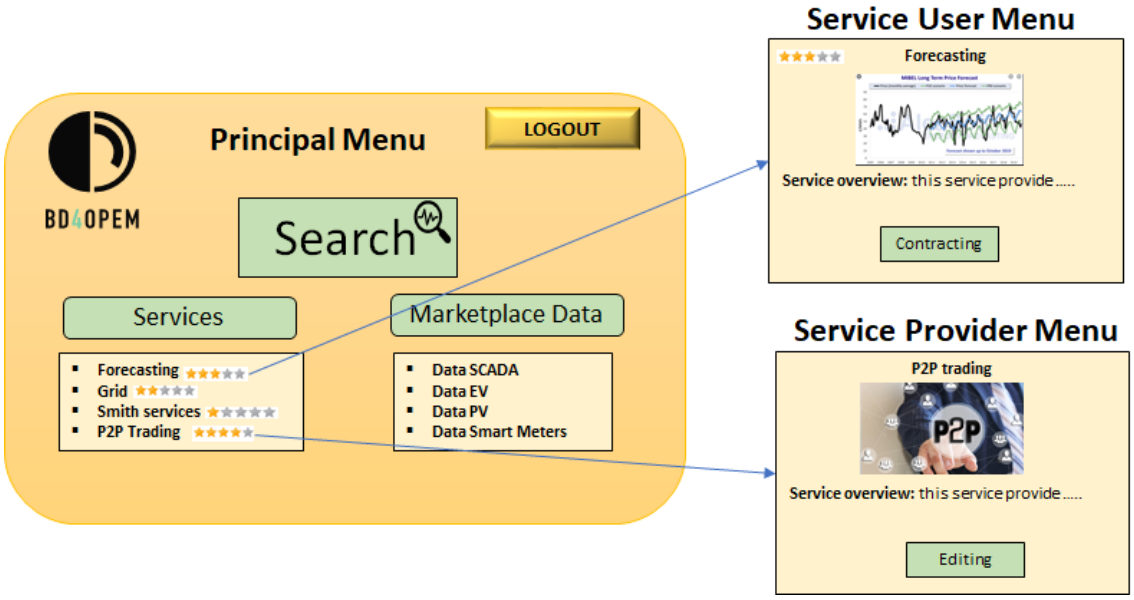


Figure 13. BD4OPEM Marketplace main menu proposal for a service or data user

5 BD4OPEM reference Architecture. Implementation View

This section covers the implementation view of BD4OPEM platform, providing an overview of the different functionalities, presented as assets in the project, that will be implemented in the five pilot sites. In essence, this view is referenced to the previous logical view and aims to show how the software system is divided into components, the technical requirements and the dependencies between those components. Table 6 shows the template used for collecting all this information:

Module_Id	Module Name
General Description	Main features of the asset
Functionality	Describe the functionalities provided
Input data	Describe the data required by the asset for operating (e.g. template, cost, data granularity, specification of the service, etc..)
Output data	Detail the output data
Interconnections	List the modules with it interacts
Dependencies	Detail if there is any dependency with other modules (e.g. one services is able to start if another is already running)
UML diagram	Depict the UML diagram of the component

Table 6. Template for BD4OPEM modules

Subsequently, sections describe all the components (Table 1, section 4) involved in BD4OPEM platform following the previous template. Here, it shall mention that some fields are not be able to provide since the project is at the initial phase of developments.

5.1 Physical Layer. Implementation View

5.1.1 Field Devices

General description. Field devices provide the source of information to be used in the BD4OPEM platform. Each pilot site will provide the communications and access methods for their integration, according to the use cases specifications.

Functionality. As previously mentioned, field devices generate the data that will be exploited in the platform. Main functionalities shall cover these fields are:

- Being accessible in order to enable retrieved data
- Provide the measurements established for the project
- Fulfil with the UC requirements in terms of frequency and data quality

BD4OPEM considers, up to now, the following type of field device, which could vary depending on the final pilot sites' configuration

- Distributed Energy Resources (DER) devices
  - EV charging
  - PV data
- Weather Data
- Distribution
  - Market data
  - Electricity Price Data
  - System Deviations
- GIS
  - Grid connectivity
  - Grid Asset characteristics
- AMI
  - Smart Meters (SM)
- SCADA
  - Primary Substation Measurements
  - Secondary Substation Measurements
  - Substations Alarms

Input data. N/A

Output data. Field devices will provide near/real-time and historical data.

Interconnections. Field devices are directly connected with their respective Device Adaptor

Dependencies. N/A

5.1.2 Open Data Sources

General description. Open Data sources refer to data that are public and available. Some of the services that are developed in the analytical layer of the BD4OPEM platform require open data (for example, climatological) that will be ingested through APIs or Web services.

Functionality.

Input data. N/A

Output data. Open Data Sources will provide open data to BD4OPEM platform.

Interconnections. These fields are directly connected with their respective Device Adaptor

Dependencies. N/A

Below Figure 14 presents both interconnections with BD4OPEM platform.

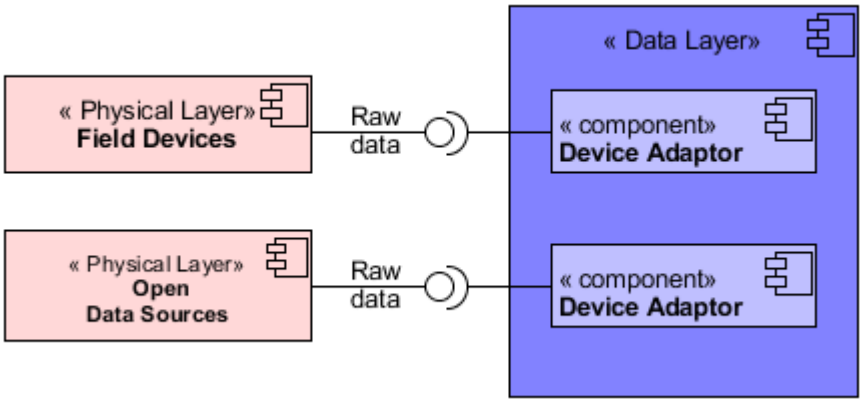


Figure 14. UML of Physical Layer

5.2 Data Layer. Implementation View

Following Table 7**Table 7.** Modules of Data Layer

lists the eight modules that compose the Data Layer and the responsible partner.

Data Layer		
#2.1	Device Adaptors	ATOS
#2.2	Data Harmonization–Common Information Model	ATOS
#2.3	Context Broker	ATOS
#2.4	Data Cleansing	ICOM
#2.5	Storage	ICOM, WEP
#2.6	Data Lake	ICOM
#2.7	Unified API	ATOS
#2.8	Data Management and Monitoring	ICOM

Table 7. Modules of Data Layer

5.2.1 Device Adaptors

General description: These modules are software components that enable both, the communication and data exchange between devices envision in the pilot sites and BD4OPEM platform, providing the end-to-end interoperability. In other words,

devices adaptors act as a middleware, ensuring the data ingestion and making them accessible to the other layers.

Functionality: the main functionalities of these modules are:

- Ensure compatibility communication with BD4OPEM platform
- Acting as a middleware enabling the data ingestion
- Read the data generated from the energy assets settled in the pilot sites

Input data: these modules receive the data, periodically and near/real-time, provided by the field devices planned in the five pilot sites, as well as, Open Data.

In this sense, Device Adaptors ingest three types of Data, divided into:

- *Data for developing services* coming from the Field Devices from the PS
- *Data for selling*. This sort of data will not be used for developing any service of BD4OPEM platform
- *Open Data* which derives from applications, such as weather apps.

Output data: N/A (they do not have a specific output data, because their implementation is included in Data Harmonization module)

Interconnections. These modules are related to:

- Field Devices
- Open Data Sources
- Data Harmonization – CIM (Common Information Model)
- SIEM

Dependencies:

- Field Devices should be generating the data to ingest and being accessible data

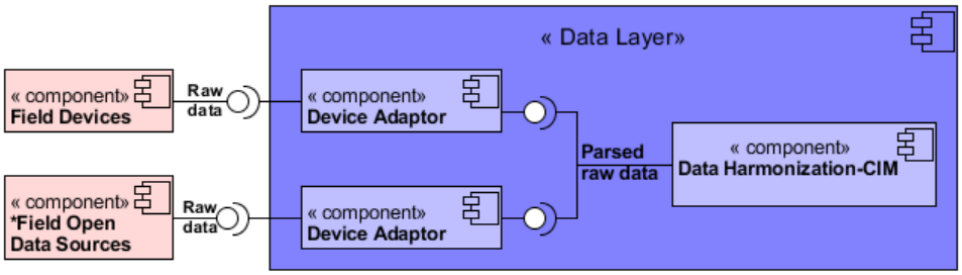


Figure 15. Device Adaptors UML diagram

5.2.2 Data Harmonization–Common Information Model

General description. Data harmonization module is a software component in charge of harmonization of all data ingested by the Device Adaptors from the Field devices. The harmonization is accomplished by the implementation of standards and

ontologies (e.g. SAREF[1], SAREF4NER[2], FIWARE[3], CIM[4]) framed within ICT and energy domain, offering a common structure for data and fully defined in D3.2.

Functionality. The main functionality of Data Harmonization - CIM is ensuring interoperability between the collected data of Device Adaptor and Analytics Layer. Furthermore, other functionalities to highlight in the architecture are:

- Facilitate uniform access to data regardless of source
- Allow reducing the number of errors
- Simplify data exchange
- Improve automation by ensuring interoperability.
- Provide a common structure acting as the information wrapper for the proprietary protocol.

Input data. As above cited, Data harmonization module receives the data from different Device Adaptors, one per asset, in a raw (crude) format and applies it correspond IEC standard, such as CIM (Common Information Model).

Output data. Both, Context Broker and Data Lake, modules get the harmonized data.

Interconnections. Data Harmonization is connected, on one side, with all adaptor, so as to receive the data, reading it and applying the correspondent CIM. Whilst, on the other side, Data harmonization is interacting with the Context Broker for real/near time data and Data Lake.

Dependencies. Both Device adaptor and devices (Field Devices and Open Data Sources) should be running in order to provide the input data.

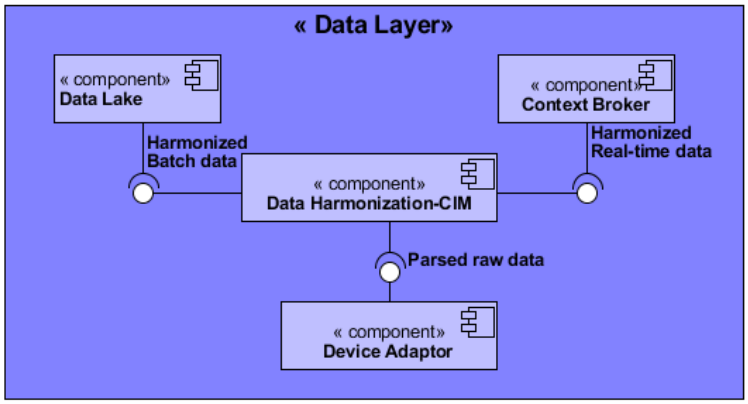


Figure 16. Data Harmonization-CIM UML diagram

[1] https://ontology.tno.nl/saref/

[2] https://saref.etsi.org/saref4ener/v1.1.2/

[3] https://www.fiware.org/

[4] https://www.dmtf.org/standards/cim

5.2.3 Context Broker

General description. The module of Context Broker handles the real time context information, so that this information could be available for other interested entities. Basically, this module uses standards (e.g. NGSI-v2[5]) for processing all data transmitted from the devices, updating the current state of those entities represented in the platform. Context Broker enables to the rest of platform's components access to data synchronously or asynchronously (through subscriptions).

Functionality. This module provides the following functionalities:

- Generate, collect, manage and publish the context information allowing its availability
- Analyse this information enabling to inform third parties and consequently acting through the context data (altering or enriching the current state)

Input data. Context Broker receives real time data straight forward from the Data Harmonization – CIM module.

Output data. Afterwards, it transmits it to Unifies API and/or Data Cleansing.

Interconnections. Context Broker is connected, on the one hand, with the Data Harmonization-CIM module, from where it receives the harmonized data generated in Field Devices and/or Field Open Data Sources and Data Lake in where the metadata is stored. On the other hand, this module is associated with Unified API, providing the real time data.

Dependencies. Data Harmonization – CIM module have to be running in order to ingest data to the Context Broker.

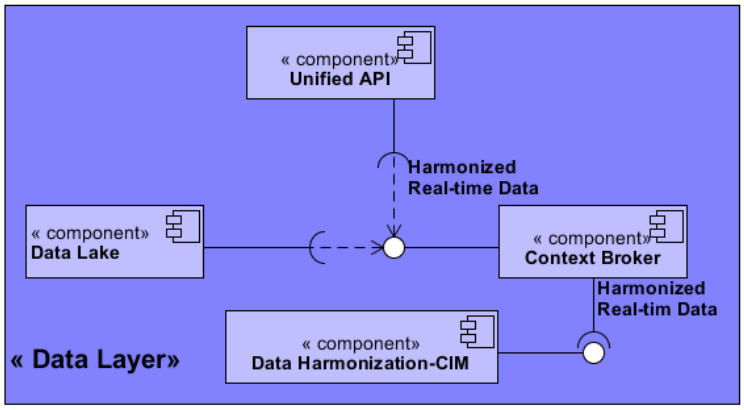


Figure 17. Context Broker UML diagram

5.2.4 Data Cleansing

General description. A data cleansing is a process that takes place after the data is ingested in the data lake and ensures better data quality.

Functionality.

[5] https://fiware.github.io/specifications/ngsiv2/stable/

- Duplicate removal
- Structural errors
- Correct data types
- Removal of inconsistent data records

Input data. Harmonized data

Output data. Cleansed data

Interconnections. This module has connections with the following modules:

- Data Lake
- Data Management and Monitoring

Dependencies. The data needs to be available/stored in harmonized form in the Data Lake.

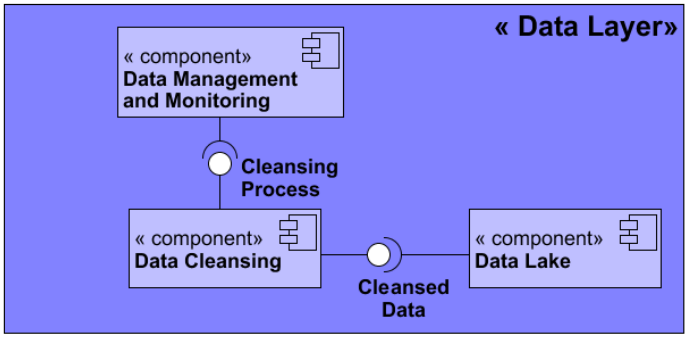


Figure 18. Data Cleansing UML diagram

5.2.5 Storage

Storage, as its name suggests, will store the operational data of the Marketplace layer such as service and data template or contract (permissions); as well as, the temporary storage of data of the services results for facilitating integrations. This module is divided into three different storages: analytics storage, marketplace storage and 3rd party database (DB).

In contrast, the data lake stores the data from data sources (e.g. field devices, open data sources, external APIs) once these data are harmonized.

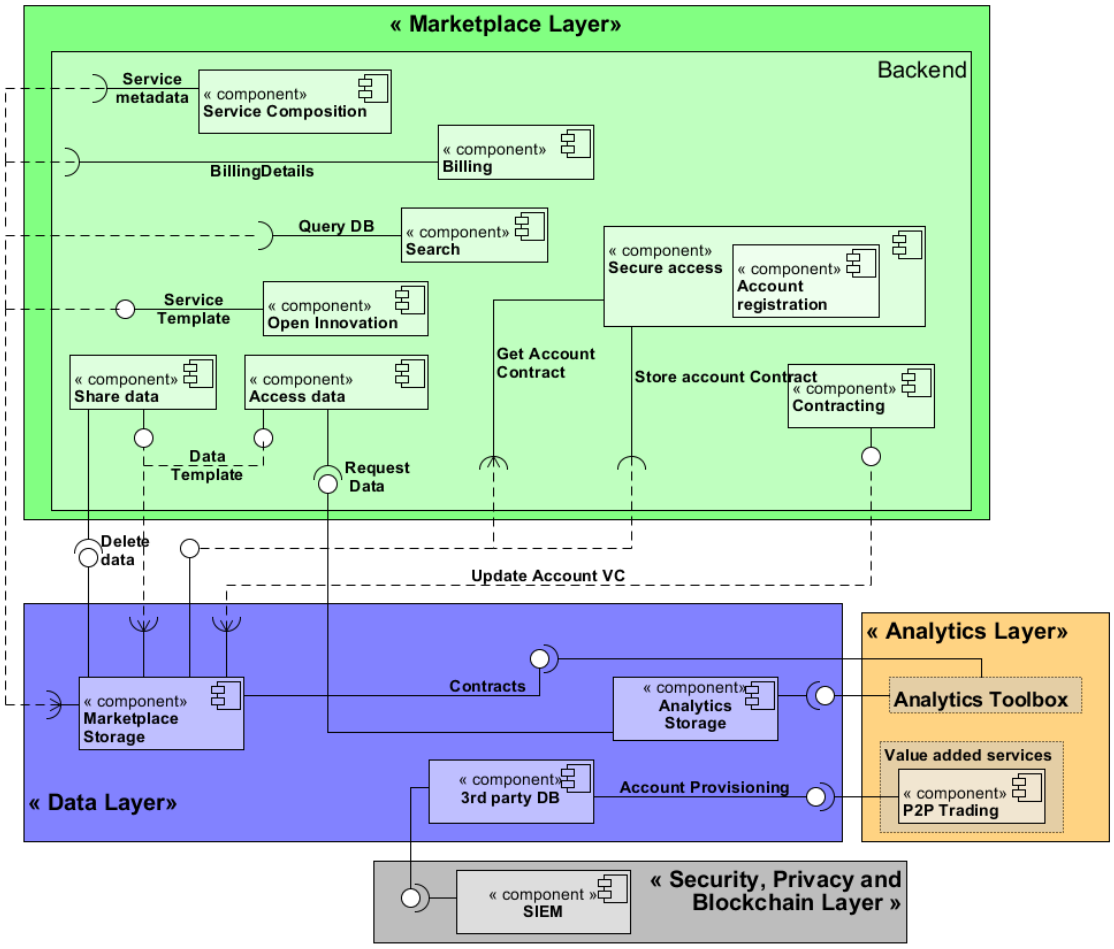


Figure 19. Storage UML diagram

5.2.6 Data Lake

General description. A scalable storage solution of data for facilitating data sharing and building analytic services shall be provided. The solution should enable the storage of diverse data types. It will support multiple replicas of the data facilitating distributed processing and high availability.

Functionality. Distributed storage: Storage support for different dataset environments

- Filed data in harmonised form - Harmonised data
- Data after the cleansing process - Cleansed data

Input data.

- Harmonized data

Output data.

- Harmonised data
- Cleansed data
- Metadata

Interconnections.

- Data Harmonization – CIM
- Context Broker
- Data Management and Monitoring
- Data Cleansing
- Unified API
- Distributed Analytics Engine
- SIEM

Dependencies. N/A

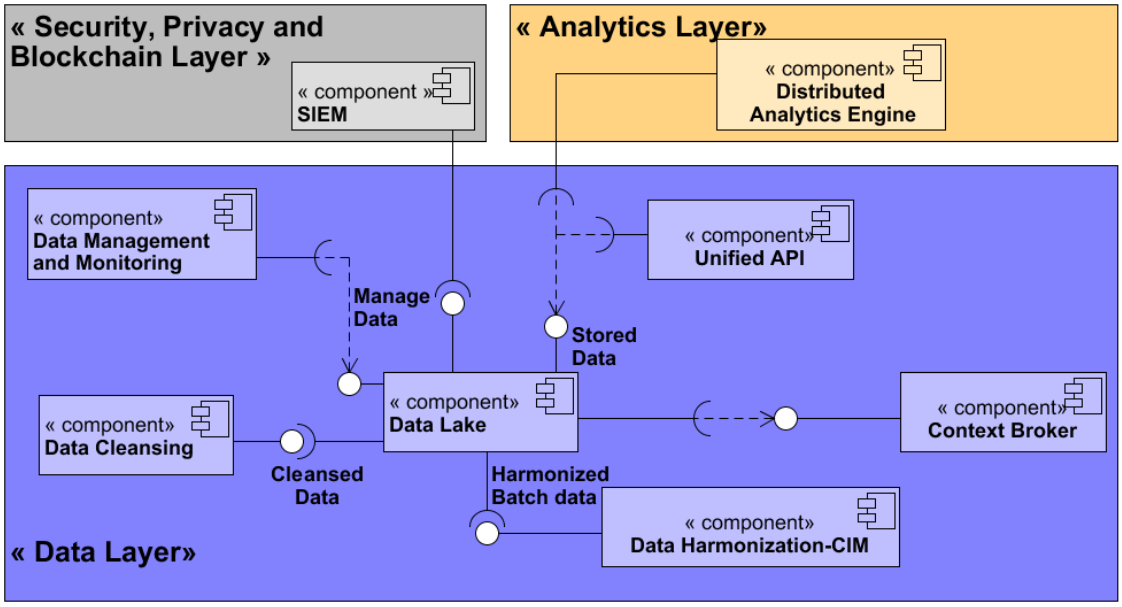


Figure 20. Data Lake UML diagram

5.2.7 Unified API

General description. Unified API module represents an abstraction layer that serves as a unified communication between Data Layer and both, Analytics Layer and Marketplace Layer.

Functionality. This module ensures data interoperability providing a harmonized way to request and send data. In short, it enables the exchange of requests and data that is carried out between the advance energy services, deployed in Data Analytics, and the data hosted in Data Layer, avoiding the complexity that different protocols request could ~~be~~ represent.

Input data. It receives harmonized real time and historical/batch data from the Context Broker and Data Lake, respectively.

Output data. It delivers harmonized and/or harmonized cleansed data

Interconnections. Unifies API interacts with:

- Data Lake
- Context Broker
- P2P Trading
- Service not utilizing DAE
- Access Data

Dependencies.

- Context Broker
- Data Lake

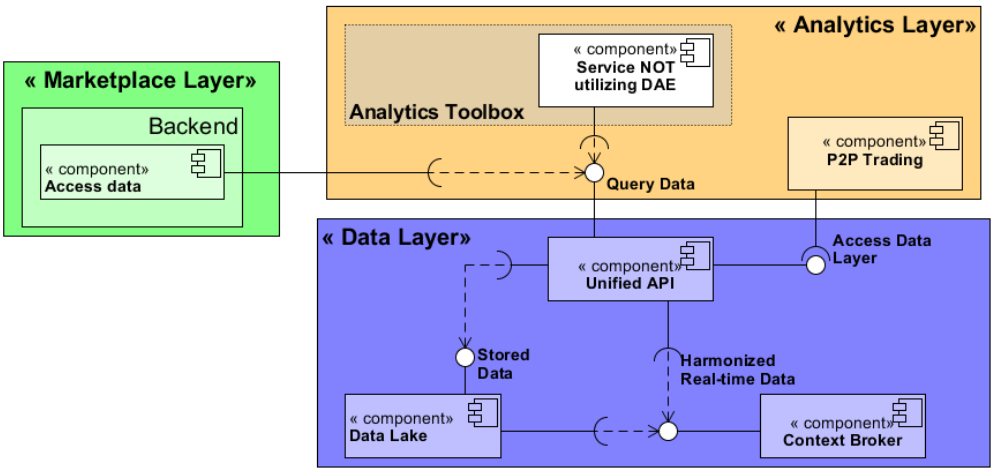


Figure 21. Unified API UML diagram

5.2.8 Data Management and Monitoring

General description. This module includes tasks that ensure the organization and supervision of the data lake. The module is in charge of modifying data permissions, deleting data and handling of the data after ingestion. All of the tasks create reports or metadata, so that everything happening in the data lake is documented.

Functionality. The main functionalities of this module are:

- Modify Data Permissions (grant/revoke access)
- Post-ingestion pre-processing (invoke Data Cleansing Module)
- Data Deletion (remove data when it is no longer needed)

Input data. Template with data info such as origin of data, type of data, privacy status, permission for accessing data, etc.

Output data. Metadata report (data location and basic info), cleansing process parameters, permissions change report (access granted, or access revoked)

Interconnections.

- Data Lake
- Data Cleansing

- Marketplace Storage
- Share Data

Dependencies. N/A

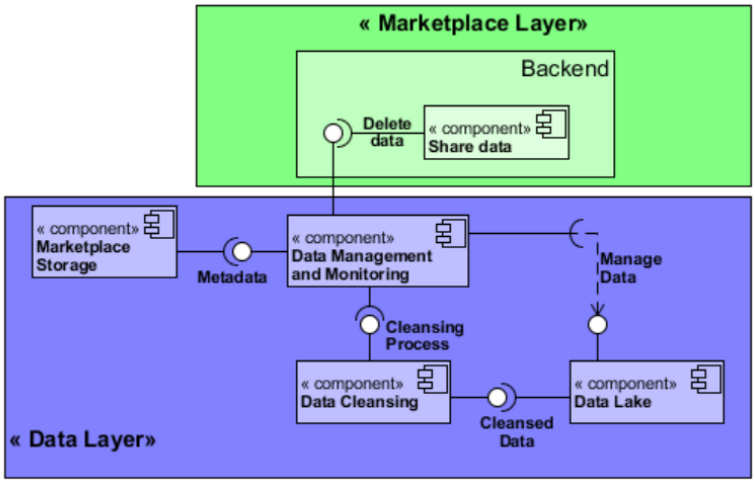


Figure 22. Data Management and Monitoring UML diagram

5.2.9 Data Layer Summary

Next Figure 23 presents the whole Data Layer UML Diagram.

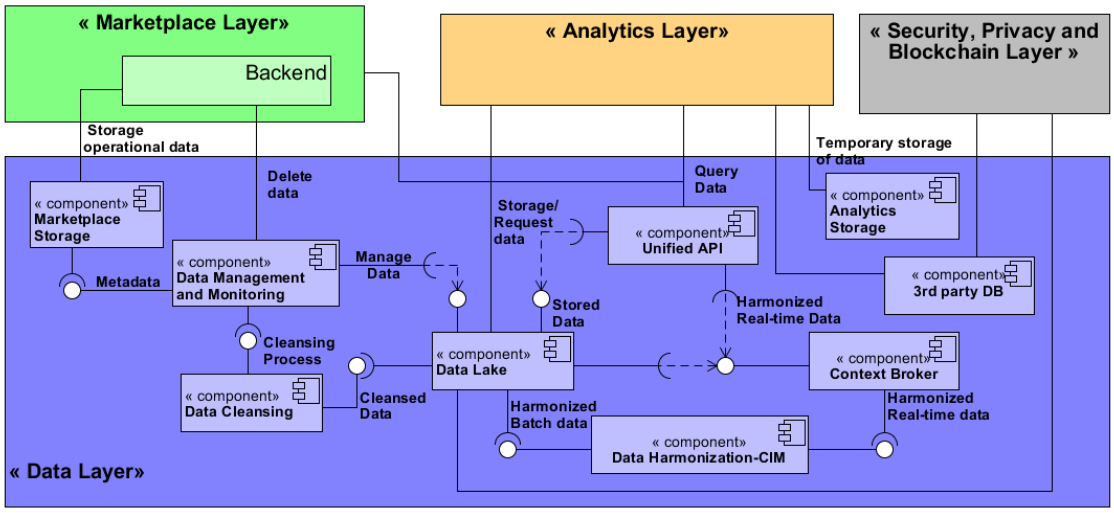


Figure 23. Data Layer UML diagram

5.3 Analytics Layer. Implementation View

Table 8 lists all modules that conforms the Data Analytics layer, including its responsible partner.

Analytics Layer		
#3.1	Analytics Toolbox	ALL
#3.1.1	E1.1-Topology	ODT
#3.1.2	E1.2-Observability	ODT
#3.1.3	E1.3- Predictive maintenance in electrical power systems	UPC, JSI
#3.1.4	E2.1 Measurement errors detection	ICOM, UPC
#3.1.5	E3.1- Grid disturbance simulations	UPC, JSI
#3.1.6	E3.2- Impact study PV, EV & new loads	ODT
#3.1.7	E4.1- Inconsistencies in energy balance and power-voltage	ODT
#3.1.8	E4.2- Fraud patterns detection	UPC
#3.1.9	E5.1- Flexibility forecast	JSI, UPC
#3.1.10	E5.2- Flexibility aggregated services for BRPs	JSI, UPC
#3.1.11	E5.3- Flexibility aggregated services for DSOs	JSI, UPC
#3.1.12	E5.4- EV to Grid	NUVVE
#3.1.13	E6.1- Energy management at household or at community level	ICOM, JSI, UPC
#3.1.14	E6.2- Energy forecasting	ICOM, JSI, UPC
#3.1.15	E8.1- Asset and investment planning	UPC
#3.1.16	E8.2- Asset estimation optimization for microgrids	VUB
#3.2	Distributed Analytics Engine	ICOM
#3.3	Analytics Service & Management	WEP
#3.4	Resource monitoring	UPC
#3.5	E9.1-Service KPIs	UPC
#3.6	Value added services	ATOS

#3.6.1	E7.1- P2P trading	ATOS

Table 8. Modules of Analytics Layer

5.3.1 Analytics Toolbox

Analytics Toolbox represents the core of the Analytics Layer. This module contains the majority analysis tools that will be offered through Marketplace as an energy service.

These services can be integrated into the BD4OPEM platform, like Flexibility forecast or Grid disturbance simulations services, or run on a third-party platform, such as Topology or Observability.

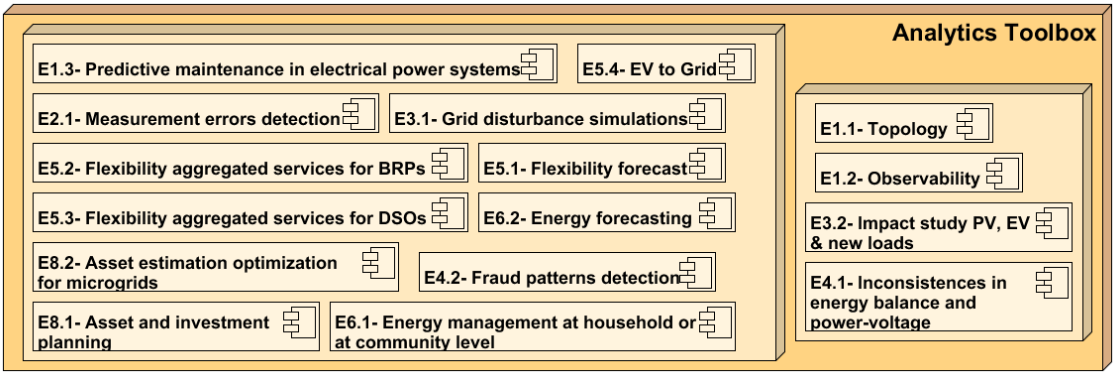


Figure 24. Analytics Toolbox logical diagram

Following subsections describe the services included in the Analytics Toolbox.

5.3.1.1 E1.1-Topology

General description. Low Voltage networks topology is often poorly known, impeding effective network management. The proposed service makes available, on the platform developed during the project, the network real topology: meter-to-substation associations (level 1), and meter-to feeder/phase associations within the substations (level 2).

Functionality. Retrieve the topology (connectivity information smart-meters / phase / feeder / substation) of the Low Voltage network.

Input data.

- Historic Smart Meter Measurement Data
- Historic Secondary substation
- GIS coordinates of assets (smart meters, secondary substation)

Output data.

- Connectivity information describing the links between assets in the low voltage network.
- Offline analysis

5.3.1.2 E1.2-Observability

General description. Smart meters data are not available in real time, yet real time knowledge of the LV network state, so-called "Observability" is essential for network operation. The proposed solution provides a statistical estimation of the LV network state that can be run using data from the primary substation only, opening the way to real time management for LV networks. Primary substation data can be taken directly from the MV SCADA. The solution can be run after a training period, where smart meter data are required in order to analyse the network behaviour in regard to external conditions. After this training period, only primary substation data is required for the operation.

Functionality. Provide real time state estimation of one point of the low voltage network which is not monitored in real time.

Input data.

- Historic Smart Meter Measurement Data
- Historic Primary Substation Measurement Data
- Primary Substation Measurement Data (SCADA)
- Meteorological data
- For display purpose
- GIS coordinates of assets (smart meters, secondary substation)

Output data.

- State estimation of the monitored point of the network.
- Online analysis

5.3.1.3 E1.3- Predictive maintenance in electrical power systems

General description. The system is trained with historical component failure data, including different variables and the date of failure of the component. Once the system is trained by monitoring these variables, the algorithm can predict the probability of OLTC and circuit breakers failure in different time windows. Finally, following a reliability-centered maintenance, a list is returned with the different system components and their remaining useful life is sorted by criticality, time and probability of failure.

Functionality. To perform predictive maintenance of the following distribution grid components: OLTC and circuit breakers.

Input data. This module uses two types of input data depending on the stage of the running:

For training:

- Historical meteorological/ weather data
- Historical sensor values of the components
- Historical operating values of the components
- Historical short-circuit current
- Historical calendar data
- Historical failure date of the components

- Asset ID

For operation:

- Model ID
- Sensor values of the components
- Operating values of the components
- Calendar data

Output data. In the same manner of the input data, the output data is divided into:

For training:

- Forecast Model ID for asset

For operation:

- List of assets with their remaining useful life, criticality and probability of failure

5.3.1.4 E2.1 Measurement errors detection

General description. This module is a service that seeks to identify, detect and solve eventually anomalies, errors or missing values coming from data sources, in order to ensure the data quality and usability for the services. Depending on the type of anomaly detected, a correction will be automatically performed, or the anomaly will be used as input to other modules.

Functionality.

- Outliers removal
- Missing values removal
- Interpolation
- Other requests by the Service User

Input data. The input data are:

- Cleansed Data
- Specifications of Service User

Output data.

- S2.1 output data

5.3.1.5 E3.1- Grid disturbance simulations

General description. This module aims to offer services the chance to identify possible congestions or other PQ problems in a MV/LV distribution grid in long-term, day-ahead, or real-time scenarios.

Functionality.

- Identify the possible congestions in a MV distribution grid in real-time.

- Analyse in the off-line mode for forecasted scenarios the congestion or PQ related issues.

Input data.

For training:

- Forecast
  - Historical data of consumption and generation
  - Weather data
  - Calendar data
  - Time range
- Congestions Forecast
  - Historical data of loads and generators
  - Topology and Layout
  - Lines and transformers parameters
  - Substation data

For operation:

- Forecast
  - Forecasting Model
- Congestions Forecast
  - Forecasted demand
- Congestions Real Time
  - Topology and Layout
  - Lines and transformers parameters
  - Substation data
  - Smart meter measurements (active and reactive power) in real time

Output data.

For training:

- Forecast
  - Forecasting Model for consumption (day-ahead, long-term)
- Congestions Forecast
  - Congestions Model

For operation:

- Forecast
  - Demand forecast for the desired time range
- Congestions Forecast
  - Probability of congestions in lines and transformers offline
- Congestions Real Time
  - Congestions existence online

5.3.1.6 E3.2- Impact study PV, EV & new loads

General description. One service of the Analytic Toolbox within the Analytic Layer. Low voltage networks must adapt to the energy insertion

Functionality. Provide an impact study of the integration of a new asset in a given low voltage network

Input data.

- Historic Smart Meter Measurement Data
- New facility characteristics
- Weather data
- For display purpose:
- GIS coordinates of assets (smart meters, secondary substation)

Output data.

- Insertion capacity map.
- Offline analysis

5.3.1.7 E4.1- Inconsistencies in energy balance and power-voltage

General description. One service of the Analytic Toolbox within the Analytic Layer. The aim of this service proposal is to make available, on the platform developed during the project, an effortless way to perform a Non-Technical Losses study.

Functionality. Provide a location and quantification of the non-technical losses which occurs on a given low voltage network

Input data.

- Historic Smart Meter Measurement Data
- Historic Secondary substation
- For display purpose:
- GIS coordinates of assets (smart meters, secondary substation)

Output data.

- Weighted non-technical losses map

5.3.1.8 E4.2- Fraud patterns detection

General description. This module is aimed at detecting inconsistencies in the energy balance, subtract the technical losses (TL) and identify the non-technical losses (NTL). Studying the NTL patterns, the service will return the type of grid users (grouped in clusters) who may be responsible in order to carry out a field inspection.

Functionality. This module provides the following functionalities:

- Detect inconsistencies in the energy balance between secondary transformers and consumers (LV grid).
- Analyse what type of consumer might be committing fraud in order to execute a targeted field inspection.

Input data.

For training:

- Historical SM's load curve (active and reactive power) and voltage profile
- Historical calendar data

- LV grid topology (S1.1 output)
- Metering point details connected to the substation
- Historical frauds (ID of the consumer, estimation of power, number of hours, etc) and NTL data
- Weather data
- Historical local transformer power and voltage profile
- Historical SM logs

For operation:

- Model ID for DSO/country
- SM's load curve (active and reactive power) and voltage profile
- Calendar data
- Local transformer power and voltage profile
- SM logs

Output data.

For training:

- Model ID for DSO/country

For operation:

- Location and estimation of the NLT
-
- Potential metering points responsible for NTL with estimation of NTL type and probability of classification correctness

5.3.1.9 E5.1- Flexibility forecast

General description. Flexibility of consumers or distributed energy resources (DER) in the system will be estimated according to the system and environmental conditions, weather prediction and consumption of consumers and prosumers. The forecast will be provided on a short-term horizon (day-ahead, intra-day), to help the operation of the distribution network and the imbalances settlement of the BRP. The forecast will be provided considering the aggregation of the end-users' flexible assets.

Functionality. Provide an aggregated flexibility prediction of consumers and DER resources flexibility in the system.

Input data.

For training:

- Environmental data – weather data
- National calendar data
- GIS coordinates – SM location
- Load profiles (SM data)
- DERs data – mainly PV (SM data)

- Flexibility devices, their consumption data – submetering data
- Zone ID

For operation:

- Model ID(s)
- Target period
- Target system scope/area
- Weather prediction
- Environmental data
- Load profiles (SM data)
- DER (SM data)
- Flexibility devices, their consumption data – submetering data

Output data

For training:

- Forecasting Model ID for zone or area

For operation:

- Aggregated flexibility forecast value [kWh] per time period
- Flexibility forecast accuracy estimation/probability per time period

5.3.1.10 E5.2- Flexibility aggregated services for BRPs

General description. This module aims to provide services for day-ahead and self-balancing optimization of the BRP portfolio.

Functionality. Calculate a flexibility request to be used by the BRP for the following objectives:

- Minimize the BRP overall electricity purchase costs in the day-ahead market.
- Minimize the BRP imbalances in its portfolio to avoid imbalances penalizations.

Input data.

For training:

- Historical weather data
- Historical electricity market data
- Historical consumption/generation of the BRP portfolio
- Historical BRP portfolio imbalances
- Forecasting type
- Historical calendar data
- "Link" (data lake reference) to historical data for training
- Area ID

For operation:

- Model ID(s)
- Past BRP portfolio consumption/generation data
- Past BRP portfolio imbalances data
- Past electricity system imbalance data
- Weather forecast data
- Scheduled electricity market data (day ahead)
- Calendar data
- "Link" (data lake reference) to data for operation

Output data.

For training:

- Forecasting Model ID for BRP service (Area)

For operation:

- Flexibility request of the BRP
- Flexibility offer by the Aggregator (TBD)

5.3.1.11 E5.3- Flexibility aggregated services for DSOs

General description. This module aims to provide flexibility services for DSO in the operation phase (short-term), namely congestion management, grid capacity management and voltage control. The main objective of the service is to help the DSO to calculate the Flexibility Request (FR) to be activated for the network operation and congestion management by means of flexibility.

Functionality. Calculate a flexibility request to be used by the DSO for the following objectives:

1) Prevent grid system overload by reducing peak loads
2) Optimize grid operational performance and asset dispatch by reducing peak loads.
3) Prevent excessive voltage push up in areas with a large number of renewables.

Input data.

- Topology data: node and substations connections
- Grid characteristics: lines, transformers parameters
- Metering data: historical consumption,
- Flexibility offer by the aggregator (output from S5.1)
- Congestion problem in the network of study (output from S3.1)
- Network demand forecast (calculated in service S3.1)

Output data.

- Flexibility request of the DSO: value (up-regulation or down-regulation) + time period

5.3.1.12 E5.4- EV to Grid

General description. This module aims to perform flexibility services for the local DSO in order to avoid the congestion of the grid. Flexibility from the aggregated EV fleet at the congested meters will be pooled in order to assist a balanced operation of the grid. In parallel, this service will help to postpone and reduce the need of grid expansion investment. Additional services might be provided via EVs (e.g. Time-of-Use optimization).

Functionality. This service aims to cope with the congestion issues of the distribution grid level. The fleet power provision will be activated according to DSO requests and/or forecasts of congestion (TBC). The fleet will react by absorbing or injecting electricity (mainly absorbing in order to avoid overloading of the cables) in order to obtain a smooth grid operation.

Therefore, the optimal provision of this service requires the current flexibility and forecasted flexibility capacity that can be provided by the fleet of EVs in order to provide the flexibility services for DSOs. Furthermore, information about the flexibility provided upon request/trigger is required to evaluate the correct execution of the flexibility service.

Input data. The input data are:

- EV/EVSE data (internal of Nuvve),
- Grid data from local DSOs,
- Optimized schedule through forecasts (internal of Nuvve and from UPC)

Output data. The output data are the service provision availability, forecasted signals for service provision

5.3.1.13 E6.1- Energy management at household or at community level

General description. The module provides a service to prosumers to utilize their own flexibility behind-the-meter, which allows optimal scheduling of the electrical appliances at end-user premises through energy management systems.

Functionality.

- Optimal scheduling of controllable assets
- Support for cost optimization either by time-of-use optimization, by reducing maximum load or to perform self-balancing.
- Controlling of assets behind-the-meter

Input data.

For operation:

- Metering data from the EMS (sub-metering)
- Metering data from the SM
- Energy Forecasting data (generation and consumption)
- Controllable Assets' parameters
- User Preferences
- Flexibility "signals"
- Electricity tariff

- Target period
- "Link" (data lake reference) to data for operation

Output data.

For operation:

- Optimized schedule
- Price signals in case of manual control

5.3.1.14 E6.2- Energy forecasting

General description. This module aims to predict the baseline, flexible assets demand and generation curves at household or community level within a short-term horizon. Predict the household or community baseline demand, flexible assets consumption and generation curves in order to schedule loads, energy storage system and generation in an optimal manner, through the energy management system optimization. The energy training model application is in charge of training the forecasting models.

Functionality.

- Predict baseline demand at household or community level
- Predict flexible assets' consumption and generation
- Model training for different assets, forecast types and levels of aggregation
- Support for automatic training/maintenance of models

Input data.

For training:

- Historical consumption/generation data
- Historical weather data
- Historical calendar data
- Asset Information
- Forecasting type
- "Link" (data lake reference) to historical data for training

For operation:

- Past metered load consumption data
- Past metered generation data
- Weather data
- Calendar data
- Model ID(s)
- "Link" (data lake reference) to data for operation

Output data.

For training:

- Forecasting Model ID for asset

For operation:

- PV generation forecasting
- Flexible loads consumption forecasting
- Disaggregated loads consumption forecasting
- Baseline consumption forecasting

5.3.1.15 E8.1- Asset and investment planning

General description. This module aims to develop optimal investment strategies that contribute to the long-term planning using traditional assets combined along with flexible assets to drive into an optimal decision-making, minimizing the capital expenditure and operational expenditure costs (CAPEX/OPEX).

Functionality. Minimize investment costs based on the optimal management of Energy Storage Systems and EV Charging Stations installed in the MV network, which can provide flexibility in the demand peaks, considering the growth in a long-term planning horizon.

Input data.

- Metering data from the SMs
- Topology data
- Grid disturbance data.
- Financial data
- Assets data

Output data.

- Optimal investment plan.

5.3.1.16 E8.2- Asset estimation optimization for microgrids

General description. Asset Management and optimization in microgrid is a significant task performed by utility companies to extend the lifetime of the critical distribution assets and to accordingly ensure grid reliability by preventing accidental and unintentional outages. A microgrid is an evolving distribution technology that incorporates a variety of distribution technologies, including distributed generation, demand response, and energy storage. Moreover, the substation transformer, as the most critical component in a distribution grid, is selected as the component of choice for asset management studies[11].

Functionality. The main functionality in microgrid is optimal scheduling problem that determines the least-cost schedule of available resources (DERs and loads) while minimising the cost of distribution transformer loss of life subjected to the fundamental operational constraints.

Another relevant functionality is to minimize the microgrid annual operation cost, including the local generation cost and the cost of energy exchange with the utility grid [11].

Input data.

Information required at different time horizons on the status of the grid and its assets like

- Smart Meters Data from PV and EV
- Power Generation Data
- EV Charging Station Data
- Battery Storage
- Measurement Errors
- Data from Network Stability Indicators
- Power Losses
- Number of Faults

Output data.

Minimize the microgrid annual operation cost, including the local generation cost and the cost of energy exchange with the utility grid.

5.3.1.17 Services utilizing DAE and Services not utilizing DAE

Following sections will refer to these (16) services as "Services utilizing DAE" (Distributed analytics engine) or "Services not utilizing DAE". Table 9 lists the services that belong to each group.

	Service utilizing DAE	Service NOT utilizing DAE
E1.1 Topology		X
E1.2-Observability		X
R1.3- Predictive maintenance in electrical power system	X	
E3.1- Grid disturbance simulations	X	
E3.2- Impact study PV, EV & new loads		X
E4.1- Non-Technical losses- Inconsistences in energy balance and power-voltage		X
E4.2- Fraud Patterns Detection	X	
E5.1- Flexibility Forecast	X	
E5.2- Flexibility Aggregated BRPs	X	
E5.3- Flexibility Aggregated DSOs	X*	
E5.4- EV to Grid		X
E6.1- Energy management at household or at community level	X*	
E6.2- Energy forecasting (Demand estimation)	X	

E8.1- Asset and investment planning	X*	
E8.2- Asset estimation optimization for microgrids		X

X* = TBC

Table 9. List of services utilizing and not utilizing DAE

The services not utilizing DAE are accessible on the BD4OPEM platform the same way the other one are. There are two options for the connection of those services to the platform:

Case 1: Historical Data received in files (e.g. csv, excel ...) will be used (e.g. in S1.1 Topology).

The service access to the platform through the unified API which sources the data from the data lake. After the provision of the service, the same operation is carried out in reverse meaning the results are pushed back to the BD4OPEM platform through the API and store in the relevant module.

Case 2: Real-time data is required (e.g. S1.2 Observability).

The service is connected to the platform through the unified API as in case 1 but instead of using the data lake, the API sources the real time data of the service user from the context broker.

Those service will communicate with the Analytics Service & Management module and the Resource Monitoring module the way there specifications require to.

Figure 25 illustrates the representative UML for the preceding services.

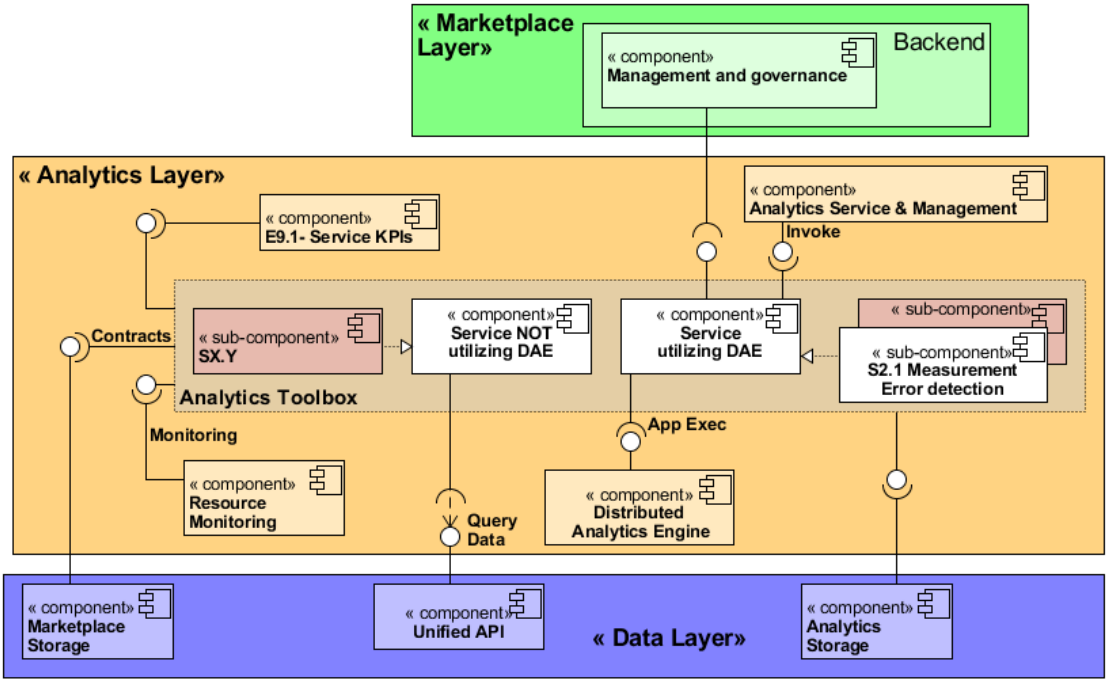


Figure 25. UML diagram of Services utilizing and not utilizing DAE

5.3.2 Distributed Analytics Engine

General description. Distributed analytics spreads the workload of the services over multiple nodes in a cluster of servers, rather than asking a single node to tackle a big problem. The same algorithms run across each of the nodes, processing a subset of the data. When the processing concludes, the data sets are aggregated, or brought back together, to generate collective insight. The engine will be able to host different applications developed/modelled by the solutions providers[12].

Functionality.

- Support a variety of analytics processes
- Distribute analytics workload over multiple clusters
- Improved performance and efficiency

Input data. Application id, application parameters

Output data. Result of analysis (e.g. file)

Interconnections.

- Data Lake
- Analytics Toolbox: Service utilizing DAE
- Resource Monitoring

Dependencies.

- Resource monitoring

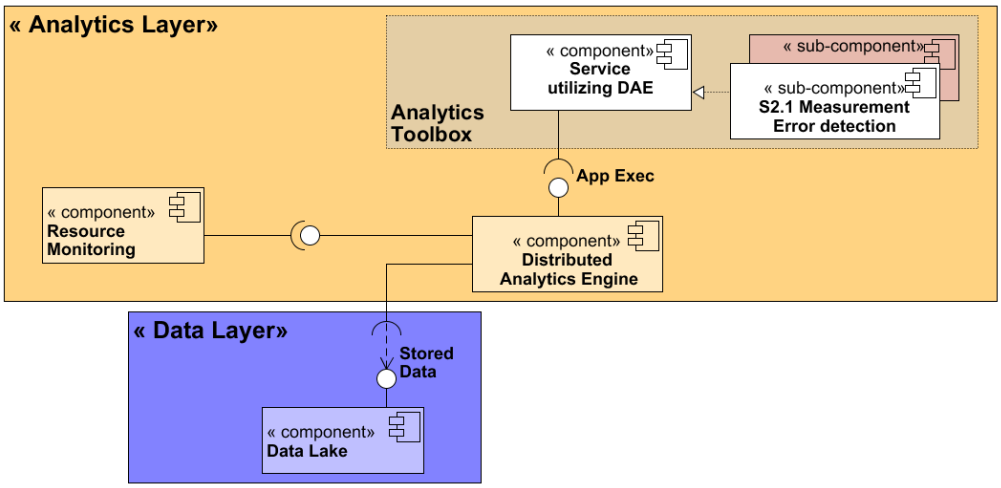


Figure 26. Distributed Analytics Engine UML diagram

5.3.3 Analytics Service & Management

General description. Analytics Service and Management provides the connection between the Marketplace Layer and the Analytics layer. It can act as a trigger, calling/activating/launching/scheduling any module of analytics toolbox, as well as, processing and generating the demanded outputs.

Functionality.

- Smart scheduling module.
- Asynchronous trigger for other modules, i.e.:
  - Invokes service composition functionalities
  - Invoke management and governance functionalities

Input data. This module receives the input data from the Marketplace

Output data. Replies go back to the Marketplace.

Interconnections.

- Management and Governance
- E7.1- P2P Trading
- Resource Monitoring
- Analytics Toolbox: Service utilizing DAE

Dependencies.

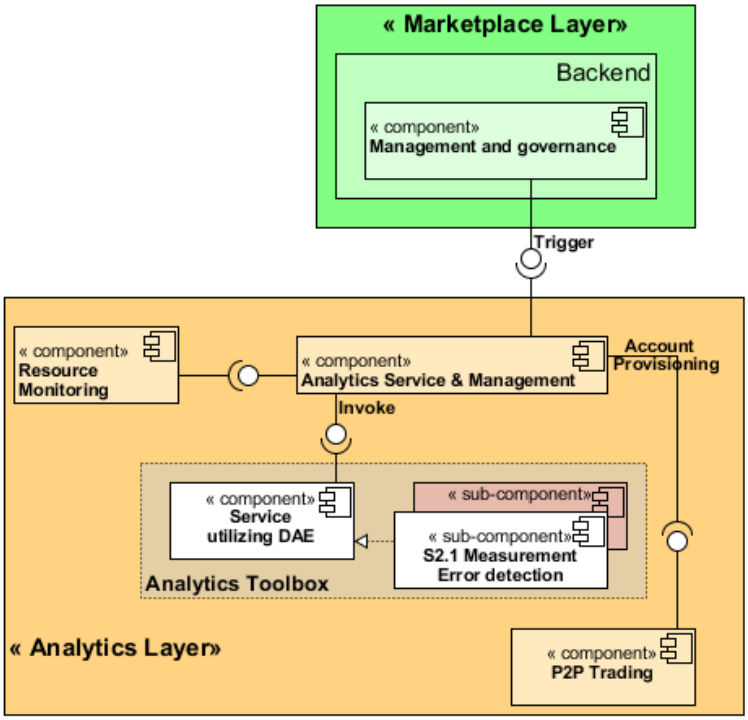


Figure 27. Analytics Service & Management UML diagram

5.3.4 Resource monitoring

General description. Resource monitoring is a supervision tool in charge of verifying the analytics tools performance.

Functionality. The module has the following main functionalities:

- Log activity to record when resources are used
- Use metrics to tell how the service is performing and the resources that it is consuming
- Manage Analytics layer alerts, troubleshooting and reactions to critical situations.

Input data.

- Performance data Analytics Service and Management, Distributed Analytics Engine, Analytics Toolbox, and SIEM modules
- Errors and alerts from Analytics Service and Management, Distributed Analytics Engine, Analytics Toolbox, and SIEM

Output data.

Interconnections.

- Analytics Service & Management
- Distributed Analytics Engine
- Analytics Toolbox

- SIEM

Dependencies.

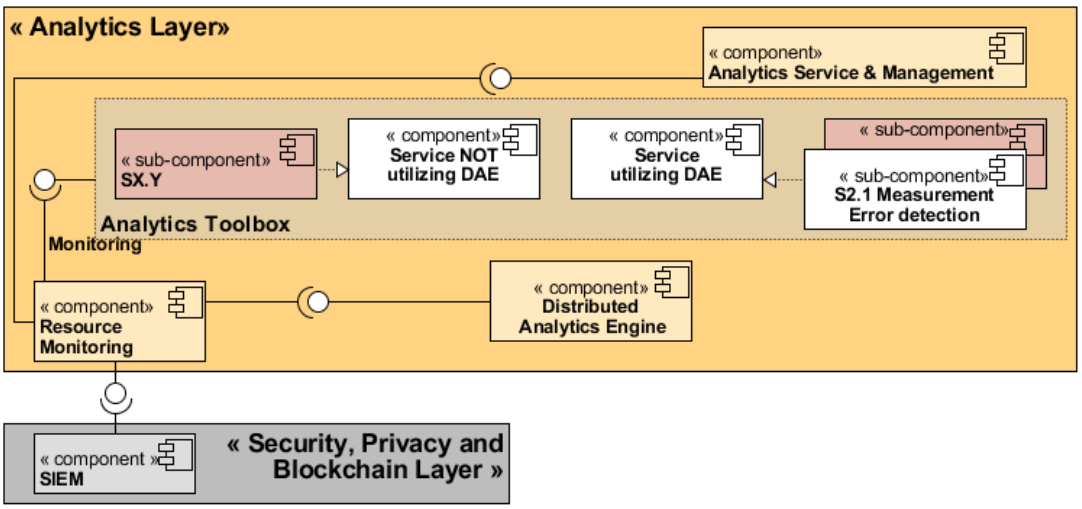


Figure 28. Resource monitoring UML diagram

5.3.5 E9.1-Service KPIs

General description. Several key performance indicators (KPIs) are collected from different services. They are displayed on the platform at different time horizons through the Publish module. Service KPIs include status of service (operating/idle), success, economic and environmental KPIs, performance quality or errors rate.

Functionality.

- Calculate service KPIs.
- Manage the calculation of service KPIs to evaluate their performance

Input data. To be determined based on each service KPIs

Output data. Service KPIs

Interconnections.

- Analytics Toolbox
- Publish

Dependencies. Services need to be executed to be able to calculate their KPIs.

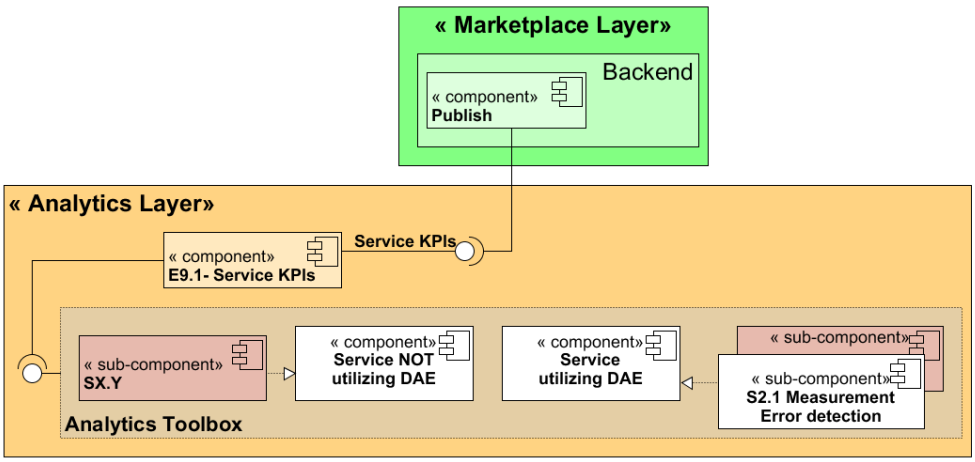


Figure 29. E9.1-Service KPIs UML diagram

5.3.6 Value added services

Value added services module includes the P2P Trading, as a non-analytic service that facilitates the purchase and sale of energy between two or more parties connected to the grid (homes, communities, EVs, companies, etc.) in a safe and transparent context and without intermediaries or central authority.

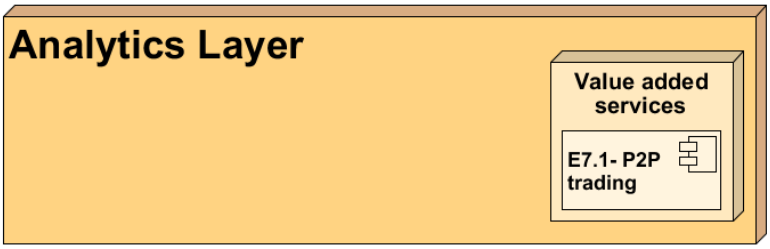


Figure 30. Value Added Service logical diagram

5.3.6.1 E7.1- P2P trading

General description. This module performs local trading of energy between end-users of the service (prosumers, consumers and small-scale community producers), on a non-commercial basis, promoting DRES. All participants´ smart meter readings (detailing overall production/consumption energy flow) are obtained from the BD4OPEM Data lake.

The P2P Trading service enables these transactions in a trusted and automated fashion by initiating respective smart contracts on the blockchain.

Resultant production and consumption figures are made available to the service users for billing.

Functionality. The module starts the execution of the CEC P2P Energy Trading on behalf of a CEC Administrator which then configures and starts a Smart Contract (SC)

to perform the collection CEC Members´ smart meter readings to monitor the flow of energy to and from end-users that contract the trading service per hour.

The trading tariff(s) are obtained from the contract rules and can require real-time access to commercial trading tariffs, and thus the SC needs access to contract parameters set up for the CEC, and real-time commercial tariffs.

The SC is responsible for capturing all energy traded by its members each hour and issuing periodic statements. The Periodic statements are made available outside the blockchain for settlement of the energy traded between CEC Members. The periodic statement billing data for all members is made available to the CEC Administrator for managing the billing between the members of the CEC, so to bill net consumers and remunerate net producers [3].

- CEC P2P Trading Algorithm:

  Having a CEC with pre-defined trading rules makes it simpler for the user´s to manage trading energy with their CEC peers in a fully transparent and automated way rather than having to make individual deals with the different members of the community which can make the whole enterprise unmanageable for individuals [3].

  A potential simple example could be: solar residential prosumers A, B & C are members of a CEC, along with consumers D, E, & F and also a small-scale producer wind generator H is run by a Billing agent on behalf of the community. Billing data is updated hourly as rates can vary during the day. So, a specific hour CEC P2P Energy Trading billing data could be as follows:

  - +ve energy balance = 7kWh/h (A: 3 kWh/h, B: 1kWh/h, H: 3 kWh/h)
  - -ve energy balance = -10kWh/h (C: -1 kWh/h, D: -3kWh/h, E: -3kWh/h F: -3kWh/h)

  CEC Member Accounts:

  - A = 3 kWh/h: 3 kWh/h @ productionCEC tariff
  - B = 1 kWh/h: 1 kWh/h @ productionCEC tariff
  - C = -1 kWh/h: -1 kWh/h @ consumptionCEC tariff
  - D = -3 kWh/h: -2 kWh/h @ consumptionCEC tariff, - 1 kWh/h @ commercial tariff
  - E = -3 kWh/h: -2 kWh/h @ consumptionCEC tariff, - 1 kWh/h @ commercial tariff
  - F = -3 kWh/h: -2 kWh/h @ consumptionCEC tariff, - 1 kWh/h @ commercial tariff
  - H = 3 kWh/h: 1 kWh/h @ productionCEC tariff

  Note CEC energy produced is shared in a fair manner between the members that are consuming it.) Overall, the CEC generated 7 kWh/h and a total of 4 members consumed 10 kWh/h. So working on an even split algorithm (another algorithm could be used), C will be charged the CEC rate for the 1 kWh/h (leaving 6 kWh/h) used by D, E & F so they will be charged 2 kWh/h to the CEC rate and the remaining 1 kW/h to the retailer [3].

In summary, the power produced/consumed per hour will be captured in the smart contract for all members of the CEC. In this way, user´s will not be charged for electricity in the hours from the commercial vendor when they are in surplus. Also,

in the hours where the whole CEC is in surplus and any member is in negative (including naturally consumers) then the surplus is shared between the members (not in surplus that hour) so that they benefit from the renewable energy. Note the local renewable energy rate could be less than buying from their commercial vendor to make it attractive, or they charge it at a higher rate for eco-conscious consumers. This depends on the national/regional business cases and policy.

The above example endeavors to provide for a fair billing scheme, to incentivize citizens to join up to local energy trading to buy/sell their energy within a local community, getting a better return for their money than selling surplus to the DSO and also enabling them to purchase local green energy at potentially a better tariff than that provided by their commercial vendor over the local distribution.

Different CEC P2P trading models should be explored with the possibility to offer more than one type of trading model.

Input data.

Account Provisioning

- Service-User provisioning interface for the service (CRUD)

Access Data layer

- Access Smart Meter Data (R)

Output data.

Account Provisioning

- Service-User provisioning interface for the service (CRUD)

SC Management

- SC Management and provisioning interface (CRU)

Smart Meter Output Data

- Smart Meter input data interface (CRU)

Interconnections.

- Analytics Service Management (ASM)
- 3rd Party DB
- Unified API
- Blockchain

Dependencies.

- Analytics Service Management (ASM)
- 3rd Party DB
- Unified API
- Blockchain

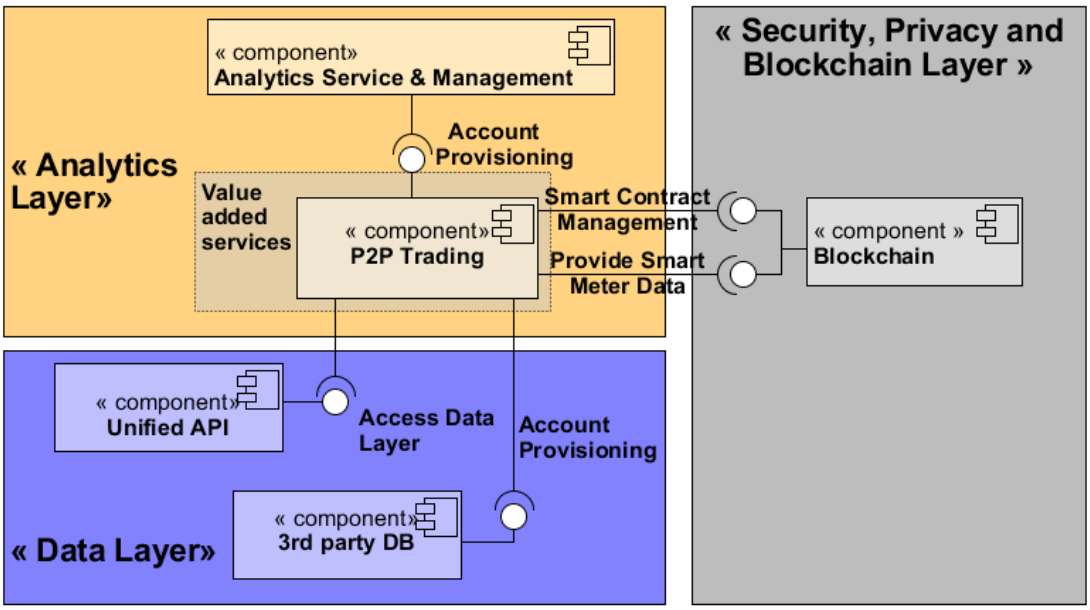


Figure 31. E7.1- P2P Trading UML diagram

5.3.7 Analytics Layer Summary

Next Figure 32 presents the whole Data Layer UML Component Diagram.

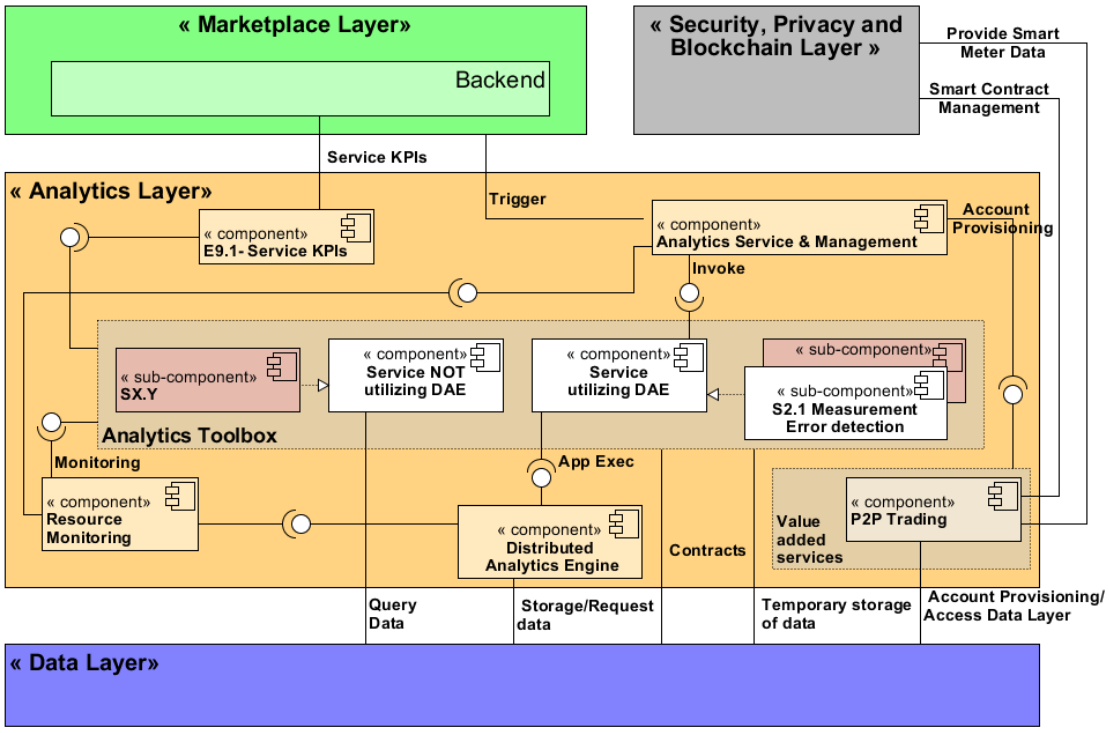


Figure 32. Analytics Layer UML diagram

5.4 Marketplace Layer. Implementation View

Table 10 lists all modules that conform the Marketplace layer including their responsible partner.

Marketplace Layer		
#4.1	Contracting	ATOS
#4.2	Search	WEP
#4.3	Provisioning	JSI
#4.4	Billing	WEP
#4.5	Management and governance	WEP
#4.6	Advertising	WEP
#4.7	Publish	WEP
#4.8	Open innovation	ICOM
#4.9	Secure access	ATOS
#4.10	Share data	ICOM
#4.11	Access data	ICOM
#4.12	Service composition	WEP
#4.13	Notifications	ALL
#4.14	Account registration	ATOS

Table 10. Modules of Marketplace Layer

5.4.1 Contracting

General description. Contracting supports the issuing of a contract as a W3C Verifiable Credential to a user, as requested by a business logic function. Upon the user accepting the contract, it is issued as a VC to the user's SSI Wallet / Agent. The contract details are actually stored inside the user´s account information, and thus every time the user contracts a new service, this is specified under her account, and the user is issued with a new Account VC.

The updated Account VC is requested to be presented back (signed by the user) to the contracting function to provide proof that the contract is agreed by both parties in the updated account and is stored in the BD4OPEM Marketplace storage (detailing in section 6.1.3.4). As well as keeping a record of the account´s contracted service(s) it is able to serve as validation of the user's contracts when needed.

Different business logic functions can request to issue contract to different users that access the marketplace e.g. Provision module calls the Contracting for a Service-User & Data-User; Publish module calls the Contracting module for an SP administrator, Share module calls the Contracting module for a DP Administrator etc.

Functionality.

- Issue contract as a Verifiable Credential
- VC presentation signed by the user to seal the contract

Input data.

- Access Token (with account data)
- Claims received from other BD4OPEM business logic requested to be issued in the contract VC for the redirected user.
    - Redirected user
    - SSI Wallet / Agent DID Auth
- Account VC presented by user

Output data.

- Updated Account VC issued to a user (with new contract details added)
- Updated Account VC presented by a user to be stored in the Marketplace DB (with new contract details added)

Interconnections.

- Marketplace Dashboard: BD4OPEM business logic module requesting a contract (e.g. Provisioning for Service-Users)
- Marketplace Storage
- SSI Agent
- Open innovation
- Provisioning
- Publish
- Share Data

Dependencies.

- All account contract claims are fully specified and supported by Contracting
- Blockchain to support the SSI
- Registration of all actor types (Service User, Service Provider administrator, Data Provider Administrator etc.)

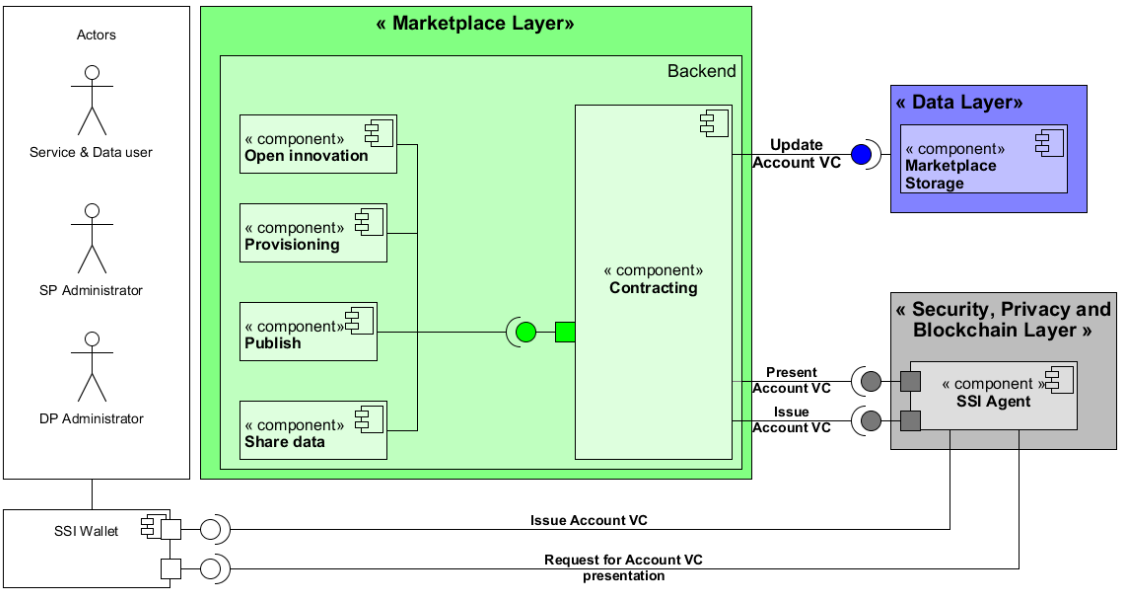


Figure 33. Contracting UML diagram

5.4.2 Search

General description. The BD4OPEM Marketplace shall provide a unified interface to search for data and services. A customer requesting data from the Marketplace will need to pass the authentication/authorization mechanisms to search for data and services.

Functionality. Search data, search services

Input data.

- Search query (e.g. words to search)

Output data.

- List of data and services that match the search query

Interconnections.

- Marketplace Storage
- Marketplace Dashboard

Dependencies.

- Data catalogue
- Service catalogue

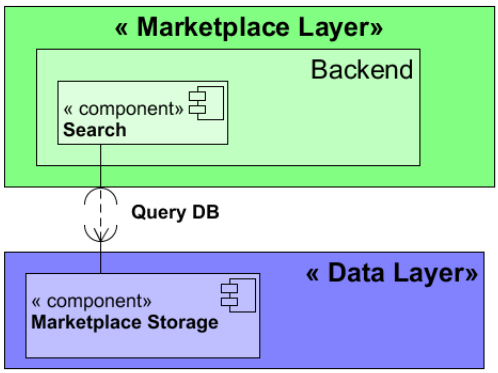


Figure 34. Search UML diagram

5.4.3 Provisioning

General description. Provide a service for a customer. Three types of service are envisioned: data service, analytic service and energy grid service

Functionality. The module has the following abilities:

- Provides a user customized instance of a service to the user according to the contract
- Notifies the user of service availability
- Notifies the service provider that the user has received the service

Input data.

- Service contract template/filled; defines the service specification and needed data for the service provisioning

Output data.

- Service information - API

Interconnections.

- Contracting
- Notification
- Management and governance
- Marketplace Dashboard

Dependencies.

- Contract

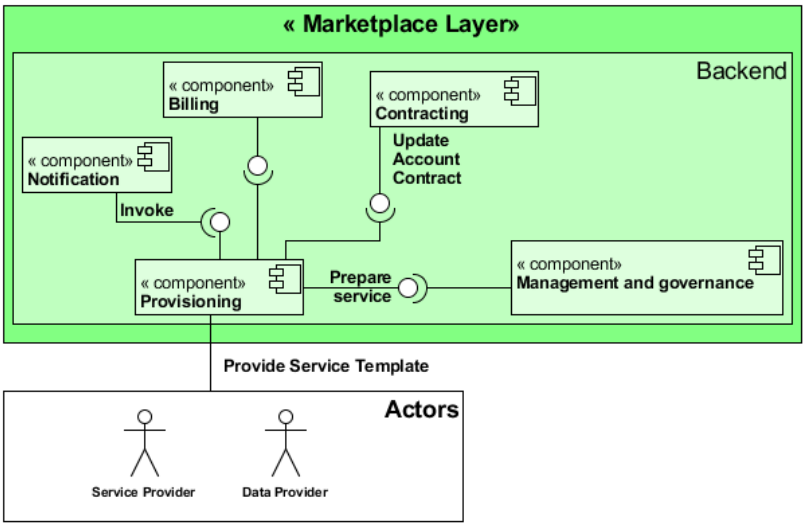


Figure 35. Provisioning UML diagram

5.4.4 Billing

General description. After the contract has been established, the customer needs to proceed with the payment. The payment formula can be different and based, for example on number of accesses. The customer then can access data or services provided through the marketplace. Some data or services could be granted over one-time payment.

Functionality. Provide a way to process payments

Input data.

- Payments details
- Authentication details

Output data. Payment processing

Interconnections.

- Marketplace Storage
- Payment Gateway
- Provisioning
- Marketplace Dashboard

Dependencies.

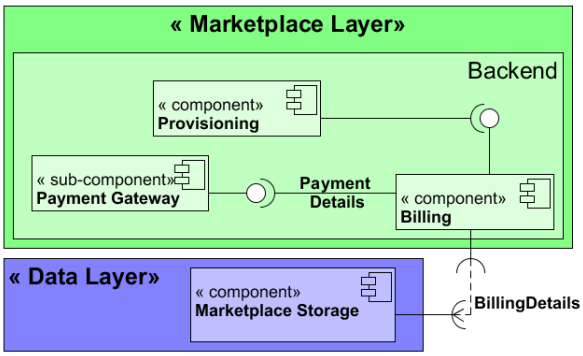


Figure 36. Billing UML diagram

5.4.5 Management and governance

General description. How the services are started, administered, revoked, who is the owner, security policies for the services, the life cycle of the services etc. Life cycle could include several sub-use cases (e.g. assign, deploy, start, stop, revoke)

Functionality. Manage and handle services through:

- Administer a service
- Define ownership of services
- Give access to services (assign)
- Revoke access to services
- Define security policies for accessing services
- Define services life cycle (initiation, deployment, assign, revoke)
- Delete services from the marketplace

Input data. Service to be managed

Output data. Complete control over service

Interconnections.

- Provisioning
- Analytics Service and Management
- Analytics toolbox: service utilizing DAE

Dependencies. services

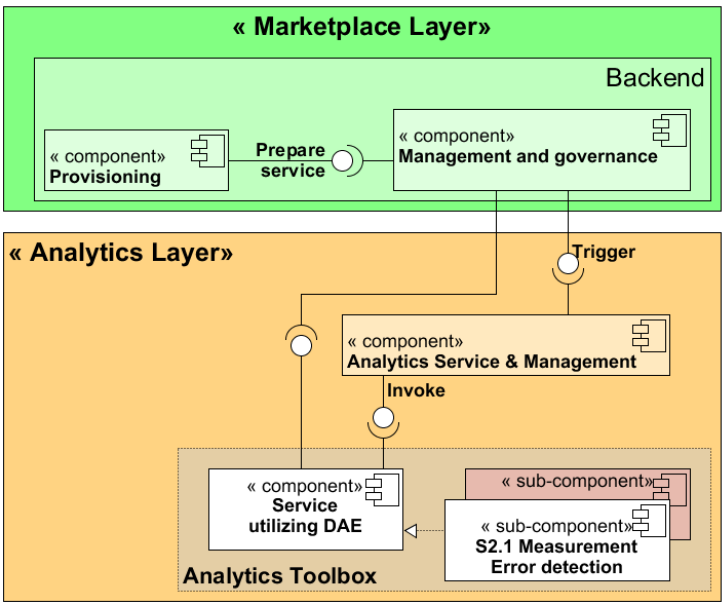


Figure 37. Management and governance UML diagram

5.4.6 Advertising

General description. Service and data providers may want to advertise their data and services. The marketplace should provide through an ads mechanism a way to create landing pages which summarize the data and services proposed, in a way that is appealing to customers.

Landing pages are created by services and data providers for the sake of advertising their data and services.

Functionality. Provide a way to advertise data and services provided by the marketplace.

Input data. Services to be advertised

Output data. Landing pages for service advertising

Interconnections.

- Publish
- Ads mechanism
- Marketplace Dashboard

Dependencies. Service catalogue

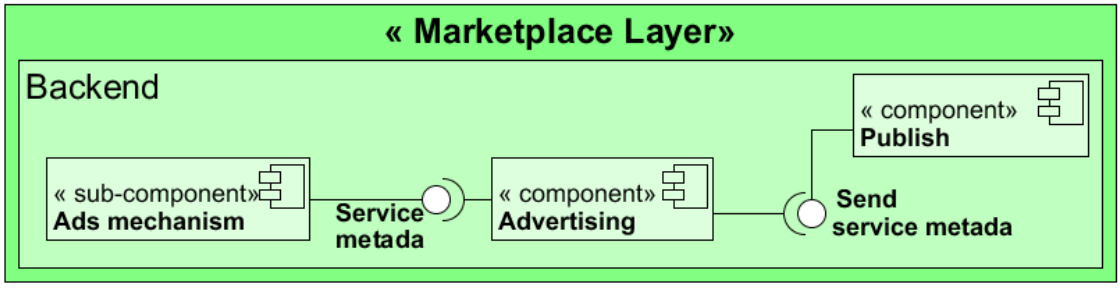


Figure 38. Advertising UML diagram

5.4.7 Publish

General description. The BD4OPEM platform should allow service providers to provide their own services and publish them to the marketplace

Functionality. Provide a way to publish services to the marketplace

Marketplace Operator:

- Define the service publishing process
- Provide a descriptive document including requirements (e.g. technical, functional and non-functional), contract and billing details.
- Define the service approval process
- Integrate approved services into a portfolio (either manually or by automatic procedure)
- Promote in portfolio
- Remove a service (e.g. due to violations to the guidelines, request by the service provider, etc.)

Service Provider:

- Prepare service
- Apply for publishing
- If approved and agreed, sign contract
- Submit service
- Publish service
- Update service

Input data. Service to be published

Output data.

- Service status
- Update Account contract

Interconnections.

- Contracting
- E9.1- Service KPIs
- Advertising
- Marketplace Dashboard

Dependencies. Services

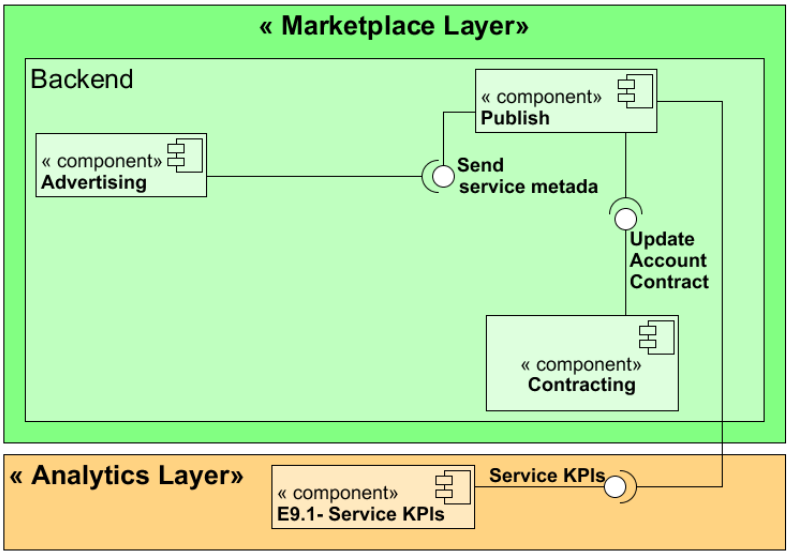


Figure 39. Publish UML diagram

5.4.8 Open innovation

General description. Open innovation exposes an interface for requesting a new service from the Marketplace. The potential Service User can make an Open innovation Service Request, whilst Service Providers will be informed through the Marketplace about the new Open innovation Service Request, and they can make their proposal (Open innovation Service Procurement). The potential Service User can review the proposals and make a choice. Service User and Service Provider can then sign a contract and proceed to sharing data and implementing the service.

Functionality.

- Ability to make an Open innovation Service Request
- Ability to make an Open innovation Service Procurement
- Support of a Q&A section for providing clarification
- Trigger data sharing process
- Trigger service contracting process

Input data.

- Open innovation Service Request Template
- Open innovation Service Proposal Template

Output data. N/A

Interconnections.

- Contracting
- Marketplace Storage
- Marketplace Dashboard

Dependencies. N/A

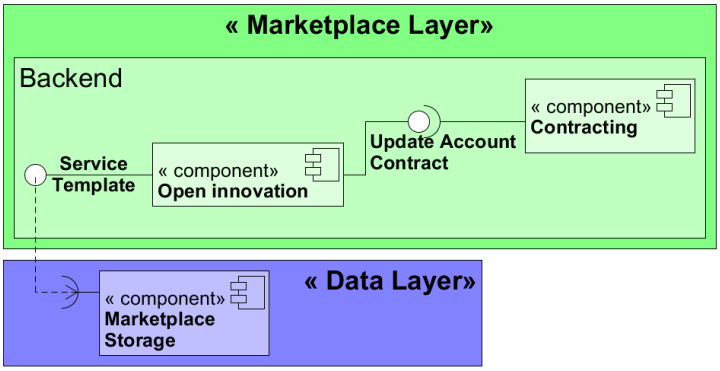


Figure 40. Open innovation UML diagram

5.4.9 Secure access

General description.

Functionality. To provide account registration and secure access for the marketplace and its services, with the following functionality:

1) To register accounts upon first entry to the Marketplace so that registered account holders can subsequently identify themselves and their associated account to be able to contract services in the Marketplace.
    - Note, as described in section 4.6, account holders will primarily be companies and organistaions, and they will themselves be responsible for issuing company and organisation credentials to their employees and members with a role that will authorise them to enter the Marketplace, and act on behalf of their company / organisation. Account registration is a sub component of Secure Access and as such is described in its own section **Error! Reference source not found.**.
2) To protect access to the marketplace through secure authentication, based on Decentralised Identifiers (DIDs) [9] and Verifiable Credentials (VCs) [10].
3) Provide capability for services and any platform modules to authorize access based on the presented Verifiable Credentials.
    - This is facilitated by the creation of a self-describing Access Token that includes all identity and account claims, so that they are made available to the functional modules as they are accessed in the platform by the user.

Aside: Secure access of data entering from Data Provider systems could potentially also be provided by DIDs and Verifiable Credentials etc., however considering the range of different devices that could ingress data it is decided not to support DIDs and VCs for this ingress of data from the Data Providers. Instead, it is suggested this will be managed by Data Provider administrators logging into the marketplace and configuring the Public Keys of the DP systems that will be uploading data to the data layer, with secure access being recommended to be provided either through Mutual TLS Authentication or HTTPS Signature, depending on what is

supported by the Data Provider. This will be addressed in the next iteration of the deliverable in the Data Layer.

All users and actors in the system that access the marketplace and its services will have a Decentralized Identifier and identity and associated account claims to access protected resources. Company employee representatives and organisation members will be issued with identity claims in a Verifiable Credential that will authorise them to act on behalf of their company / organisation with a specific role.

Roles:

- Administrator - Able to perform any of the actions by other roles
- DP - Able to request & manage Data Provider contracts
- SP – Able to publish & manage Service Provider contracts
- Service User – Able to provision & manage Marketplace Value Added Services
- Data User – Able to provision & manage Marketplace Data Services

When a Company Representative or Organisational Member accesses the Marketplace the presented claims will identify the person and the company / organisation DID, and it will be checked that an active Account Contract exists for the company / organisation they represent.

The Account Contract VC will detail all services that may be contracted under this account under e.g SP, DP , Marketplace Services, and the representative will be able to access them as per the role they presented.

Example scenarios for a private individual and a company representative accessing the Marketplace is described with a sequence diagram in section 6.1.3.4 for sceanrios 3 and 4 respectively.

Once a user and associated company / organisation have been authenticated and proved they have the necessary Marketplace identity and account claims an Access Token will be generated to authorise the access. This is described in more detail below:

- The Account Contract VC signature is verified against the BD4OPEM public key and the Account contract stored in the Marketplace DB; subsequently, a JWT Token including all the validated account information is created and is signed (JWS) and then encoded (JWE). The resultant token serves as a self-describing Access Token that is returned back to the callback address of the dashboard / App that called the authentication.
- If the user did not have the Account Contract VC, the user would be redirected to Account registration to be issued with an Account Contract VC.
- If there is a mismatch in the Account Contract VC copy of the Marketplace Storage and the Account Contract VC, the user is asked to check this with the registration management to get a new verified Account Contract VC that matches the stored VC. A mismatch could have occurred if the user or administrator removed a service or changed the status.
- The Access Token generated will hold all relevant information that is in the Account Contract VC and any identity claims previously presented, so that any module in the platform that is called for this user can decode the Access Token and handle the user appropriately according to the identity and account

claims. This will take a JSON format and a non-normative example is given below:

```
{
  "iss": "https://server.example.com",
  "sub": "did:example:123456789abcdefghi",
  "exp": 1311281970,
  "iat": 1311280970,
  "auth_time": 1311280969,
  "profile": {"Role": "Service-User", "Name": null, "Organisation":"example.com"},
  "charging": {"accountName": "xxx", accountNum: "112376278782348787"},
   "Services": ["Service": {"Name": P2Ptrading, dataSetType: SM,
dataSetId:12345678353, "dataRole": "owner"}]
}
```

The secure ccess flow works in the same manner, no matter if the Service-User is a person accessing a web page, web app, mobile app, standalone application or is a machine accessing the platform automatically.

The SSI Agent will support various DID Authentication mechanisms to support all scenarios.

Note the time of the Access Token can be kept fairly low to what is expected of a typical user scenario and would then expire, necessitating the re-authentication of the user.

Input data.

DID Authentication Information:

- Register account request with account information
- Company Representative / Organsation Representative / private Individual identity claims
- Marketplace Account Claims
- Decentralised Identifier

Output data.

Account Contract VC

- Issued to the waalet of the individual / company / organsiation
- Stored in the Marketplace Storage

Access Token:

- JWT token claims e.g. expiry
    - Includes identity and account claims

Interconnections.

- Marketplace Storage
- Account registration
- SSI Agent

Dependencies.

- Blockchain

- Service Provisioning
- Marketplace Storage
- SP or DP Company SSI Agent to issue their company Rep Credentials

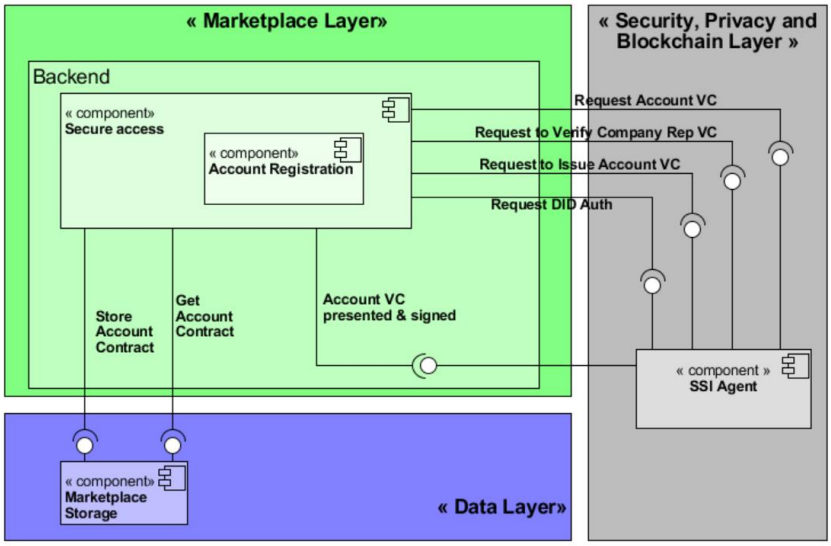


Figure 41. Secure access UML diagram

5.4.9.1 Account registration

General description. This module will register an account for the Marketplace so that the acciount holder can subsequently identify themselves using a Marketplace Account Contract VC, so to be able to contract services.

The main purpose of the Account Registration module is to issue an Account Contract VC so that the holder will be able to present this to securely access the Marketplace and identify the services contracted in the Marketplace as described in section 5.4.9.

Once the Account Contract VC is issued to the holder, it is requested that the holder accepts this and presents with the holders signature and it is then s saved in the marketplace storage so as to being able to validate the identification of the holder and associated services upon future access to the Marketplace.

Example account registration for a a company representative and private individual are described with a sequence diagram in section 6.1.3.4 for sceanrios 1 and 2.

Functionality. This module is responsible for managing the following actions:

a. DID Authentication between the actors wallets and the Marketplace SSI Agent
b. Issue an Account Contract VC to the account holder
c. Immediately after issuing the Account Contract VC, the holder is asked to confirm and present an Account Contract VC which is then stored in the Marketplace Storage.

Input data.

- Register account request

- Company / organisaation Representative VC (In case the user is also a company administrator)

Output data.

- Account Contract VC

Interconnections.

- Marketplace Dashboard
- SSI Agent
- Secure Access

Dependencies.

- Marketplace Storage
- SP or DP Company SSI Agent to issue their Company / Organsiation Rep Credentials

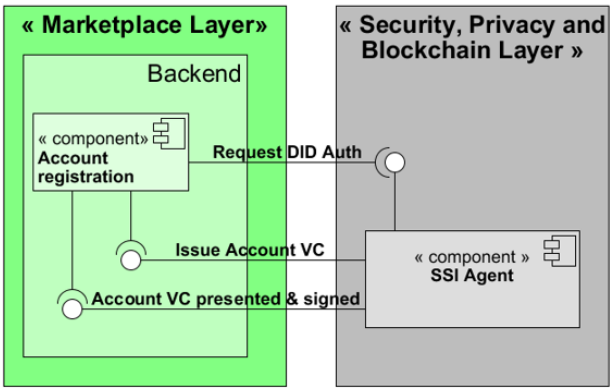


Figure 42. Account registration UML diagram

5.4.10 Share data

General description. Marketplace enables the sharing of data among stakeholders for the realization of the various services offered or for the creation of new services. This module defines how data providers' data is exported to the marketplace and in what manner could be shared with other actors in the system. The data needs to be exported in a pre-defined and standardized way.

Functionality.

- Ability to add data for various purposes (i.e. sell, open)
- Manage data

Input data.

- Request to "publish" data

Output data.

- Data Contract info

Interconnections.

- Marketplace Storage
- Data Management and Monitoring
- Contracting
- Marketplace Dashboard

Dependencies.

- Marketplace Dashboard (Data Marketplace)
- Contracting
- Data Management and Monitoring

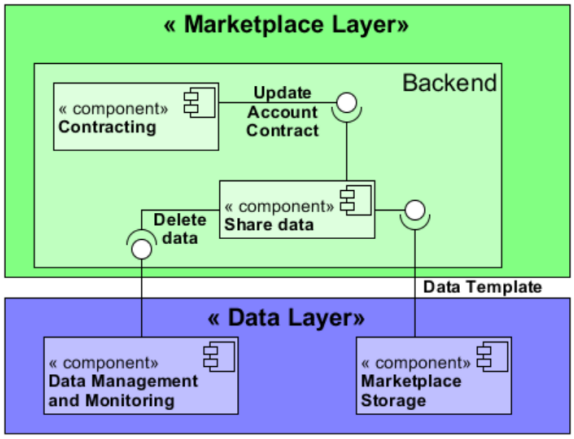


Figure 43. Share data UML diagram

5.4.11 Access data

General description. This module should allow data user to view their contracted data and download (possibly) data sets from the marketplace. The Data User can access the description of already contracted data (metadata query). The Data User can then make a request to download data, by making a query through Unified API.

Functionality.

- Ability to view data information (metadata)
- Ability to download data

Input data.

- Request to view or download data (optional query parameters)

Output data.

- Data Template, Data File (or url to download data)

Interconnections.

- Marketplace Storage
- Unified API
- Analytics Storage

Dependencies.

- Marketplace Dashboard (Data Marketplace)
- Contracting

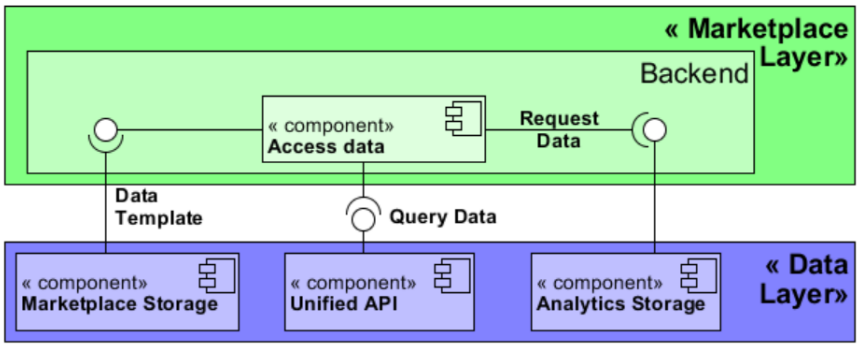


Figure 44. Access data UML diagram

5.4.12 Service composition

General description. Customer may need to access the marketplace to request a combination of different services. The marketplace should allow such combination.

Interconnection between services and which services could be combined should be defined by the marketplace, with support of service(s) provider(s).

Customer needs access to all required services to benefit from service composition.

Access to the service(s) should be granted with access rights through a dedicated authorization service.

Functionality. Combining different services

Input data. Services to be combined

Output data. Combined services

Interconnections.

- Marketplace Storage
- Marketplace Dashboard

Dependencies. Service catalogue

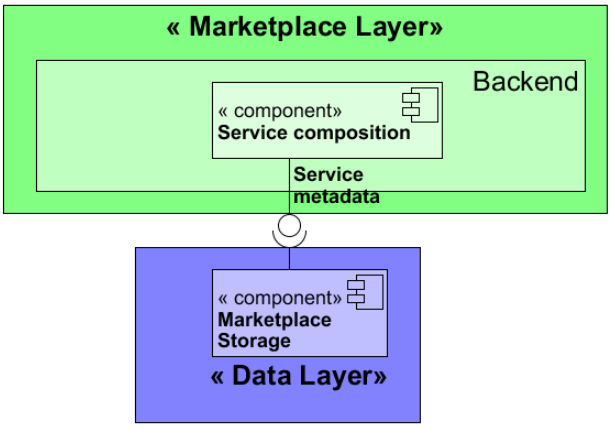


Figure 45. Service Composition UML diagram

5.4.13 Notifications

General description. This module will handle the different notifications between platform-stakeholders and module-module.

Functionality. Exchange the notifications from-to provisioning

Input data. Provisioning requests

Output data. Request

Interconnections.

- Provisioning

Dependencies.

- Provisioning

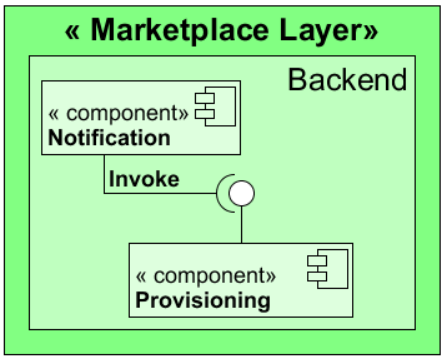


Figure 46. Notifications UML diagram

5.4.14 Marketplace Layer Summary

Next Figure 47 presents the whole Data Layer UML Diagram

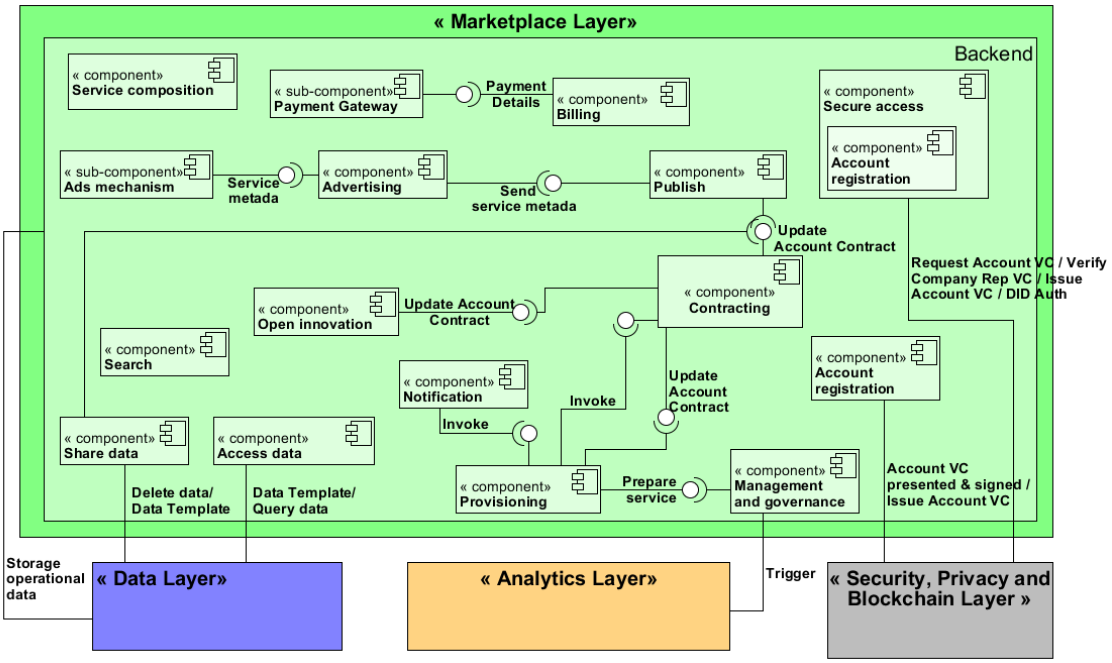


Figure 47. Marketplace Layer UML diagram

5.5 Security, Privacy and Blockchain Layer. Implementation View

Table 11 lists all modules that conform the Marketplace layer, including also its responsible partner.

Marketplace Layer		
#5.1	SSI Wallet	ATOS
#5.2	Blockchain	ATOS
#5.3	SSI Agent	ATOS
#5.4	SIEM	WEP

Table 11. Modules of Security, Privacy and Blockchain Layer

5.5.1 SSI Wallet

General description. The SSI Wallets are external to and independent of the Marketplace. Users use them to store W3C Verifiable Credentials issued by the marketplace. Other systems external to the marketplace can also update the SSI Wallet with any other type of VCs.

Functionality.

- DID Auth
- Store Account VCs issued by Marketplace SSI Agent
- Present Account VCs to the SSI Agent

Input data.

- DID
- Account VC

Output data.

- DID
- Updated SSI Wallet with the new Verifiable Credential
- Account VC

Interconnections.

- SSI Agent
- Blockchain

Dependencies. A device to host the SSI Wallet app

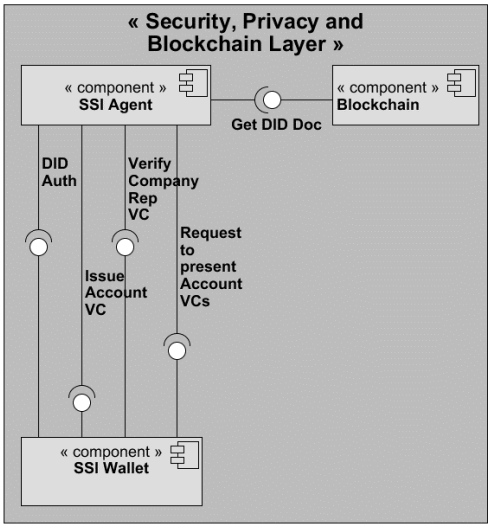


Figure 48. SSI Wallet UML diagram

5.5.2 Blockchain

General description. A Blockchain is a single, agreed upon and distributed record that enables the storage of information grouped into cryptographically signed blocks. The information stored in previous blocks can't be remove or edited and variables can be only modified by adding subsequent blocks to the existing blockchain. As new records are created, they are first checked and validated by the network nodes and then added to a new block that is linked to the chain. This property allows the "blockchain" data structure to act as a database containing an irrefutable history of

information. It will be used to store essential data about the BD4OPEM SSI Agent and its credential definitions, and to execute the P2P Trading Smart Contract.

Regarding the SSI Agent data, in order not to store personal information in the blockchain, it is stored in the form of verifiable credentials and is managed only by the wallet of the person or company.

The information stored in the blockchain is the minimum necessary to check the reliability of the SSI Agent and therefore of its issued credentials. This is Public DIDs, Schemas, Credential Definitions [10].

Public DIDs

Every company is allowed to issue credentials. However, verifiers can only be certain the validity of a VC, if the issuer's DID and public keys are previously stored in a blockchain.

Schemas

The SSI Agent will define in the blockchain the structure of the issued BD4OPEM VCs. This makes it easier for verifiers to check that the VCs being used contain the necessary information to proceed. The scheme can be developed from scratch or use an existing scheme in the blockchain.

Credential Definitions

Before the SSI Agent can issue a credential, a credential definition must be defined in the blockchain. The credential definitions make use of the public DIDs and previously defined schemes to give validity to the issued VCs. Therefore, it is a requirement for the SSI Agent to store a DID, a schema, and a credential definition in the blockchain.

In addition, The P2P Trading module will develop a Smart Contract that has to be uploaded to the blockchain. A smart contract is a computer code that, contrary to what is usual, is not installed on a personal computer or a server. Once the code is running in a blockchain, it cannot be deleted or edited.

The Smart Contract is executed when the conditions established in the contract are met. Because the rules of the SC have been previously accepted and everything is transparent to all parties, the results of the SC execution cannot be repudiated.

For more details on the P2P Trading module, please refer to its corresponding section.

Functionality.

The functionalities of the blockchain can be divided into two different areas

1. Store the necessary public information about the SSI Agent and the credentials:
    a. SSI Agent Public DID
    b. VC Schemas
    c. VC Definition
2. Execute the P2P Trading developed Smart Contract in an autonomous manner

Input data.

- A transaction or series of transactions including the Public DID, the Schema and the VC Definition
- The P2P Trading Smart Contract

- Smart Meter Data

Output data.

- Blockchain transactions publicly stored in the blockchain and visible for external agents.
- Trust-based billing information for the companies integrating the Smart Grid.

Interconnections.

- P2P trading Module
- SSI Agent

Dependencies. A blockchain physical layer consisting of servers, edge nodes, IoT devices which act as nodes on the blockchain network. These are connected as a P2P network where Peers are equally privileged, equipotent participants in the network.

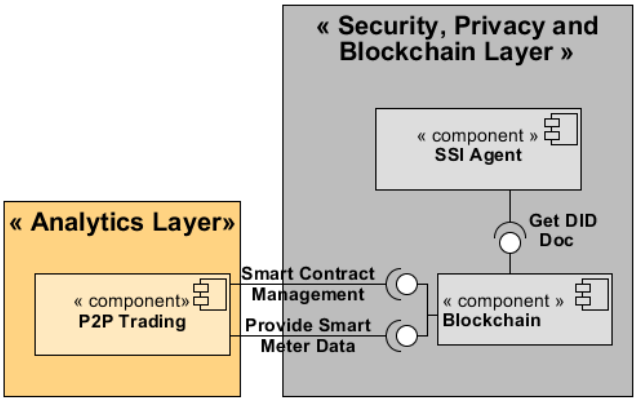


Figure 49. Blockchain UML diagram

5.5.3 SSI Agent

General description. The SSI Agents are responsible for performing DID Auth, issuing and verifying VCs. These actions are done by means of a low-level protocol named DIDCOMM. This protocol has a series of predefined messages between two agents to perform actions such as issuing credentials or verifying them.

The BD4OPEM SSI Agent will be the invoked by other modules for carrying out these functions e.g. to issue an Account VC on behalf of the Account Registration module. Account VC will be issued the first time a user registers in the Marketplace and will be subsequently updated when a service is contracted by the user.

Functionality.

- Performing DID Auth
- Issuing VCs
- Verifying VCs

Input data.

- Account data to be issued in the VC
- DID

Output data.

- Account VC
- DID Auth result

Interconnections.

- SSI Wallet
- Blochckain
- Secure access - Account Registration
- Contracting

Dependencies.

- Blockchain

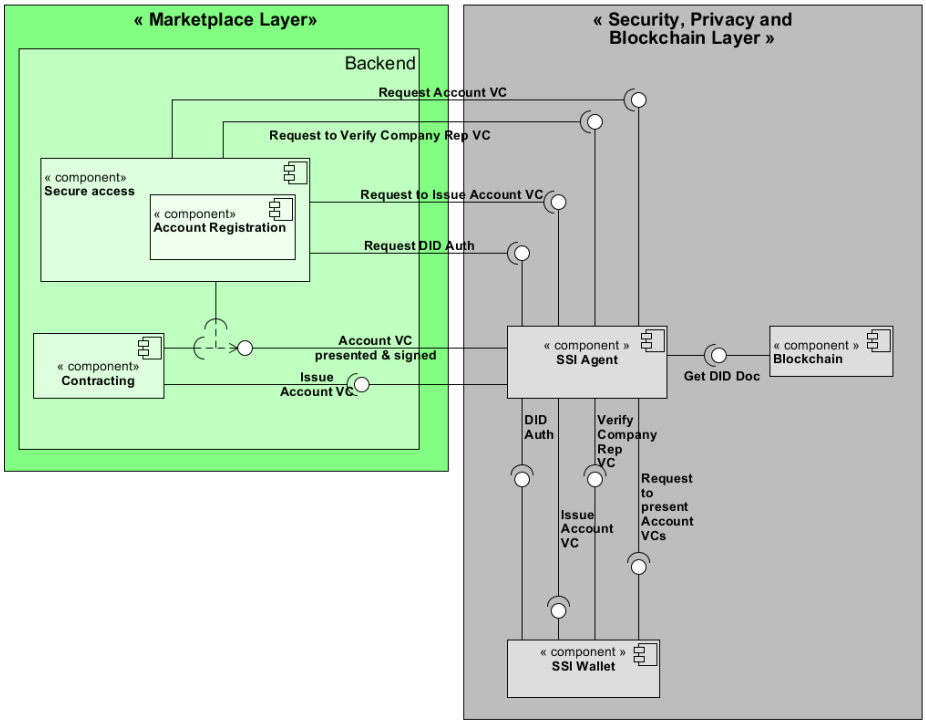


Figure 50. SSI Agent UML diagram

5.5.4 SIEM

General description. SIEM stands for Security Information and Event Management. A SIEM implementation aims to provide real-time analysis for security alerts and can also help security managers in responding to threats and incidents. The "SIM" side of SIEM is basically log data analysis and reporting, whilst "SEM" focuses on monitoring and correlated real-time events.

BD4OPEM SIEM implementation will be external and independent of the Marketplace. It will be able to aggregate logs produced by each component of the architecture, correlate events raised by different actors (both human and technical), turn them in information useful for Marketplace operators and/or raising alerts based on defined rules.

All data, collected and produced by the SIEM module will be subjected to long-term retention in order to to allow further analysis and pattern discovery.

Functionality.

- Collect and aggregate logs
- Collect and correlate events
- Raise alerts

Input data.

- Logs in various format
- Events raised by the platform

Output data.

- Alerts
- Threat reports

Interconnections.

- Data Lake
- 3rd party DB
- Resource Monitoring
- Potentially every BD4OPEM component
- Device Adaptor

Dependencies. N/A

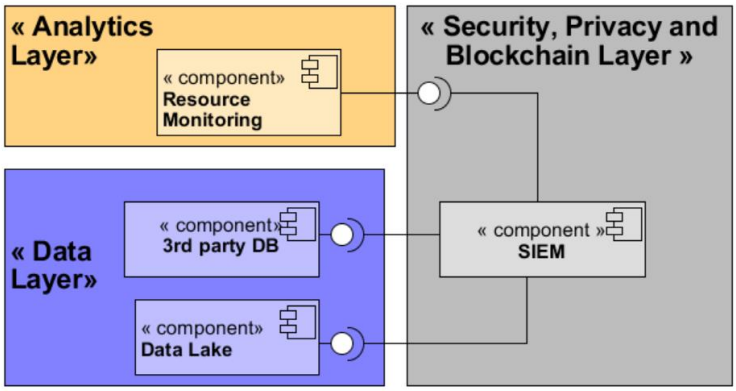


Figure 51. SIEM UML diagram

5.6 Implementation View Summary

Figure 52 below, presents the complete UML diagram of the BD4OPEM platform.

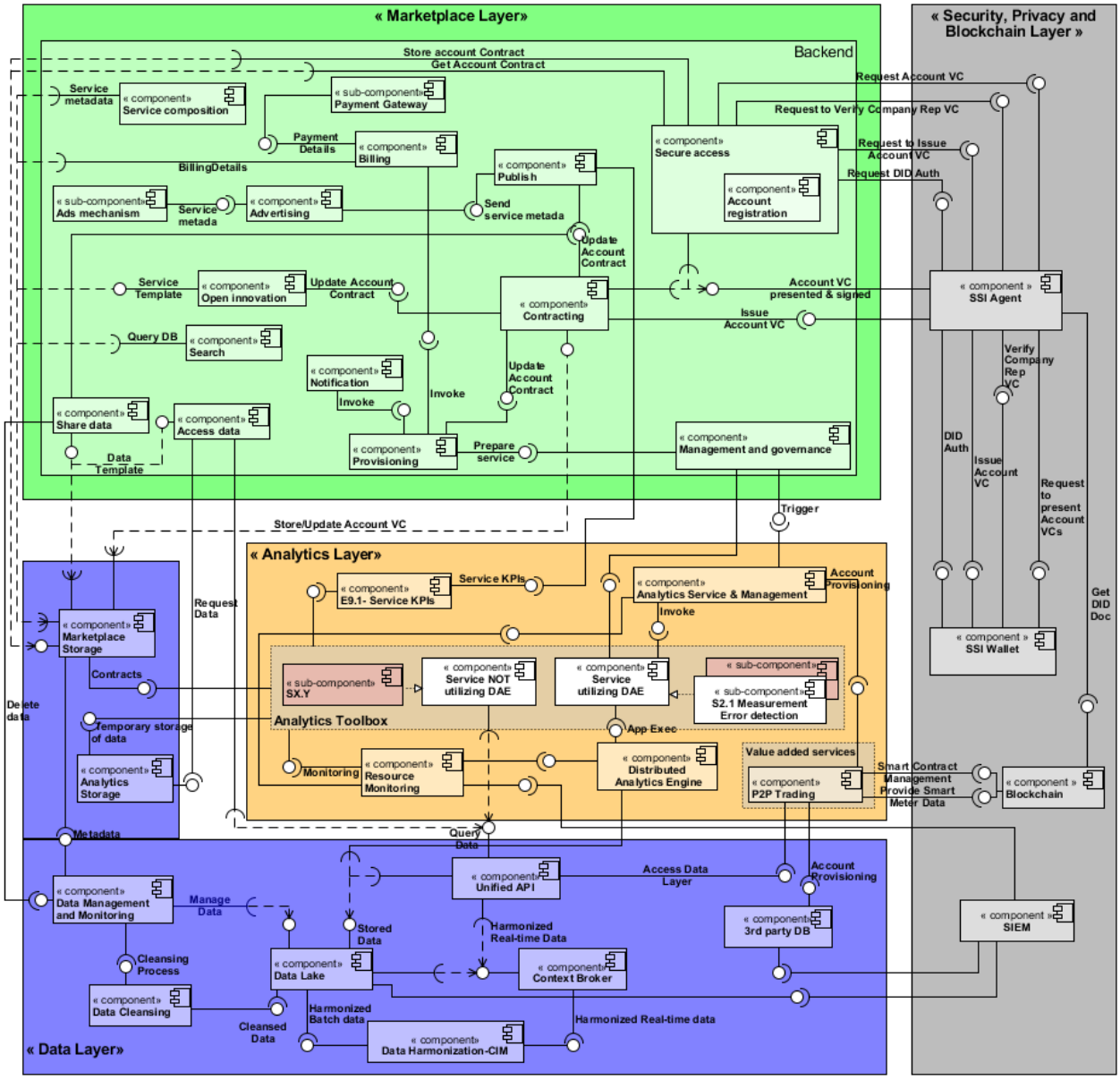


Figure 52. BD4OPEM UML diagram (Backend)

6 BD4OPEM reference Architecture. Process View

This section is dedicated to describing the dynamic aspect of BD4OPEM platform addressing the running systems behaviour and the way in which these processes communicate.

Process view takes as input the Use Cases, the "process perspective with the actors involved", and details steps-by-steps the operational workflow of the components that make up the system. For that, the following subsections present the sequence diagrams where the READER is able to identify the different implementation view in process groups that interact with each other.

BD4OPEM project has identified 10 business use cases (BUC) and 24 system use cases (SUC), as shown in Table 12**Table 12.** BUC and SUC of BD4OPEM

. BUCs are focused on those goals achieved through business processes played by the business actors, previously detailed in section 5. In contrast, SUCs pivot on the interactions among the relevant system parts for achieving the prior processes.

BUC	BUC M1: Share data on the marketplace BUC M2: Publish service on the marketplace BUC M3: Provision a service BUC M4: Open innovation proposal for a new service BUC E1: Flexibility services for BRPs BUC E2: Flexibility services for DSOs BUC E3: Cost optimization BUC E4: P2P energy trading BUC E5: Detect non-technical losses BUC E6: Optimized investment management
SUC	SUC M1: Data export SUC M2: Secure access SUC M3: Contract service SUC M4: Billing SUC M5: Service composition SUC M6: Service advertisement SUC M7: Service management and governance SUC M8: Search SUC M9: Service provisioning SUC M10: Access data in the marketplace SUC M11: Data management and governance SUC M12: Requesting service through Open innovation SUC M13: Service Procurement through Open innovation SUC E1.1: Topology

	SUC E1.2: Observability
	SUC E1.3: Predictive maintenance in electrical power systems
	SUC E1.4: Measurement errors detection
	SUC E1.5: Grid disturbance simulations
	SUC E1.6: Impact study PV, EV & new loads
	SUC E1.7: Fraud patterns detection
	SUC E1.8: Flexibility Forecast
	SUC E1.9: EV to Grid
	SUC E1.10: Energy forecasting
	SUC E1.11: Asset and investment planning

Table 12. BUC and SUC of BD4OPEM

6.1 Use cases and sequence diagrams

Following subsections tackle the presented use cases (BUCs and SUCs) through a template that includes: the number and name of the UC, the short description of the UC already provided in D2.1 and the sequence diagram.

The sequence diagrams describe in a graphical way the interactions between the actors and the modules of BD4OPEM platform. These interactions are detailed by the messages that flow among the modules, and the lifeline refers, in this case, to the elements that represent each participant involved in the process.

Some use cases present more than one sequence diagram since they cover different scenarios of the process for the same use case.

6.1.1 Data Layer sequence diagrams

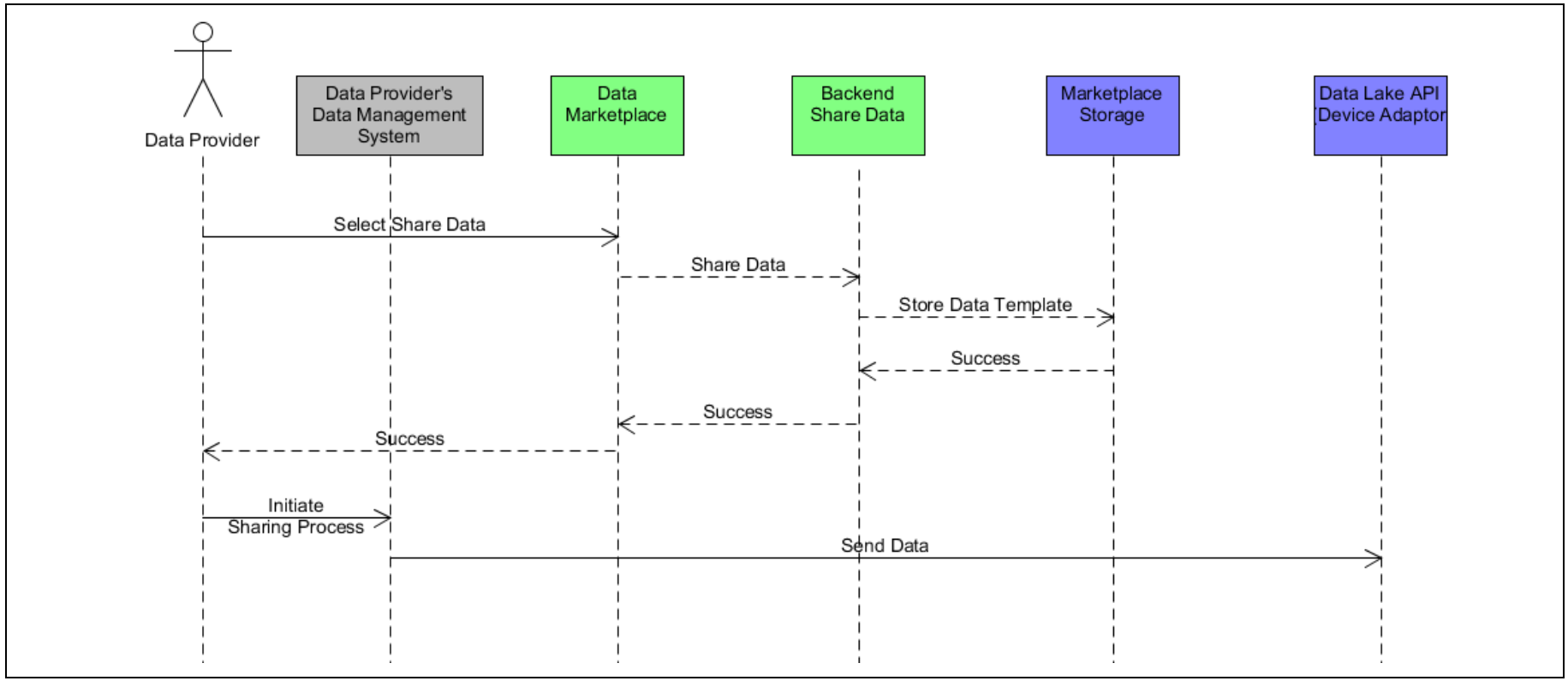
The Data Layer is mainly involved in the following four use cases.

No	Name
BUC M1 / SUC M1	Share data on the marketplace and Data export
SUC M10	Access data in the marketplace
SUC M11	Data management and governance

Table 13. Use cases of Data Layer

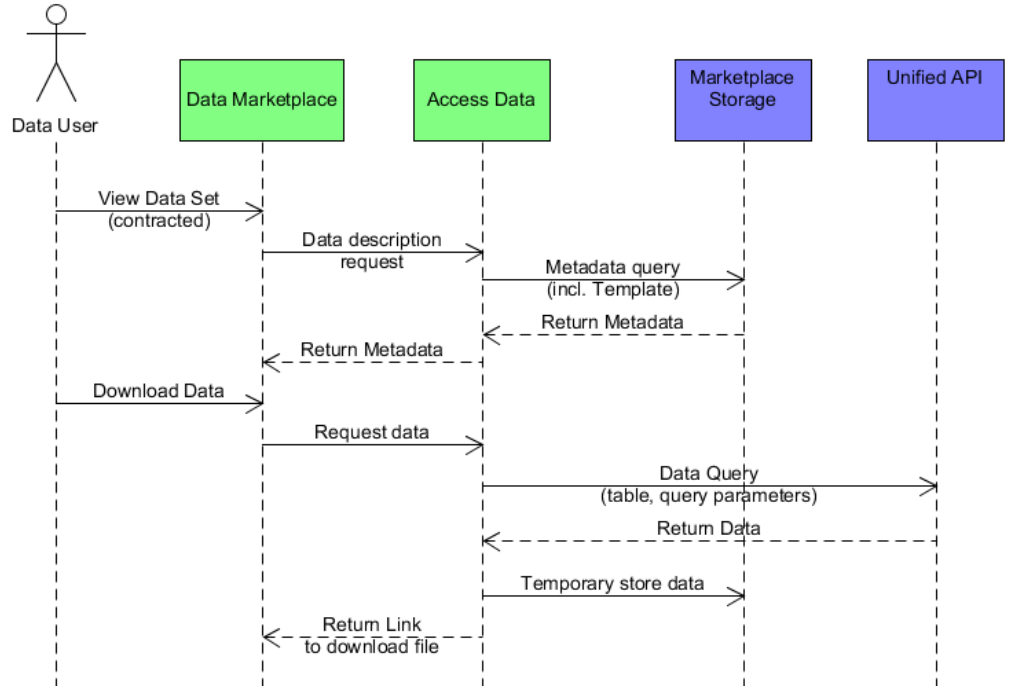
6.1.1.1 BUC M1 / SUC M1

No	Name
BUC M1 /SUC M1	Share data on the marketplace and Data export
Short description	The BD4OPEM Marketplace shall enable data sharing among registered users of the system. Data can be shared with the Marketplace by relevant stakeholders for various reasons:• Due to a contract for enabling services offered via the Marketplace ▪ Due to a request for open innovation ▪ For facilitating new services (e.g. open data, satisfy regulation) The Data Provider is a party sharing the data, who need to provide a description of the data providing information on sovereignty (open/free, restricted), constraints related to data storage (e.g. country of origin), as well as technical information related to data (i.e. velocity, volume) and termination conditions. Data Provider should be the owner of the data or authorized entity for sharing the data, whilst a contract need to be established with the platform. Upon successfully registering a data source, the Data Provider can initiate the data sharing process, where data can be exported to the Marketplace in a pre-defined / standardized way.


Data Provider
Data Provider's Data Management System
Data Marketplace
Backend Share Data
Marketplace Storage
Data Lake API Device Adaptor
Select Share Data
Share Data
Store Data Template
Success
Success
Success
Initiate Sharing Process
Send Data

6.1.1.2 SUC M10

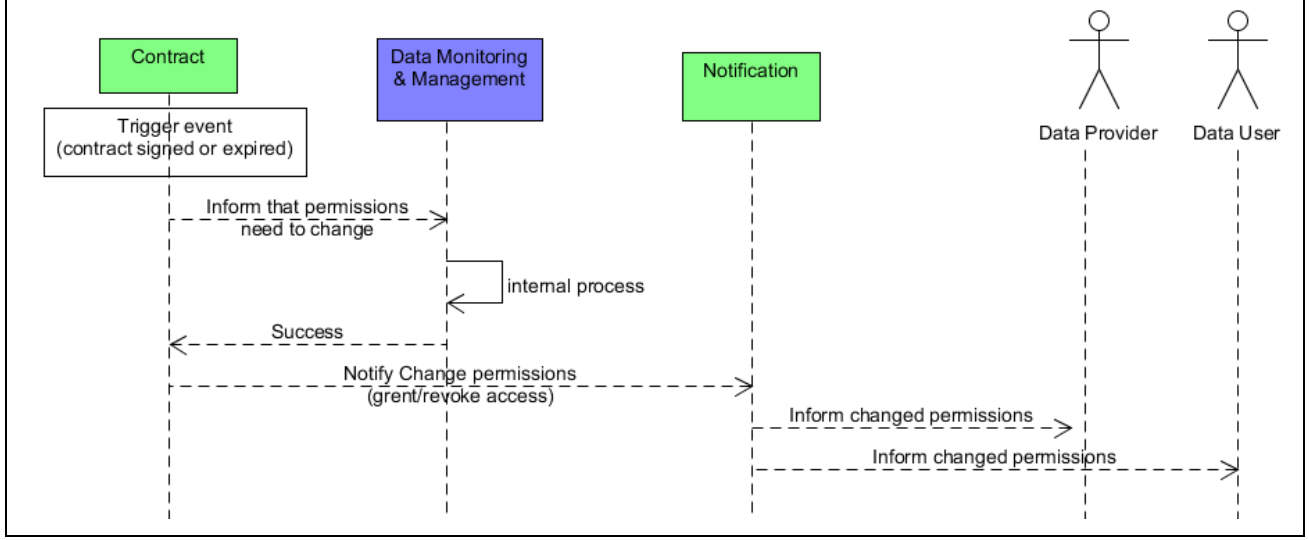
No	Name
SUC M10	Access data in the marketplace
Short description	The Marketplace shall provide the capability to a Data User for manually downloading data stored in the data lake. A Data User requesting data from the Marketplace will need to pass the authentication/authorization mechanisms in order to query and access data whilst a contract for accessing data must be already signed.The Data user can view their contracted data sources and request access to data sets stored in the data lake of the marketplace, by providing specific queries. Following this query request the Marketplace will provide a link to download the requested data from a temporary data store.



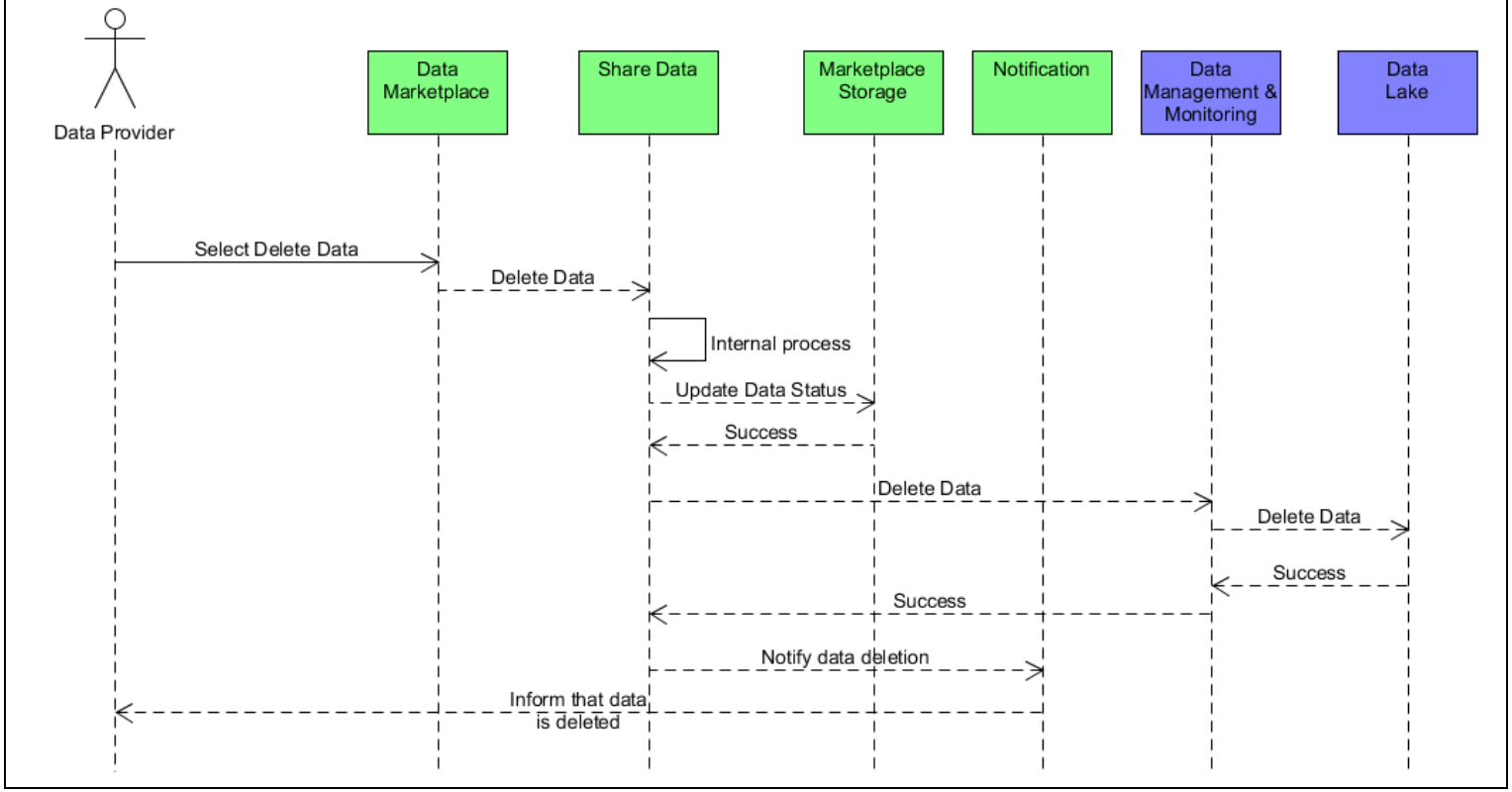
6.1.1.3 SUC M11

No	Name
SUC M11	Data management and governance
Short description	Data Management and Monitoring includes tasks that ensure the organization and supervision of the data lake. The module is in charge of modifying data permissions (grant or revoke access to data, according to contract changes), deleting data and handling of the data after ingestion (invoking the data cleansing process and arranging the storing of data). All of the tasks create reports or metadata, so that any operation happening in the data lake is monitored, safeguarded and documented.

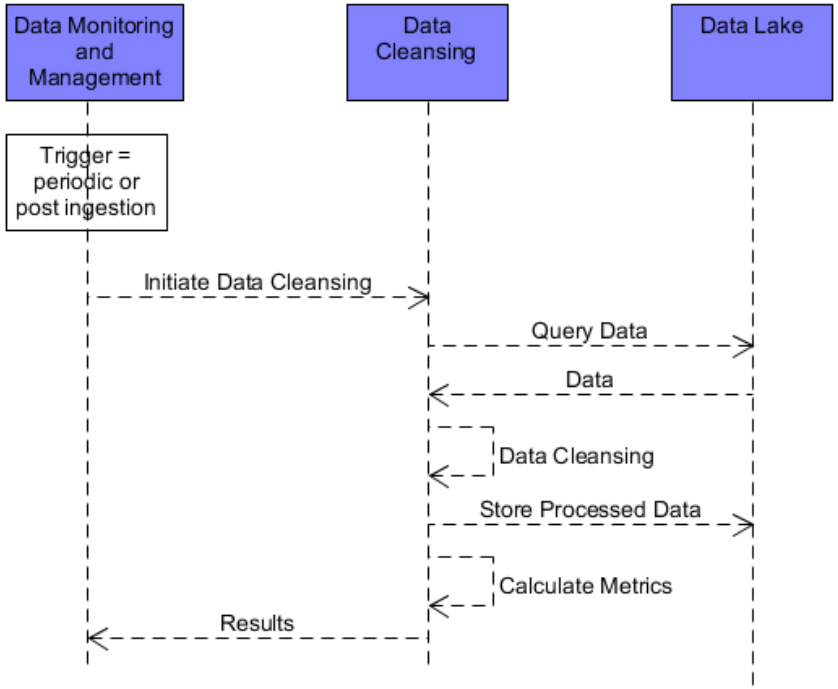
No	Name
SUC M11 (Scenario 1)	Data management and governance (Delete Data)



No	Name
SUC M11 (Scenario 2)	Data management and governance (Delete Data)



No	Name
SUC M11 (Scenario 3)	Data management and governance (Post-ingestion Processing)



6.1.2 Analytics Layer sequence diagrams

Analytics Layer involves all the use cases listed in Table 14 below, representing all the services that will be offered to the stakeholders through the Marketplace.

Considering the significant amount of running specifications that each service has although the project is still in its early stages, the following use cases, except for P2P trading are addressed by one representative use case, named *Service used by a service user*.

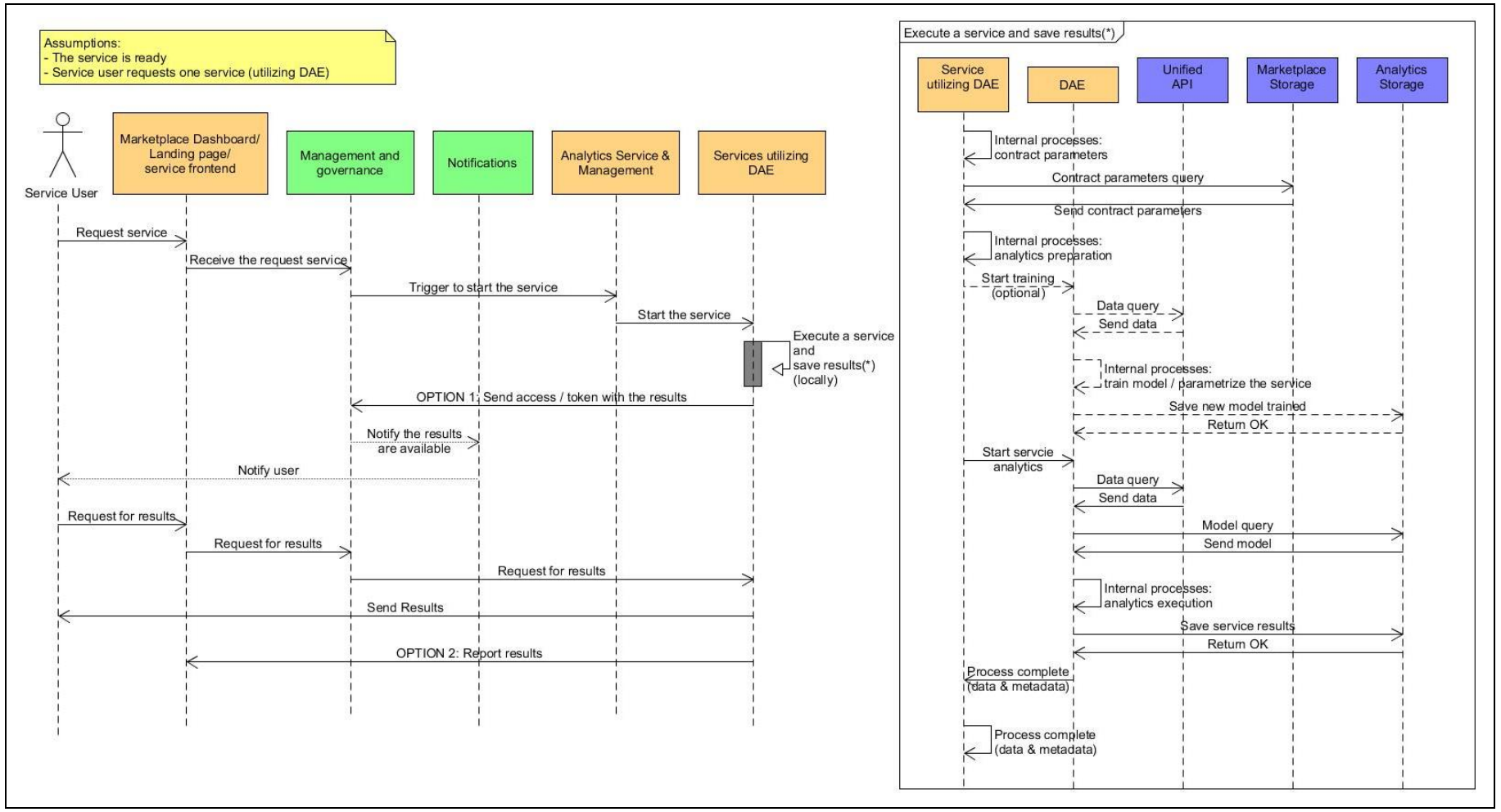
This use case details the dynamic process of the system so that a service user can use it. In fact, it comprises two possible cases, depending on whether the service uses or does not use DAE. The former, in turn, includes three possible scenarios depending on how the service returns his results to the Marketplace.

Use Cases of Analytic Layer	
BUC E1	Flexibility services for BRPs
BUC E2	Flexibility services for DSOs
BUC E4	P2P energy trading
BUC E5	Detect non-technical losses
SUC E1.1	Topology
SUC E1.2	Observability
SUC E1.3	Predictive maintenance in electrical power systems
SUC E1.4	Measurement errors detection
SUC E1.5	Grid disturbance simulations
SUC E1.6	Impact study PV, EV & new loads
SUC E1.7	Fraud patterns detection
SUC E1.8	Flexibility Forecast
SUC E1.9	EV to Grid
SUC E1.10	Energy forecasting
SUC E1.11	Asset and investment planning

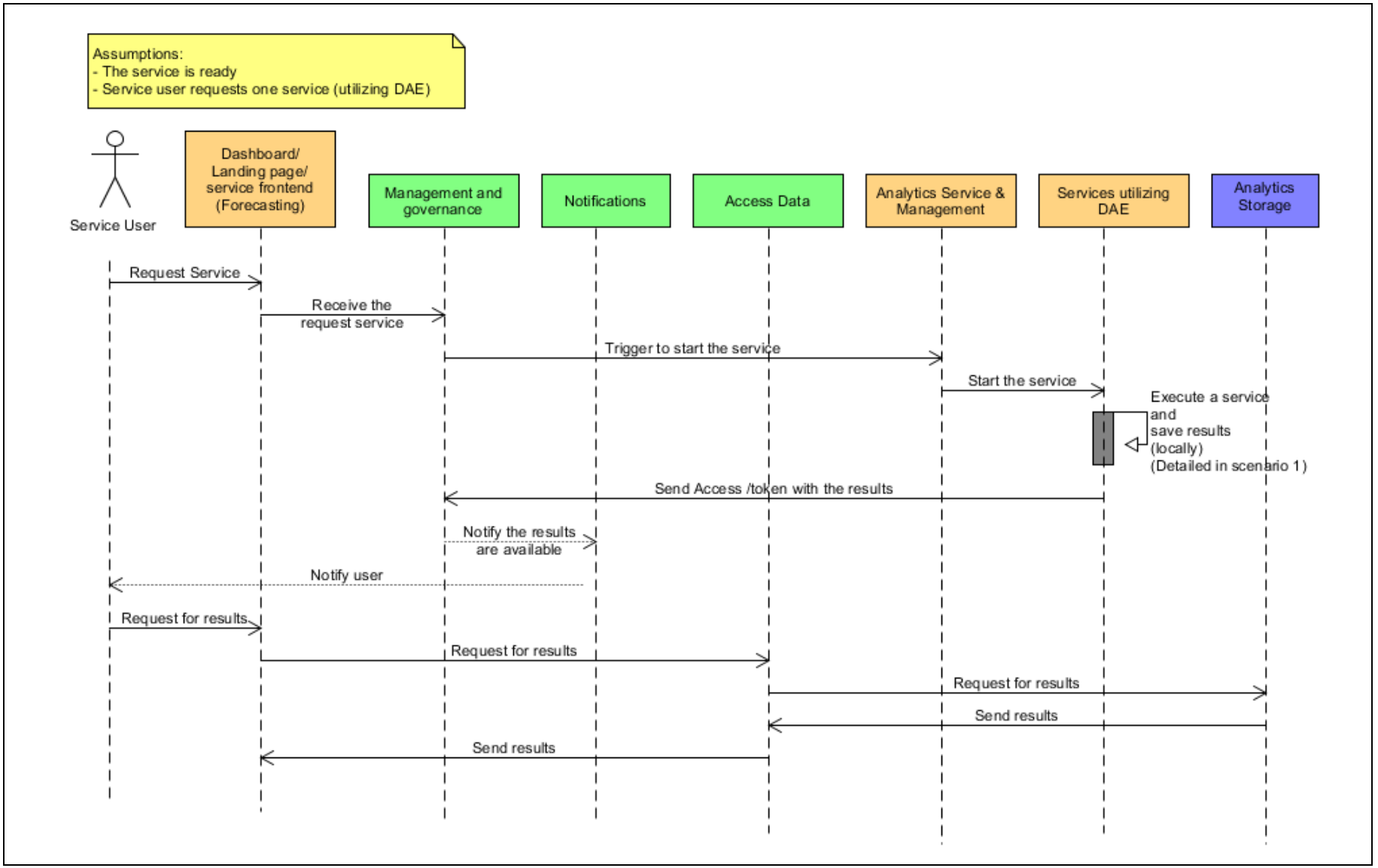
Table 14. Use cases of Analytic Layer

6.1.2.1 CASE 1: Service utilizing DAE

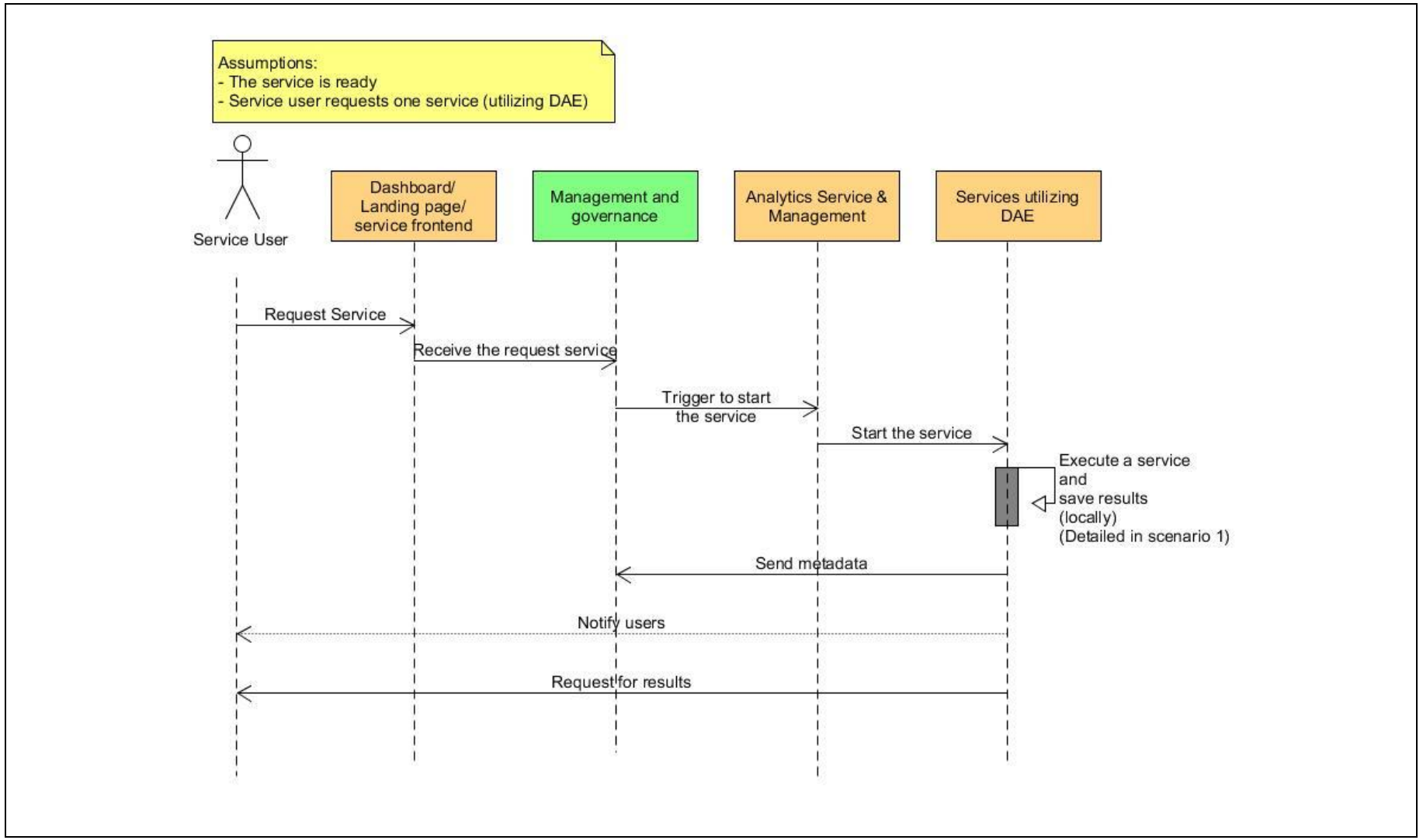
No	Name
BUC EX.X (Scenario 1)	Service used by a service user (Sharing results)
Short description	A Service User can utilize a service being offered in the Marketplace. Services that make use of the distributed Analytics Engine (DAE), like E4.2 or E5.1, first receive a request to execute the desired service. Then, the service is executed through the DAE and its results are saved. In this scenario, results can be finally shared with the Service User directly with the marketplace Dashboard or landing page, or with the help of an access token, as shown in the following sequence diagram.


Assumptions:
- The service is ready
- Service user requests one service (utilizing DAE)
Service User
Marketplace Dashboard/ Landing page/ service frontend
Management and governance
Notifications
Analytics Service & Management
Services utilizing DAE
Request service
Receive the request service
Trigger to start the service
Start the service
Execute a service and save results(*) (locally)
OPTION 1: Send access / token with the results
Notify the results are available
Notify user
Request for results
Request for results
Request for results
Send Results
OPTION 2: Report results
Execute a service and save results(*)
Service utilizing DAE
DAE
Unified API
Marketplace Storage
Analytics Storage
Internal processes: contract parameters
Contract parameters query
Send contract parameters
Internal processes: analytics preparation
Start training (optional)
Data query
Send data
Internal processes: train model / parametrize the service
Save new model trained
Return OK
Start servcie analytics
Data query
Send data
Model query
Send model
Internal processes: analytics execution
Save service results
Return OK
Process complete (data & metadata)
Process complete (data & metadata)

No	Name
BUC EX.X (Scenario 2)	Service used by a service user (Sharing results using Access data module)
Short description	A Service User can utilize a service being offered in the Marketplace. Services that make use of the distributed Analytics Engine (DAE), like E4.2 or E5.1, first receive a request to execute the desired service. Then, the service is executed through the DAE and its results are saved. In this scenario, results can be finally shared with the Service User making use of the Access data module to manage result requests and collect results from the Analytics Storage module.

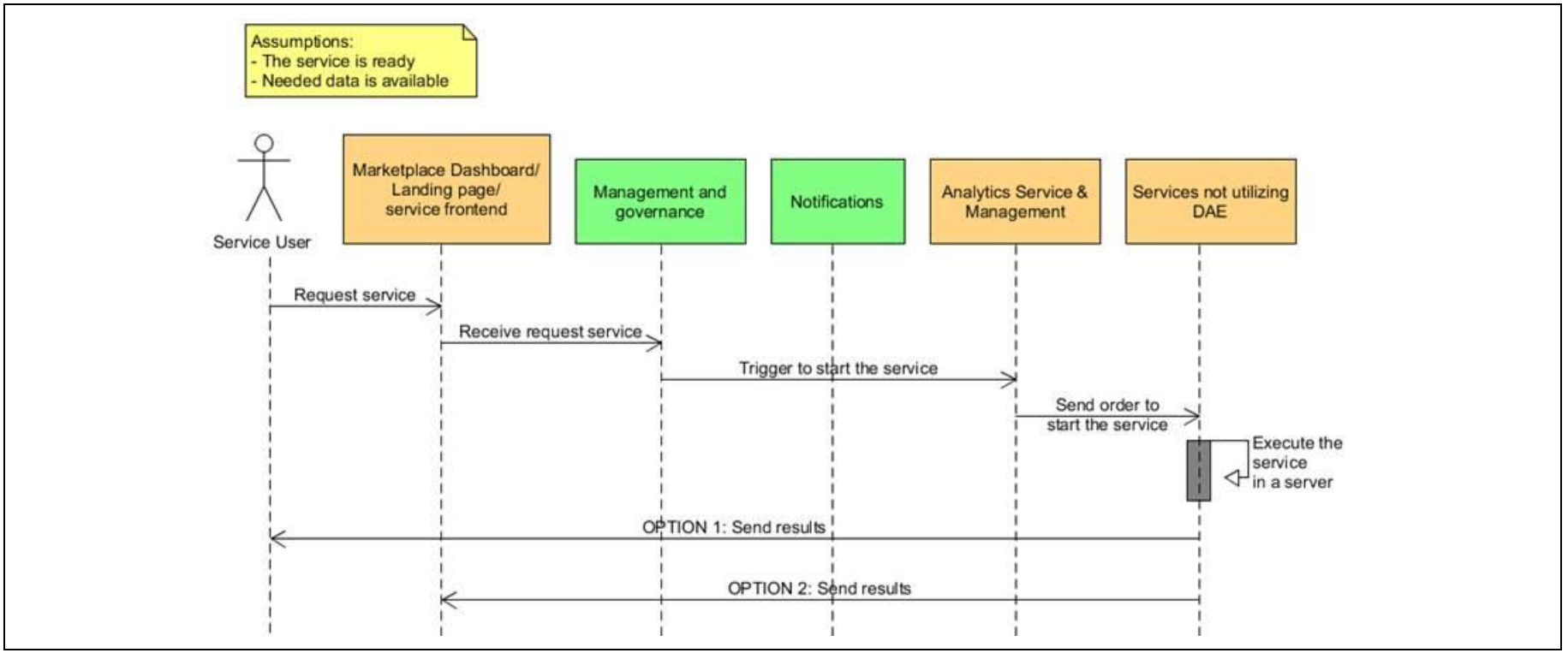

Assumptions:
- The service is ready
- Service user requests one service (utilizing DAE)
Service User
Dashboard/ Landing page/ service frontend (Forecasting)
Management and governance
Notifications
Access Data
Analytics Service & Management
Services utilizing DAE
Analytics Storage
Request Service
Receive the request service
Trigger to start the service
Start the service
Execute a service and save results (locally) (Detailed in scenario 1)
Send Access /token with the results
Notify the results are available
Notify user
Request for results
Request for results
Request for results
Send results
Send results

No	Name
BUC EX.X (Scenario 3)	Service used by a service user (Not sharing results)
Short description	A Service User can utilize a service being offered in the Marketplace. Services that make use of the distributed Analytics Engine (DAE), like E4.2 or E5.1, first receive a request to execute the desired service. Then, the service is executed through the DAE and its results are saved. In this scenario, results are not directly shared with the Service User and only a notification may be sent to inform that the service has been executed.


Assumptions:
- The service is ready
- Service user requests one service (utilizing DAE)
Service User
Dashboard/ Landing page/ service frontend
Management and governance
Analytics Service & Management
Services utilizing DAE
Request Service
Receive the request service
Trigger to start the service
Start the service
Execute a service and save results (locally) (Detailed in scenario 1)
Send metadata
Notify users
Request for results

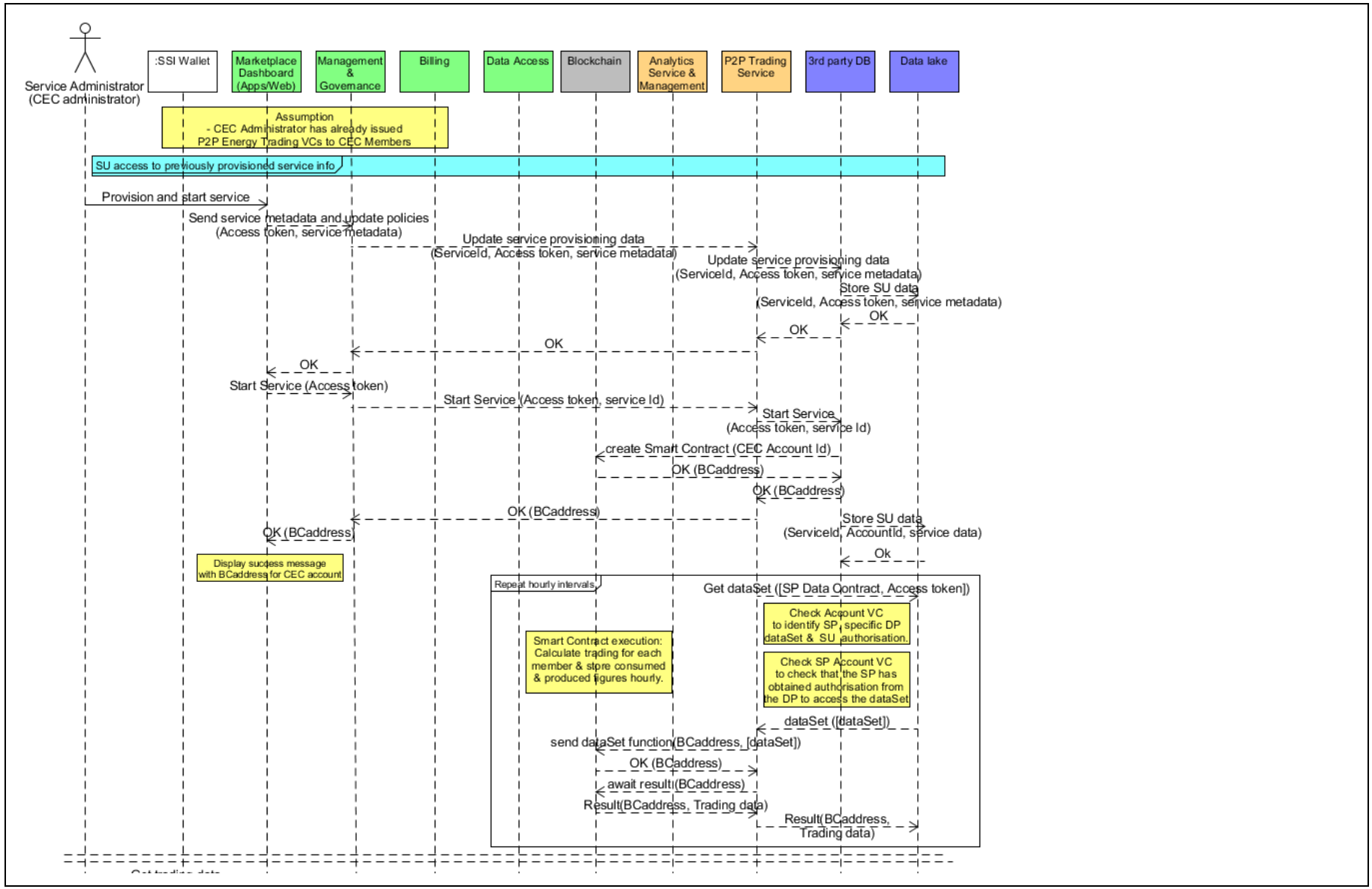
6.1.2.2 CASE 2: Service not utilizing DAE

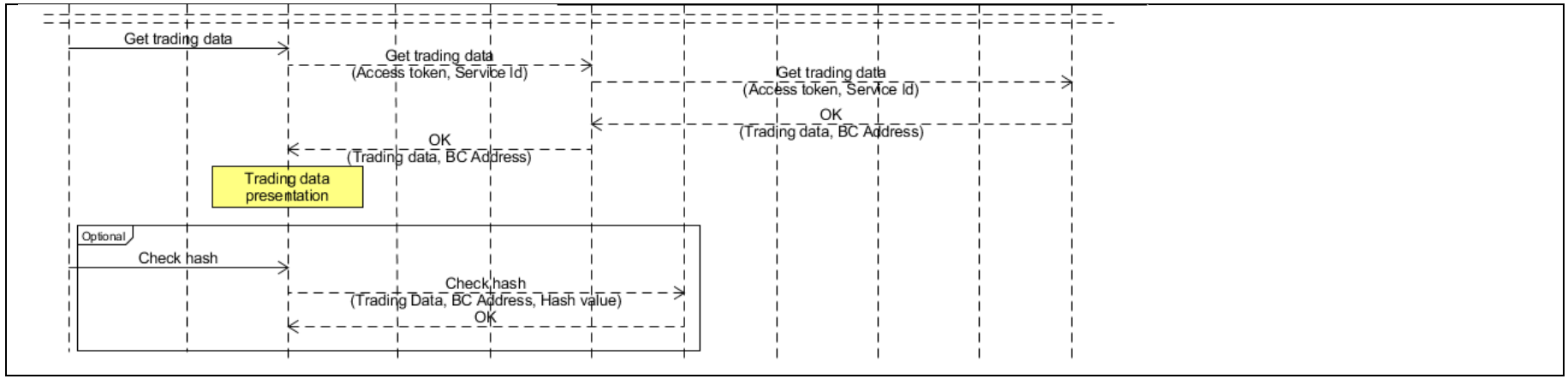
No	Name
BUC EX.X	Service used by a service user
Short description	The marketplace offers energy services to the different stakeholders. These services that not use the distributed analysis engine as SUC E1.1, SUC E1.2, E1.6, and BUC E5 present the option to share their results with the marketplace via connection through the unified API. As shown in the following sequence diagram. The service user first requests the desired service (eg,). Management and governance receive the request and send to start the service. Analytics Service & management Once the request is confirmed, the SU requests the results of that service, obtaining the data on the landing page / dashboard of the service directly. The connection is made using the API located in the data layer


Assumptions:
- The service is ready
- Needed data is available
Service User
Marketplace Dashboard/ Landing page/ service frontend
Management and governance
Notifications
Analytics Service & Management
Services not utilizing DAE
Request service
Receive request service
Trigger to start the service
Send order to start the service
Execute the service in a server
OPTION 1: Send results
OPTION 2: Send results

6.1.2.3 BUC E4

No	Name
BUC E4	P2P energy trading
Short description	Previous to the execution of the service, the CEC Administrator would set up the Service Account for the CEC and issue CEC Member Veriable Credentials to all member participants of the CEC organisation. The CEC Members would also be required to access the P2P Trading Service and give their authorisation to access their smart meter data by presenting their Data Source VC. In a first step to start the service, the CEC Administrator will click on start the service, and then the Smart Contract is initiated and the data related to the instance (CEC Account ID, the address of the SC in the blockchain) is saved in the 3rd party DB. Next, the Smart Contract is executed and obtains users' smart meter readings from the datalake as needed. The smart contract saves the whole CEC balance and the cryptographic hashes of the individual CEC balances in the blockchain. Also the SC saves all resulting trading data in the data lake. Finally, all trading data can be accessed from the data lake and individual balances can be corrobolated with the blockchain.


Service Administrator
(CEC administrator)
:SSI Wallet
Marketplace Dashboard (Apps/Web)
Management & Governance
Billing
Data Access
Blockchain
Analytics Service & Management
P2P Trading Service
3rd party DB
Data lake
Assumption
- CEC Administrator has already issued P2P Energy Trading VCs to CEC Members
SU access to previously provisioned service info
Provision and start service
Send service metadata and update policies (Access token, service metadata)
Update service provisioning data (ServiceId, Access token, service metadata)
Update service provisioning data (ServiceId, Access token, service metadata)
Store SU data (ServiceId, Access token, service metadata)
OK
OK
OK
OK
Start Service (Access token)
Start Service (Access token, service Id)
Start Service (Access token, service Id)
create Smart Contract (CEC Account Id)
OK (BCaddress)
OK (BCaddress)
OK (BCaddress)
OK (BCaddress)
Store SU data (ServiceId, AccountId, service data)
Ok
Display success message with BCaddress for CEC account
Repeat hourly intervals
Get dataSet ([SP Data Contract, Access token])
Check Account VC to identify SP, specific DP dataSet & SU authorisation.
Smart Contract execution: Calculate trading for each member & store consumed & produced figures hourly.
Check SP Account VC to check that the SP has obtained authorisation from the DP to access the dataSet
dataSet ([dataSet])
send dataSet function(BCaddress, [dataSet])
OK (BCaddress)
await result(BCaddress)
Result(BCaddress, Trading data)
Result(BCaddress, Trading data)


Get trading data
Get trading data
(Access token, Service Id)
Get trading data
(Access token, Service Id)
OK
(Trading data, BC Address)
OK
(Trading data, BC Address)
Trading data
presentation
Optional
Check hash
Check hash
(Trading Data, BC Address, Hash value)
OK

6.1.3 Marketplace Layer sequence diagrams

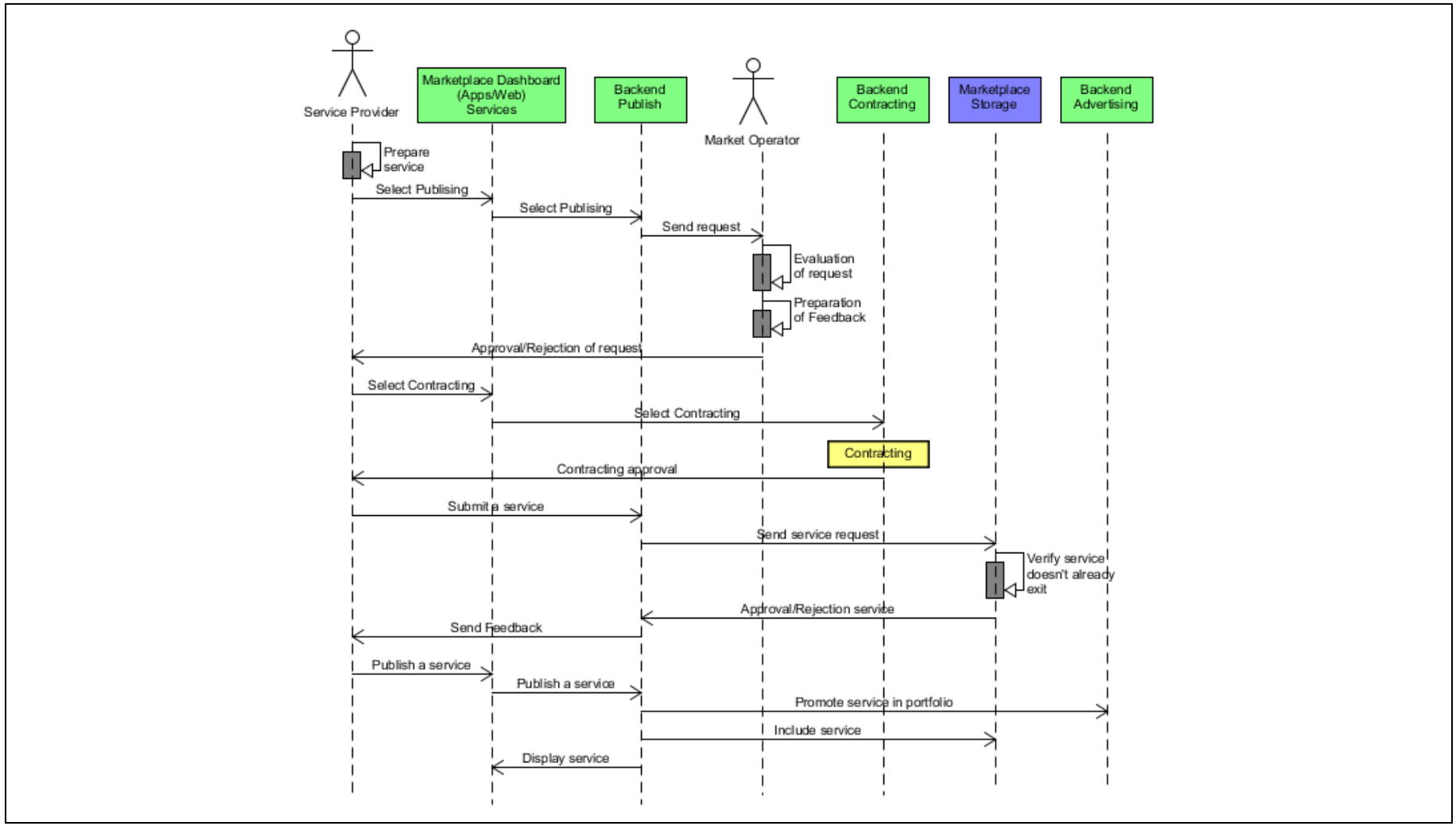
Sequence diagrams related with Marketplace Layer are presented afterwards.

Use Cases of Analytic Layer	
BUC M2	Publish service on the marketplace
BUC M3	Provision a service
BUC M4 / SUC M12 / SUC M13	Requesting and producing a service through Open innovation
BUC E3	Cost optimization for prosumers (TBD)
BUC E6	Optimized investment management (TBD)
SUC M2	Secure access
SUC M3	Contract service
SUC M4	Billing
SUC M5	Service composition
SUC M6	Service advertisement
SUC M7	Service management and governance
SUC M8	Search
SUC M9	Service provisioning (TBD)

Table 15. Use cases of Marketplace Layer

6.1.3.1 BUC M2

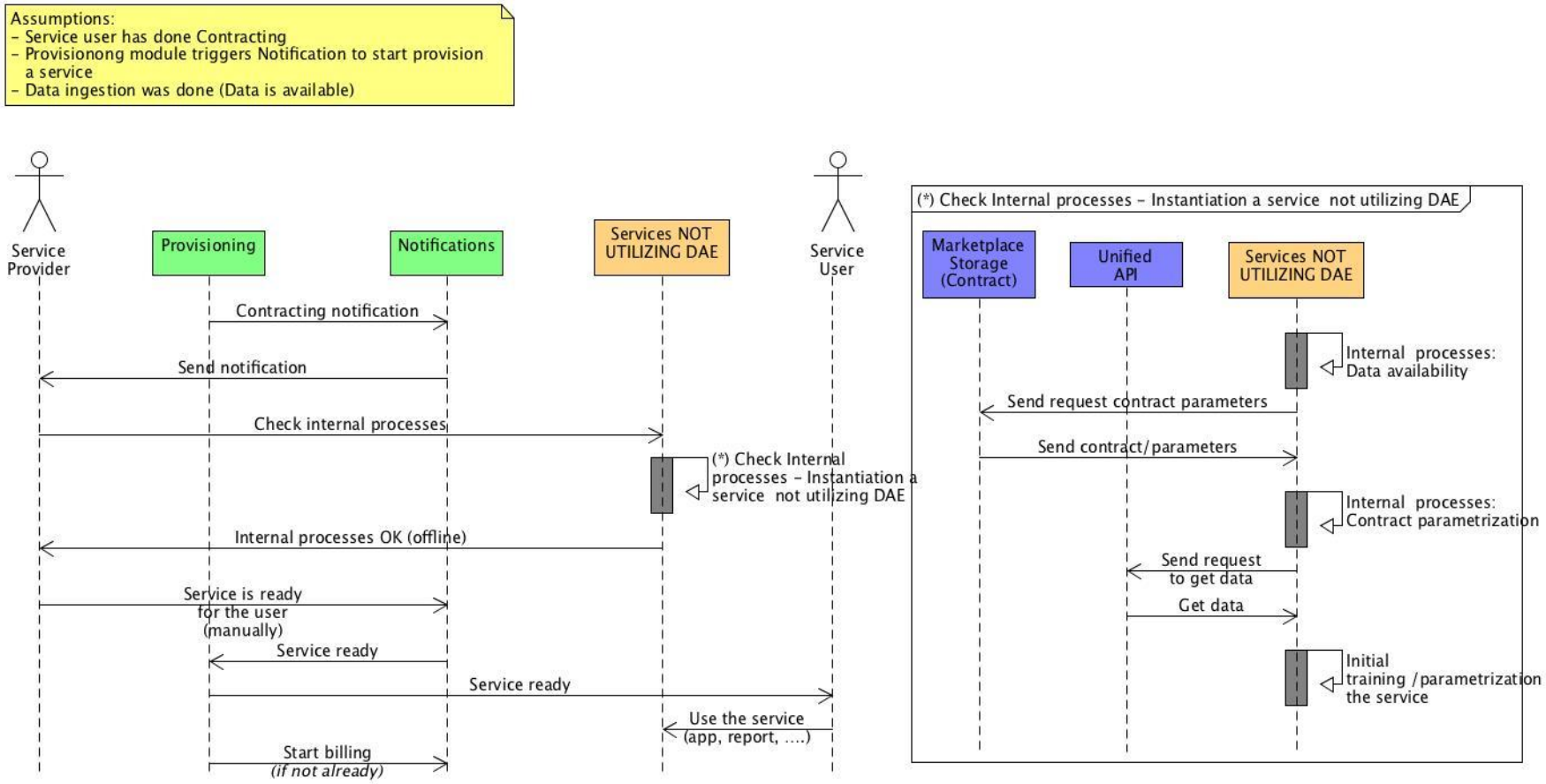
No	Name
BUC M2	Publish service on the marketplace
Short description	The Publish service on the marketplace business use case goal is to describe the way a service provider can publish a new service on the marketplace. After the service has been developed and tested internally on SP's premises, it is ready to be promoted in the market. Via a specific marketplace dashboard interface, the service provider creates its request to publish it, providing all the necessary information. This request is evaluated by a marketplace operator who can approve or reject it in case the information provided is not completed or clear or incorrect. In both cases the service provider is notified by the system about the operator decision. If the request is rejected, the service provider can decide to fix it and submit it again, or to restart the process from scratch. After operator's request approval, service provider must virtually sign the contract with the marketplace so that both parties are guaranteed about rights and duties. As soon as the contract has been signed, the service provider can deploy the service, add it to the marketplace services portfolio, providing all the information requested by the system (contracts for users, terms of use, how to use the service, urls…) and promote it so that marketplace users can be notified of this new service.


Service Provider
Marketplace Dashboard (Apps/Web) Services
Backend Publish
Market Operator
Backend Contracting
Marketplace Storage
Backend Advertising
Prepare service
Select Publising
Select Publising
Send request
Evaluation of request
Preparation of Feedback
Approval/Rejection of request
Select Contracting
Select Contracting
Contracting
Contracting approval
Submit a service
Send service request
Verify service doesn't already exit
Approval/Rejection service
Send Feedback
Publish a service
Publish a service
Promote service in portfolio
Include service
Display service

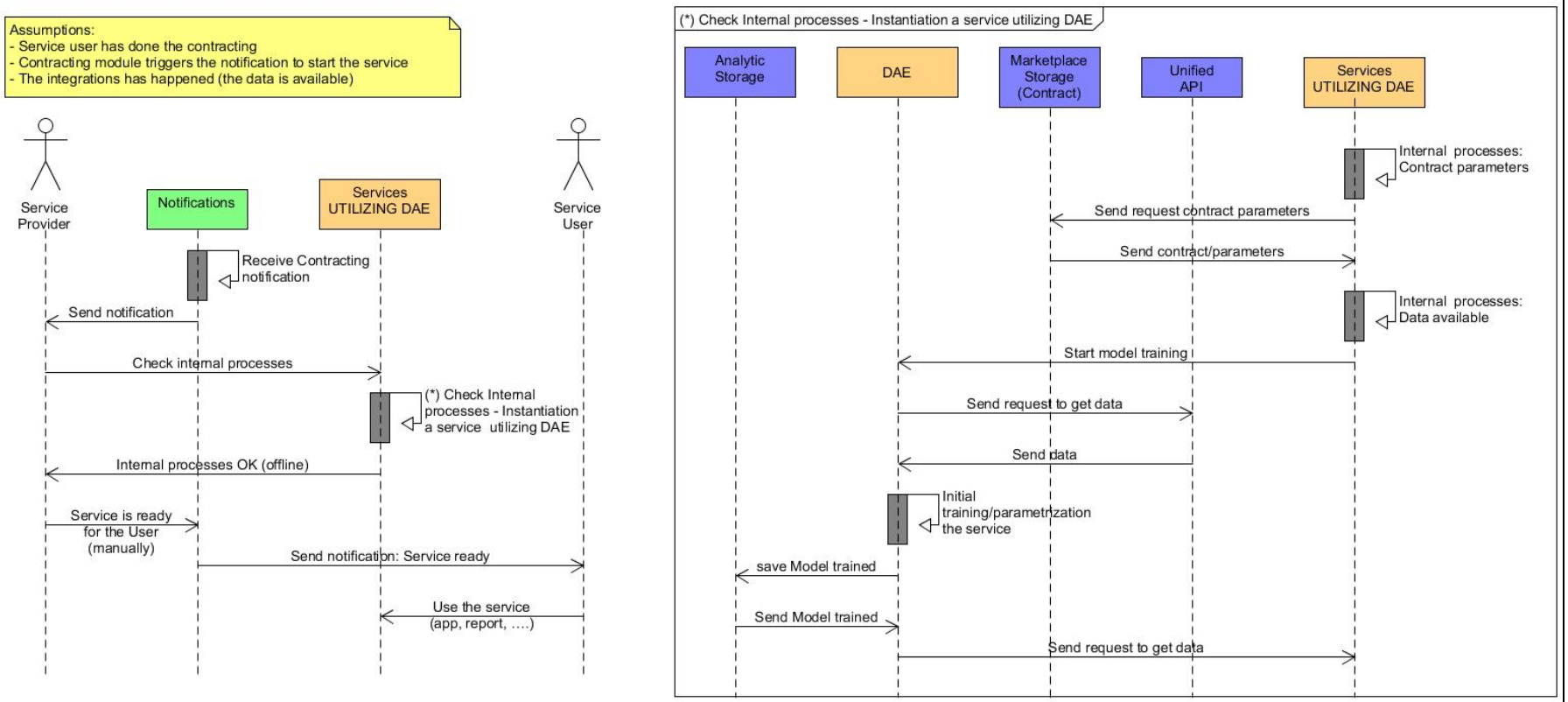
6.1.3.2 BUC M3

No	Name
BUC M3	Provision a service
Short description	A customer is interested in new services offered on the market. The customer enters the marketplaces and searches for new services. When suitable services are found, the customer analyses the offerings and decides to buy a service. He/She logs into the marketplace and orders the service. The marketplace facilitates contracting the service with the service provider and supports the service provisioning. During the contracting process the parameters of the service are selected by the user and embedded in the contract. When the service has been contracted, the service provisioning process can start. A notification is send to the service provider who starts the process of service provisioning. During the provisioning process the service provider checks for the data needed to provide the service, and then adapts the service according to the service user needs specified through the contracting parameters. When the service provider provisioning process is finished the user gets notified and can start using the service.

No	Name
BUC M3 (Scenario 1)	Provision of a service not utilizing DAE

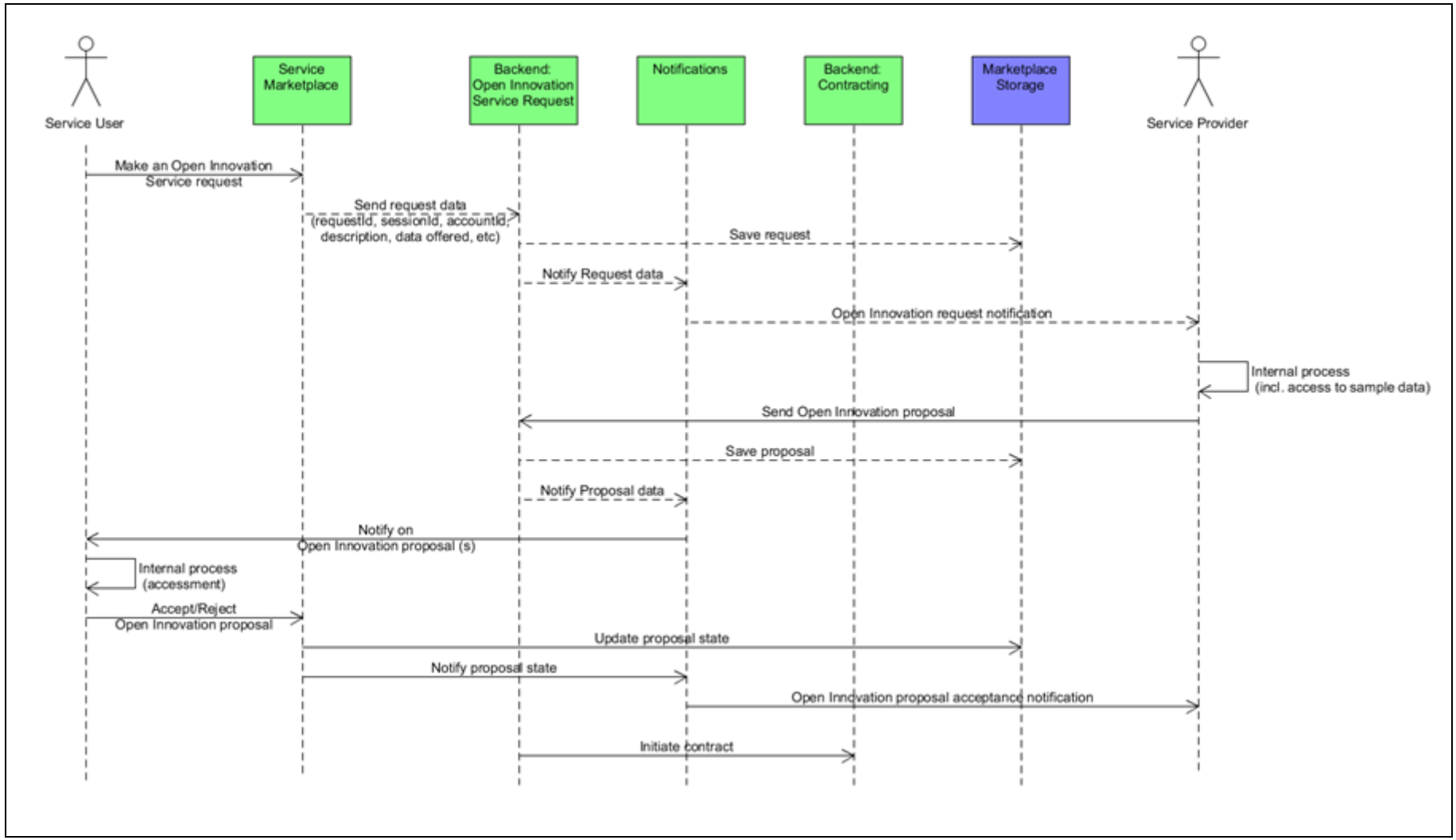


No	Name
BUC M3 (Scenario 2)	Provision of a service utilizing DAE



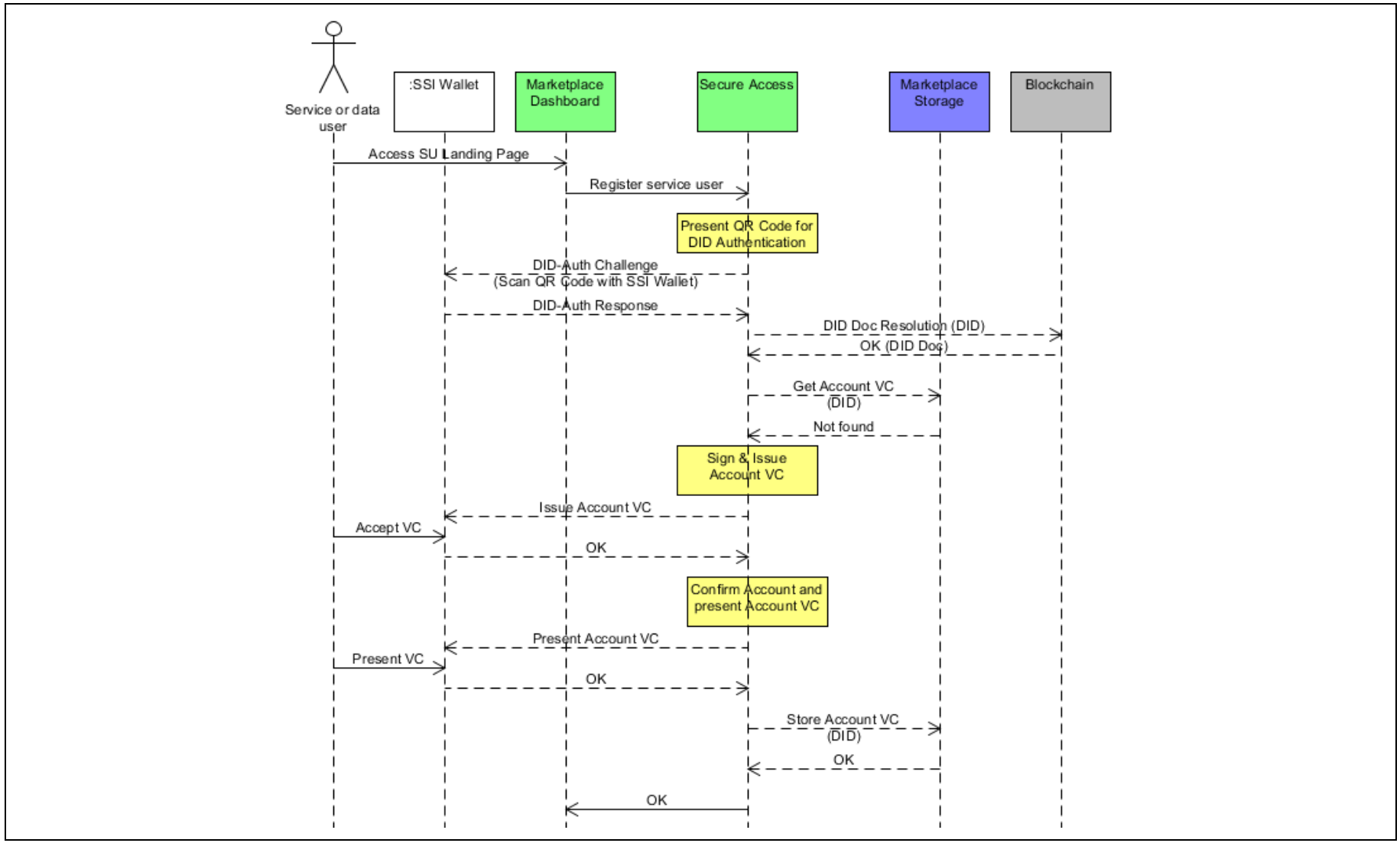
6.1.3.3 BUC M4 / SUC M12 / SUC M13

No	Name
BUC M4 / SUC M12 / SUC M13	Requesting and producing a service through Open innovation
Short description	Open innovation exposes an interface for requesting a new service from the Marketplace. The potential Service User can make an Open innovation Service Request by describing the service needed. Service Providers that are registered in the Marketplace will be informed about the new Open innovation Service Request, and they can make their proposal (Open innovation Service Procurement). The potential Service User can review the proposals and make a choice. Service User and the chosen Service Provider can then sign a contract and proceed to sharing data and implementing the service.

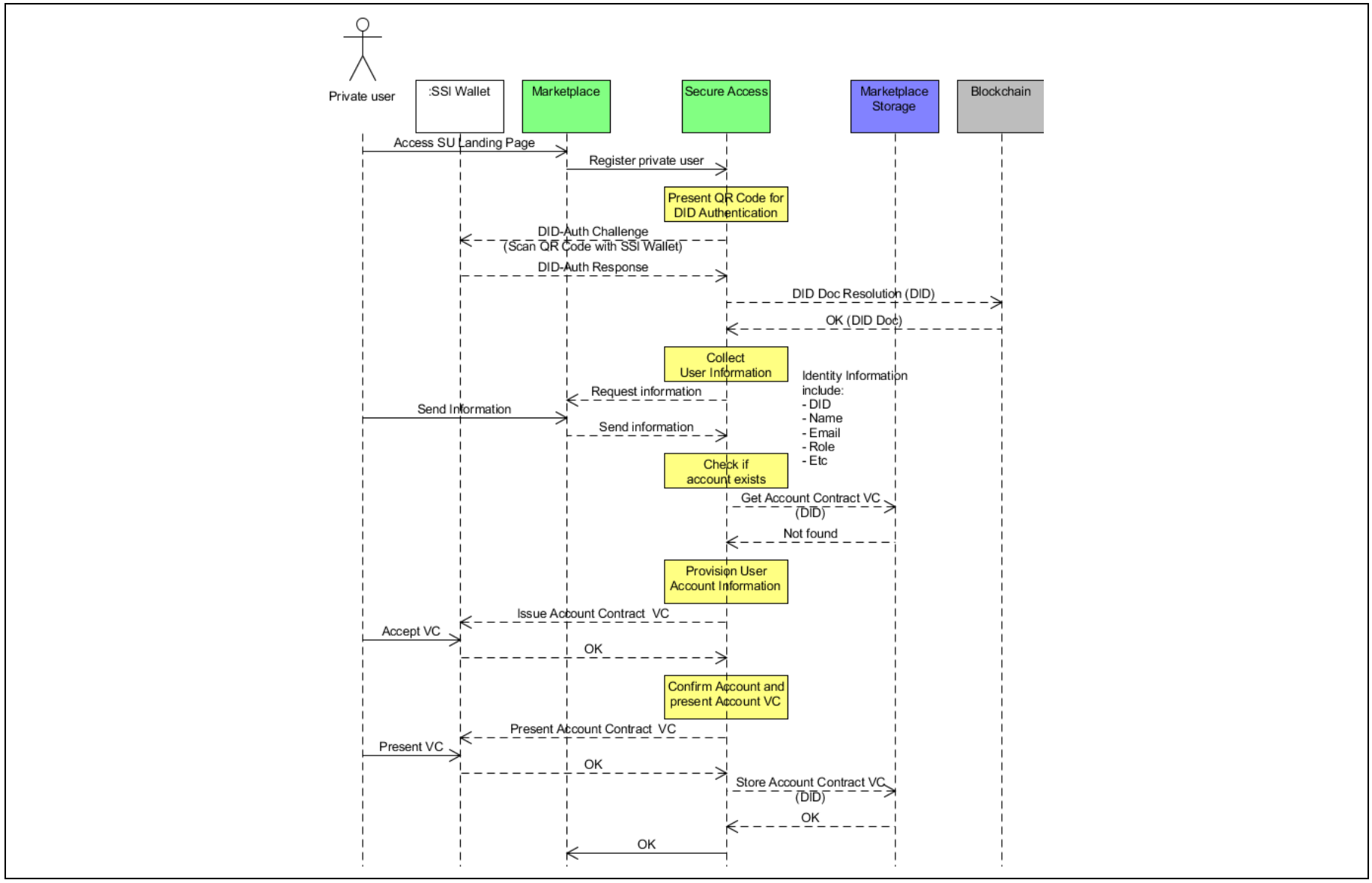

Service User
Service Marketplace
Backend: Open Innovation Service Request
Notifications
Backend: Contracting
Marketplace Storage
Service Provider
Make an Open Innovation Service request
Send request data (requestId, sessionId, accountId, description, data offered, etc)
Save request
Notify Request data
Open Innovation request notification
Internal process (incl. access to sample data)
Send Open Innovation proposal
Save proposal
Notify Proposal data
Notify on Open Innovation proposal (s)
Internal process (accessment)
Accept/Reject Open Innovation proposal
Update proposal state
Notify proposal state
Open Innovation proposal acceptance notification
Initiate contract

6.1.3.4 SUC M2

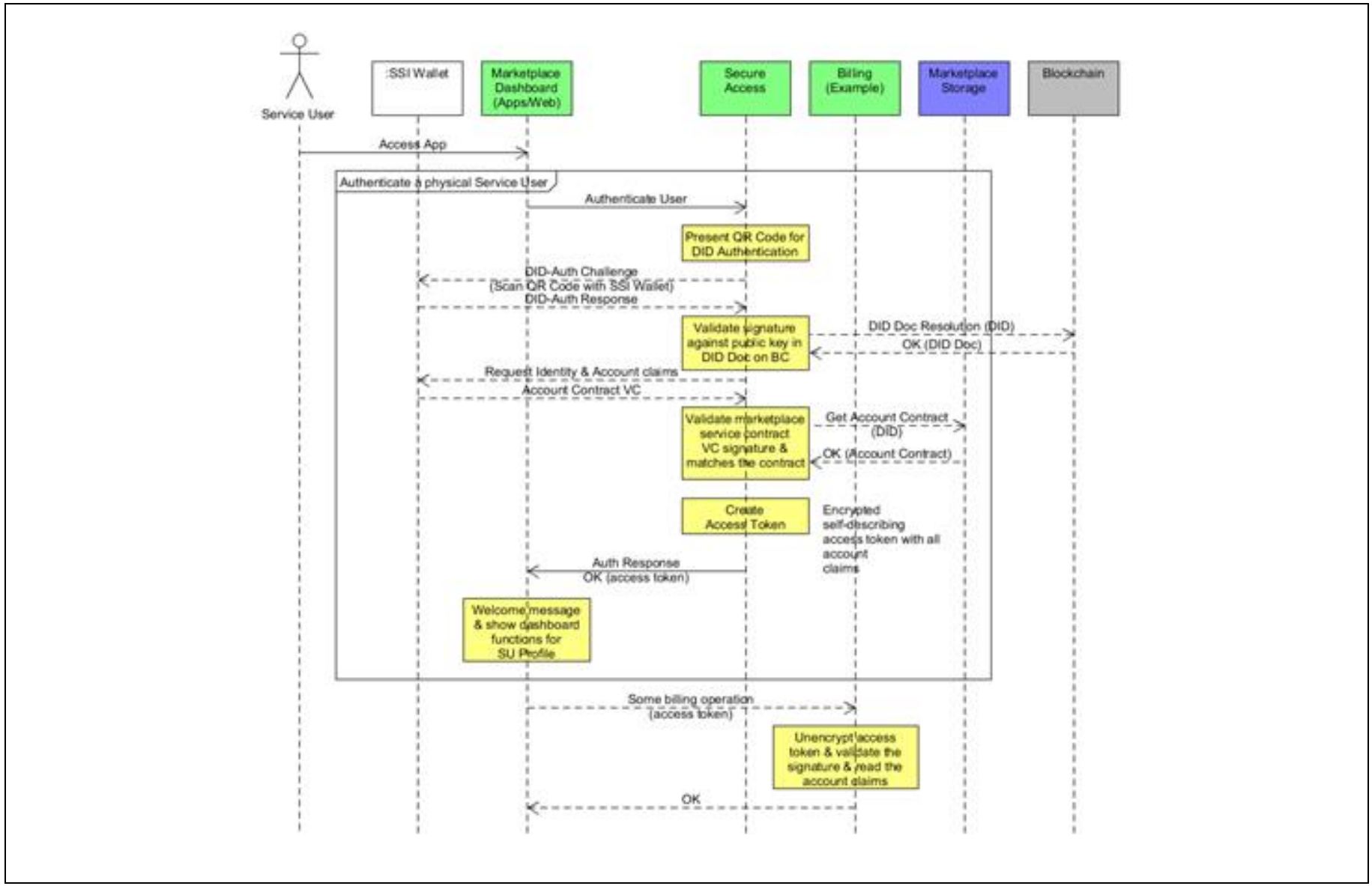
No	Name
SUC M2 (Scenario 1)	Secure access – Register a company
Short description	Previosly a company has already issued a company representative VC with role as a administrator so he or she can register the company in the marketplace. The company administrator accesses the Marketplace with their Company Rep VC after performing DID Authentication with the SSI Agent of the Secure Access module. Next, it is checked that the company has not been previously issued with a Company Account VC by checking the marketplace storage with the Company DID. Finally, the Company administrator fill the required information to set up account and the Company Account VC is issued to the company's online wallet and signed and ultimately, the Company Account VC is confirmed by presenting it again and saved in the Marketplace.


Service or data user
:SSI Wallet
Marketplace Dashboard
Secure Access
Marketplace Storage
Blockchain
Access SU Landing Page
Register service user
Present QR Code for DID Authentication
DID-Auth Challenge
(Scan QR Code with SSI Wallet)
DID-Auth Response
DID Doc Resolution (DID)
OK (DID Doc)
Get Account VC
(DID)
Not found
Sign & Issue Account VC
Issue Account VC
Accept VC
OK
Confirm Account and present Account VC
Present Account VC
Present VC
OK
Store Account VC
(DID)
OK
OK

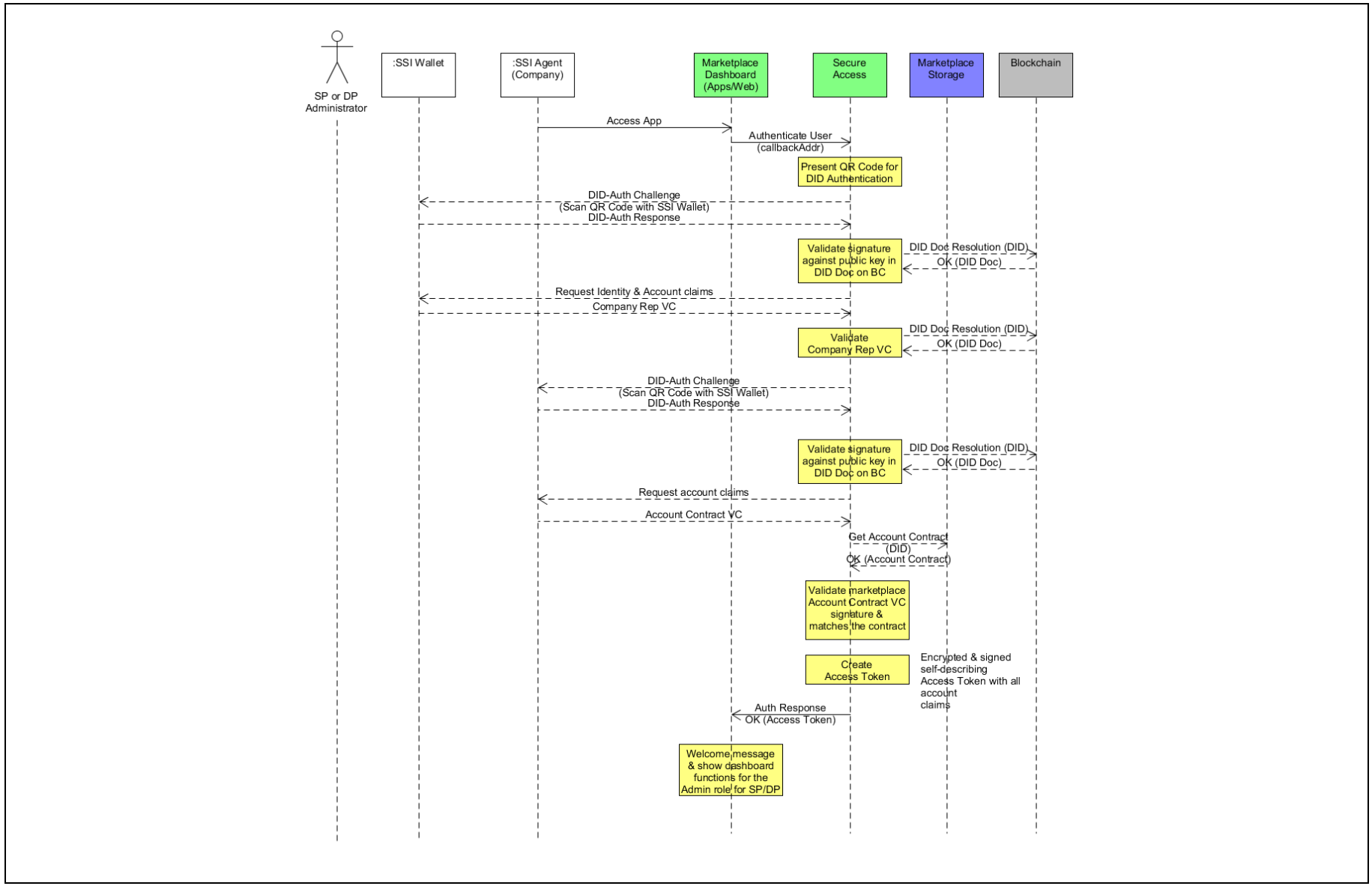
No	Name
SUC M2 (Scenario 2)	Secure access – Register a private user
Short description	A private user accesses the Marketplace and establishes a DID Authentication with the SSI Agent included in the Secure Access module. Next, the user is requested for identity and account claims and it is checked that the user has not been previously issued with a VC Account Contract. Finally, the Account VC is issued to the user's wallet and signed and ultimately, the Account VC is confirmed by presenting it again and saved in the Marketplace.


Private user
:SSI Wallet
Marketplace
Secure Access
Marketplace Storage
Blockchain
Access SU Landing Page
Register private user
Present QR Code for DID Authentication
DID-Auth Challenge (Scan QR Code with SSI Wallet)
DID-Auth Response
DID Doc Resolution (DID)
OK (DID Doc)
Collect User Information
Identity Information include:
- DID
- Name
- Email
- Role
- Etc
Request information
Send Information
Send information
Check if account exists
Get Account Contract VC (DID)
Not found
Provision User Account Information
Issue Account Contract VC
Accept VC
OK
Confirm Account and present Account VC
Present Account Contract VC
Present VC
OK
Store Account Contract VC (DID)
OK
OK

No	Name
SUC M2 (Scenario 3)	Secure access – Authenticate a service user (private citizen)
Short description	SUC M2 (Scenario 3), an individual (private citizen) who previously registered with the BD4OPEM as a Service User is accessing the Marketplace and is asked to perform DID Authentication and present identity and account claims from his/her SSI Wallet. Upon successfully performing DID Authentication and validating the received Account Contract VC for an individual user, the Account Contract VC is checked against the Marketplace records to make sure its status is active, and not another status such as revoked. Finally, the user's Dashboard / App User Agent (UA) is returned with a successful authentication including a self-describing Access Token that is encrypted and signed with all the Account Contract claims. Each time, the user accesses the Marketplace from the Dashboard, the UA must send the Access Token and the receiving business logic module will decipher it using the Marketplace Secure Access private key, validate its signature and check that it is not expired. The token will have an expiry time (1h hour , 1 day etc.) that will need the user to re-authenticate when it is expired. The business logic module in the above example is billing and will examine the deciphered claims to obtain the role of the user and any other account claims it needs to decide how to treat the user. Note a Service User may also be a company or organisation in which case the following scenario 4 flow would apply.

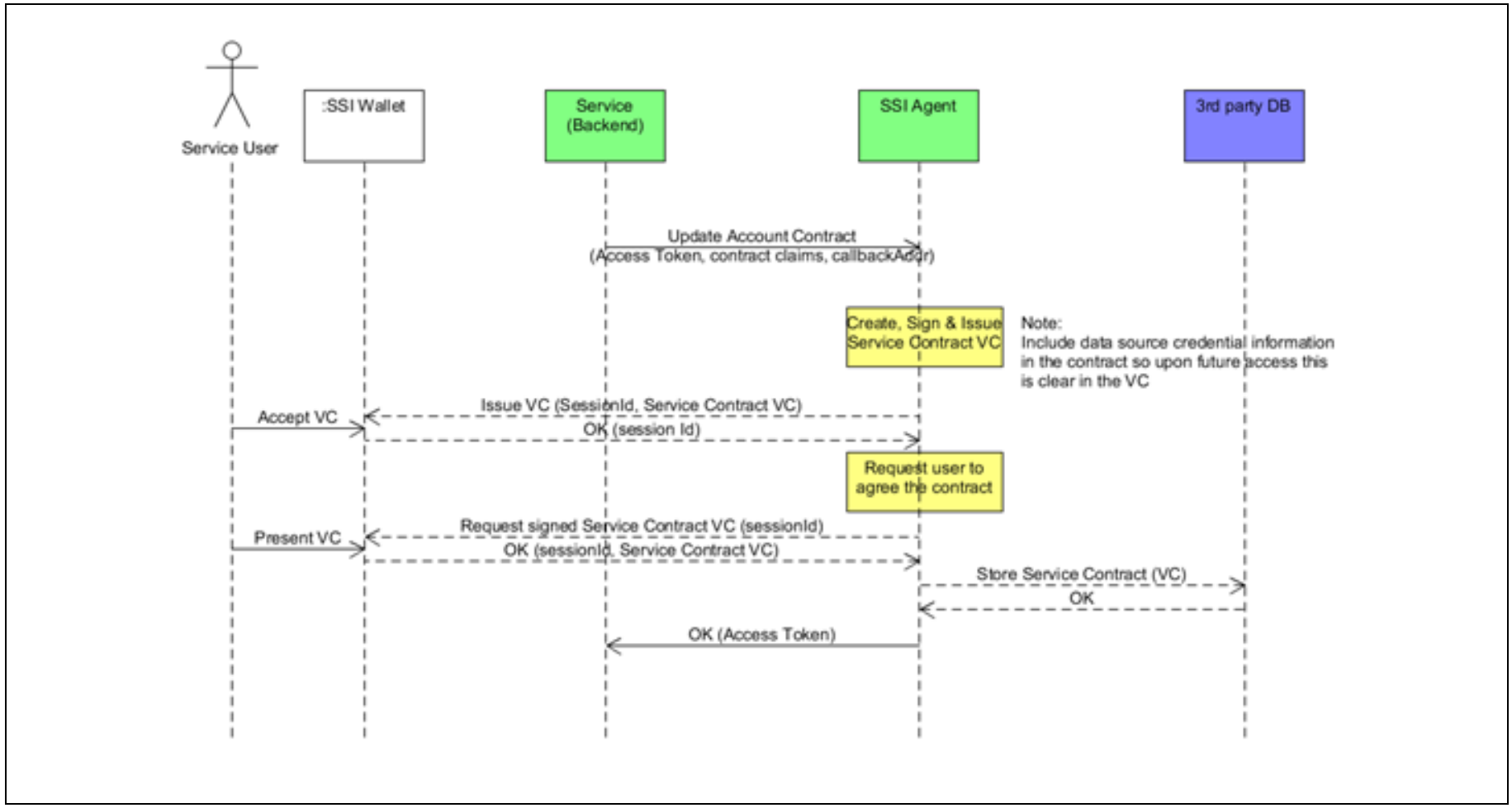

Service User
SSI Wallet
Marketplace Dashboard (Apps/Web)
Secure Access
Billing (Example)
Marketplace Storage
Blockchain
Access App
Authenticate a physical Service User
Authenticate User
Present QR Code for DID Authentication
DID-Auth Challenge (Scan QR Code with SSI Wallet)
DID-Auth Response
Validate signature against public key in DID Doc on BC
DID Doc Resolution (DID)
OK (DID Doc)
Request Identity & Account claims
Account Contract VC
Validate marketplace service contract VC signature & matches the contract
Get Account Contract (DID)
OK (Account Contract)
Create Access Token
Encrypted self-describing access token with all account claims
Auth Response OK (access token)
Welcome message & show dashboard functions for SU Profile
Some billing operation (access token)
Unencrypt access token & validate the signature & read the account claims
OK

No	Name
SUC M2 (Scenario 4)	Secure access – Authenticate a service provider (SP) or data provider (DP) administrator
Short description	SUC M2 (Scenario 4), a Company Reprepesentative is accessing the BD4OPEM platform on behalf of their company. In this example they are accessing as an Administrator for an SP or DP. Upon accessing the Marketplace, the user is redirected to Secure Access and is asked to perform DID Authentication and present identity and Marketplace account claims from his/her SSI Wallet. As the user is a company representative, he/she will present the requested claims from his/her Company Rep VC which includes the DID of their company. The Secure Access module will then proceed in this case to request the Company's SSI Agent to perform DID Auth and request the company Account Contract VC. Upon successfully perfroming DID Authentication and validating the received Account Contract VC, the Account Contract VC is checked against the Marketplace records to make sure its status is active and not revoked. Finally, the user's Dashboard / App User Agent (UA) is returned with a successful authentication including a self-describing Access Token that is encrypted and signed with all the Account Contract claims. Each time, the user accesses the Marketplace from the Dashboard, the UA must send the Access Token and the receiving business logic module will decipher it using the Marketplace Secure Access private key, validate its signature and check that it is not expired. The token will have an expiry time (1h hour , 1 day etc.) that will need the user to re-authenticate when it is expired. Any business logic module that is subsequently accessed by the Company Rep will examine the deciphered claims to obtain the role of the user and any other account claims it needs to decide how to treat the user. In this example the Company Rep is an Adminsitrator and is able to therefore access all services contracted by the company.


SP or DP Administrator
:SSI Wallet
:SSI Agent (Company)
Marketplace Dashboard (Apps/Web)
Secure Access
Marketplace Storage
Blockchain
Access App
Authenticate User (callbackAddr)
Present QR Code for DID Authentication
DID-Auth Challenge (Scan QR Code with SSI Wallet)
DID-Auth Response
Validate signature against public key in DID Doc on BC
DID Doc Resolution (DID)
OK (DID Doc)
Request Identity & Account claims
Company Rep VC
Validate Company Rep VC
DID Doc Resolution (DID)
OK (DID Doc)
DID-Auth Challenge (Scan QR Code with SSI Wallet)
DID-Auth Response
Validate signature against public key in DID Doc on BC
DID Doc Resolution (DID)
OK (DID Doc)
Request account claims
Account Contract VC
Get Account Contract (DID)
OK (Account Contract)
Validate marketplace Account Contract VC signature & matches the contract
Create Access Token
Encrypted & signed self-describing Access Token with all account claims
Auth Response
OK (Access Token)
Welcome message & show dashboard functions for the Admin role for SP/DP

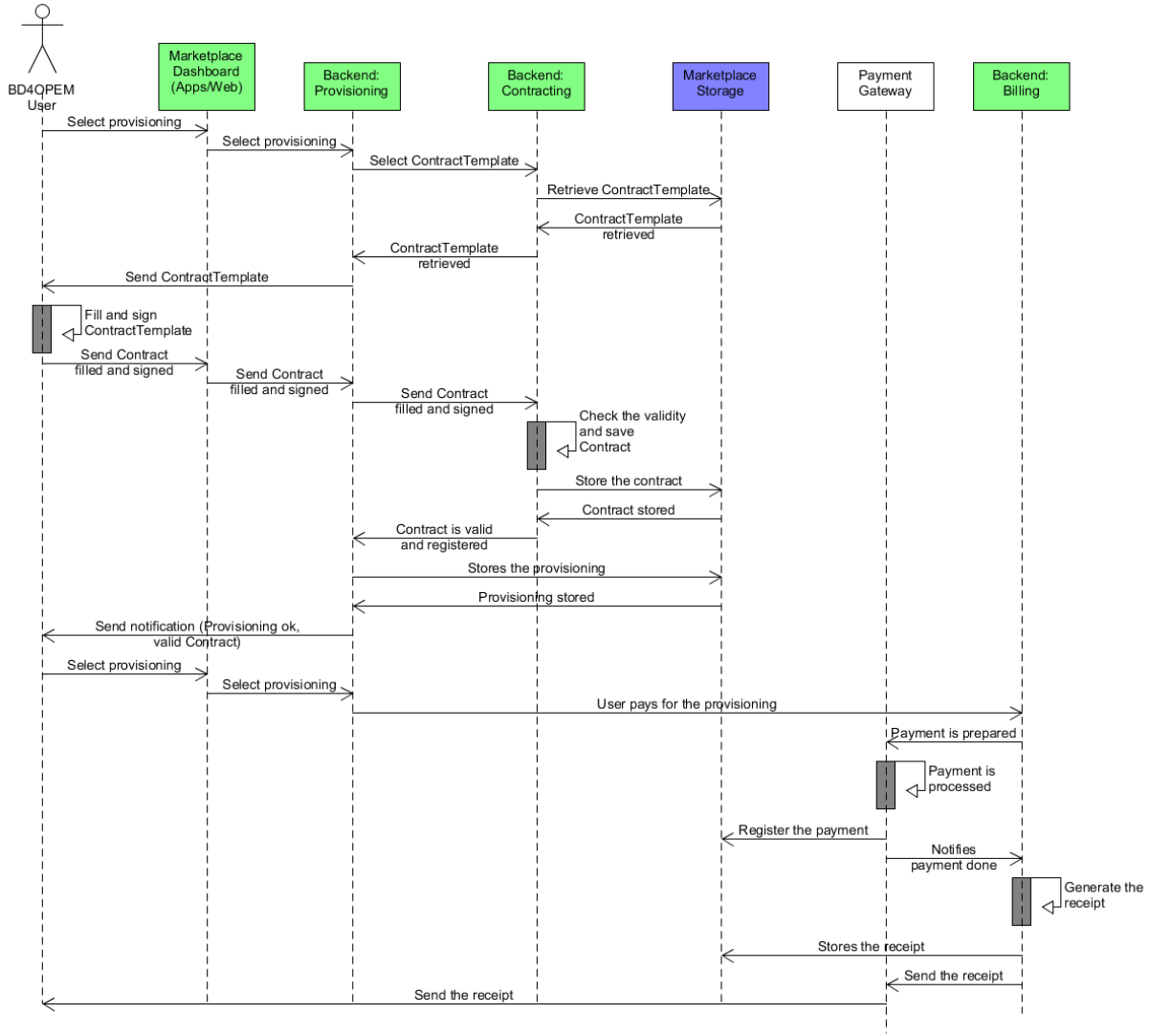
6.1.3.5 SUC M3

No	Name
SUC M3	Contract service
Short description	The contracting module will issue updated Account Contract VCs on behalf of business logic modules (Share data, Provisioning Publish & Open innovation) to capture the new contract details. The figure above show's an example of a Service User, agreeing to contract for a new service. In this example, the Provisioning module business logic will specify all the details of the service with the Service User and will call the contracting module to issue a contract to the user for the service the user is signing up to. For example, a user signing up to a marketplace service could specify the following set of Service Attributes in the contract: • Service: e.g. CECp2pEnergyTrading • credentialType: MarketplaceService • ServiceTermsConds: url to services Termas and Conditions • National identity: V7877898782H • Role: e.g. Service User • Data Provider: DSO_123.com • Data Source Type: Smart Meter • Data Source Id: 123e4567-e89b-12d3-a456-426614174000 • Data Source Role: Data Subject, Data Provider, Data Consumer etc. • DSO local Distribution ID: 843894389 • Service Profile: CEC Member • Service Profile ID: CEC Id • Billing Type: Bank Account • Once completed, the Contract Service will issue service attribute claims as a W3C Verifiable Credential to the Service User. This is a non-nomative example and the exact list of attribute claims needed will be identified per business logic calling the Contracting module.


Service User
:SSI Wallet
Service (Backend)
SSI Agent
3rd party DB
Update Account Contract
(Access Token, contract claims, callbackAddr)
Create, Sign & Issue Service Contract VC
Note:
Include data source credential information in the contract so upon future access this is clear in the VC
Issue VC (SessionId, Service Contract VC)
Accept VC
OK (session Id)
Request user to agree the contract
Request signed Service Contract VC (sessionId)
Present VC
OK (sessionId, Service Contract VC)
Store Service Contract (VC)
OK
OK (Access Token)

6.1.3.6 SUC M4

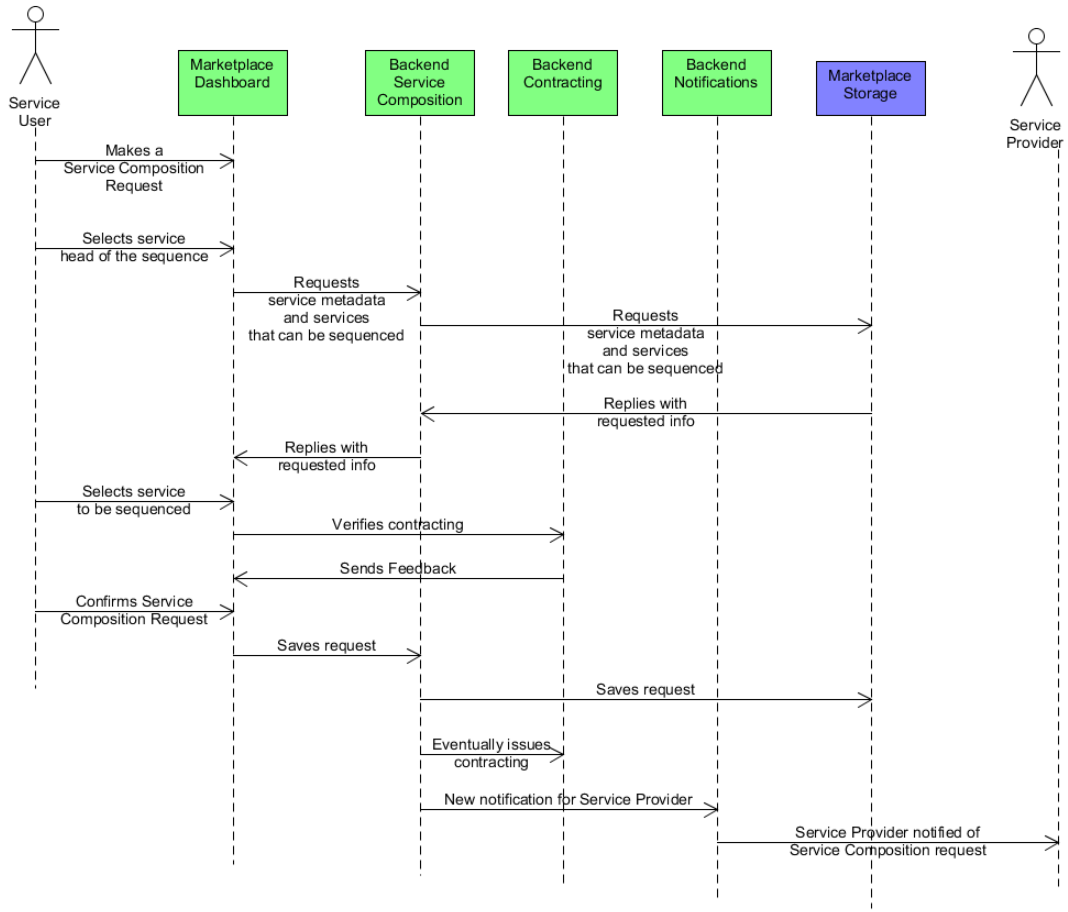
No	Name
SUC M4	Billing
Short description	When a service user or a data user wants to access a service or some data, a contract among the provider and the user must be in place in order to establish the provisioning. The user must fill in and sign the contract template corresponding to the desired provisioning, then the system checks the contract validity and stores it in the database. The provisioning contract, among other things, defines payment features and timings, for example if payment is one-time or recurring. Via a specific function of the marketplace, the user can proceed with the payment. When the payment succeeds, the system gives the corresponding receipt to the user. After that, the customer (service user or data user) can finally access service or data according to the technicalities requested.


BD4OPEM User
Marketplace Dashboard (Apps/Web)
Backend: Provisioning
Backend: Contracting
Marketplace Storage
Payment Gateway
Backend: Billing
Select provisioning
Select provisioning
Select ContractTemplate
Retrieve ContractTemplate
ContractTemplate retrieved
ContractTemplate retrieved
Send ContractTemplate
Fill and sign ContractTemplate
Send Contract filled and signed
Send Contract filled and signed
Send Contract filled and signed
Check the validity and save Contract
Store the contract
Contract stored
Contract is valid and registered
Stores the provisioning
Provisioning stored
Send notification (Provisioning ok, valid Contract)
Select provisioning
Select provisioning
User pays for the provisioning
Payment is prepared
Payment is processed
Register the payment
Notifies payment done
Generate the receipt
Stores the receipt
Send the receipt
Send the receipt

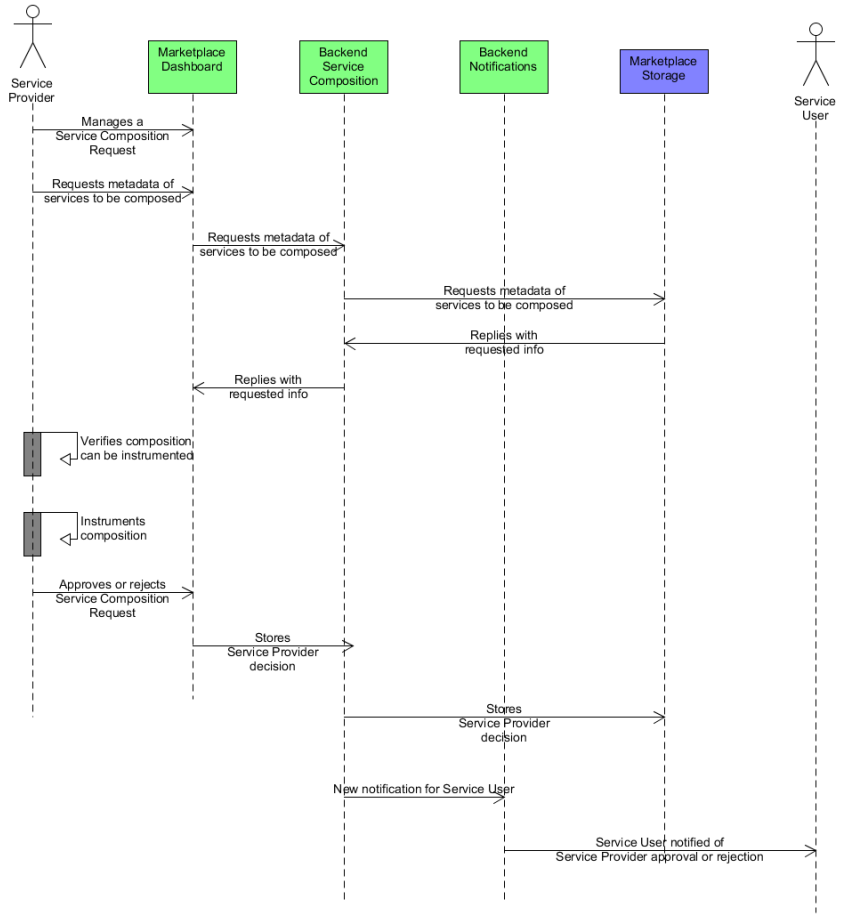
6.1.3.7 SUC M5

No	Name
SUC M5	Service composition
Short description	This use case's objective is to allow a service user to benefit from the chaining of different existing services. The service user who thinks to have an advantage from this chaining must have contracted and be allowed to use each single service. Not all the services in the marketplace can be chained at user's will, but each service "knows" which other services can come after or before it in the chain. These information are predefined for the user because are defined by the marketplace operators with support of service providers, i.e. service creators. First scenario regards the service composition request. Process starts with a service user that creates a service composition request on the marketplace. After choosing the service first in chain, the marketplace suggests which services are suitable for being the next in chain, according to the info mentioned above: the service user can then select from this list the next item in the chain. After this selection, the user can decide to terminate the requested chain or to continue it, querying again the marketplace about which services are suitable as next item in chain. Once the user is done with the service composition request, the system notifies the service provider(s) about this request. Second scenario is about service composition request management. Service provider(s) is(are) notified about the user request and manage(s) the request checking all the logical and technical constraints of all the services involved in the proposal in order to verify the request's feasibility. In case the service provider(s) decide(s) not to approve the user's request, it(they) reject(s) the request and the marketplace provides a notification to the service user who created the request. In case of approval, the service user gets notified and a new process starts in order to put in place the new composed service.

No	Name
SUC M5 (Scenario 1)	Service composition request

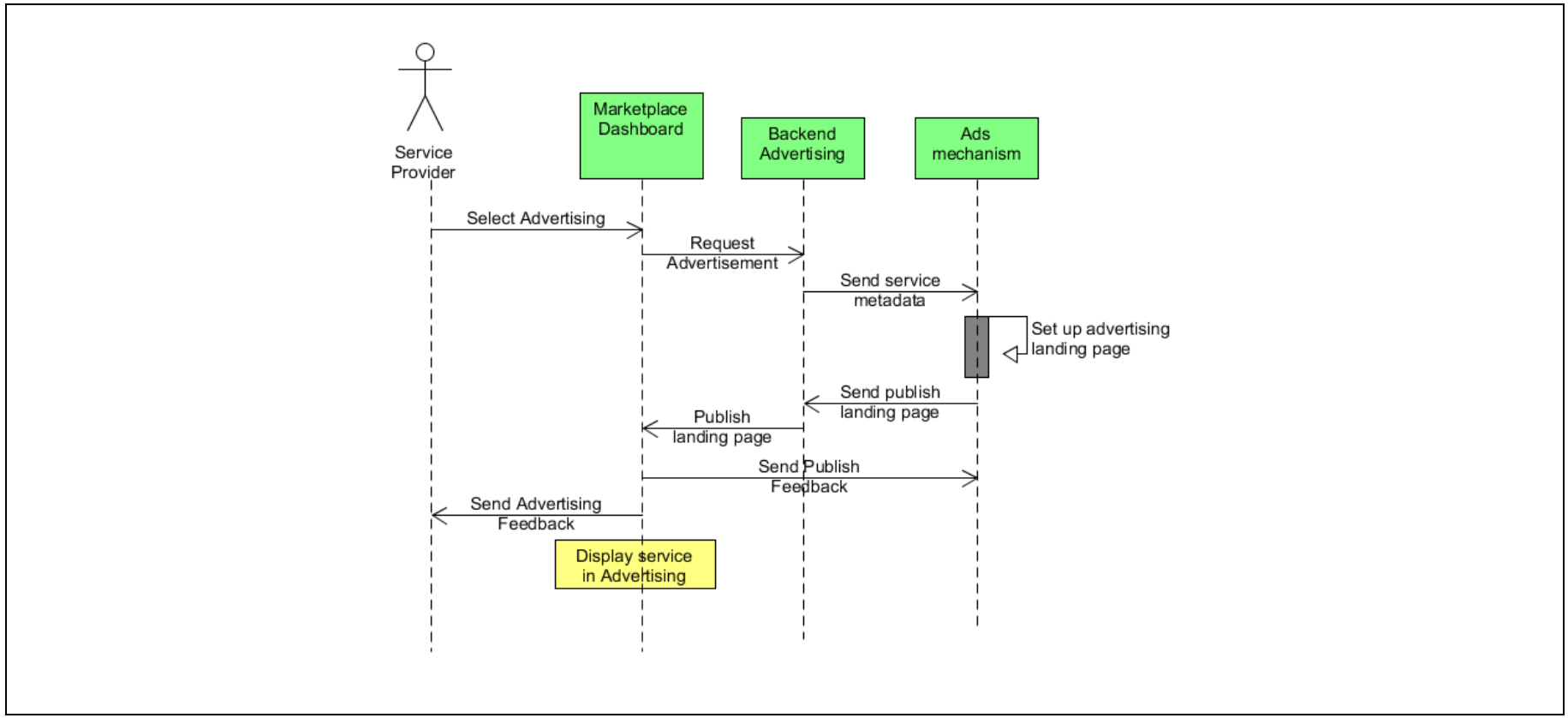


No	Name
SUC M5 (Scenario 2)	Service composition procurement



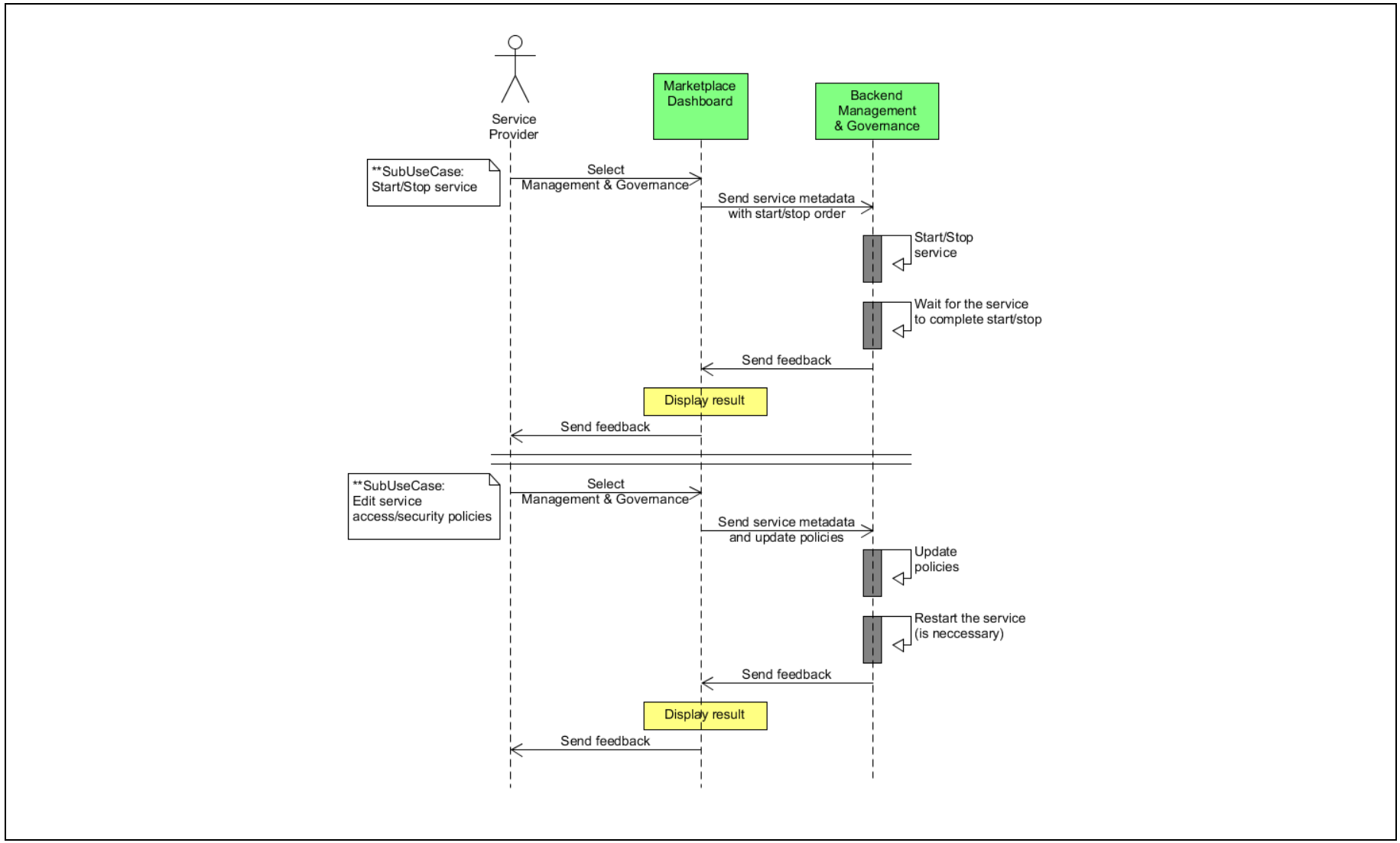
6.1.3.8 SUC M6

No	Name
SUC M6	Service advertisement
Short description	This use case aims to describe how a service provider can promote one of its existing services to the other marketplace users. The idea is that at the end of the process a new advertising landing page dedicated to a service exists in the marketplace. The process starts with the service provider decision to promote in the marketplace one of its already existing services. Via a specific marketplace function the service provider fills in a form specifying which service must be advertised to other users, the text and the pictures to be displayed in the advertising page. After being completed by the service provider, the request is processed by the backend of the Advertising module and the Ads mechanism in order to generate the new landing page requested. At the end of this process the new landing page is created. Once the landing page is active, service provider can intervene and correct or improve texts and graphics.


Service Provider
Marketplace Dashboard
Backend Advertising
Ads mechanism
Select Advertising
Request Advertisement
Send service metadata
Set up advertising landing page
Send publish landing page
Publish landing page
Send Publish Feedback
Send Advertising Feedback
Display service in Advertising

6.1.3.9 SUC M7

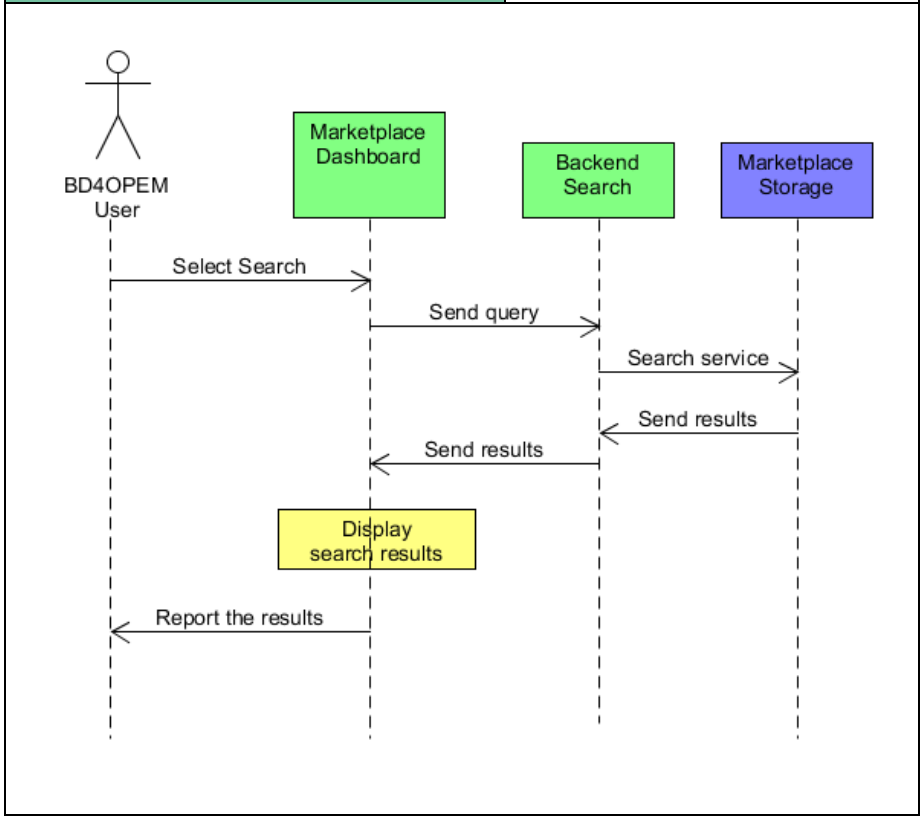
No	Name
SUC M7	Service management and governance
Short description	Via this use case implementation, the service provider is able to manage the lifecycle of a service, and specifically to ▪ manage service access and security policies (which users can access the service) ▪ stop and start the service ▪ administer / configure service aspects Service provider can select one of its services and, via specific marketplace interfaces, manage these mentioned technicalities. Each functionality informs the user about the success or failure of the user activity.


Service Provider
Marketplace Dashboard
Backend Management & Governance
**SubUseCase: Start/Stop service
Select Management & Governance
Send service metadata with start/stop order
Start/Stop service
Wait for the service to complete start/stop
Send feedback
Display result
Send feedback
**SubUseCase: Edit service access/security policies
Select Management & Governance
Send service metadata and update policies
Update policies
Restart the service (is neccessary)
Send feedback
Display result
Send feedback

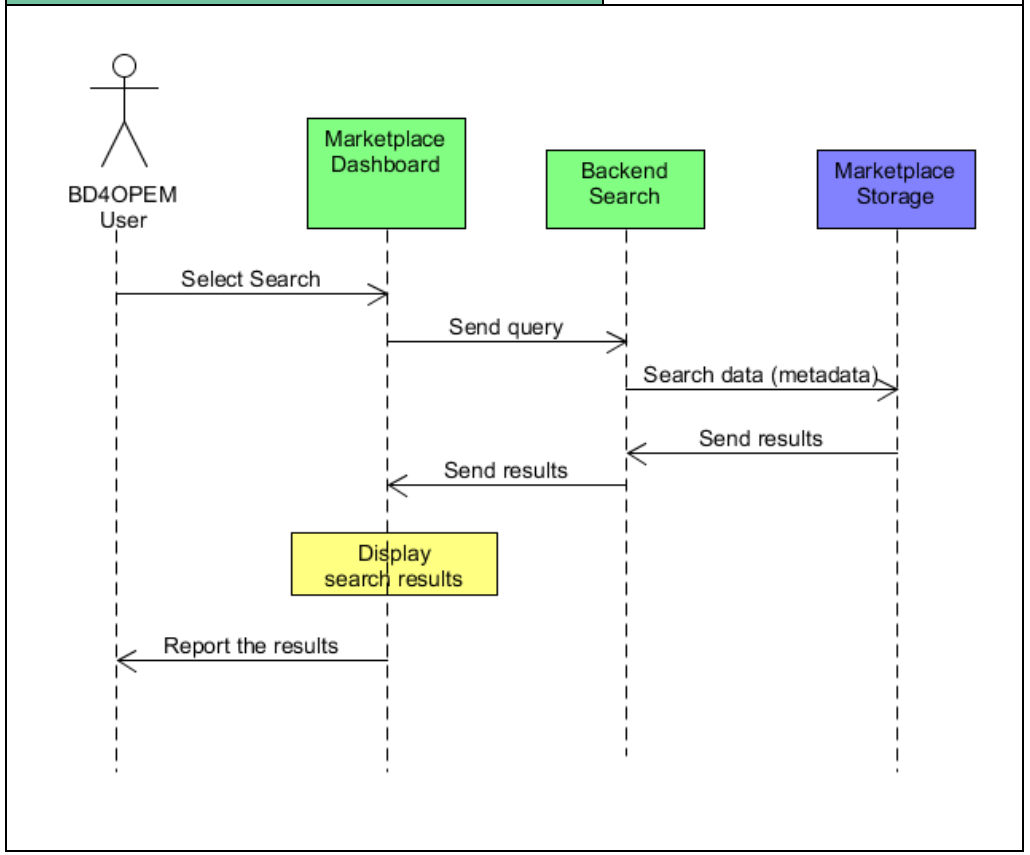
6.1.3.10SUC M8

No	Name
SUC M8	Search for services
Short description	SUC M8 is about searching information about services and data in the marketplace. Two different scenarios are to be considered, first one related to services querying, second one related to data querying. For each of these two functionalities, marketplace offers to the user a specific page that allows the user to specify a query and submit it to the system. In case of services search, the system queries metadata of all the services defined in the marketplace and retrieves the list of services that match the user query. In case of data search, the system queries metadata of all the datasets existing in the marketplace storage and retrieves the list of results that match the user query. From both the results page, user is able to move and visualize detail page of the desired object.

No	Name
SUC M8 (Scenario 1)	Search for services



No	Name
SUC M8 (Scenario 2)	Search for data



7 BD4OPEM reference Architecture. Deployment View

The physical or deployment view of the '4+1'Architectural View Model addresses the physical components envision in the five pilots of BD4OPEM project. This view defines the topology of software entities in the physical layer (hardware), as well as the physical connections between these components and physical constraints.

This view will be covered in the next deliverable, D2.4.

8 BD4OPEM reference Architecture. Use Case ("+1) View

Use Case or "+1" scenarios View is used to represent the use cases from the point of view of the different stakeholders (integrator, developers, users, etc,). In this context, this view aims at joining and putting in relation all other 4 views, through the functionalities of BD4OPEM platform that will be translated into concrete requirements.

Therefore, this "+1" scenario view is devoted to describing:

- The actors involved and how they interact in the platform. An introduction was detailed in the aforementioned section 4.6 (End Users and their high level interaction with BD4OPEM)
- Tools/assets/modules used by each pilot site
- Functional goals linking to the reason why a given user might act over a particular tool.

Given that, at the time this report is being written, the technical maturity of BD4OPEM components and services is not developed enough to provide a full linking to use cases and requirements, this view will be addressed in the next updated report for the architecture (D2.4).

9 Conclusions

In summary, this deliverable details the overview of BD4OPEM platform as an interoperable and scalable architecture combining relevant features from both ICT and energy domains. It is built upon five main layers, fully mapping the SGAM reference architecture and presented under the 4+1 View Model. This first architectural release covers the logical, implementation and process views.

It is the result of a collaborative work between all partners involved, with the goal to set the baseline for future iterations and implementation tasks. In fact, this deliverable includes the inputs and viewpoints from:

- *Concept design and use-cases described in D2.1*. This report is the major input to BD4OPEM architecture, since it provides the initial use cases. They focus on the development of an energy marketplace based on advanced analytics, with innovative services and solutions in the big data domain. D2.3 uses this as a baseline and provides an extensive layer and module description, as well as implementation schemas and process sequence diagrams.
- *Initial BD4OPEM architecture at proposal stage*. This draft proposal was used as a guideline and aiming at refining the proposed framework, taking SGAM Model as a reference, and detailing the interfaces and communication between the different modules.
- *Technical perspective*. BD4OPEM marketplace includes advanced and innovative analytical solutions that require the vision of component providers. This way, D2.3 captures the perspective of developers who validate the changes made as the different iterations of the architecture are processed.

These previous inputs, along with feedback received during this task, enabled the consolidation of this dedicated report which describes the logical design of the conceptual architecture addressing the concerns of all the stakeholders implicated in the following views:

- *Logical view* provides a full description of each of the five layers of BD4OPEM platform (Physical Layer, Data Layer, Analytics Layer, Marketplace Layer and Security, Privacy & Blockchain Layer), detailing their logical connections and being useful for end users.
- *Implementation view* describes the functionalities and internal and external specifications (interconnections, input-output data, dependencies) of the modules that make up each layer. This view shows the system from the perspective of a programmer and deals with software management.
- *Process view* depicts the sequence diagrams for each of the use cases, dealing with the dynamic aspect of the system represented from the perspective of a system integrator. Different UML diagrams show, step-by-step, the business and operational workflow between the modules of the whole architecture.
- Finally, both *Deployment view* and *Use cases view*, which address the physical components envisioned in BD4OPEM pilot sites and represent the use cases from the point of view of the different stakeholders will be covered in deliverable D2.4.

With the release of this report, BD4OPEM architectural design team will switch the focus towards BD4OPEM technical WPs, so that the detailed definition and development of components can be aligned with the agreements already included in this report. This way, those future implementations will be also easily mapped in the envisaged update of the architecture (D2.4), together with the aforementioned future inputs regarding deployment view and use case view.

10 References

[1] P. Kruchten (1995). Architectural Blueprints—The "4+1" View Model of Software Architecture. Paper published in IEEE Software 12 (6) November 1995, pp. 42-50

[2] CEN-CENELEC-ETSI Smart Grid Coordination Group. SG-CG/ M490/F_ Overview of SG-CG Methodologies. Version 3.0. 2014

[3] "D2.1 Concept design and use cases" (Confidential), BD4OPEM project (H2020, GA 872525), 2020.

[4] "ISO/IEC/IEEE 42010:2011 — Systems and software engineering — Architecture description," (https://www.iso.org/standard/50508.html).

[5] P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, P. Merson, R. Nord, and J. Stafford, "Documenting Software Architectures: Views and Beyond", Boston: Addison-Wesley, 2002 (https://pdfs.semanticscholar.org/6fbd/b44c124afa469a5a00c47bd375de633e9f3c.pdf)

[6] P. Eeles and P. Cripps, "The Process of Software Architecting". Addison-Wesley Professional, 2010

[7] T. B. Callo-Arias, P. America, and P. Avgeriou, "Defining execution viewpoints for a large and complex software-intensive system", Proceedings of WICSA/ECSA 2009. http://www.cs.rug.nl/paris/papers/WICSA09.pdf

[8] N. Rozansky and E. Woods, "Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives", Addison-Wesley, 2005

[9] https://www.w3.org/TR/did-core/

[10] https://www.w3.org/TR/vc-data-model/

[11] M. Mohsen, A. Majzoobi, A. Khodaei, "Distribution Asset Management through Coordinated Microgrid Scheduling". IET Smart Grid, September 2018, doi: 10.1049/iet-stg.2018.0076

[12] C. Freedman, "Distributed Analytics: Enabling More Insightful Solutions", November 2016.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 872525