SOFIE - Secure Open Federation for Internet Everywhere

779984

DELIVERABLE D5.2

Initial Platform Validation

Project title	SOFIE – Secure Open Federation for Internet Everywhere
Contract Number	H2020-IOT-2017-3 – 779984
Duration	1.1.2018 – 31.12.2020
Date of preparation	3.7.2019
Author(s)	Dmitrij Lagutin (AALTO), Yki Kortesniemi (AALTO), Antonio Antonino (AALTO), Tommaso Bragatto (ASM), Francesca Santori (ASM), Francesco Bellesini (EMOT), Michele Pagliaccia (EMOT), Giuseppe Raveduto (ENG), Vincenzo Croce (ENG), P. Anton (GT), M. Haavala (GT), M. Mardin (OPT), Antonis Gonos (OPT), Elias Kanakis (OPT), Asimakis Christodoulopoulos (OPT), David Mason (ROV), Ahsan Manzoor (ROV), Ioannis Oikonomidis (SYN), Sotiris Karachontzitis (SYN)
Responsible person	Ioannis Oikonomidis (SYN), oikonomidis@synelixis.com Sotiris Karachontzitis (SYN), karachontzitis@synelixis.com
Target Dissemination Level	Public
Status of the Document	Completed
Version	1.00
Project web-site	https://www.sofie-iot.eu/

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 779984.

Table of Contents

List of Figures

List of Tables

List of Acronyms

AMI	Advanced Metering Infrastructure
AWS	Amazon Web Server
BoEU	Block of Energy Unit
BLE	Bluetooth Low Energy
BP	Business Platform
DEDE	Decentralized Energy Data Exchange
DEFM	Decentralized Energy Flexibility Marketplace
DER	Distributed Energy Resources
DID	Decentralized Identifiers
DLT	Distributed Ledger Technology
DR	Demand Response
DSO	Distribution System Operator
EV	Electric Vehicle
EVSE	Electric Vehicle Supply Equipment
FSC	Food Supply Chain
GSM	Global System for Mobile communication
GUI	Graphical User Interface
GW	Gateway
IAA	Identification, Authentication and Authorization
IoT	Internet of Things
ML/VL	Medium Voltage / Low Voltage
MRMG	Mixed Reality Mobile Gaming
NORM	Next generation Open Real time smart Meter
PoC	Proof of Concept
PoI	Point of Interest
PV	PphotoVoltaic
QR	Quick Response
RBAC	Role Based Access Control
RES	Renewable Energy Sources
SM	Supermarket
SWS	Supervisor Web Server
TSO	Transmission System Operator
TR	Transportation
V2G	Vehicular to Grid
WH	Warehouse

1 Introduction

1.1 Scope of this document

This deliverable represents the first report about the system software architecture and initial validation results for the four SOFIE pilots. For each pilot, a proof of concept prototype has been implemented and demonstrated in the lab environment to prove the feasibility of SOFIE's innovation. The main objective has been to validate the key functionality of the SOFIE federation architecture and framework components in a number of well-defined scenarios, and to test specific use cases which are mapped to the application domain and the security requirements of each pilot, i.e. the food supply chain, the decentralized energy data exchange, the decentralized flexibility energy marketplace and the mixed reality mobile gaming sectors. Moreover, the document updates "D5.1 – Baseline System and Measurements" to clarify the application context of each pilot, to highlight constrains and unsolved issues of the corresponding application domains, to summarize KPIs and metrics expected to be fulfilled per pilot, as well as the capabilities and benefits that SOFIE can provide.

1.2 Structure of the deliverable

An individual section has been devoted to each one of the four SOFIE pilots to report its status, methodology, time plan and initial validation results. These sections are structured in the same way by using the following three subsections:

- Subsection X.1 presents the overview of the pilot describing its application context, SOFIE's added value, and the pilot's status and planned steps.
- Subsection X.2 updates the scenarios and the use cases considered in the pilot by incorporating the feedback that was collected through SOFIE communications activities since the first release of scenarios and use cases in D5.1. Based on the updated use cases, the domain and the security requirements of the pilot have been elicited and the system software architecture has been presented with the use of Unified Modeling Language (UML), describing both the static and dynamic behaviour of the system under development.
- Subsection X.3 presents the proof of concept prototype and the initial SOFIE validation results.

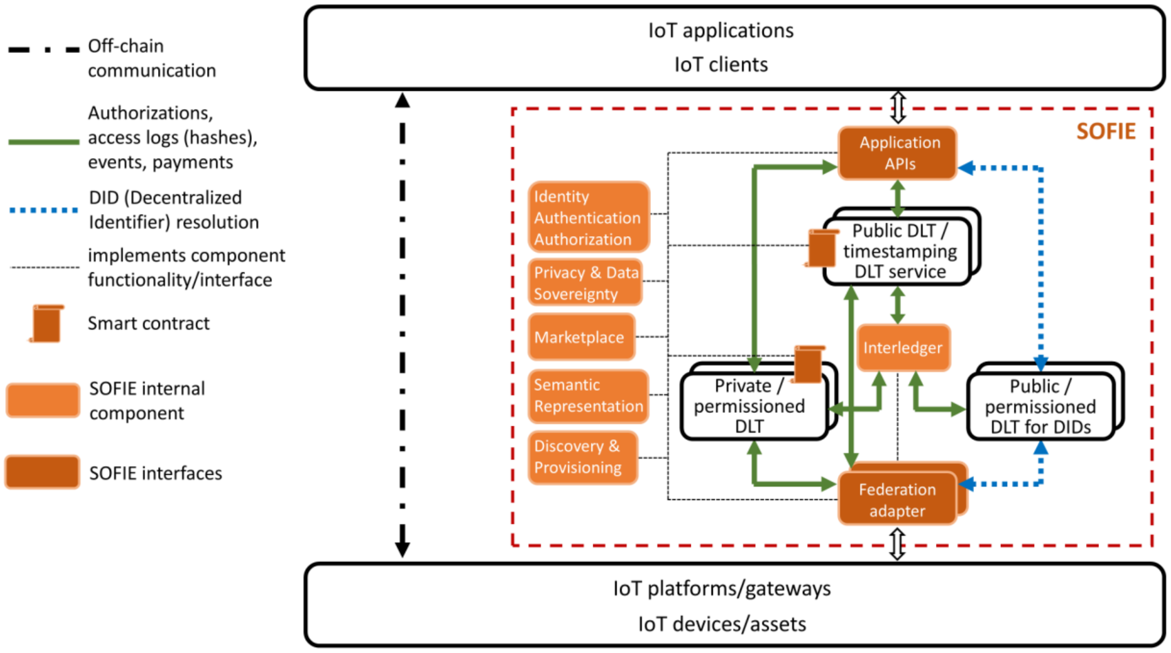
The final chapter concludes the deliverable.

1.3 Relation to other activities

In the second half of the SOFIE project, WP5 validation and testing activities will be further extended to cover and demonstrate the full SOFIE functionality, as the proof of concept prototypes which are presented in this document advance to minimum viable products supporting the corresponding business platforms. Aiming to deliver a final result of high quality assurance and control, these validation activities will progress and make use of all important outcomes of the rest SOFIE WPs, i.e. the most updated releases of SOFIE framework components implemented in WP2, the SOFIE CI/CD software development environment established in WP3, and the guidelines resulted by the qualitative and quantitative functional evaluation of SOFIE techniques in WP4. The results will be included in Deliverable "D5.3– End-to-end Platform Validation" following the final implementation of SOFIE federation framework and architecture.

2 SOFIE Reference Architecture

Figure 1 provides the overview of the SOFIE reference architecture containing the main functional components of the SOFIE framework (orange boxes), along with their cross-domain interactions with external domains/components (white boxes) as defined SOFIE D2.4[1].



Off-chain communication
Authorizations, access logs (hashes), events, payments
DID (Decentralized Identifier) resolution
implements component functionality/interface
Smart contract
SOFIE internal component
SOFIE interfaces
IoT applications
IoT clients
SOFIE
Application APIs
Identity Authentication Authorization
Privacy & Data Sovereignty
Marketplace
Semantic Representation
Discovery & Provisioning
Public DLT / timestamping DLT service
Interledger
Private / permissioned DLT
Public / permissioned DLT for DIDs
Federation adapter
IoT platforms/gateways
IoT devices/assets

Figure 1. The SOFIE framework architecture

The lowest level of the architecture contains IoT assets (or resources), that include IoT sensors for sensing the physical environment, actuators for acting on the physical environment, and boxes with RFID tags that are used to transport products. IoT assets can be connected to or integrated in devices. *IoT platforms* include platforms with data stores, where measurements from sensors are collected and made available to third parties, as well as servers providing IoT services.

The *federation adapters* interface the IoT platforms with the SOFIE framework and implement supportive functionality for various SOFIE components. Note that a part of the adapter's functionality can be implemented in smart contracts. Moreover, different scenarios and pilots can utilize different types of federation adapters, which implement different functionality.

The architecture figure emphasizes the *interledger functionality*, which is responsible for interconnecting different types of DLTs that can have different features and functionality in terms of transaction throughput, latency, cost, security, and access rights. The interledger framework component will contain a collection of interledger techniques over different DLTs which will be customized per case to fulfil intreledger scenarios which are considered in pilots. Providing interledger mechanisms to interconnect different DLTs allows companies and consortiums to select private/permissioned distributed ledgers based on their specific requirements and constraints. Hence, interledger mechanisms can enhance the interoperability across different IoT platforms that utilize different distributed ledger technologies.

[1] SOFIE D2.4: SOFIE Federation Architecture, 2nd version, June 2019. Available from: https://media.voog.com/0000/0042/0957/files/SOFIE_D2.4-Federation_Architecture_2nd_version_v1.00.pdf

The other SOFIE framework components are: *Identity, Authentication, and Authorization (IAA)*, which provides identity management and supports multiple authentication and authorization techniques; *Privacy and data sovereignty*, which provides mechanisms that enable data sharing in a controlled and privacy preserving way; *Semantic representation*, which provides tools for describing services, devices, and data in an interoperable way; *Marketplace*, which allows participants to trade resources by placing bids and offers in a secure, auditable, and decentralized way; and *Discovery & provisioning*, which provides functionality for the discovery and bootstrapping of services.

Finally, the upper component of the architecture are the *application APIs*, which provide the interfaces for IoT clients and applications to interact with the SOFIE framework.

The current SOFIE architecture is explained in more detail in SOFIE Deliverable “D2.4 - Federation Architecture, 2nd version”, while the framework components will be described in more detail in SOFIE deliverable “D2.5 - Federation Framework, 2nd version” due in August 2019.

3 Food Supply Chain Pilot

3.1 Pilot overview

3.1.1 Application context, constraints and unsolved issues

The initial design and application context of the Food Supply Chain (FSC) pilot has been reported in D5.1[2]. As stated there, the pilot considers the field-to-fork route of a food product (i.e. table grapes) over a number of heterogeneous business segments and corresponding IT infrastructures which are deployed along the FSC path. The pilot aims to show how SOFIE research and innovation advances in IoT federation and interoperability, data safety and privacy can be applied over a number of decentralized platforms to enhance the traceability, quality control and safety of products, as well as the mutual trust among the participants (companies) of the FSC.

The high-level overview of the FSC pilot is shown in Figure 2. The pilot aims to establish and demonstrate a provenance chain Business Platform (BP) that covers the farming, storage, distribution (logistics), and retail subdomains of the FSC. The setup is comprised of a decentralized distributed system, where data is collected from a number of heterogeneous, federated IoT platforms. The system makes use of SOFIE federation architecture and framework components to securely manage, store and transform the collected data, so as to optimize supply chain collaboration and collaborative advantage among participating companies, e.g. the gathering, exchanging and improving of resources and their management.

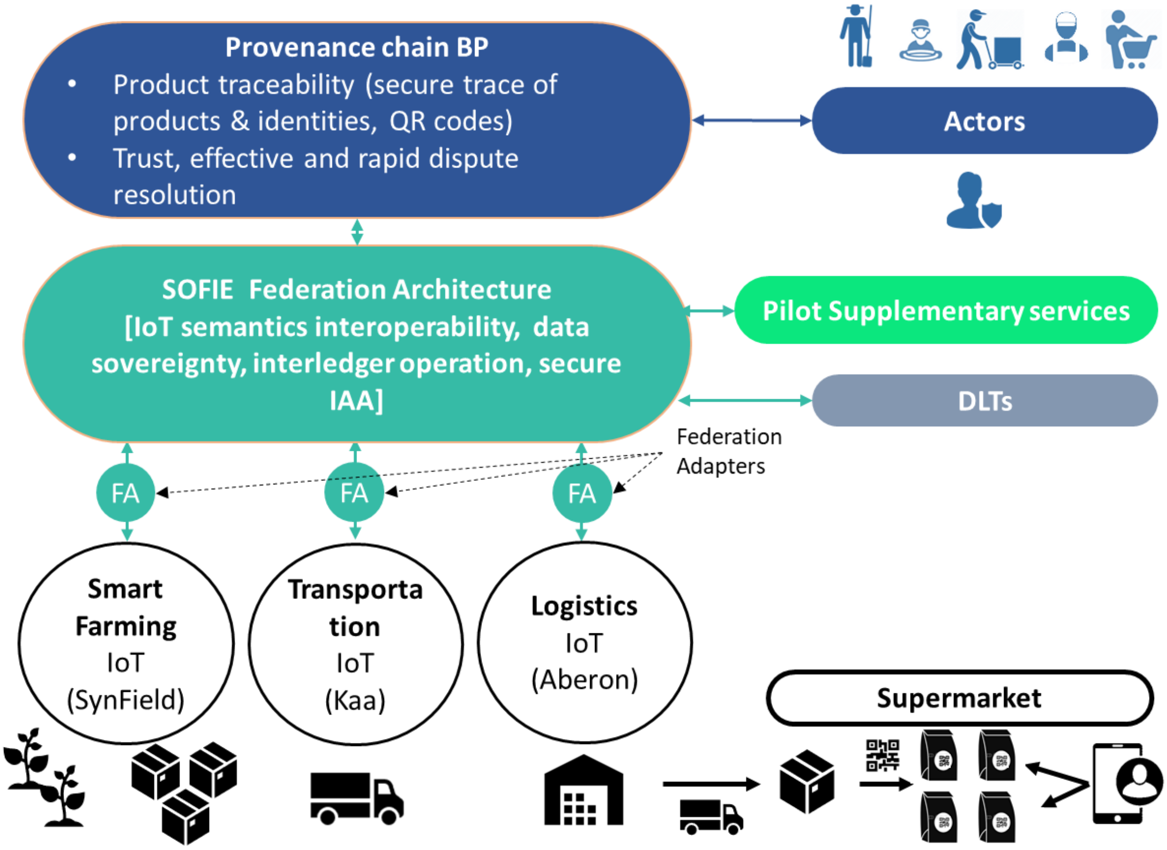

Provenance chain BP
• Product traceability (secure trace of products & identities, QR codes)
• Trust, effective and rapid dispute resolution
Actors
SOFIE Federation Architecture [IoT semantics interoperability, data sovereignty, interledger operation, secure IAA]
Pilot Supplementary services
DLTs
Federation Adapters
FA
FA
FA
Smart Farming IoT (SynField)
Transporta tion IoT (Kaa)
Logistics IoT (Aberon)
Supermarket

Figure 2. Overview of the food supply chain pilot

[2] SOFIE D5.1: Baseline System and Measurements, June 2018. Available from: http://media.voog.com/0000/0042/0957/files/SOFIE_D5.1-Baseline_System_and_Measurements.pdf

As was defined in D2.3[3], the main asset in the established BP is the box that carries products from the field to the selling point. In the SOFIE pilot setup, these boxes are the property of the transportation company which has the responsibility to transfer them form one site to another, e.g. from the field to the warehouse or from the warehouse to the supermarket. By using RFID technology (i.e. by attaching an RFID tag to each box), assets (and the included products) are mapped to virtual entities which represent them digitally in the data model of the system. As assets move over the chain, transactions over them as well as various activities of involved actors and organizations are collected and processed to enable product tracking from the field to the selling point. The pilot takes advantage of SOFIE innovation to demonstrate the following two important services:

i) A reliable and data integrity preserving product traceability service for final customers using QR codes that provides complete history of products from the field to the supermarket shelf.

ii) A trusted audit service for companies participating in the BP that enable fast and effective product quality audits to identify safety risks and points of failure in cases where a safety or quality issue is detected in the final product.

In the FSC pilot, the deployed provenance chain BP ensures wide visibility of supply chain information, traceability of assets, and secure data exchange among the decentralized, federated IoT environments, without forcing additional changes to their infrastructures, equipment and security policies. The pilot leverages a hierarchical topology of cooperating DLTs to improve transparency and traceability of assets and build a robust and secure data management framework that verifies integrity of exchanged data and ensures identity and authenticity control of involved entities. The challenge is to achieve federation and interoperability at the data level and to introduce common semantic rules and API models which are independent of the underlying protocols and technology.

3.1.2 SOFIE added value

In the current FSC management, the provision of traceability services is based on a centralized authority to manage and share information (that may also include sensitive or private data), possible with limited automation and reliability. As a result, in many cases, it is quite challenging to provide each interested party a common view of the assets' conditions and transactions over the chain since, usually, this information is stored in multiple locations, with each one being accessible by just one entity (or a small number of entities). In SOFIE, the provenance chain BP does not rely on a single entity or organization to establish trust among the participating organizations. On the contrary, the pilot establishes a hybrid, decentralized data organization and management topology, where part of data is stored in local IoT silos and another part (alongside with related metadata) is stored in a shared system which is composed of a hierarchical topology of cooperating DLTs. This way, the pilot aims to enhance the integrity and transparency of assets' traceability, and to provide advanced automation in managing transactions over them.

Despite the fact that DLTs are continually gaining ground in the transformation of the supply chain domain[4], an easy-to-use and non-disruptive technical solution to federate these systems and enable them to securely interoperate in composing and analysing all the valuable information that flows over the chain has not been presented so far. In contrast to other approaches that combine blockchain technology with IoT in supply chain[5], the advantage of SOFIE is that it is agnostic to the technology and technical specification of the integrated IoT

[3] SOFIE D2.3: Federation Framework, 1st version, October 2018. Available from: https://media.voog.com/0000/0042/0957/files/SOFIE_D2.3-Federation_Framework_1st_version_v1.00.pdf

[4] Blockchain ready manufacturing supply chain using distributed ledger

[5] M. Montecchi, K. langger, M. Etter, "It's real, trust me! Establishing supply chain provenance using blockchain," Elsevier, Business Horizons, Volume 62, Issue 3, pp. 283-293, May–June 2019

environments, i.e. segments of the FSC. In this scope, interoperability across these environments is achieved by applying SOFIE's common semantic model for IoT resources and services. Beyond this, the pilot also leverages the SOFIE interledger operations and Identity Authentication and Authorization (IAA) services to guarantee i) end-to-end security for IoT data transactions, ii) integrity in data processing and management over the federated environment, and secure access in IoT resources and measurements[6]. Based on this approach, the following table summarizes the added value of SOFIE for the food chain pilot:

Table 1. SOFIE's added value compared to the existing traceability systems in supply chain

Legacy FSC systems	**SOFIE added value**
Centralized data management (without using any DLTs)	• Increases traceability of products and ensures integrity of critical data without a centralized authority. • Increases trust among companies and transparency in data management. • Automates interaction/transactions over heterogeneous IoT ecosystems corresponds to the various segments of the FSC. • Reduces the chances of fraud, cutting out corresponding mediation expenses and transaction costs and demonstrates proof of interaction between different parties.
Systems that make use of DLT and IoT technology + IoT[7,8]	• Introduces a transparent data adaptation layer for IoT platforms. • Proposes an easy to use and non-disruptive solution to federate heterogeneous IoT environments. • Proposes a solution which is agnostic to the existing DLT and IoT technology. • Enables anonymity and privacy protection of sensitive information.
Audits, quality checks (sample-based)	• Improves safety and quality of the delivered products by opening up more effective proactive audits (automation) for product quality assurance and improves reactive solutions in the cases of detected issues. • Increases transparency and trust in auditing processes in cases of safety/quality issues and disputes among the companies of the food provenance BP.

The KPIs of the pilot are summarized in Table 2:

Table 2. KPIs in food supply chain pilot

Technology Level	
Enabling hardware deployment	• Devices deployment effort and configuration time • Compatibility with existing infrastructure
Enabling software and development of applications	• Deployment effort for 3rd parties' application • Behavioural UX for SOFIE web/mobile applications
Federating platforms	• Degree of interoperability • Federate ≥ 3 IoT platforms • Transaction over ≥ 3 DLTs

[6] SOFIE Deliverable D2.4, Federation Architecture, 2nd version – section 4

[7] https://diamonds.everledger.io/

[8] https://www.provenance.org/

Business Level	
Financial metrics	• Items of goods make it to the market • Cost of transaction fees in using DLT to store information
Customer metrics	• Customer satisfaction • Time and effort for QR creation • Percentage of product defects and disputed resolved

The above KPIs will be properly quantified and measured during the final validation of the SOFIE platform in the FSC pilot and the results will be reported in "D5.4 – Final Validation and Replication Guidelines".

3.1.3 Pilot status and time plan

At this stage, FSC is in the development phase. As planned in WP5, all SOFIE pilots will try to keep the same timeline and be aligned in each WP deliverable, as summarized in Table 3.

Table 3. Timeline and main outcomes per reporting period for all SOFIE pilots

Period	Achievement	Reported
M1-M6	• Definition of the baseline technology, scenarios and use cases.	D5.1
M6-M18	• Definition of pilot software architecture • Initial implementation of supplementary components used in the pilot • Proof of concept validation of SOFIE federation architecture and components (as presented in §3.3)	D5.2
M18-M30	• Engagement and training of end users • First implementation and deployment of pilot architecture (including integration of SOFIE components) • Initial on-site validation of SOFIE architecture/framework (second release) through deployment and testing pilot's end-to-end services (i.e. QR usage, product audit)	D5.3
M30-M36	• Update pilot architecture and its deployment based on feedback from M18-M30 period. • Final on-site validation of SOFIE architecture/framework (final release) • KPIs quantification and overall pilot assessment • Provision of replication guidelines	D5.4

3.2 Pilot scenarios, requirements and system architecture

The actors in the FSC pilot are the following:

- **Producers:** Producers are responsible to enter key input data relate to the farming and cropping phase of the product. This information may include cultivation type of the product, location/timezone of the field, dates when pesticides were used etc.
- **Transportation employees:** Transporters transfer assets from one side to another. They also register and release assets which will be used to carry products.
- **Warehouse employees:** They handle assets in the premises of the warehouse. In particular, they store product according to its safety specifications and also packetize content in sellable packets.
- **Supermarket employees:** Supermarket employees receive and open assets to put packets on the shelf. For each packet they create a QR code that contains the full history of the contained product.

- **Customers:** Customers scan QR codes by using their smartphones to access product history. In the case they detect any quality issue, they report to the supermarket employee.
- **Consortium certifier organization:** administrates the common ledger system that stores data related to assets' tracking. It acts as an authority that grants and enables access to participants' data. It is also the actor who activates and supervises the process of audit and dispute resolution in the case of a breach or when a customer reports an issue about the product quality.

In the pilot setup, the data entry points are: a) the various IoT platforms/services that collect data about assets as those are moving from the field to the selling point, as well as b) a web application, which is used by the actors to interact with assets and inject more information into the system, e.g. additional specifications for products, confirmations of responsibility transfers, metadata about products etc. All data sources are authenticated by using the SOFIE IAA services before being able to push data into the data management layer of the system. This way data exchange is secured between all data entry point and the management layer. Actors' and IoT platforms' credentials (and unique identifiers) are created by the system when the corresponding participant company registers in the provenance business platform. According to the applied data sovereignty rules, actors' profiles are anonymized in the system data management layer and their personal information (e.g. position in companies, qualifications, association with products etc.) stays at local business level. This way, actors are both certified to interact with the system but they also remain completely anonymous to entities which act outside of their business environment.

In the following subsections, an updated version of the demonstrated scenarios and use cases in the FSC pilot is presented. This update to what was originally described in D5.1 was necessary to improve the modeling of activities and to include further operational aspects that take place in the real FSC industry. The updated scenarios and used cases are later analyzed to identify pilot domain and security requirements, based on which the first version of the pilot system architecture is designed.

3.2.1 Scenarios and use cases

The pilot covers two scenarios called Food Quality Monitoring and Food Quality Audit, respectively. The first scenario summarizes the sequence of actions in preparing and accessing the QR codes, which are attached to the product packages on the supermarket shelf (containing the history of the product from the field to the shelf). In the second scenario, the story is extended to include an audit case among the business companies participating in the food chain.

3.2.1.1 Food supply chain pilot scenarios

SCENARIO	Food Quality Monitoring
History	v0.2
Key Actors	Producer, transporter A, warehouse employee, transporter B, supermarket employee, supermarket customer
Assumptions / Dependencies	1) All business companies have registered in the provenance chain BP. 2) The type of the product which will be transferred from the producer to the seller has been agreed. 3) IoT platforms are installed and running, including: • SynField field nodes and Cloud platform • Transportation platform with boxes equipped with RFID tags, and vehicles equipped with RFID readers and temperature sensors • Aberon IoT platform that monitors temperature in storage rooms 4) SOFIE framework components have been installed and are running. 5) FSC web application is operational, and actors have been registered.

Objective(s)	1) Collect, filter and manage data and metadata from various IoT environments and other data entry points (i.e. web application). 2) Trace the change of responsibility over assets, as they move across the whole supply chain. 3) Create QR codes to encode product history. 4) Respect the privacy requirements of the involved actors and organizations and guarantee the integrity of the exchanged data.
Description	**Step 1)** The producer owns a field, where she grows table grapes. In this field, a SynField smart farming installation exists to help her to control and monitor crop production and quality. The SynField Head Node collects and transmits both climate and crop status data to the SynField IoT cloud platform. The producer has an agreement to sell her goods to a specific warehouse or supermarket which is also a member of the provenance chain BP. At the suitable time, she calls a transportation company that also participates in the BP to obtain boxes, which will be used for the transportation of grapes when they are ready to be harvested. Boxes are the property of the transportation company and each one is identified by a unique RFID tag that is attached to it. When the producer fills a (group of) box(es), she accesses the FSC web application to insert additional (meta)data about the contained product, e.g. define type and origin of product, location of the field, dates and types of fertilizers used, total weight of the product inside the box, etc. **Step 2)** When the producer's goods are ready to be transported, she arranges for a transportation vehicle to arrive at her field. The transportation vehicle is equipped with a unique RFID reader and a temperature sensor both deployed in the cabin of the truck. Once the boxes are filled with grapes, they are loaded into the truck, so the RFID reader detects their presence. The producer and the transportation employee/driver (transporter A) access the FSC web application to initiate and confirm the transfer of responsibility for the detectable boxes. The FSC web application flow guides both to agree about the status of the product inside each box (e.g. weight of each box, ripening level of the product etc.). Once this has happen, the boxes are sealed. **Step 3)** Transporter A drives the vehicle to the Warehouse (WH). The temperature sensor inside the truck cabin continually measures and transfers temperature values to the transportation IoT cloud platform. Once the vehicle reaches the WH, the transporter A accesses the FSC web application to determine the boxes that will be delivered to the warehouse. The WH employee also accesses the FSC web application to confirm the transaction. In the warehouse, the product is packetized in packets of the same weight, e.g. 1Kg, and the packets are placed again inside the boxes. The WH employee accesses the FSC web application to specify which boxes will be used to store packets and define the storage location for each box based on the specific quality and safety specifications of the product (e.g. appropriate temperature, ripening level of the product etc.). While the boxes are stored at the WH, Aberon IoT collects information regarding the location of boxes, the temperature in the storage room and the storage duration. **Step 4)** When one or more boxes should be transferred from the warehouse to the supermarket, the WH employee calls the transportation company to send a transportation truck. Once the boxes have been loaded into the truck, the transporter B and the WH employee access FSC web application to confirm the transaction and to transfer responsibility of boxes. The boxes arrive at the supermarket, where the supermarket employee unloads them from the truck. The Transporter B and the supermarket employee access the FSC web application to confirm the transfer of responsibility for the delivered boxes. Sometime later, the supermarket employee opens the boxes and accesses the FSC web application to create a QR code per box that contains the complete history of the box (thus, the product inside the contained packages) from the field up to that time. Each QR code encodes also the ID of the box to which it refers to. For every packet of each specific box, a copy of the corresponding QR code is attached to its

	surface. Finally, packets are placed on the appropriate spot inside the customer area and boxes are returned to the transportation company. **Step 5)** A customer scans a QR label attached to a package containing grapes with his smartphone and reads the full product history from the field to the shelf containing information about origin and type of the product, harvesting date, date the product reaches the supermarket, max/min temperature value during transportation and storage etc.
Services	• Representation of IoT data based on common IoT semantics. • Data privacy and secure communication in data and metadata exchange between internal and end-to-end services. • Interledger operations. • IAA services of both actors, IoT environments and internal communication processes.
Metrics	• Federation of ≥ 3 IoT platforms • Transactions over ≥ 3 DLTs • Behavioural UX for SOFIE FSC web application (measured by feedback from employees) • Items of goods sold (compared to goods without QR labels) • Customer satisfaction (measured by questionnaires)
SCENARIO	**Food Quality Audit**
History	v0.2
Key Actors	Producer, transporter A, warehouse employee, transporter B, supermarket employee, supermarket customer, consortium certifier organization
Assumptions / Dependencies	1) … 5) as in the previous scenario. 6) Audit service is operational.
Objective(s)	Audit reasons of having defective products and resolve disputes between the companies participate in the provenance chain BP.
Description	The description extends FoodChain_Scenario_1 by including also the following activities: **Step 3)** While being at the WH, the temperature in a room (let room 123), where a number of boxes has been placed rises above the level that has been set by the WH employee for that specific room. Consequently, an alert is generated and sent to the monitoring application of the Aberon IoT platform. The WH employee decides not to act, since the temperature shortly recovers to the desired levels. **Step 5)** The customer scans the QR label of a package containing grapes from a box which was stored in room 123 of the WH when the incident described in step3 took place. The customer notices that the maximum temperature of the product is unusually high and reports that to the supermarket employee. **Step 6)** The employee confirms that the attached QR code is a valid one and that the temperature issue appears when someone scans it. He scans the QR labels of several packages that exist on the shelf to find out whether the issue holds for other packages too, as it is the case. He records the IDs of the boxes which were carrying "problematic" packages and then accesses the FSC web application to activate the audit service. He describes the incident by providing information about reception date of "problematic" boxes and he requests to know whether other boxes have been received with the same problem. **Step 7)** Upon the reception of the request for audit, the companies/members of the provenances chain BP involved in the transfer and storage of the specific box from the field to the supermarket are identified. The consortium certifier organization (who supervises the audit process) requests the identified transportation and WH companies to search and report any issue with respect to the carrying/storage conditions of the specific box.

	Step 8) Transportation companies check the history data about the temperatures during transportation of the box from the field to the WH and from the WH to the supermarket, and they report that no issue was detected. They provide both temperature records to the consortium certifier organization and the block numbers in the public blockchain used by the BP where their hashes have been uploaded. The WH company informs the consortium certifier organization that about what the problem was, when happened, and which other boxes may also have been affected.
Services	• As in the previous scenario
Metrics	Additional to the metrics of FoodChain_Scenario_1 • Degree of interoperability • Percentage of product defects • Percentage of resolved disputes

3.2.1.2 Food supply chain pilot use cases

The following UML use case diagram summarizes how the various actors interact with the FSC pilot. Each illustrated use case is linked with some activity on behalf of the corresponding actor(s) and implies some actual functionality that must be performed by the system.

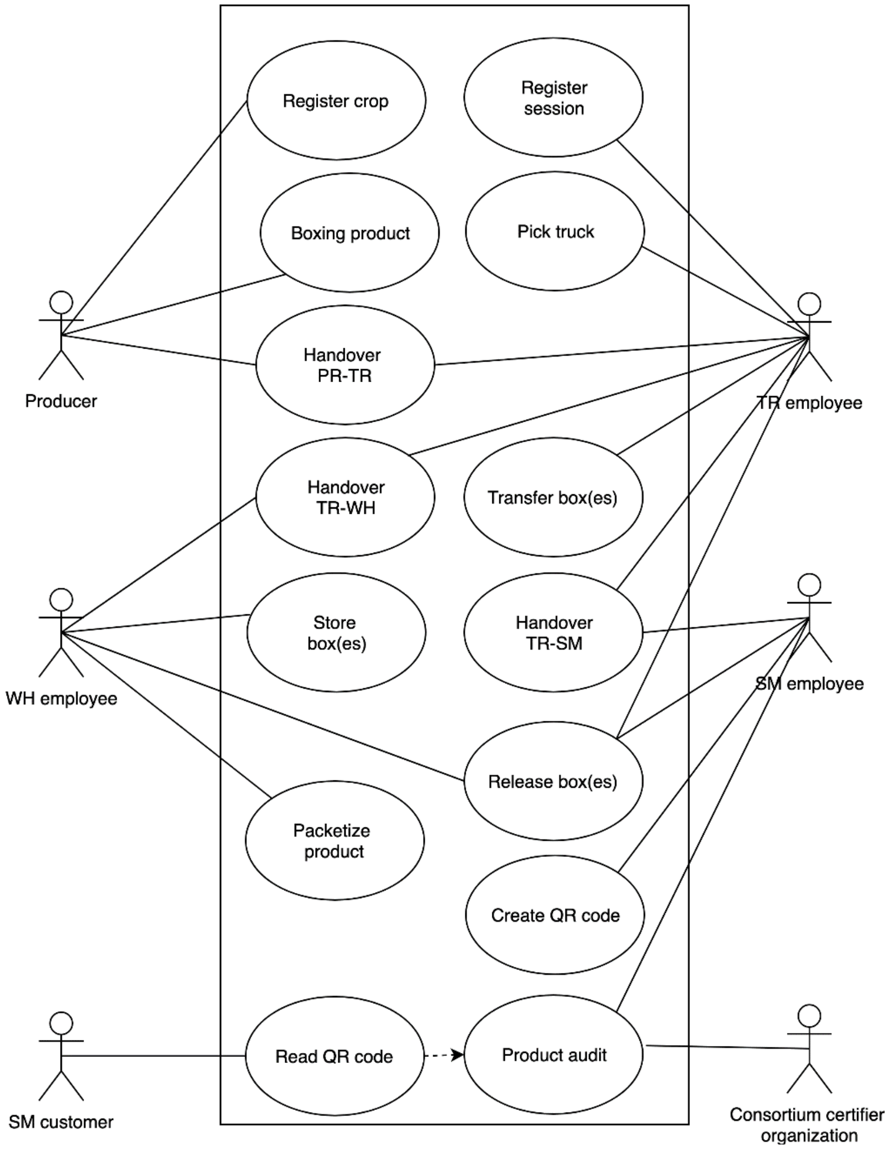


Figure 3. Actors' interaction with the food supply chain system

The use cases which are shown in Figure 3 are analyzed further in Table 4 by using the template that has been introduced in D5.1.

Table 4. Use cases in the food supply chain system

ID	Name	Actors
FSC_UC1	Register crop	Producer
FSC_UC2	Box product	Producer
FSC_UC3	Hand over product: PR-TR	Producer, TR employee
FSC_UC4	Hand over product: TR-WH	TR employee, WH employee
FSC_UC5	Register session	TR employee
FSC_UC6	Pick truck	TR employee
FSC_UC7	Transfer box(es)	TR employee
FSC_UC8	Hand over product: TR-SM	TR employee, SM employee
FSC_UC9	Place box(es)	WH employee
FSC_UC10	Packetise product	WH employee
FSC_UC11	Create QR code	SM employee
FSC_UC12	Release box(es)	SM employee, WH employee
FSC_UC13	Read QR code	SM customer
FSC_UC14	Product audit	Consortium certifier organization

USE CASE Description	
ID	FSC_UC1
Name	Register crop
Actors	Producer
Storyline	The producer accesses the FSC web application to register the field and the product variety which will be transferred to the warehouse. He provides information about farm location, crop establishment date and product variety.
Trigger events	Product traceability history must be able to include farming data for the whole growing season.
Preconditions	There is an agreement between the producer and the warehouse and/or the supermarket company about the product which will be transferred from the former to the latter one.
Postconditions	The date is defined from which the farming/growing conditions of the product crop are monitored.
Related scenarios	Food quality monitoring Food quality audit.

USE CASE Description	
ID	FSC_UC2
Name	Box product
Actors	Producer
Storyline	The producer accesses the FSC web application to specify the actual box(es) which will be used to carry the product to the warehouse. He also provides information about farming process/history of the product which is deposited into the box/boxes (e.g. harvesting date, used fertilizers etc.).
Trigger events	One or more boxes should be used to carry grapes from the field to the warehouse.
Preconditions	The used box/boxes should have been registered (by the transportation employee) and delivered to the producer.
Postconditions	Growing history of the product (as it is monitored by the SynField IoT) is linked to the specific assets (boxes).
Related scenarios	Food quality monitoring Food quality audit.

USE CASE Description	
ID	FSC_UC3
Name	Hand over product: PR-TR
Actors	Producer, TR employee
Storyline	The producer accesses the FSC web application to specify the actual box(es) which will be delivered to the TR employee. Then, the TR employee accesses the FSC web application to accept the responsibility of these boxes and confirm the transaction. Parameters such as the weight of boxes, ripening level of product etc. are also agreed on and the boxes may be sealed.
Trigger events	Boxes carrying grapes should be delivered to the TR employee.
Preconditions	The used boxes have been registered by the TR employee. The boxes have been filled with grapes and are loaded into the transportation truck (detected by the RFID sensor).
Postconditions	The responsibility of box/boxes has been shifted to the TR employee.
Related scenarios	Food quality monitoring. Food quality audit.

USE CASE Description	
ID	FSC_UC4
Name	Hand over product: TR-WH
Actors	TR employee, WH employee
Storyline	The TR employee accesses the FSC web application to specify the actual

	box(es) which will be delivered to the specific WH. The WH employee accesses the FSC web application to accept the responsibility for these boxes and to confirm the transaction. In the opposite direction: The WH employee accesses the FSC web application to specify the actual box(es) which will be received by the TR employee and activate the transaction. The TR employee accesses the FSC web application to confirm the transfer of responsibility for these boxes.
Trigger events	The TR employee delivers a number of boxes to the WH employee. In the opposite direction: The WH employee delivers a number of boxes to the TR employee.
Preconditions	The boxes have been loaded into the transportation truck and they are detected by the deployed RFID sensor.
Postconditions	Responsibility of boxes has been shifted from the TR employee to the WH employee. In the opposite direction: Responsibility of boxes has been shifted from the WH to the TR employee.
Related scenarios	Food quality monitoring. Food quality audit.
USE CASE Description	
ID	FSC_UC5
Name	Register session
Actors	TR employee
Storyline	The TR employee accesses the FSC web application to establish a session linked to the product transfer from the field to the fork. The employee specifies one or more boxes (by using their RFID tags) to be used for carrying the specific product from the specific producer. The boxes are delivered to the producer.
Trigger events	Boxes are needed to be used to carry products from the field to the supermarket
Preconditions	Each selected box is attached to a unique RFID tag. The ID of the producer who will use the boxes should be known to the TR employee.
Postconditions	Assets (boxes) are linked to the specific product and producer.
Related scenarios	Food quality monitoring. Food quality audit.
USE CASE Description	
ID	FSC_UC6
Name	Pick truck

Actors	TR employee
Storyline	The TR employee access the FSC web application to declare the truck that will be used to carry boxes from an origin site (field or WH) to a destination site (WH or supermarket).
Trigger events	Transportation company has been called to transfer boxes
Preconditions	An RFID reader has been installed in each transportation truck.
Postconditions	The IDs of the RFID reader and the temperature sensor of the truck are linked to the planned transfer.
Related scenarios	Food quality monitoring. Food quality audit.
USE CASE Description	
ID	FSC_UC7
Name	Transfer box(es)
Actors	TR employee
Storyline	TR employee drives the transportation truck to deliver boxes either to the warehouse or the supermarket.
Trigger events	Boxes need to be transferred between origin and destination sites.
Preconditions	The boxes have been loaded into the transportation truck and they are continually tracked by the RFID reader.
Postconditions	Measurements collected by the sensors installed inside the truck trolley are stored in the local ledger/DB of the IoT transportation platform.
Related scenarios	Food quality monitoring. Food quality audit.
USE CASE Description	
ID	FSC_UC8
Name	Hand over product: TR-SM
Actors	TR employee, SM employee
Storyline	The TR employee accesses the FSC web application to specify the actual box(es) which will be delivered to the Supermarket and activate the transaction. Then, the Supermarket employee accesses the FSC web application to accept responsibility of these boxes and confirm the transaction.
Trigger events	The TR employee delivers a number of boxes to the supermarket.
Preconditions	The boxes have been loaded into the transportation truck and they are detectable by the RFID reader.
Postconditions	Responsibility of boxes has been shifted from the TR employee to the Supermarket employee.
Related scenarios	Food quality monitoring.

	Food quality audit.
USE CASE Description	
ID	FSC_UC9
Name	Store box(es) in the WH
Actors	WH employee
Storyline	The WH employee accesses the FSC web application to specify the storage rooms where each box is placed, based on the quality specification of the product (e.g. ripening level, temperature etc).
Trigger events	The WH employee places the boxes in the WH.
Preconditions	Box(es) have been received by the WH employee. Temperature is measured in each used storage room.
Postconditions	Boxes are placed in storage rooms where temperature conditions are continually monitored.
Related scenarios	Food quality monitoring. Food quality audit.
USE CASE Description	
ID	FSC_UC10
Name	Packetise product
Actors	WH employee
Storyline	The WH employee deposits the product from one or more boxes into the food packaging automation system where packages are created. The packages are placed inside the boxes and stored in the WH facilities. By accessing the FSC web application, the WH employee specifies the boxes which will be used to feed product into the packaging system and which boxes will be used to store created packages.
Trigger events	Product inside the boxes which were delivered to the WH must be packetized in packets of the same weight.
Preconditions	The boxes contain raw products.
Postconditions	There are some boxes that contain packetized products.
Related scenarios	Food quality monitoring. Food quality audit.
USE CASE Description	
ID	FSC_UC11
Name	Create QR code
Actors	SM employee
Storyline	The SM employee accesses the FSC web application to create a QR code per box that records all information about the contained product from the field to the

	supermarket. The created QR code includes also information about which box was used to carry product and when QR code was created.
Trigger events	Product packages should be created and placed on the shelf.
Preconditions	The box(es) have been delivered to the supermarket.
Postconditions	QR labels are attached to every package containing part of the product which was transferred inside a box. Created QR codes contain all the product history from the field to the supermarket and also the RFID tag of the box which was used to carry the product.
Related scenarios	Food quality monitoring. Food quality audit.
USE CASE Description	
ID	FSC_UC12
Name	Release box(es)
Actors	WH employee, SM employee, TR employee
Storyline	The WH and the SM employee releases one or more boxes after removing from inside all the contained product (either raw or packetized). The TR employee releases the boxes which finally haven't been used to carry products from the field to the warehouse.
Trigger events	Boxes have become empty or not used at all. They may be used again after they are returned back to the transportation company.
Preconditions	The WH employee has emptied the product from the boxes to packetize it. The SM employee has produced attached a QR label to every packet which was contained in the box
Postconditions	Released boxes should be delivered to the transportation company. After that, they are ready to be used again to carry products from the field to the fork.
Related scenarios	Food quality monitoring. Food quality audit.
USE CASE Description	
ID	FSC_UC13
Name	Read QR code
Actors	SM customer
Storyline	The SM customer uses his smartphone to scan a QR code and get the full history of the product which is contained in a package.
Trigger events	The SM customer wants to access history information of a product.
Preconditions	A QR label have been attached to the package. A QR reading application is running in the smartphone of the customer.
Postconditions	None
Related scenarios	Food quality monitoring.

	Food quality audit.
USE CASE Description	
ID	FSC_UC14
Name	Product audit
Actors	Consortium certifier organization, SM employee
Storyline	The SM employee access the FSC web application to report a quality issue related to one or more boxes. The consortium certifier organization tracks the history of the box(es) to identify the issue and also specify the business companies that had the responsibility of the product as it was transferred from the field to the supermarket. The certifier may request these companies to send additional (IoT) data about how they have handled the product. The certifier finally informs the SM employee about the audit results.
Trigger events	A customer has reported an issue in the history data of a product. The SM employee identifies the box(es) which was related to that product.
Preconditions	All data relate to the product history over the FSC are stored in an immutable way and they are accessible by the audit service.
Postconditions	None
Related scenarios	Food quality audit.

3.2.2 Requirements and software architecture

In this subsection, the system architecture of the FSC pilot is presented. Firstly, the domain and security requirements of the pilot are identified based on the expertise of the consortium members in the corresponding application domains (i.e. SYN and OPT in farming and warehouse environments), as well as the feedback collected from food supply experts of the SOFIE advisory board members (i.e. Pegasus company). Based on these requirements, the first version of the FSC system architecture is designed in line to the SOFIE framework architecture as this is defined in D2.4 – SOFIE Federation Architecture, 2nd version.

3.2.2.1 Pilot domain requirements

Table 5 summarizes the domain and end-user requirements of the FSC pilot.

Table 5. List of requirements in the food supply chain pilot

ID	Name	Description	Priority	Related use case
REQ_FSC0.1	FSC Web application	The services must be provided (to the actors) through the same web application.	MUST	ALL
REQ_FSC0.2	RBAC over provided services	The services must be accessible (by the actors) under a Role-based Access Control (RBAC) policy.	MUST	ALL
REQ_FSC0.3	Actors unique identifiers	Each actor must be identified in a unique way	MUST	ALL

REQ_FSC0.4	IoT environments unique identifiers	Each federated IoT environment must have a unique identifier in the system architecture.	MUST	ALL
REQ_FSC0.5	Authentication management	Authentication and access control logic must be applied to common storage resources.	MUST	FSC_UC3, FSC_UC4, FSC_UC8 FSC_UC10 FSC_UC13
REQ_FSC1.1	Timestamped crop registration	Registration of a crop must be timestamped.	MUST	FSC_UC1
REQ_FSC2.1	Farming (meta)data of boxes	The QR code that summarizes product history must include farm location, harvesting date, used fertilizers (dates), and the type of the product (from the perspective of the farming system),	MUST	FSC_UC2
REQ_FSC3.1	Record of handovers	Handovers must be recorded in an immutable way where all federated IoT environments must have access.	MUST	FSC_UC3 FSC_UC4 FSC_UC8
REQ_FSC3.2	Sealing of boxes	The boxes could be sealed upon the delivery to the transportation company (from the producers).	COULD	FSC_UC3
REQ_FSC4.1	Unsealing Box(es) at the WH	Upon delivery to the WH employee, boxes could be unsealed by the TR employee.	COULD	FSC_UC4
REQ_FSC5.1	Box unique identifier	Each box must have a unique RFID tag identifier.	MUST	FSC_UC5
REQ_FSC5.2	Boxes as Things	Boxes must be considered as things of the transportation IoT platform.	MUST	FSC_UC5
REQ_FSC5.3	Box registration	Box registration in the supply chain must define also the producer from whom it will be used.	MUST	FSC_UC5
REQ_FSC5.4	Timestamped box registration	Registration of a box must be timestamped.	MUST	FSC_UC5
REQ_FSC6.1	Transportation truck connectivity	Transportation trucks must have internet connection to communicate and exchange data with the transportation IoT platform.	MUST	FSC_UC6
REQ_FSC6.2	Use of transpor-tation truck	A TR employee (driver) must be able to use different transportation trucks on different occasions.	MUST	FSC_UC6

REQ_FSC7.1	Local storage of IoT data	Measurements from IoT devices are stored locally in the corresponding IoT platform.	MUST	FSC_UC7 FSC_UC9
REQ_FSC8.1	Unsealing Box(es) at the SM	Upon delivery to the SM employee, boxes could be unsealed by the TR employee.	COULD	FSC_UC8
REQ_FSC9.1	Tracking warehouse conditions	The temperature within each storage room of the WH must be continually monitored.	MUST	FSC_UC9
REQ_FSC9.2	Warehouse alarms	In the WH, a notification appears in the monitoring service of the Aberon IoT platform each time a predefined temperature range is violated.	MUST	FSC_UC9
REQ_FSC10.1	Packetizing products	The (unreleased) boxes in the WH must contain either raw or packetized products.	MUST	FSC_UC10
REQ_FSC11.1	QR code creation	QR codes must include data which is collected from the federated IoT environments, as well as provided by the actors through the FSC web application	MUST	FSC_UC11
REQ_FSC11.2	QR labels of packets	The same QR label must be attached to every packet containing grapes which were transferred into the same box.	MUST	FSC_UC11
REQ_FSC11.3	Vocabulary of QR labels	Labeling of products must be based on a common vocabulary for the food supply domain that maximises reuse of data and acceptance by the customers.	MUST	FSC_UC11
REQ_FSC11.4	Self-contained QR codes	The QR codes must be self-contained, so internet connection is not needed to read their content.	MUST	FSC_UC11
REQ_FSC11.5	Information recorded in QR codes	The QR codes must contain product information relate to all the segments of the chain.	MUST	FSC_UC11
REQ_FSC12.1	Boxes reuse	Boxes must be able to be re-used in the future (to carry other products) after they have been released of the current transfer.	MUST	FSC_UC12
REQ_FSC13.1	QR code reading	QR labels must be accessible by everyone by using a smartphone device.	MUST	FSC_UC13
REQ_FSC14.1	Traceability of historic data	In the case of and audit, requested organizations must be	MUST	FSC_UC14

		able to provide proof of their claims about the historic data of assets which are stored locally.		
REQ_FSC14.2	Timestamps of handovers	Transfer of responsibility over boxes (assets) must be timestamped.	MUST	FSC_UC3, FSC_UC4, FSC_UC8, FSC_UC14
REQ_FSC14.3	Confirmation of transactions	A transaction must be confirmed by both transacting parties.	MUST	FSC_UC3, FSC_UC4, FSC_UC8, FSC_UC14
REQ_FSC14.4	Retrieve of past transactions	Both parties of a transaction must be able to access the details of the transaction at any time.	MUST	FSC_UC3, FSC_UC4, FSC_UC8, FSC_UC14

3.2.2.2 Pilot security and privacy specifications

Security analysis of the SOFIE federation framework has identified three important security and privacy challenges which are critical for all pilots of the project. Table 6 summarizes how the FSC pilot aims to address these challenges.

Table 6. Security and privacy specifications in the food supply chain pilot

Transactions must be immutable and verifiable
One of the key features of DLTs is that they provide immutability of the recorded data. The FSC pilot will make use of a private Ethereum, named as consortium ledger, to store all performed transactions, as well as other critical data and metadata, which is necessary to achieve traceability of products over the supply chain. Each recorded transaction will be linked with various metadata (as those result after transforming data which is collected from IoT segments and actors), so being able for an interested (and authorized) entity not only to get full details of the transaction at any later time but also correlate it with other relative information. Apart from the consortium ledger, the pilot will also make use of other DLTs (through SOFIE interledger operations) to provide strong data verification means to entities which are not participating in the consortium ledger but they have a great interest in data genuineness, e.g. the supermarket companies. These DLTs will be used to create and permanently store hashes or anchors of the data which is recorded in the consortium ledger. Being publicly available, these records can be used by any entity to verify claims of the members of the consortium ledger in the cases of disputes as products are distributed to the selling points.
Support for transactions where only authorised entities can participate
The business segments (IoT environments) who record data in the consortium ledger will register themselves by using an Ethereum client such as Geth to create accounts by using built-in encryption policies. The communication with the client will be done with JSON protocol and the only thing these entities should do is to pass “passphrases” when creating or signing with their account. Once the created keystore files received by them and corresponding accounts on the created wallets are unlocked, these entities will use their private keys to sign any transaction and data transfer initiated by them. Upon receiving request to record data into the consortium ledger, the smart contract which will be deployed therein will implement a number of checks to verify both the identities of the entities requesting transactions and also data integrity. Overall, only registered entities in the SWS will be able to push and recall data in/from the consortium ledger. Apart from this, the FSC web application will enable a role-based access control (e.g. by using Keycloak server) to both authenticate and assign permissions to actors about how they can access the exposed APIs (thus how they can act

over assets) based on their role and their organization.
Privacy issues and business secrets must be considered carefully when deciding what data (including authentication/authorization information, logs etc.) is collected, stored or exchanged between parties.
Depending on the type of the data object that is recorded into the consortium ledger, a part of the corresponding datum may be encrypted. For example, in the case of a transaction, both the IDs of the employees and the IDs of the corresponding platforms (if applicable) who transact each other will be encrypted (anonymity), thus only the corresponding organizations being capable to verify the transaction (and who persons from their own stuff got involved). The same holds for other important aspects of the transaction which should not be revealed to the other members of the BP, e.g. total weight of the product that was transferred between the two parties.

3.2.2.3 Pilot system architecture

Based on the scenarios, use cases and application/domain requirements stated above, as well as the specifications of the provenance chain BP which were defined in D2.3, the FSC system architecture is shown in Figure 4.

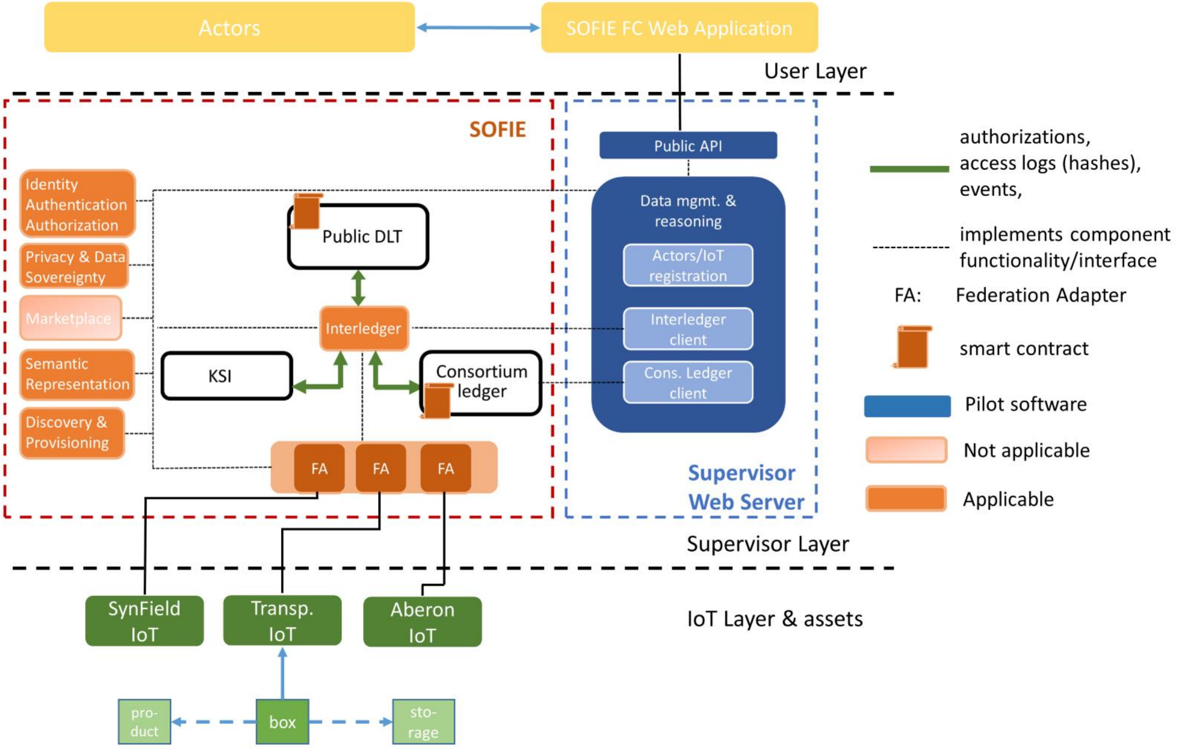


Figure 4. Overview of system architecture for the food supply chain pilot

As shown in Figure 4, the architecture defines three operational layers:

i) **the user layer** which provides an interface to actors to interact with the system, activate services and feed (meta) data.

ii) **the supervisor layer** which controls the execution of the services which are provided by the SOFIE framework components and establishes a data management layer to deliver the two main services of the BP, i.e. QR code creation and product quality audit. In summary, the supervisor layer is responsible to: i) manage, transform (create data mappings) and store in the deployed network of DLTs all data and metadata which are used to guarantee reliable traceability of products (income into its environment either from the federated IoT environments or the web application), ii) supervise the state of the registered/active assets (i.e. boxes) and the proper execution of transactions over them (e.g. handovers), iii) expose a public API to the actors to enable their interaction with the system based on their roles (performing also authentication and authorization), and iv) support the execution of services which are provided by the SOFIE federation framework components by preparing any necessary data mapping between them.

iii) **the IoT layer** which includes the IoT platforms which are federated in the provenance food BP to transfer data and metadata to the supervisor layer. As already mentioned, three IoT platforms are federated in the pilot setup, which corresponds to the business segments of the producer (SynField platform), the transportation company (the transportation IoT platform), and the warehouse (the Aberon IoT platform).

Data organization and management: The pilot architecture defines a hierarchical data organization and management over the considered three layers. At the bottom layer, each IoT environment uses its own data management and storage infrastructure (which can be either a database or a DLT) to handle and store measurements collected by the various deployed devices, as well as any other dataset is critical for its business interests. A part of this data, as well as relative metadata (which are created either by the IoT environments or the actors through the web application) are transferred to the supervisor layer, where they are transformed and combined according to the system data model to create datums (data objects) which are stored in the considered DLT network. The system architecture introduces a network of three DLTs, i.e. the consortium ledger, the KSI[9] and a public ledger, to securely store data and guarantee their integrity and immutability (the role of each DLT is explained in §3.2.2.3.2). This data is used to track product history, thus it is used in both the QR codes which are attached to the product packages but also to identify reasons of product quality deterioration (when such are reported) and resolve disputes between members of the BP. Note that the information that is recorded in pilot's DLTs network will ultimately be as valuable as the data that goes into the supervisor layer. If the data feed in the supervisor layer is incomplete or inaccurate, then the records held within the DLTs network will be similarly incomplete or inaccurate. The complete data model which is defined by the system architecture will be provided in "D5.3: End-to-end Platform validation". Table 7 summarizes the various categories of data which are considered in the two levels of the system data management layer.

Table 7. Datasets considered in the food supply chain pilot

	Farming	Transportation	Warehouse
Data collected /stored in the local level	- Climate values (temperature, humidity, wind speed/orientation etc.), - Ripening tracks	- Temperature within the truck - Presence of boxes in the trolley of the truck - Timestamps of handovers based on RFID	- Location/room where each box is placed, - Temperature of storage rooms -
(Meta)Data stored in DLTs	- Farm location, - Crop establishment date, - Harvesting date, - Crop type, - Pesticides products, - Farm audit specifications, - Ripening level at	- Duration of box(es) transportation, - High/Low temperature during transportation of box(es), - Box(es) delivery timestamps	- Duration of box(es) storage, - Box(es) delivery timestamps, - High/Low temperature during storage of box(es)

[9] Described in detail in SOFIE D2.1-chapter 4.9. Available from: https://media.voog.com/0000/0042/0957/files/SOFIE_D2.1-State_of_the_Art_Report.pdf

	harvesting day, - Box(es) delivery timestamps		

In the following, the three layers of the pilot architecture shown in Figure 4. Overview of system architecture for the food supply chain pilot are presented in more detail.

3.2.2.3.1 User layer

In the user layer, a web application (supported by supervisor web server) enables the various actors to interact with the system and access the services which are summarized in Table 8.

Table 8. Services provided to the actors through FSC web application[10]

Actor	Service Type
Producer	• Push (meta)data for crop and also the product which is deposited in boxes into the supervisor layer • Bind product with box(es)
Transportation employee	• Register boxes to a specific field • Register trucks to a specific transportation • Transfer responsibility of assets (boxes) [from the producer to the transportation] • Transfer responsibility of assets (boxes) [from the warehouse to the transportation]
Warehouse employee	• Define storage location for boxes within the area of the WH • Transfer responsibility of assets (boxes) [from the transportation to the warehouse]
Supermarket employee	• Transfer responsibility of assets (boxes) [from the transportation to the supermarket] • Create QR codes • Release boxes • Initiate traceability audit

3.2.2.3.2 Supervisor layer

The supervisor layer makes use of several components of the SOFIE architecture, as well as the Supervisor Web Server. The functionality of each part is as follows:

Supervisor Web Server (SWS): The SWS establishes a data management layer which is responsible to:

- Prepare and manage data objects (datums) which are stored in the DLTs by transforming and adapting data and metadata which are inserted into the system through the federated IoT platforms and the SOFIE FSC web application. The information flow which is implemented by the SWS is shown in Figure 5.
- Supervise and control the status of each registered asset (i.e. boxes) from the time it is created up to the time it is released. The status of each asset is controlled by a smart contract which is executed in the consortium ledger, the state of which is updated based on the actors' activities, e.g. when a handover is executed between two actors.
- Authenticate IoT platforms and actors which request access to the DLTs as well as data objects/streams which are sent from entities to the consortium ledger. In particular, a RBAC mechanism is implemented as part of the backend services of the SWS to restrict access of actors based on their role in the supply chain. SWS also collaborates with IAA component (of the SOFIE architecture) to register/verify IoT

[10] Note that QR labels are self-contained in the sense that they encode all the product history information from the field to the selling point. Thus, QR reading by the supermarket customer is not considered as a service of the SOFIE FSC web application, although it is a service which is provided by the BP to this type of actor.

environments in the consortium ledger and authorize the storage of each piece of information therein.

- Implement mechanisms to push and retrieve information from the consortium ledger. As part of the consortium ledger client, an event monitoring and explore mechanism is implemented to monitor and search events of interest (i.e. events related to the status of assets) which are recorded in the blocks of the consortium ledger.
- Support and supervise interledger operations based on the applied interledger networking policy and model. A detailed description of the implemented interledger patterns over the considered network of ledgers will be included in “D5.3: End-to-end Platform validation”.

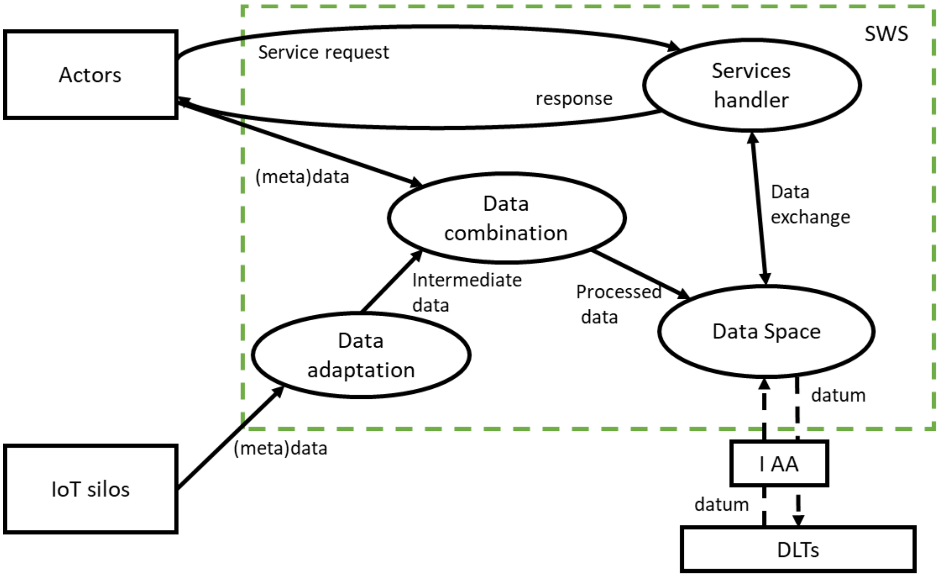


Figure 5. Information flow in food supply chain pilot

- Expose a public API to the actors (accessible by the SOFIE web application) and serve their requests by coordinating the appropriate resources and application data.

SOFIE architecture: The FSC pilot architecture makes use of the following core components of the SOFIE architecture:

- **SOFIE Federation adapter:** This component is used to adapt data representation of the corresponding IoT environment to the Things Description (TD) model which is defined by SOFIE architecture (W3C WoT TD [11]). It is also used to execute supplementary data transformation services on behalf of the local IoT environments during authentication and interledger operations (so their federation be completely transparent).
- **IAA**: This component is used to register IoT platforms in the deployed DLTs. For instance, upon registration in the consortium ledger, an IoT platform requests the creation of a keystore file from the IAA component. Once the Ethereum wallet (keystore file address) is stored in the consortium ledger and the keystore file is received by the IoT platform (in particular by the corresponding federation adapter), the private key is used to digitally sign any data objects/stream which is sent to the SWS.
- **Interledger**: This component is used to enable transactions between the various DLTs which are used by the business logic of the system. In particular, the deployed ledger network includes the following ledgers: i) the **consortium ledger** where all critical data and metadata are stored which are used to track products’ history from the field to the supermarket. The storage of this (meta)data is handled by a smart contract that

[11] https://www.w3.org/TR/wot-thing-description/

pushes the appropriate types of events (i.e. data objects) to the consortium ledger based on actors' activity and their interaction to the system. ii) The **KSI ledger** which is used to create timestamped signatures/hashes of the data which is stored in the consortium ledger, either periodically or event based. These signatures can be used as anchors which verify the genuineness of the recorded data in the consortium ledger to external parties or members of the BP that has no access there. iii) The **public ledger** which is used to store the hashes which are created by the KSI and make this information public, so as to protect the BP itself from any potential manipulation by the members of the consortium ledger.

- **Privacy and data sovereignty:** This component defines the most appropriate mechanisms with respect to the pilot domain requirements to apply SOFIE privacy and data sovereignty policies in the data management processes of the SWS (e.g. protection of sensitive information in DLTs transactions, implementation of access control mechanism to access services etc.)
- **Discovery and provisioning:** This component applies the SOFIE service discovery and provisioning mechanism to the pilot system architecture. The integration details will be defined in "D5.3: End-to-end Platform validation".
- **Semantic representation:** This component defines the data models that should be implemented by the various software components of the pilot architecture (e.g. the way the data must be structured) as well as how data should be translated between them to enable interoperability and data exchange.

3.2.2.3.3 IoT layer

The IoT reference architecture for the three IoT platforms integrated in the FSC pilot is shown in Figure 6.

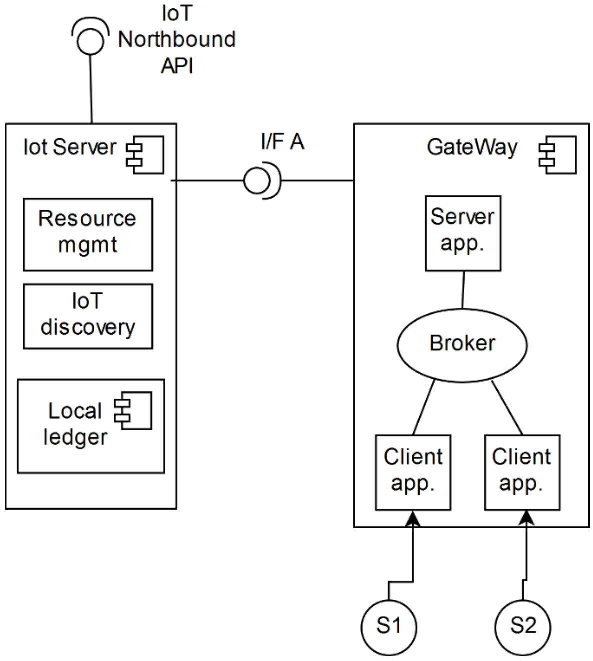


Figure 6. The reference architecture for federated IoT platforms

As shown in Figure 6, the considered IoT platform architecture model defines the following two major components:

- **IoT server:** The IoT server is responsible to aggregate and process measurements which are collected from the sensing devices, control resources and devices, execute further functionality, such as applying rules, managing events and filtering data, provide data to connected applications and guarantee the proper execution of connectivity protocols and security mechanisms. Typically, it integrates a local ledger/DB to store measurements and metadata for the IoT environment. Note that the functionality of the IoT server of each specific IoT stack which is used in the pilot may be extended to other services also. Typically, the services provided by the IoT server can be accessed using APIs (Northbound API).

- **IoT gateway:** The role of each IoT gateway is to assist data aggregation at the IoT cloud server side by hosting messaging endpoints that translate and transfer measurements of the connected devices. Such endpoints abstract the communication protocols used by the devices and provide functionalities to facilitate the transfer of data to/from the IoT server. Typically, an interface exists between each IoT gateway and the IoT server (I/F A) that makes the former accessible from any other deployed endpoint or internal service of the IoT platform.

Table 9 summarizes the devices of the IoT platforms which are considered in the pilot setup.

Table 9. Devices of the IoT platforms integrated in the FSC pilot.

IoT Platform	SynField IoT	Transportation IoT	Aberon IoT
Devices	• Weather station (temperature, humidity, anemometer, rain collector) • Soil moisture • Soil electrical conductivity • Leaf wetness • Light sensor • Solenoid valve	• RFID readers and tags • Temperature	• Temperature

3.2.2.3.4 Sequence of interaction for pilot use cases

The two sequence diagrams in Figure 7 and Figure 8 show the sequence of interaction between various objects of the architecture for the two main services in the FSC pilot, i.e. the QR code creation and the product audit processes.

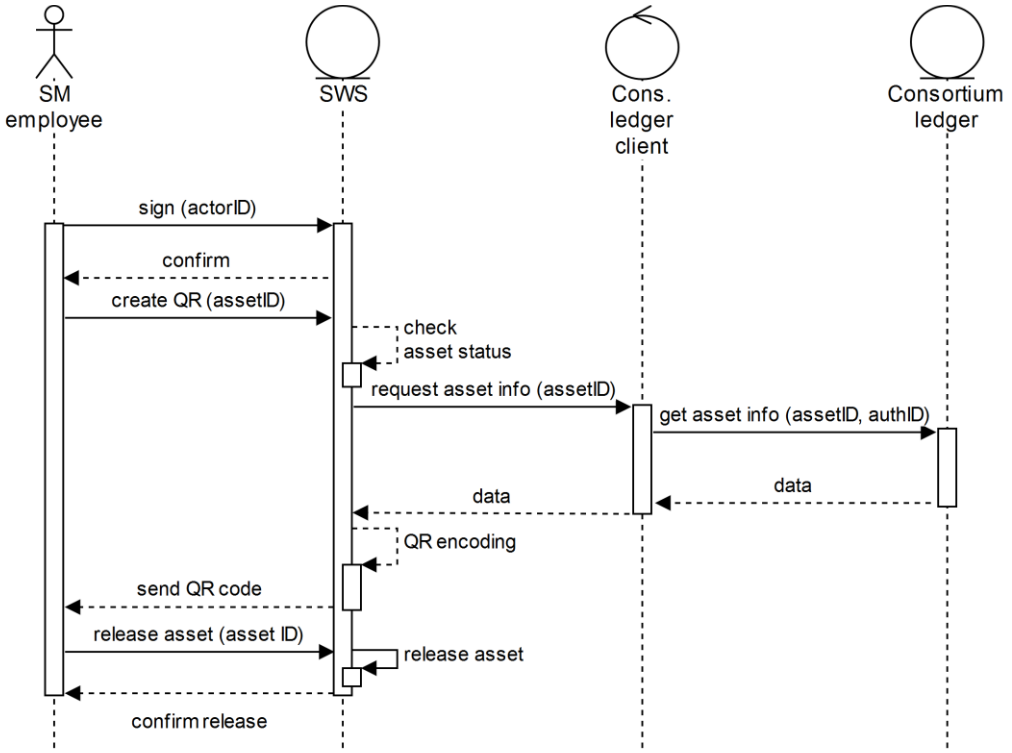


Figure 7. Sequence diagram for QR code creation

In Figure 7, the steps the SM employee follows to create the QR code for a specific asset are shown. First, the SM employee should access the FSC web application and activate the QR creation service, providing also the unique ID of the box that contains the product. After the SWS checking the state of the corresponding asset (e.g. that ownership belongs indeed to the SM employee responsibility), it activates the consortium ledger client to recall from the consortium ledger all data relate to the specific asset. The client ledger authenticates the request to the consortium ledger, recalls all relative data and transfer it to the SWS. After the QR code is prepared, it is transferred back to the SM employee for printing, while the box is released for possible future use.

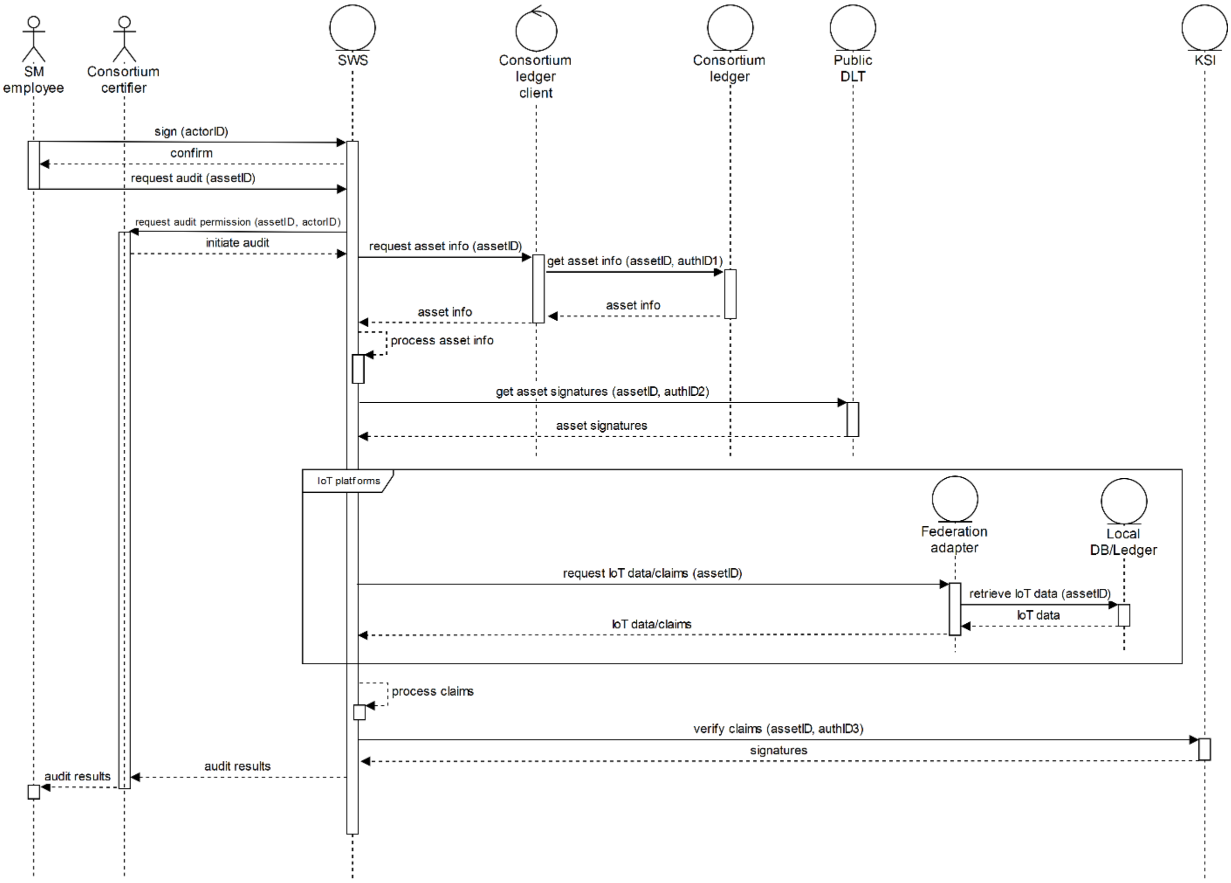


Figure 8. Sequence diagram for product audit

Figure 8 shows how the product audit process is performed. As shown, the audit process is initiated by the SM employee as described in scenario 2. Once informed, the consortium certifier retrieves all relative information from the consortium ledger, as well as the relative signatures which have been stored in the public DLT. In the next step, the information where these signatures are referred to is requested by the federated IoT environments and its verification is performed by using the services of the KSI. The final results that identify the reasons of this product deterioration and the organization that is responsible for that are sent back to the consortium certifier organization.

3.3 Initial SOFIE validation

Figure 9 illustrates how the SOFIE advances are validated in the FSC pilot. Leveraging on pilot specifications and system architecture, the aim is to implement and elaborate a prototype that determines and deploys the technology which is necessary to realize domain requirements and verify that the system under design satisfies the actors and business aspects it is designed for. In this subsection, we describe the Proof of Concept (PoC) implementation of the prototype.

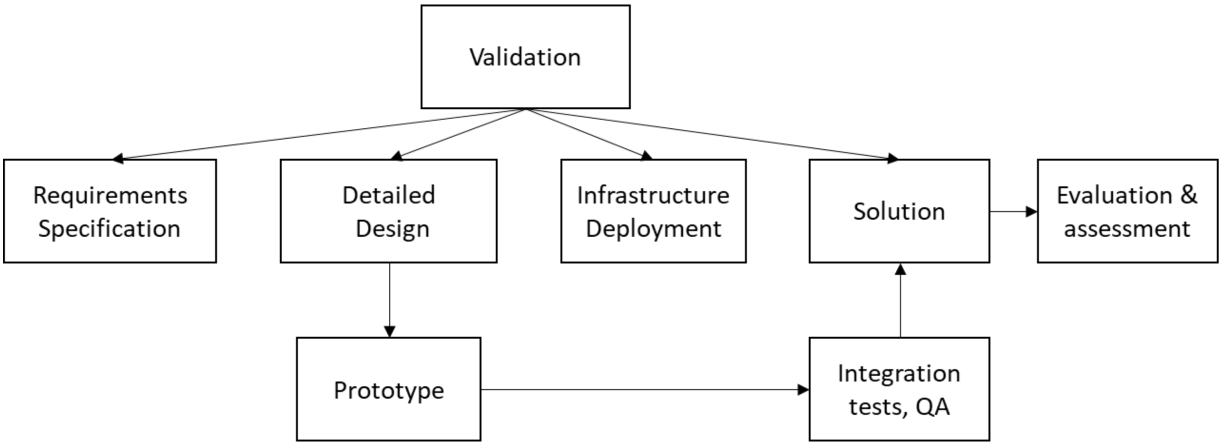
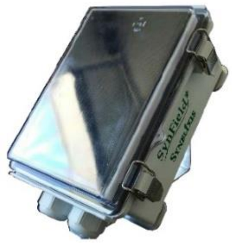
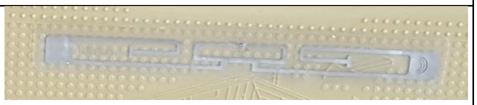


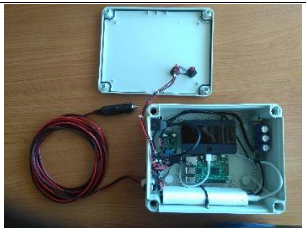
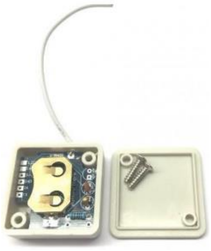
Figure 9. SOFIE validation process in the food supply chain pilot

3.3.1 Installed infrastructure

Table 10 summarizes the infrastructure used in the PoC prototype. The used equipment relates mostly with the sensing devices and the hardware components which are used by the two IoT platforms which are integrated in the PoC prototype, i.e. the SynField platform and the Kaa-based transportation IoT platform. The first is a production environment (uses operational data) that has been deployed in a vineyard field in the area of Kiato, Greece. The second is a lab environment that integrates all the devices which will be finally deployed on site for the transportation IoT platform. Note that both the IoT platforms, as well as the backend services of SWS and the used DLTs are deployed in Synelixis cloud.

Table 10. Equipment used in PoC prototype

Equipment	Role in pilot	View
SynField Head Node (HN)	It is installed in the field and acts as a gateway to collect and transfer farming data into the cloud SynField IoT platform. It is provided by Synelixis Solutions.	
Weather Station	It is connected to the SynField HN to measure temperature, humidity and other climatic factors (wind speed/direction, rain collector etc.). It is provided by Davis (Vantage Pro 2) and it includes SynField adaptation kit.	
RFID tags (boxes IDs)	An RFID tag is attached over each box.	

IoT transportation GW	A raspberry which will be installed inside the truck cabin is used as a GW for the transportation IoT platform. A 4G router is used to guarantee that the GW is always connected to the internet.	
Long range RFID (CAEN ion – R4301P)	Long range RFID which is installed in the truck trolley and it is connected to the IoT transportation GW. It features embedded HW architecture (x86) and standard operating system (Linux) to enable the development of custom software that detects every onboarded box. It is provided by CAENRFID.	
Temperature sensor (ds18b20)	Analogue temperature sensor which is deployed inside the transportation truck trolley. It is connected with the truck GW.	

The physical placement of equipment used by the SynField IoT platform is shown in Figure 10.

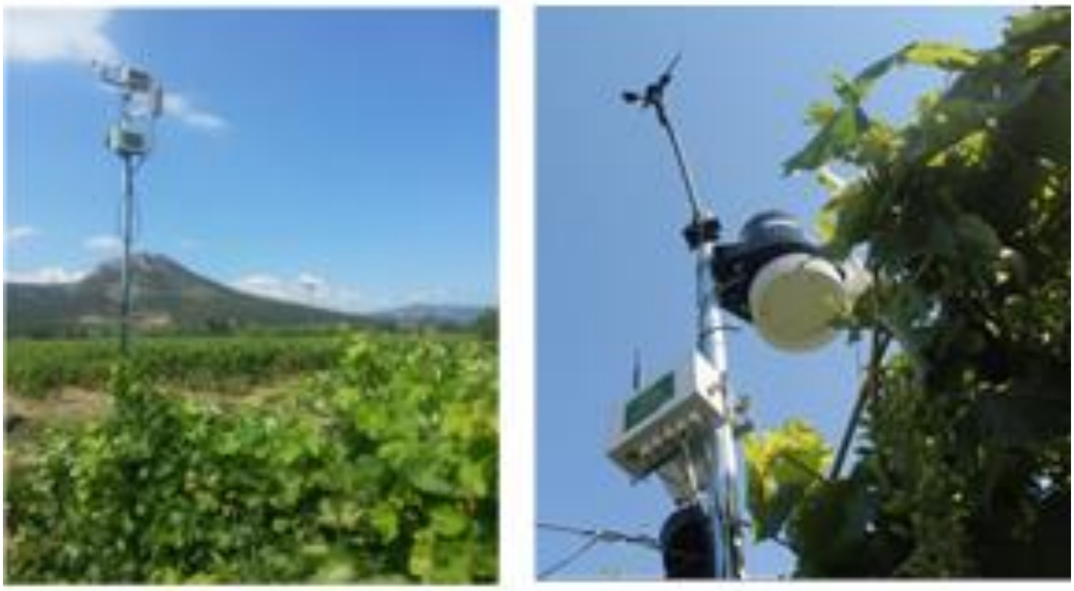


Figure 10. SynField HN and sensors deployed in a vineyard field in Soulinari (Kiato region)

3.3.2 Proof of concept prototype

The PoC prototype focuses on data management at the supervisor level and validates how data from different IoT environments are combined and recorded in the consortium ledger to ensure reliable traceability of assets, data integrity and privacy of entities' information.

A first release of the Supervisor Web Server (SWS) has been implemented to manage and adapt data which is received from different IoT platforms and actors. The consortium ledger (i.e. an Ethereum blockchain) has been deployed and the first release of the smart contract which is responsible to record events, i.e. (meta)data which correspond to actors' activities, has been implemented. Once an asset (box) is registered, the smart contract establishes a unique session bound to it and controls its state in relation to the actors' activities. Thus, in a later step (e.g. during QR creation), the SWS needs just to track this session to recall all relative information to the asset. The consortium ledger client of the SWS exposes an initial list of REST endpoints enabling SWS to interact with the smart contract and push into the consortium ledger the appropriate data objects per event based on the input which is received by the IoT environments and the actors. The SWS has also been integrated with a Keycloak authorization and access management server to enable actors' registration in the supervisor data management layer and establish role-based accessibility to the provided services (i.e. endpoints of the smart contract).

The SynField and the transportation IoT platforms (and corresponding devices) have been integrated in the PoC prototype to model how data from the farming and the transportation business segments is adapted before entering the pilot system architecture. Both platforms have been registered in the consortium ledger by using the client geth of the Ethereum[12]. The keystore files and the related private keys which are created upon registration are managed by the corresponding federation adapters and they are used to sign any data transaction from the local IoT environments to the supervisor layer. Note that in the initial PoC prototype, no UI exists for the FSC web application, so all data and transactions are injected into the SWS by using command line.

The objective of the PoC validation is to explore how feasible are certain advances of the SOFIE innovation and technology in the food supply domain and ensure that corresponding functionality can be integrated in the processing chain through development. In this scope, the core functionality which is validated in the initial proof of concept is summarized in Table 11.

Table 11. Core functionality which is integrated in the PoC prototype

ID	Title	Description
MSI	Metering Services and Interoperability	1) Development of northbound APIs for the integrated IoT platforms. 2) Adaptation of data which is sent by the IoT platforms according to the Things Description (TD) model defined by the SOFIE representation model. This functionality is implemented by the federation adapters. 3) Secure communication and data/metadata exchange between the integrated IoT segments and the SWS. (meta)data transactions from the IoT segments are signed by using the private keys they have received during their registration in the consortium ledger.
SDAM	Services, (meta)Data and Assets Management	1) Backend data management services in the SWS (actors' authentication and access control, functionality and interface of consortium ledger client, etc.). 2) Data transformation and mapping in the SWS based on the designed information flow (preparation of datums) 3) Smart contract-based session management in the consortium ledger

[12] https://geth.ethereum.org/downloads/

		(enables traceability).
DPSA	Data Protection and Security Aspects	1) Protection of the data residing within the consortium ledger (usage of SOFIE sovereignty policies to support transactions from only authorized entities and protect integrity of data). 2) Privacy of sensitive information relate to the IoT platforms, performed transactions and actors' identities. 3) Role based authentication and access control in SWS services.

3.3.3 Validation results

In this subsection, the functionality of the PoC prototype is demonstrated by focusing on the following two use cases of the pilot (see Table 4 for the complete list of pilot use cases) to show how (meta)data is managed by the deployed smart contract in the consortium ledger and how datums are prepared by the SWS before events are recorded into the ledger:

- FSCP_UC2 – Box product: here how an actor (i.e. the producer) interacts with the SWS services based on its role and how data which is streamed into the pilot platform from different sources are combined before they are recorded to the consortium ledger are validated.
- FSCP_UC3 – Hand over product: PR-TR (i.e. handover between the producer and the TR employee): here how asset's handovers and related (meta)data are recorded into the consortium ledger and how the corresponding IoT environments and actors are verified according to SOFIE data privacy and sovereignty policies before they allowed to push data in the ledger are validated.

In Figure 11, the endpoints which are exposed by the consortium ledger client of the SWS (first release) are shown. These endpoints are used to register entities in the deployed smart contract, as well as record and retrieve data objects (datums) from the consortium ledger. The implemented endpoints are grouped in the following five categories:

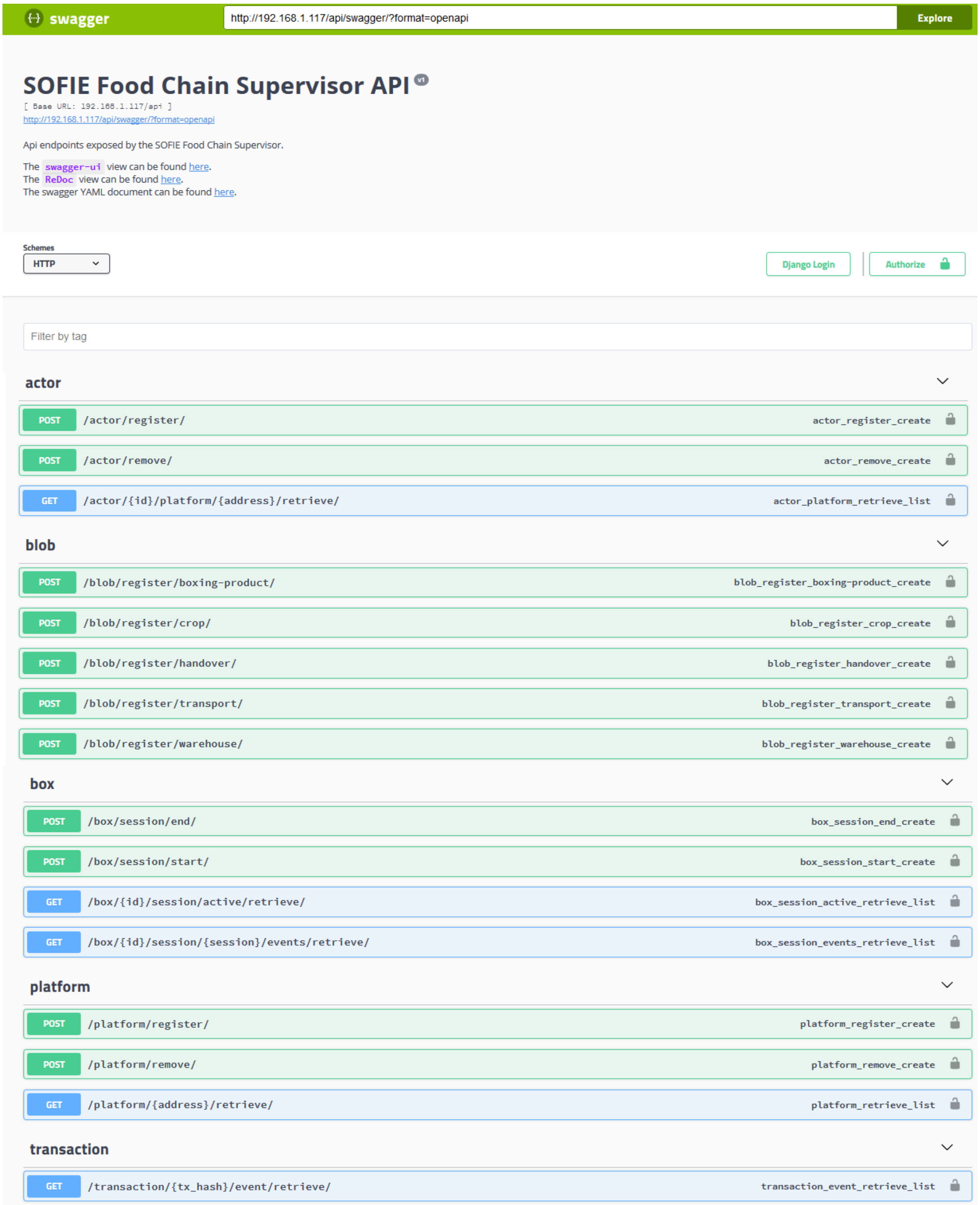
- Actor: these endpoints are used to register actors in the smart contract and also overview which of them have already been registered therein.
- Blob: these endpoints are used to push data objects relate to the following use cases: register crop, box product, perform handovers among actors, transport boxes from one site to another, packetize boxes and store them in the warehouse.
- Box: these endpoints are used to register sessions and release boxes.
- Platform: these endpoints are used to register IoT platforms in the smart contract. The IoT platforms must have already been registered in the consortium ledger.
- Transaction: this endpoint is used to retrieve record from the consortium ledger, e.g. upon QR creation time where all information relate to a specific box must be retrieved.

Every transaction which is submitted to the consortium ledger is linked with an IoT platform, either this is the entity that directly sends data to the smart contract or it is the IoT environment where the actor who sends the data is also registered. To ensure that only authorized entities can record data in the consortium ledger, the following three conditions must hold when a request for transaction is sent to the consortium ledger client:

1) Every entity (IoT platform or actor) who wants to use any of the endpoints of the categories Blob, Box and Transaction must first register itself in the smart contract.
2) The message payload of the transaction is signed by using the private key of the corresponding IoT environment and the signature is sent to the client.
3) The message payload of the transaction is hashed using SA3 and the hash is also sent to the client.

Leveraging on the above conditions, the deployed smart contract implements the following authentication and access control checks:

- The message payload is hashed also by the smart contract and the result is compared to the hash which is sent by the entity that requests the transaction.
- The signature and the hash of the message payload are used as input to the ecrecover() function of Solidity to verify the signature of a message and get the account address (public key) of the corresponding IoT environment.
- It is verified that the identified account address is already registered in the smart contract.
- It is verified that the IoT platform which is mentioned in the message payload is the one that has created the signature.



SOFIE Food Chain Supervisor API v1

[Base URL: 192.168.1.117/api]

http://192.168.1.117/api/swagger/?format=openapi

Api endpoints exposed by the SOFIE Food Chain Supervisor.

The swagger-ui view can be found here.
The ReDoc view can be found here.
The swagger YAML document can be found here.

Schemes: HTTP

actor

POST /actor/register/ actor_register_create
POST /actor/remove/ actor_remove_create
GET /actor/{id}/platform/{address}/retrieve/ actor_platform_retrieve_list

blob

POST /blob/register/boxing-product/ blob_register_boxing-product_create
POST /blob/register/crop/ blob_register_crop_create
POST /blob/register/handover/ blob_register_handover_create
POST /blob/register/transport/ blob_register_transport_create
POST /blob/register/warehouse/ blob_register_warehouse_create

box

POST /box/session/end/ box_session_end_create
POST /box/session/start/ box_session_start_create
GET /box/{id}/session/active/retrieve/ box_session_active_retrieve_list
GET /box/{id}/session/{session}/events/retrieve/ box_session_events_retrieve_list

platform

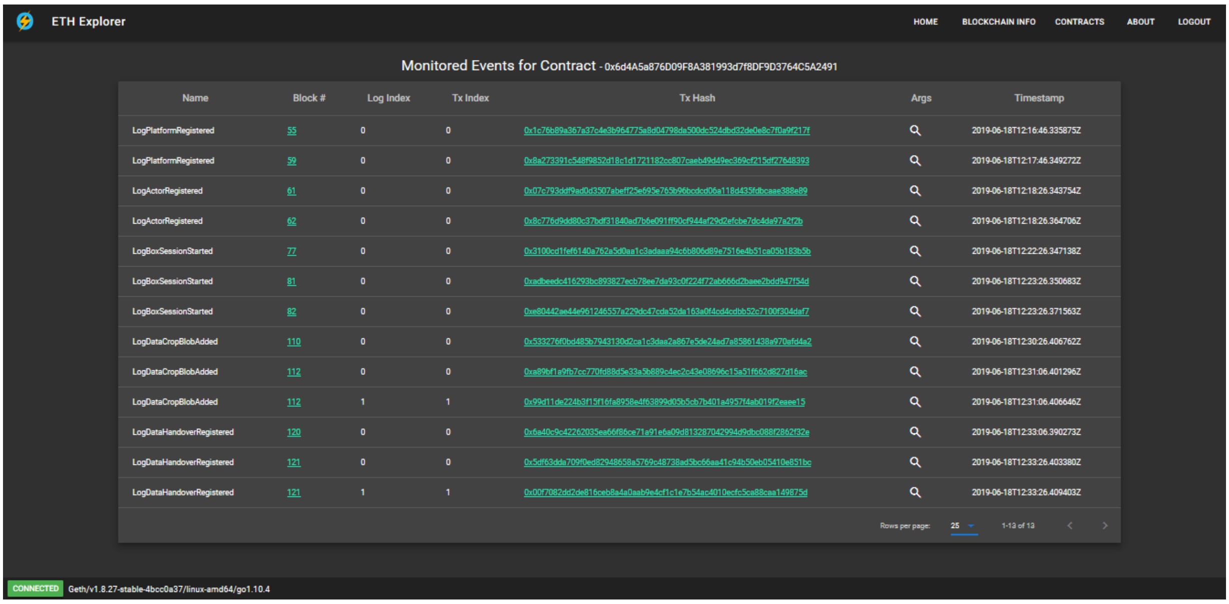
POST /platform/register/ platform_register_create
POST /platform/remove/ platform_remove_create
GET /platform/{address}/retrieve/ platform_retrieve_list

transaction

GET /transaction/{tx_hash}/event/retrieve/ transaction_event_retrieve_list

Figure 11. The API endpoints which are exposed by the consortium ledger client

The sequence of events which are recorded in the consortium ledger during the execution of the two considered use cases is shown in Figure 12. The first two events refer to the registration of the two IoT platforms in the smart contract, while the following two events refer to the registration of two actors from the corresponding business segments. Note that each engaged entity (either IoT platform or actor) has to be registered just once before being able to push data into the consortium ledger. The events that follow refer to the registration of three boxes which are used to carry products from the field to the warehouse. Then the two last events record datums relate to the “box product” and the handover between the producer and the transporter.



Monitored Events for Contract - 0x6d4A5a876D09F8A381993d7f8DF9D3764C5A2491

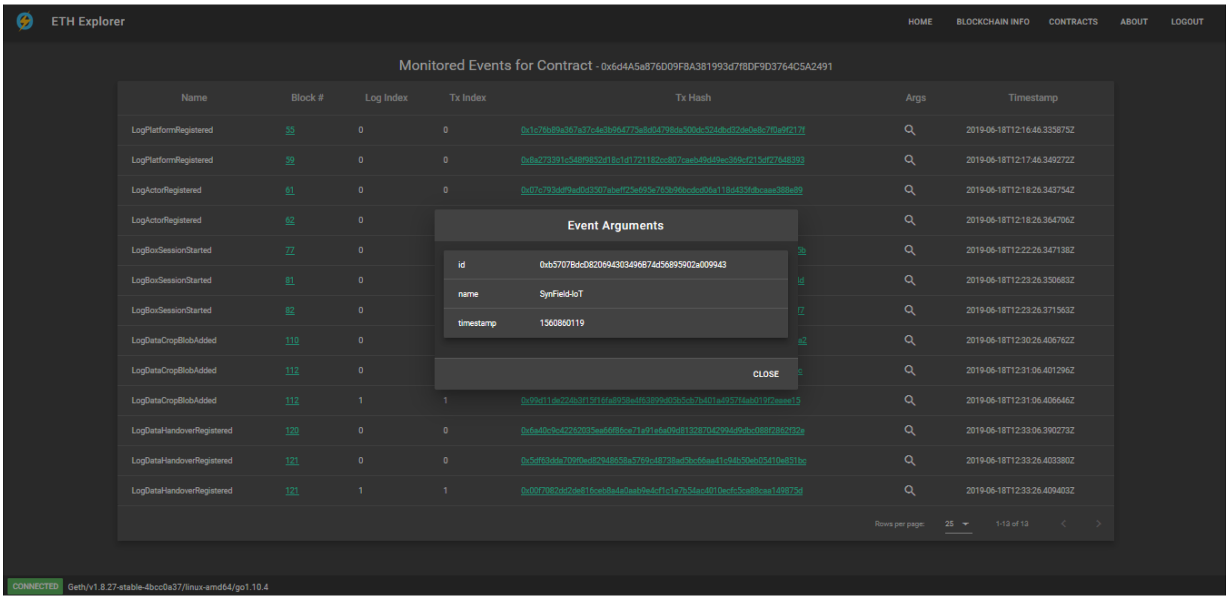
Name	Block #	Log Index	Tx Index	Tx Hash	Args	Timestamp
LogPlatformRegistered	55	0	0	0x1c76b89a367a37c4e3b964775a8d04798da500dc524dbd32de0e8c7f0a9f217f	🔍	2019-06-18T12:16:46.335875Z
LogPlatformRegistered	59	0	0	0x8a273391c548f9852d18c1d1721182cc807caeb49d49ec369cf215df27648393	🔍	2019-06-18T12:17:46.349272Z
LogActorRegistered	61	0	0	0x07c793ddf9ad0d3507abeff25e695e765b96bcdcd06a118d435fdbcaae388e89	🔍	2019-06-18T12:18:26.343754Z
LogActorRegistered	62	0	0	0x8c776d9dd80c37bdf31840ad7b6e091ff90cf944af29d2efcbe7dc4da97a2f2b	🔍	2019-06-18T12:18:26.364706Z
LogBoxSessionStarted	77	0	0	0x3100cd1fef6140a762a5d0aa1c3adaaa94c6b806d89e7516e4b51ca05b183b5b	🔍	2019-06-18T12:22:26.347138Z
LogBoxSessionStarted	81	0	0	0xadbeedc416293bc893827ecb78ee7da93c0f224f72ab666d2baee2bdd947f54d	🔍	2019-06-18T12:23:26.350683Z
LogBoxSessionStarted	82	0	0	0xe80442ae44e961246557a229dc47cda52da163a0f4cd4cdbb52c7100f304daf7	🔍	2019-06-18T12:23:26.371563Z
LogDataCropBlobAdded	110	0	0	0x533276f0bd485b7943130d2ca1c3daa2a867e5de24ad7a85861438a970afd4a2	🔍	2019-06-18T12:30:26.406762Z
LogDataCropBlobAdded	112	0	0	0xa89bf1a9fb7cc770fd88d5e33a5b889c4ec2c43e08696c15a51f662d827d16ac	🔍	2019-06-18T12:31:06.401296Z
LogDataCropBlobAdded	112	1	1	0x99d11de224b3f15f16fa8958e4f63899d05b5cb7b401a4957f4ab019f2eaee15	🔍	2019-06-18T12:31:06.406646Z
LogDataHandoverRegistered	120	0	0	0x6a40c9c42262035ea66f86ce71a91e6a09d813287042994d9dbc088f2862f32e	🔍	2019-06-18T12:33:06.390273Z
LogDataHandoverRegistered	121	0	0	0x5df63dda709f0ed82948658a5769c48738ad5bc66aa41c94b50eb05410e851bc	🔍	2019-06-18T12:33:26.403380Z
LogDataHandoverRegistered	121	1	1	0x00f7082dd2de816ceb8a4a0aab9e4cf1c1e7b54ac4010ecfc5ca88caa149875d	🔍	2019-06-18T12:33:26.409403Z

Rows per page: 25 1-13 of 13

CONNECTED Geth/v1.8.27-stable-4bcc0a37/linux-amd64/go1.10.4

Figure 12. Events which are recorded in the consortium ledger during demonstration of FSC_UC2 and FSC_UC3.

The arguments of the event which is created upon registration of an IoT platform is shown in Figure 13. The account address of the IoT platform, its name and the timestamp are included as arguments.



Monitored Events for Contract - 0x6d4A5a876D09F8A381993d7f8DF9D3764C5A2491

Event Arguments

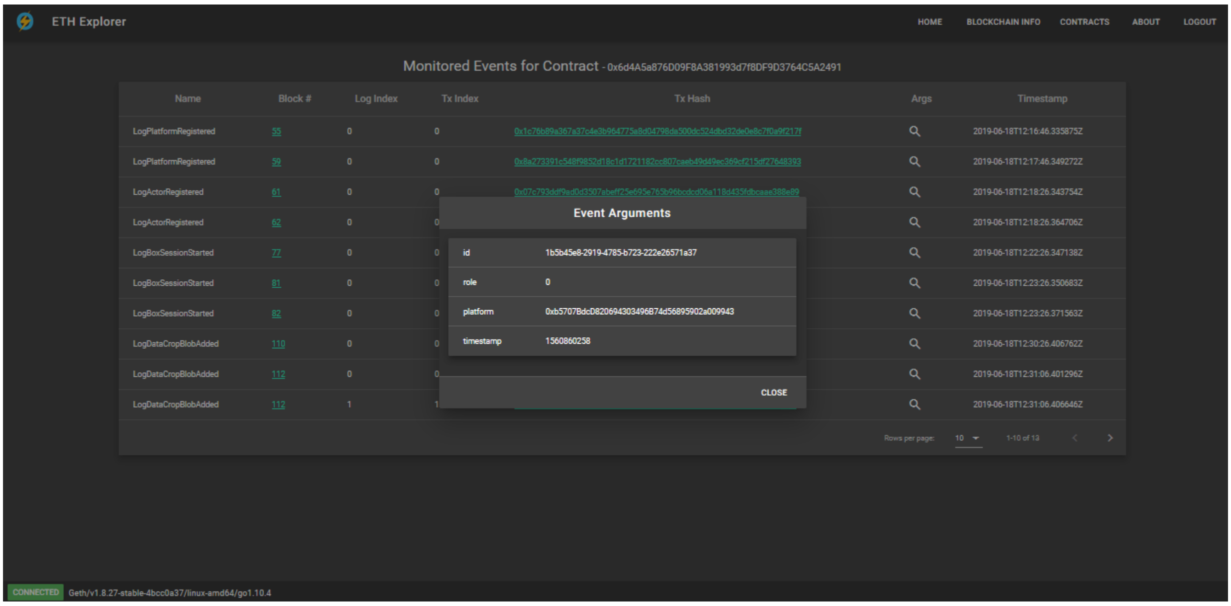
id	0xb5707BdcD820694303496B74d56895902a009943
name	SynField-IoT
timestamp	1560860119

CLOSE

CONNECTED Geth/v1.8.27-stable-4bcc0a37/linux-amd64/go1.10.4

Figure 13. Event argument for IoT platform registration in the smart contract

In Figure 14, the event arguments of actor registration in the smart contact are shown. The arguments include actor ID, IoT platform ID (where the actor is a member) and the timestamp.



Monitored Events for Contract - 0x6d4A5a876D09F8A381993d7f8DF9D3764C5A2491

Name	Block #	Log Index	Tx Index	Tx Hash	Args	Timestamp
LogPlatformRegistered	55	0	0	0x1c76b89a367a37c4e3b964775a8d04798da500dc524dbd32de0e8c7f0a9f217f	🔍	2019-06-18T12:16:46.335875Z
LogPlatformRegistered	59	0	0	0x8a273391c548f9852d18c1d1721182cc807caeb49d49ec369cf215df27648393	🔍	2019-06-18T12:17:46.349272Z
LogActorRegistered	61	0	0	0x07c793ddf9ad0d3507abeff25e695e765b96bcdcd06a118d435fdbcaae388e89	🔍	2019-06-18T12:18:26.343754Z
LogActorRegistered	62	0	0		🔍	2019-06-18T12:18:26.364706Z
LogBoxSessionStarted	77	0	0		🔍	2019-06-18T12:22:26.347138Z
LogBoxSessionStarted	81	0	0		🔍	2019-06-18T12:23:26.350683Z
LogBoxSessionStarted	82	0	0		🔍	2019-06-18T12:23:26.371563Z
LogDataCropBlobAdded	110	0	0		🔍	2019-06-18T12:30:26.406762Z
LogDataCropBlobAdded	112	0	0		🔍	2019-06-18T12:31:06.401296Z
LogDataCropBlobAdded	112	1	1		🔍	2019-06-18T12:31:06.406646Z

Rows per page: 10 1-10 of 13

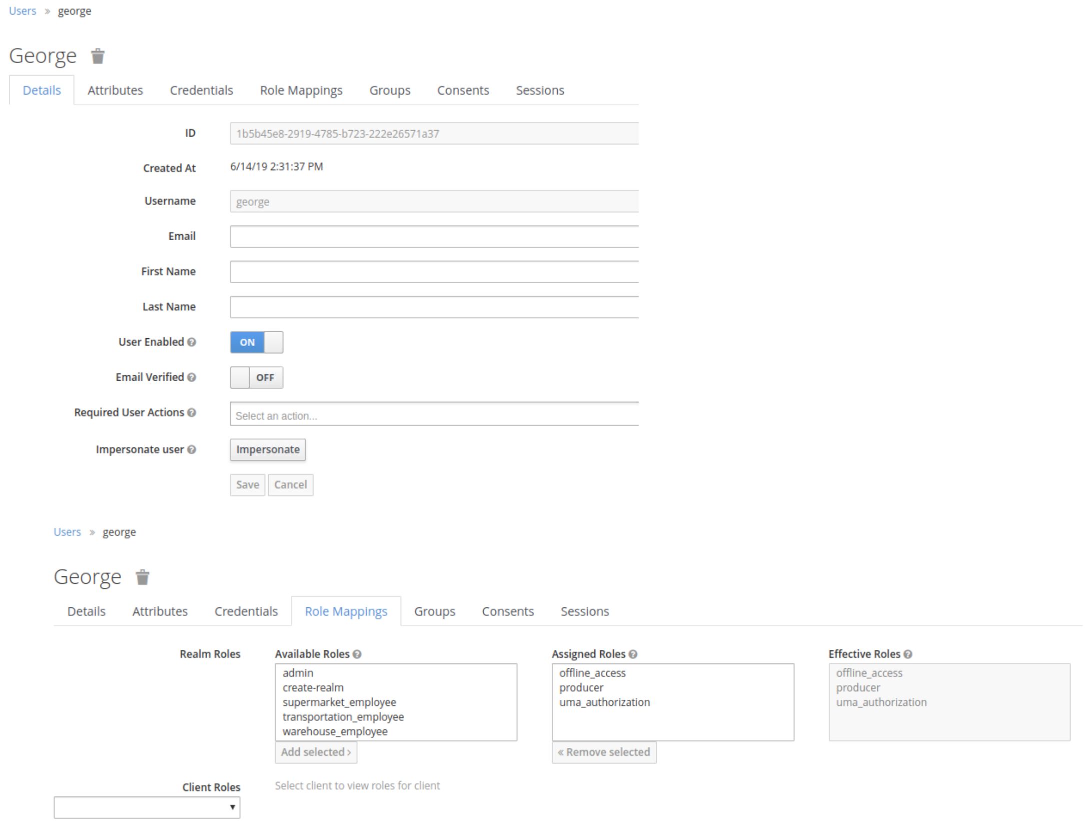
Event Arguments

id	1b5b45e8-2919-4785-b723-222e26571a37
role	0
platform	0xb5707BdcD820694303496B74d56895902a009943
timestamp	1560860258

CONNECTED Geth/v1.8.27-stable-4bcc0a37/linux-amd64/go1.10.4

Figure 14. Event argument for actor registration in the smart contract

The actor is registered into the smart contact with the ID which is produced by the keycloak authorization server when he registers in the SWS (i.e. 1b5b45e8-2919-4785-b723-222e26571a37 for a producer named as George), as seen in Figure 15. As said before, the authorization server applies a RBAC policy, thus only actors with specific roles can access specific API endpoints of the consortium ledger client, e.g. only producers can access ../register/boxing-product/ endpoint to create "box product" events.



ID	1b5b45e8-2919-4785-b723-222e26571a37
Created At	6/14/19 2:31:37 PM
Username	george
Email	
First Name	
Last Name	
User Enabled	ON
Email Verified	OFF
Required User Actions	Select an action...
Impersonate user	Impersonate

Realm Roles

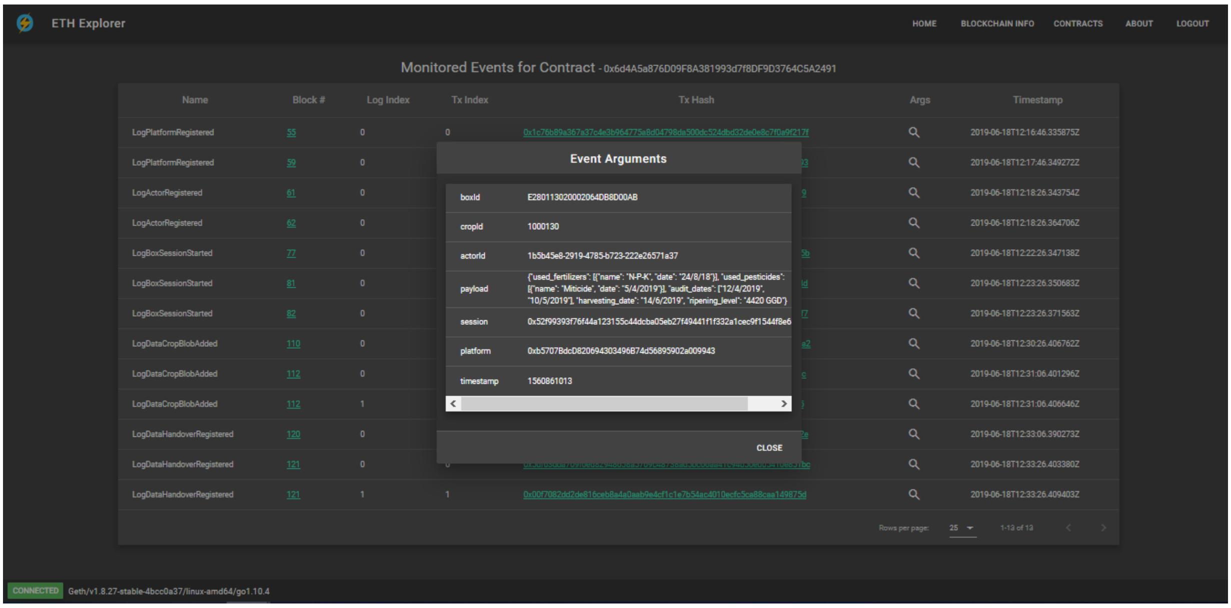
Available Roles	Assigned Roles	Effective Roles
admin create-realm supermarket_employee transportation_employee warehouse_employee	offline_access producer uma_authorization	offline_access producer uma_authorization

Client Roles

Select client to view roles for client

Figure 15. Actor registration by using Keycloak (role: producer, name: George)

In Figure 16, the arguments are shown of the event that records FSC_UC2 for one of the three boxes that have been registered. The ID of the used box is E280113020002064DB8D00AB, which is identical to the RFID tag that is attached to its surface. The box is linked with the established crop (ID 1000130), so allowing SWS to relate contained product with the field location and the product variety (which are determined by the action "register crop"). Also, the ID of the producer who performs the action is recorded (1b5b45e8-2919-4785-b723-222e26571a37). Note that only IDs and no private information for actors are written into the consortium ledger. To complete "box product", the actor should provide information about the harvesting date, audit dates and the usage of any fertilizers and pesticides during the cultivation. Note that the payload also includes information about growing degree days (GGD) of the product. This information, which is a metric of its ripening level, is not provided by the actor directly but it is retrieved through Synfied IoT platform (which tracks GGD of the product) upon preparation of the datum by the SWS. Finally, the event is also bound to the used IoT platform (ID: b5707BdcD820694303496B74d56895902a009943) and it is timestamped.



Monitored Events for Contract - 0x6d4A5a876D09F8A381993d7f8DF9D3764C5A2491

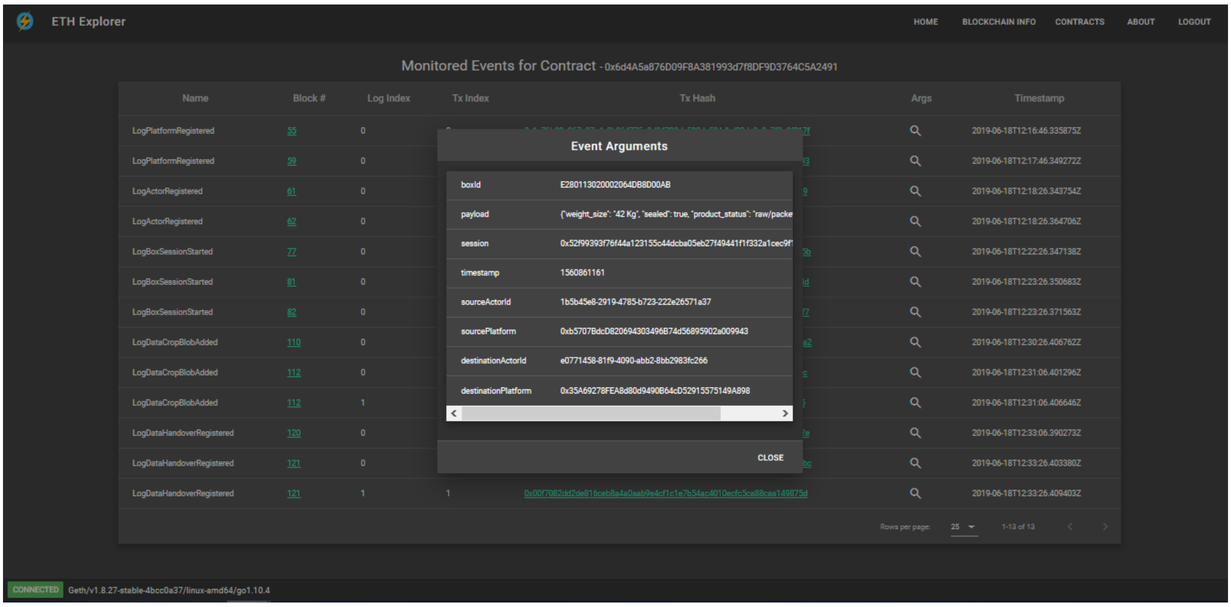
Name	Block #	Log Index	Tx Index	Tx Hash	Args	Timestamp
LogPlatformRegistered	55	0	0	0x1c76b89a367a37c4e3b964775a8d04798da500dc524dbd32de0e8c7f0a9f217f	🔍	2019-06-18T12:16:46.335875Z
LogPlatformRegistered	59	0		03	🔍	2019-06-18T12:17:46.349272Z
LogActorRegistered	61	0		9	🔍	2019-06-18T12:18:26.343754Z
LogActorRegistered	62	0			🔍	2019-06-18T12:18:26.364706Z
LogBoxSessionStarted	77	0		5b	🔍	2019-06-18T12:22:26.347138Z
LogBoxSessionStarted	81	0		ld	🔍	2019-06-18T12:23:26.350683Z
LogBoxSessionStarted	82	0		f7	🔍	2019-06-18T12:23:26.371563Z
LogDataCropBlobAdded	110	0		a2	🔍	2019-06-18T12:30:26.406762Z
LogDataCropBlobAdded	112	0		c	🔍	2019-06-18T12:31:06.401296Z
LogDataCropBlobAdded	112	1		5	🔍	2019-06-18T12:31:06.406646Z
LogDataHandoverRegistered	120	0		2e	🔍	2019-06-18T12:33:06.390273Z
LogDataHandoverRegistered	121	0	0	[illegible]bc	🔍	2019-06-18T12:33:26.403380Z
LogDataHandoverRegistered	121	1	1	0x00f7082dd2de816ceb8a4a0aab9e4cf1c1e7b54ac4010ecfc5ca88caa149875d	🔍	2019-06-18T12:33:26.409403Z

Event Arguments

boxId	E280113020002064DB8D00AB
cropId	1000130
actorId	1b5b45e8-2919-4785-b723-222e26571a37
payload	{"used_fertilizers": [{"name": "N-P-K", "date": "24/8/18"}], "used_pesticides": [{"name": "Miticide", "date": "5/4/2019"}], "audit_dates": ["12/4/2019", "10/5/2019"], "harvesting_date": "14/6/2019", "ripening_level": "4420 GGD"}
session	0x52f99393f76f44a123155c44dcba05eb27f49441f1f332a1cec9f1544f8e6
platform	0xb5707BdcD820694303496B74d56895902a009943
timestamp	1560861013

Figure 16. Event arguments for FSC_UC2

In Figure 17, the arguments are shown of the event that records a handover (FSC_UC5). In this case, the IDs of both actors who transact as well as the IDs of the corresponding IoT platforms are bound to the event datum. The payload, which must be agreed by both parties of the transaction, includes information about the weight of the box (the product which is contained in the box), whether the box is sealed or not and whether the product is packetized or not. Note that all payload information is recorded in the transaction input after being encrypted by using the Contract Application Binary Interface of the Ethereum, as shown in Figure 18.



Monitored Events for Contract - 0x6d4A5a876D09F8A381993d7f8DF9D3764C5A2491

Name	Block #	Log Index	Tx Index	Tx Hash	Args	Timestamp
LogPlatformRegistered	55	0	0	7f	🔍	2019-06-18T12:16:46.335875Z
LogPlatformRegistered	59	0		93	🔍	2019-06-18T12:17:46.349272Z
LogActorRegistered	61	0		9	🔍	2019-06-18T12:18:26.343754Z
LogActorRegistered	62	0			🔍	2019-06-18T12:18:26.364706Z
LogBoxSessionStarted	77	0		5b	🔍	2019-06-18T12:22:26.347138Z
LogBoxSessionStarted	81	0		ld	🔍	2019-06-18T12:23:26.350683Z
LogBoxSessionStarted	82	0		f7	🔍	2019-06-18T12:23:26.371563Z
LogDataCropBlobAdded	110	0		a2	🔍	2019-06-18T12:30:26.406762Z
LogDataCropBlobAdded	112	0		c	🔍	2019-06-18T12:31:06.401296Z
LogDataCropBlobAdded	112	1		5	🔍	2019-06-18T12:31:06.406646Z
LogDataHandoverRegistered	120	0		2e	🔍	2019-06-18T12:33:06.390273Z
LogDataHandoverRegistered	121	0		bc	🔍	2019-06-18T12:33:26.403380Z
LogDataHandoverRegistered	121	1	1	0x00f7082dd2de816ceb8a4a0aab9e4cf1c1e7b54ac4010ecfc5ca88caa149875d	🔍	2019-06-18T12:33:26.409403Z

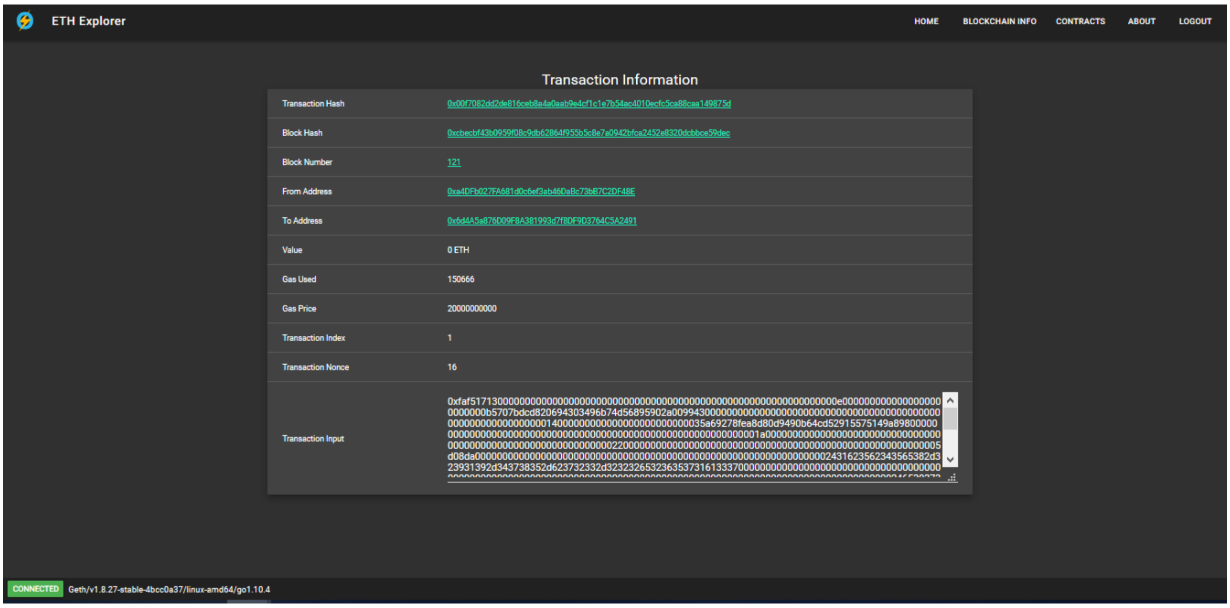
Rows per page: 25 ▾ 1-13 of 13

Event Arguments

boxId	E280113020002064DB8D00AB
payload	{"weight_size": "42 Kg", "sealed": true, "product_status": "raw/packe
session	0x52f99393f76f44a123155c44dcba05eb27f49441f1f332a1cec9f1
timestamp	1560861161
sourceActorId	1b5b45e8-2919-4785-b723-222e26571a37
sourcePlatform	0xb5707BdcD820694303496B74d56895902a009943
destinationActorId	e0771458-81f9-4090-abb2-8bb2983fc266
destinationPlatform	0x35A69278FEA8d80d9490B64cD52915575149A898

CONNECTED Geth/v1.8.27-stable-4bcc0a37/linux-amd64/go1.10.4

Figure 17. Event arguments for FSC_UC5



Transaction Information

Transaction Hash	0x00f7082dd2de816ceb8a4a0aab9e4cf1c1e7b54ac4010ecfc5ca88caa149875d
Block Hash	0xcbecbf43b0959f08c9db62864f955b5c8e7a0942bfca2452e8320dcbbce59dec
Block Number	121
From Address	0xa4DFb027FA681d0c6ef3ab46DaBc73bB7C2DF48E
To Address	0x6d4A5a876D09F8A381993d7f8DF9D3764C5A2491
Value	0 ETH
Gas Used	150666
Gas Price	20000000000
Transaction Index	1
Transaction Nonce	16
Transaction Input	0xfaf5171300000000000000000000000000000000000000000000000000000000000000e000000000000000000 0000000b5707bdcd820694303496b74d56895902a0099430000000000000000000000000000000000000000000 000000000000000000140000000000000000000000000035a69278fea8d80d9490b64cd52915575149a89800000 000000000000000000000000000000000000000000000000000000001a00000000000000000000000000000000 000000000000000000000000000000220000000000000000000000000000000000000000000000000000000005 d08da00000000000000000000000000000000000000000000000000000000000000002431623562343565382d3 23931392d343738352d623732332d3232326532363537316133370000000000000000000000000000000000000 000000000000000000000000000000000000000000000000000000000000000000000000000000000246530373

CONNECTED Geth/v1.8.27-stable-4bcc0a37/linux-amd64/go1.10.4

Figure 18. Transaction input of the event which is produced in FSC_UC5

3.3.4 Future validation activities

As the pilot system architecture is still in the development phase, Table 12 shows the status of the use cases needs to be validated in the forthcoming period.

Table 12. Status of use cases to be validated in the food supply chain pilot

ID	Name	Status
FSC_UC1	Register crop	It has been developed and validated (usage of API endpoint /blob/register/crop/ in Figure 11).
FSC_UC2	Box product	It has been developed and validated (usage of API endpoint /blob/register/boxing-product/ in Figure 11).
FSC_UC3	Hand over product: PR-TR	It has been developed and validated (usage of API endpoint /blob/register/handover/ in Figure 11).

FSC_UC4	Hand over product: TR-WH	In development: similar to the PR-TR handover, where payload information must be adjusted accordingly.
FSC_UC5	Register session	It has been developed and validated (usage of endpoint /box/session/start/ in Figure 11).
FSC_UC6	Pick truck	The data model format, the smart contract update and the API update are under development.
FSC_UC7	Transfer box(es)	The data model format, the smart contract update and the API update are under development.
FSC_UC8	Hand over product: TR-SM	In development: similar to the PR-TR handover, where payload information must be adjusted accordingly.
FSC_UC9	Place box(es)	The data model format, the smart contract update and the API update are under development.
FSC_UC10	Packetize product	The data model format, the smart contract update and the API update are under development.
FSC_UC11	Create QR code	The data model format, the smart contract update, the API update and the application to create QR are under development
FSC_UC12	Release box(es)	It has been developed and validated (usage of API endpoint /box/session/end/ in Figure 11).
FSC_UC13	Read QR code	The mobile application to read QR codes is under development.
FSC_UC14	Product audit	The data model format, the interledger client and the API endpoints update are under design.

To validate the above use cases and evolve the PoC prototype to the first release of the pilot minimum viable product (planned for Oct. 2019), the following implementation steps have been scheduled:

- Federation of Aberon IoT platform and adaption of its model according to the SOFIE data/things representation schema.
- Infrastructure testing to validate failover, high availability, and possibly scalability.
- Integration and customization of SOFIE framework components into system software architecture.
- Data models update (for the events of all the considered use cases) and update/finalization of SWS API endpoints.
- Onboarding (part of) system software components in SOFIE CI/CD environment to enable fast validation and immediate deployment of updates.
- Implementation of the SOFIE FSC web application by also taking into account aspects relate to the actors' experience (e.g. user-friendly interfaces, reliability, responsiveness etc.).
- Implementation of QR reading mobile application and end-to-end validation of the QR code creation service.
- User testing and iterative feedback to optimize user experience.
- Validation of business platform services and components interoperability using real data through SOFIE continuous integration environment.

4 Decentralized Energy Data Exchange Pilot

4.1 Pilot overview

4.1.1 Application context, constraints and unsolved issues

The decentralized energy data exchange pilot key driver is enabling the data owner (i.e. the entity/person who owns legally smart meter data) to:

- choose who gets access to the data,
- enable the data transfer supported with business logic,
- receive the proof of the parties accessing/using the data,
- receive the guarantee that activities are done according to GDPR and high security requirements.

The starting point for the decision making of access rights is fixed on the data owner side but the smart meter data storage and data processing can be different. The pilot is focusing in three different data access points:

- data from the National data hub (Estfeed platform),
- data from a regional database, energy subsystem (wind farm network),
- data from the single metering point from the household (zero-energy building)

From the legislation side in EU, the shaping of the decentralized energy data exchange pilot relies on three initiatives/regulations:

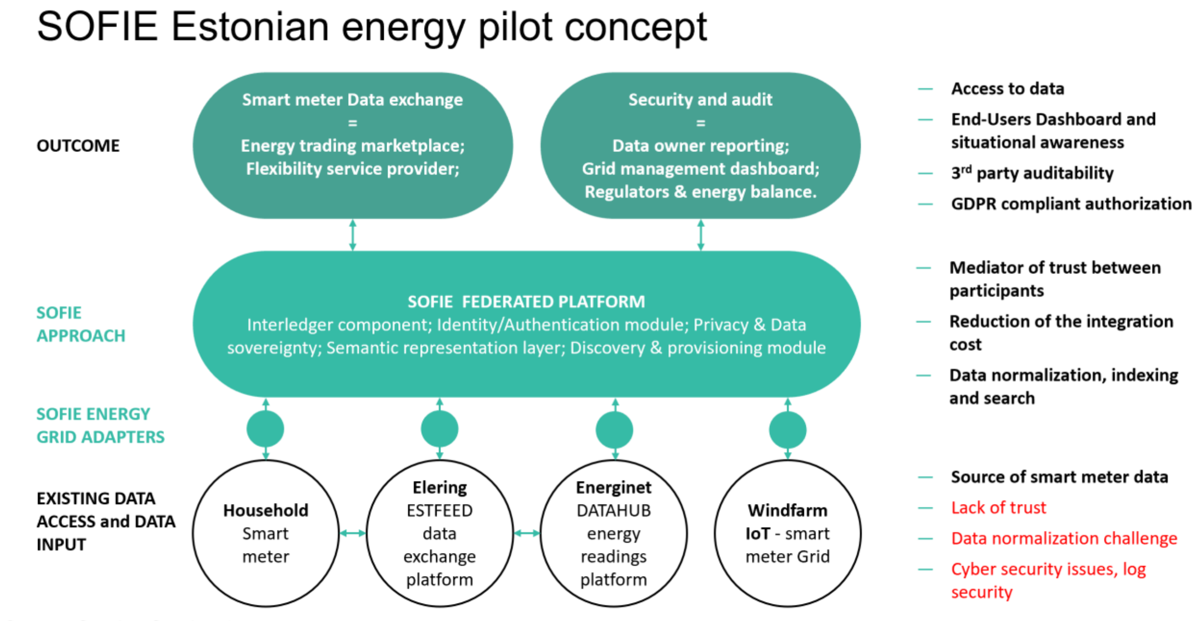
- The EU Clean energy package[13] aims at protecting the rights of consumers to receive easy and free access to data on real-time and historical energy consumption".
- The EU Smart Grids and meters initiative[14] aims at advising EU how to develop the energy market in coming years and uptake of smart grids and meters.
- GDPR[15] aims at regulating EU citizens rights to protect their personal data.

The initial design of the decentralized energy data exchange pilot is reported in SOFIE D5.1. The consultations with energy sector participants (data owners, TSOs, DSOs, Brokers etc.) have led to change the approach from the data hub (Estfeed) centric focus to a broader solution, where national data hubs are a trusted source of data (including consent, authentication handling with data owners) and the other data sources are treated equally. Nevertheless, the core idea of the pilot remains to provide a proof-of-concept for secure data exchange and agreements to data access rights between smart meter data and infrastructure owners and energy service providers (intermediaries, distributors, brokers). The pilot will develop and use the capabilities of the SOFIE federated platform for the validation and demonstration of the defined scenarios and use-cases.

The pilot will use the Estfeed open software platform (connecting 700 000 smart meters in Estonia) for energy consumption monitoring and consumers/ prosumers management as a key input. In order to demonstrate the cross-border data exchange and transfer of Trust between network grid participants the Danish Datahub (Energinet) will be the secondary input for the pilot. Besides national hubs integration, the pilot will also develop adapters and connection to two other instances: local IoT network (windfarm) and household metering point.

The concept of decentralized energy data exchange pilot is shown in Figure 19, where participants, the SOFIE approach and the added value are presented.

[13] https://ec.europa.eu/energy/en/topics/energy-strategy-and-energy-union/clean-energy-all-europeans
[14] https://ec.europa.eu/energy/en/topics/markets-and-consumers/smart-grids-and-meters
[15] https://gdpr-info.eu/



SOFIE Estonian energy pilot concept

OUTCOME

Smart meter Data exchange
=
Energy trading marketplace;
Flexibility service provider;

Security and audit
=
Data owner reporting;
Grid management dashboard;
Regulators & energy balance.

- Access to data
- End-Users Dashboard and situational awareness
- 3rd party auditability
- GDPR compliant authorization

SOFIE APPROACH

SOFIE FEDERATED PLATFORM
Interledger component; Identity/Authentication module; Privacy & Data sovereignty; Semantic representation layer; Discovery & provisioning module

- Mediator of trust between participants
- Reduction of the integration cost
- Data normalization, indexing and search

SOFIE ENERGY GRID ADAPTERS

EXISTING DATA ACCESS and DATA INPUT

Household Smart meter

Elering ESTFEED data exchange platform

Energinet DATAHUB energy readings platform

Windfarm IoT - smart meter Grid

- Source of smart meter data
- Lack of trust
- Data normalization challenge
- Cyber security issues, log security

Figure 19. The concept of the decentralized energy data exchange pilot

During the decentralized energy data exchange pilot there will be a demonstration on:

a) how a data owner can give access rights and share the data,

b) how a distributor from another country can access the smart meter data,

c) how different IoT platforms can validate the data exchange.

The goal of the decentralized energy data exchange pilot is to prove how the SOFIE federated platform can improve the existing energy sector platforms with blockchain technology in the following aspects:

- Integrity of energy consumption data and transparency mechanisms.
- Provide trust between energy market participants.
- Protect the data owner that his/her data is treated according to regulations.
- Protect the entity (Elering, Energinet) who provide access to smart meter data from court cases.
- Provide energy consumption readings which are correct beyond dispute.

Furthermore, the current energy pilot shall investigate and demonstrate opportunities on the use of new technology in order to allow more open access to metering data and therefore enable 3rd parties to develop applications on top of the smart meter readings federated platform provided by the SOFIE framework. This could initiate a really distributed energy trading marketplace to build flexible energy services.

The pilot will also use SOFIE identity and authentication module. The components involve eIDAS compliant regional authentication, using PKI, the DIDs and KSI blockchain as an ultimate trust anchor for linking personal data and the unique ID-s given to each trusted participant. Using the SOFIE federated approach the trusted list of entities on a cross European level are held in a distributed way. The entity/customer data is kept in each of the participants' premises and the IDs will be shared between all parties.

4.1.2 SOFIE added value

One of the key elements and technical challenges in the decentralized energy data exchange pilot is to make the data owners reachable for the energy service providers and vice versa, so as to provide relevant list of services to the data owner to choose from. The added value from SOFIE is to use two modules that are developed across pilots in the project: the discovery and provisioning module and the universal semantic representative layer. This will enable the interaction between new opt in participants to federated energy platform and the existing regional and national data hubs, that have trusted list of service providers and data owners. Table 13 summarizes the SOFIE added value compared to existing data exchange systems.

Table 13. SOFIE added value compared to existing national platforms and standalone data owners

Data owner current status	SOFIE added value
Minimal or no control over the selection of new services.	• Enable the indexing, search and selection of services to be granted data access rights. • Integrity of data without a centralized authority. • Reduce the chances of fraud, cutting out corresponding mediation expenses and transaction costs and demonstrate proof of interaction between different parties.
Vendor locked solution and geographically dependent on DSO, TSO	• Providing components to select preferred service provider, bypassing existing regional service provider. • Compliance with clean energy package and opening the data access to interested parties. • An easy to use and non-disruptive solution to federate heterogeneous IoT environments and enable them to interoperate in composing and analyzing valuable information.
Audits, quality checks based on the closed system willingness to cooperate.	• Built in auditing of the authentication, data access and data sharing to any 3rd party through SOFIE components. • Allow interconnection between national data hubs auditing mechanisms and standalone data sources.

The KPIs of the pilot are summarized in Table 14.

Table 14. KPIs in decentralized energy data exchange pilot

Technology Level	
Enabling software and development of applications	• Dynamic and intuitive interface for authentication and access control for data owner
Federating platforms	• Enable data import from 3 stand-alone sources to service provider • Federate 2 IoT national platforms • Consolidate 4 different data sources to the SOFIE federated platform approach.

Business Level	
Regulatory metrics	- List of countries/metering points GDPR compliant data handling - Clean energy package compliant process
Customer metrics	- Customer satisfaction through freedom to choose supplier - Time and effort for finding, choosing new service provider - Percentage of resolved disputes
Business metrics	- Number of services listed in SOFIE platforms

4.1.3 Pilot status and timeplan

Currently, the pilot is under development phase. At this stage pilot status and plan do not deviate from the timeline shown in Table 3.

4.2 Pilot scenarios, requirements and system architecture

4.2.1 Scenarios and use cases

The main objective of the pilot is to enable trust between parties who exchange energy meter readings. From the data owner side, it is critical to guarantee control of the data, as well as access in audit logs for transparent overview on whom the data access rights are given to and how private data are handled. From the smart meter system operator side (TSO or DSO), there is a need for guarantee about who is responsible after the data exchange and how to agree and prove the responsibility. From the data consumer side (brokers, aggregators, energy traders) it is important to get access to the data to support their business case and to be able to reliably verify information from the whole data provenance chain. From the auditors' side, it is required to get access to audit logs and receive tamper-proof evidence of the activities taken place in data exchange process. Main objective can be divided into the following two scenarios:

1. Data exchange scenario - covering the full chain from identification, authorisation to granting, requesting access and exchanging the smart meter data.
2. Data exchange verification scenario - to provide means for audit logs, tamper-proof evidence in case of disputes, and verification of the integrity of smart meter data.

The complexity of the scenarios stems from the fact that there are multiple different systems involved in the data exchange, different technologies being used and geographical/cross border distribution of participants. Also, there is no single specific certification authority to be the trusted party.

4.2.1.1 Actors

There are four actor roles that will appear in the pilot:

1. Smart meter system operator (SMSYS) – Entity who is responsible for some part of the energy grid that is needed for energy transaction. It is the responsible party who has authority and certificates to install, connect and operate smart meters in the grid. Depending on the country, the SMSYS can be a Transmission System Operator (TSO), a Distribution System Operator (DSO) or a separate company operating the low voltage network. SMSYS has the obligation to run the network grid and manage the smart meter infrastructure. In the decentralized energy data exchange pilot, the SMSYS can also be the smart meter data collector (when granted by data owner) and

they have the role of providing access to the smart meter data to all interested parties. SMSYS are the ones who are subject to clean energy package and open access to energy consumption data.

2. Smart meter data owner (OWNER) - Private entity or company who is legally bound to smart metering point and is interested to consume/produce energy. It has the right to give access of smart meter data and a subject to privacy when sharing, using their data.

3. Data consumer - energy service provider (SERVICE) - The entity who is responsible for providing the energy service to the end-user. It is the main communicator between customer and grid operator. It can own low, medium voltage grid and smart meter infrastructure but it is possible to have no grid/infrastructure responsibilities. It is also the main user of the smart meter data to provide cross country flexibility services in energy sector.

4. Auditor (AUDIT) – This entity can be the network controlling authority, an electricity grid expert (in case of disputes), an EU security council and data protection agency. It is the organisation that has a governing, auditing role in energy grid operations and plays a role of a middleman in case there are disputes between other parties.

4.2.1.2 Decentralized energy data exchange pilot scenarios

SCENARIO	Smart meter data exchange
History	Priit Anton– V04 – Scenario description
Key Actors	OWNER, SERVICE, SMSYS
Assumptions / Dependencies	1) Interest to exchange data is driven by business need (contract between OWNER+SERVICE, OWNER+SMYS). 2) Request is coming from data owner, who initiates the process with Service provider. 3) There is a metering point (certified and connected to the grid) with data owner. 4) SOFIE framework components implemented. 5) There is a trust authority in each country where actors are. 6) Trust authority guaranteed identity is linked with SOFIE identity.
Objective(s)	1) Configuration of metering point (SOFIE adapter installation). 2) Handling the access rights and actors in the system. 3) Removing the access rights. 4) Request and pull data. 5) Data transfer between actors
Description	1) There is a data owner with the connected smart meter to energy grid and is consuming/producing energy. 2) Data owner initiates a process to change an energy service provider, he/she connects to service providers webpage, authenticates and agrees on a contract between the data owner and service provider. 3) Data owner grants access or takes data access away from/to the service provider. 4) System operator provides data access based on access rights. 5) Service provider is able to get energy consumption data from the data owner in order to follow the contract agreements with the data owner.
Services	Discovery; Semantic representation; Privacy and data sovereignty; Interledger services, data exchange middleware
Metrics	• Three different type of data sources (standalone smart meter, National data hub, regional smart grid) integrated with SOFIE adapter. • Interledger based identity management between 4 actors in the system. • Data transfer between 3 actors in the system.

SCENARIO	**Data exchange validation**
History	Priit Anton– V04 – Scenario description
Key Actors	OWNER, SERVICE, SMSYS, AUDIT
Assumptions / Dependencies	1) The data exchange business logic and functionalities are input to validation scenario. 2) The driver to perform audit can be external (regulatory) or internal (by one of the actors) 3) The data exchange processes can be logged 4) Legislation in place in what cases Auditors can intervene with the energy consumption/production data exchange process and request data.
Objective(s)	1) Logging of participants activities (authorization, data access, data input/output, storage) 2) Enable request audit logs 3) Enable request for Personal information 4) Providing integrity of logs and metadata 5) Providing input data for dispute handling
Description	1) Data owner / System operator can view the data access history from the dashboard. 2) Data owner can request and receive information about his/her data in accordance to the GDPR 3) System operator can receive the proof of who accessed the data owner data through their system (according to consent) 4) In case of a legal/regulatory reason Auditor can gain access to data exchange, actors interaction data to preform inspection or get input data for handling disputes 5) All actors can validate the integrity of proofs without the access to SOFIE platform.
Services	KSI blockchain; Privacy and data sovereignty; Interledger services,
Metrics	• Logs written to SOFIE adapter • Proof of the logs integrity • Proof of the smart meter data integrity

4.2.1.3 Decentralized Energy Data Exchange pilot use cases

The UML use case diagram in Figure 20 shows how the various actors interact with decentralized energy data exchange system.

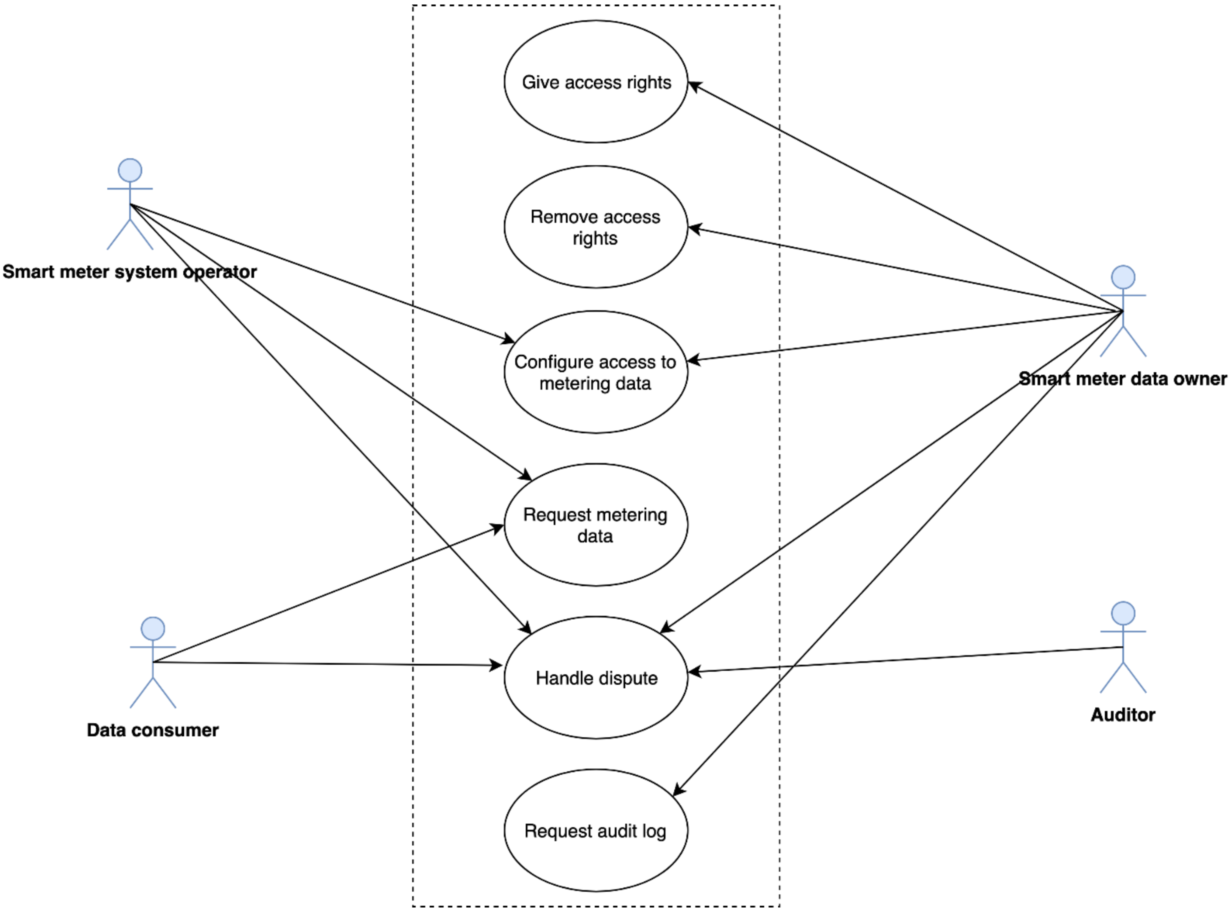


Figure 20. Actors' interaction with the decentralized energy data exchange system

In Table 15, the pilot use cases are analyzed in more detail.

Table 15. Use cases in the decentralized energy data exchange system

ID	Name	Actors
DEDE_UC1	Configure access to metering data	OWNER, SMSYS
DEDE_UC2	Request metering data	SERVICE
DED_UC3	Give access rights	OWNER
DEDE_UC4	Remove access rights	OWNER
DEDE_UC5	Request audit log	OWNER, SMSYS, AUDIT
DEDE_UC6	Handle dispute	OWNER, SMSYS, SERVICE, AUDIT
USE CASE Description		
ID	DEDE_UC1	
Name	Configure access to metering data	
Actors	OWNER, SMSYS	

Storyline	Owner initiates the connection to the SOFIE network to enable the access to his/her smart meter data and accessing rights. SOFIE adapter is installed, connections/access rights granted by SMSYS. SOFIE adapter functionality provided to OWNER
Trigger events	Owner interested to share data to SERVICE
Preconditions	Certified smart meter connected to grid approved by SMSYS SOFIE identifiers created to all actors Contract between OWNER and SERVICE
Postconditions	-
Related scenarios	Smart meter data exchange
USE CASE Description	
ID	DEDE_UC2
Name	Request metering data
Actors	SERVICE, OWNER
Storyline	SERVICE is interested to provide energy service to OWNER and needs access to the energy consumption data. After access rights are given, SERVICE can start downloading the data from OWNER and use this data to fulfill the contract.
Trigger events	Contract between OWNER and SERVICE
Preconditions	All participants have integrated to SOFIE adapters
Postconditions	-
Related scenarios	Smart meter data exchange
USE CASE Description	
ID	DEDE_UC3
Name	Give access rights
Actors	OWNER
Storyline	OWNER can grant access rights of his/her data to SERVICE and SMSYS
Trigger events	OWNER has proven the ownership of the smart meter data (through government approved authorisation).
Preconditions	All actors must have integrated SOFIE adapters and own SOFIE identifier
Postconditions	-
Related scenarios	Smart meter data exchange

USE CASE Description	
ID	DEDE_UC4
Name	Remove access rights
Actors	OWNER
Storyline	OWNER can apply GDPR regulation to any actor involved in access of his/her data
Trigger events	Request by OWNER
Preconditions	-
Postconditions	-
Related scenarios	Smart meter data exchange

USE CASE Description	
ID	DEDE_UC5
Name	Request audit log
Actors	OWNER, SMSYS, SERVICE; AUDIT
Storyline	Any actor of decentralized energy data exchange pilot can request an audit log of his/her activities and interaction with other parties. OWNER and SMSYS can use events dashboard to view and analyse the history related to data exchange (based on the consent of OWNER)
Trigger events	Request by OWNER, SMSYS, SERVICE; AUDIT
Preconditions	System logs created locally by each actor
Postconditions	-
Related scenarios	Data exchange validation

USE CASE Description	
ID	DEDE_UC6
Name	Handle dispute
Actors	OWNER, SMSYS, SERVICE; AUDIT
Storyline	In case of dispute any actor of the decentralized energy data exchange pilot can get proof of his/her activities related to data exchange and granting access. The data integrity and time can be verified by an external expert and will support the legal rights provided from contract agreements between parties, obligations in regulations side and accordance to GDPR
Trigger events	external event to start auditing, dispute process
Preconditions	Legislation in place for dispute

Postconditions	-
Related scenarios	Data exchange validation

4.2.2 Requirements and software architecture

4.2.2.1 Pilot domain requirements

Table 16 summarizes the domain and end-user requirements of the decentralized energy data exchange pilot.

Table 16. List of requirements in the decentralized energy data exchange pilot

ID	Name	Description	Priority	Related use case
REQ_DEDE1.1	Data access	Data owner can access info about his data, full visibility of data use	MUST	DEDE_UC1 – DEDE_UC5
REQ_DEDE1.2	Unique identifiers for actors	Each actor must be identified	MUST	DEDE_UC1 – DEDE_UC5
REQ_DEDE2.1	Data access	Owner must be able to decide who gets access to his/her data	MUST	DEDE_UC2 – DEDE_UC4
REQ_DEDE2.2	Auditability / Security	All user info must be GDPR compliant	MUST	DEDE_UC2
REQ_DEDE2.3	Auditability / Security	Data handover must be registered and proved at every transaction	MUST	DEDE_UC2
REQ_DEDE2.4	Data access	Service provider must be able to define the energy consumption data parameters	MUST	DEDE_UC2
REQ_DEDE2.5	Data transfer	Service provider must be able to download the energy consumption data	MUST	DEDE_UC2
REQ_DEDE2.6	Authentication	Authentication toolkit for all actors (eIDAS compliant)	MUST	DEDE_UC2 – DEDE_UC6
REQ_DEDE2.7	Auditability / Security	Processes monitoring the system must be logged, stored (in local environment)	MUST	DEDE_UC2 – DEDE_UC5
REQ_DEDE5.1	Auditability / Security	Service provider must be able to get proof of receiving the energy consumption data	MUST	DEDE_UC5
REQ_DEDE5.2	Auditability / Security	System logs integrity must be 3rd party verifiable (auditor)	MUST	DEDE_UC5, DEDE_UC6

4.2.2.2 Pilot security and privacy specifications

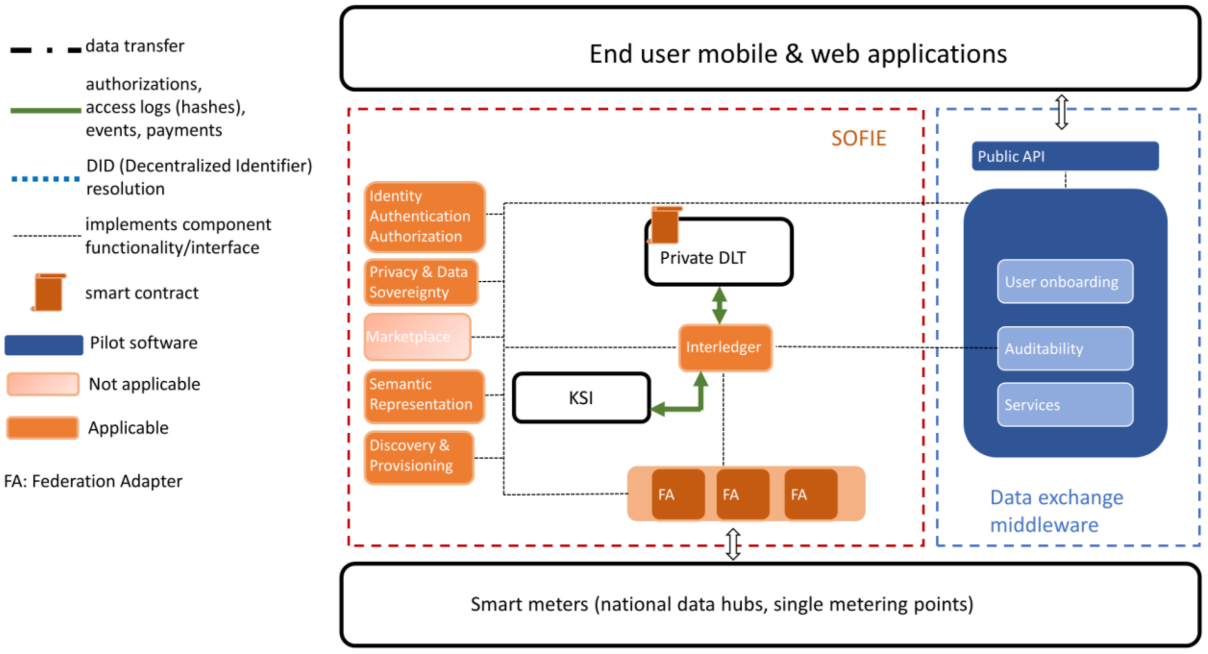
Table 17 presents how the decentralized energy data exchange pilot aims to address the critical security and privacy challenges that have been identified in SOFIE.

Table 17. Security and privacy specifications in the decentralized energy data exchange pilot

Support for transactions, where only authorised entities can participate
All participants in the interactions must have electronic identities defined. Decentralized identifiers (DID) will be used to give data owners' full control over data. Thus, actors are not dependent on 3rd party identity providers. To map real-life identities with DID-s some special issuers are needed to support such credentials. For the authorized entities a secure data exchange channel will be used.
Transactions must be authentic and verifiable
All participants will use DID-s for authentication. Participants have full control over their DIDs. Additionally, all interactions between the parties are signed by KSI blockchain - the hash of the interaction payload will be time stamped and can be later verified if required.
Privacy issues and business secrets must be considered carefully when deciding what data (including authentication/authorization information, logs etc.) is collected, stored or exchanged between parties
No metering data is stored in the system. Data is stored in data owner or in data hub side. For data exchange a secure communication channel is created between the participants, who have exclusive access to the data. Each federation adapter signs every interaction with KSI blockchain, which uses only the hash of the payload.

4.2.2.3 Pilot system architecture

The overview of the pilot system architecture is shown in Figure 21.



data transfer
authorizations, access logs (hashes), events, payments
DID (Decentralized Identifier) resolution
implements component functionality/interface
smart contract
Pilot software
Not applicable
Applicable
FA: Federation Adapter
End user mobile & web applications
SOFIE
Identity Authentication Authorization
Privacy & Data Sovereignty
Marketplace
Semantic Representation
Discovery & Provisioning
Private DLT
Interledger
KSI
FA
FA
FA
Public API
User onboarding
Auditability
Services
Data exchange middleware
Smart meters (national data hubs, single metering points)

Figure 21. Overview of system architecture for the decentralized energy data exchange pilot

SOFIE federation adapters will be used to enable data exchange with different smart meter systems:

- **National data hub** - existing information systems having non-standard integration options. The existing data hub has information about users and their consumption history. Each data hub needs to be integrated separately.
- **Single metering point** - adapter will enable requesting metering data from existing device.
- **Wind farm network** - adapter enables data exchange with a group of smart meter devices including also production data additionally to consumption data.

End users will interact through web interfaces and mobile applications. Middleware layer will provide different APIs for those applications to enable onboarding, interaction with SOFIE components and other activities required for secure data exchange. SOFIE framework components will be used to help manage service discovery, IAA, privacy and data sovereignty and following semantic representation rules. Consumption data is stored on data owner and data hub level. When a secure connection is established between the parties, data exchange will be performed point-to-point.

In the decentralized energy data exchange pilot architecture, the interledger and marketplace components are not used at the current stage. When there is a requirement from use-case and scenario side, the current architecture approach enables to incorporate these components.

4.2.2.3.1 Structural components of pilot system architecture

Figure 22 shows the main components of decentralized energy data exchange system architecture.

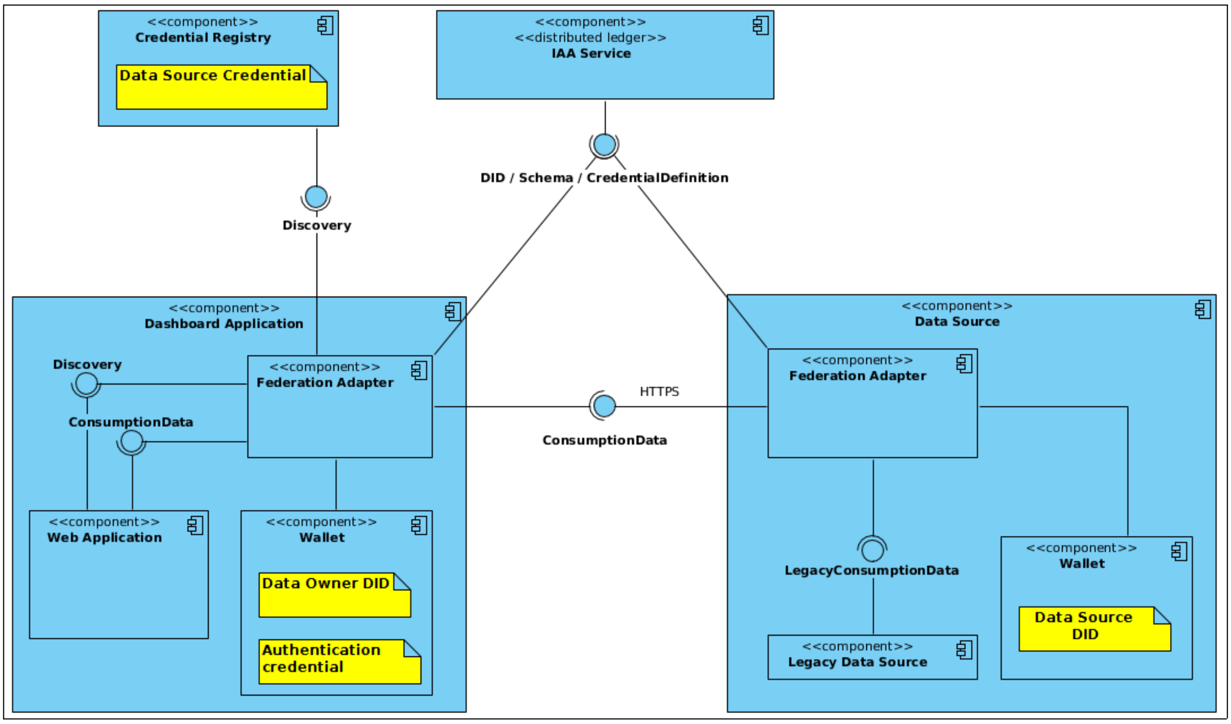


Figure 22. Components of decentralized energy data exchange system architecture

The main components of the architecture are:

- Dashboard Application - The component facing end user. It serves the role of Data Consumer. This is being run by a third party.
- Web Application - The front-end part of the Dashboard Application.

- Data Source - There will be multiple Data Source components in the pilot, all following the same basic composition shown in the diagram. This is part is operated by the data owner or a service provider authorized by a data owner.
- Federation Adapter - The central component for every participant on SOFIE. It is responsible for establishing a secure communication channel with another SOFIE Adapter.
- Legacy Data Source - Any source of electricity consumption data, with its own data format.
- Wallet - The component holding private keys for DID-s, and credentials needed to authenticate Data Owner to Data Source.
- IAA Service - A decentralized ledger service providing self-sovereign identity to SOFIE participants.
- Credential Registry - The main purpose of this component is to provide discovery service. The data Consumer needs to find Data Sources providing consumption data.

4.2.2.3.2 Sequence of interaction for pilot use cases

In Figure 23, system dynamics are represented for the main use case of the pilot. Data consumer needs access rights to smart meter data in order to provide the service. If the data consumer is not authorized, no data is exchanged. Managing access rights will generate KSI signatures, which can later be used for audit flows. Every data exchange interaction will also produce KSI signatures to store the history of data exchange interactions.

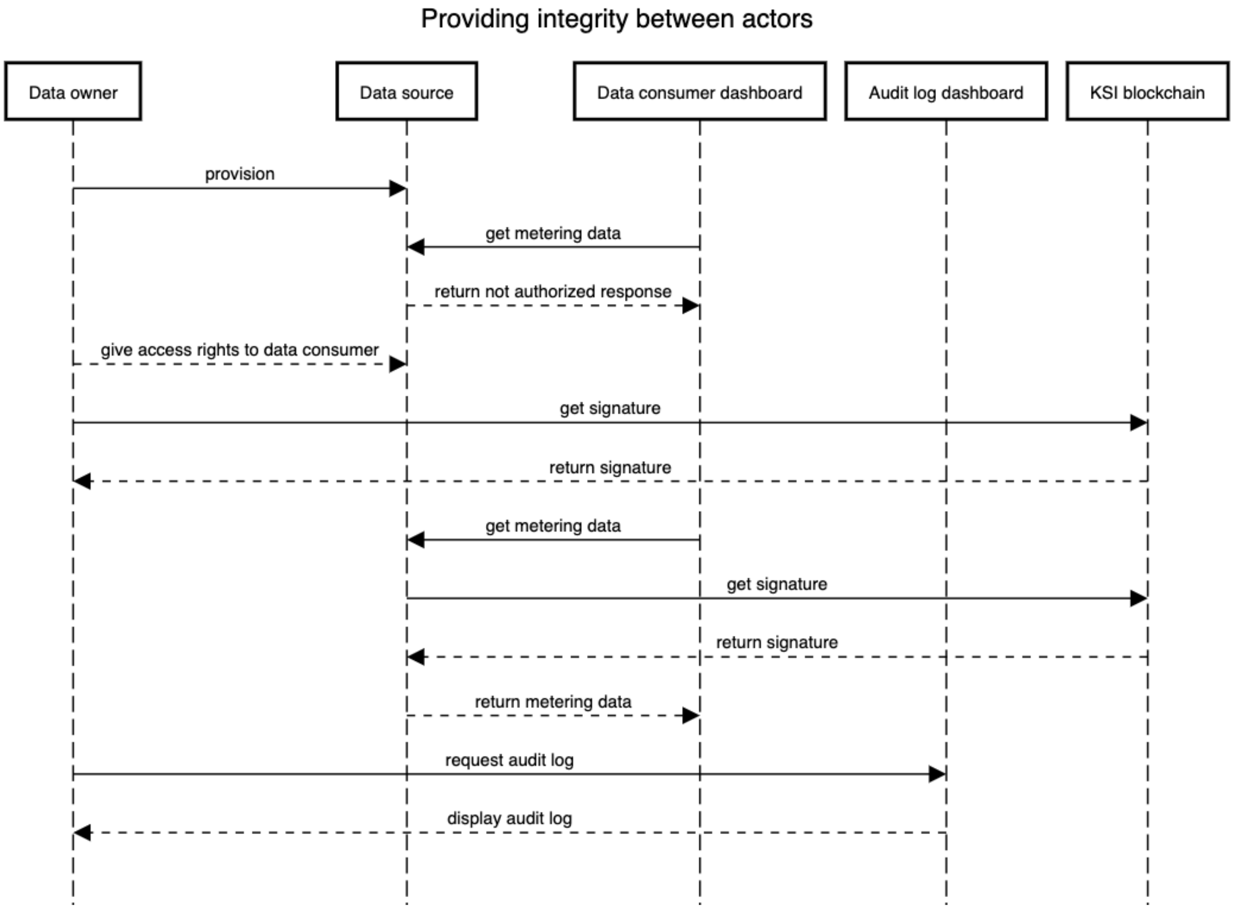


Figure 23. Smart meter data access and integrity

4.3 Initial SOFIE validation

In the first phase integration with Estonian national data hub Estfeed is validated. The goal of the integration is to make sure the services provided by the data hub and user management can be matched with the pilot approach. Additionally, for the user management different self-sovereign identity solutions were investigated. Hyperledger Indy seemed the most promising options and it's also used in other parts of the SOFIE framework.

4.3.1 Installed infrastructure

For the Estfeed integration the following steps were necessary:

- Deploying UXP security server[16]. This is needed for secure data exchange between Estfeed adapters. Only registered members can get access.
- Deploying Estfeed application adapter to interact with different Estfeed services.
- Demo application to interact with Estfeed adapter and provide a REST interface for other services.

4.3.2 Proof of concept prototype

The proof of concept prototype consists of different components that enable access to Estfeed.

4.3.3 Validation results

Based on the initial validation the following results were gathered:

- The integration with national data hub.
- The Estfeed service details regarding getting metering points per user and getting historical consumption data are confirmed.
- The data model provided by Estfeed is used as an input for the more generic SOFIE specific data model.
- The preliminary outcome of the mapping of users inside the data hub and the users in SOFIE was achieved. There is clear indication that it is challenging to combine the input from different sources. The result is that there are alternative possible approaches to be investigated.

4.3.4 Future validation tests

The most important thing for future validations is the solution for user management - users in SOFIE and users in closed systems must have mappings. The approach of DIDs will be validated across the system.

[16] https://cyber.ee/products/secure-data-exchange/

5 Decentralized Energy Flexibility Marketplace

5.1 Pilot overview

The decentralized energy flexibility marketplace pilot will demonstrate the ability to create smart micro-contracts and micro-payments in a fully distributed energy market. The pilot will include buildings, PV plants, electric vehicles (EV) and charging stations in the Terni area (Italy), each monitored in real time by the smart meter, thus having within the scenario production, distribution, storage and consumption of electricity under control. During the scenario, electricity produced from renewable sources (PV) will be fed into the low voltage electricity grid (LV). Most of this electricity will normally be consumed by energy customers adjacent to the generation plants, however, the excess of the generated power will create reverse power flow through the substation of the LV distribution network. The electrical distribution network is designed to handle only unidirectional electricity flows, so any reverse power flows cause malfunctions and reduced equipment lifetime, therefore, they cause economic damage that the DSO wants to avoid.

To remedy the abnormal functioning described above, the DSO (i.e. the operator that manages the LV and MV network) will create DR campaigns to consume the PV plants surplus of energy. DR campaigns will be directed to EV Fleet Managers due to their ability to consume large amounts of energy by recharging electric vehicle batteries. The Fleet Managers will be able to participate in the auction with the aim of obtaining the economic bonus for providing flexibility; immediately after having won the auction, the Fleet Manager will create an auction to request lower priced electricity supply to the Energy Retailers. The DR campaign thus concludes with the DSO which has avoided reverse power flow and solved the problems of electricity network, the Fleet Manager who reloaded its fleet of electric vehicles at an advantageous price and the Energy Retailer who achieved daily energy buying and selling goals. This will be possible thanks to SOFIE that enables:

- Secure identification of the actors/authors involved;
- IoT standardization of different business environments, enabling interoperability;
- Adaptation of different IoT platforms that enable coordinated data processing and market management, as well as, micro-contracts and micro-payments management.

5.1.1 Application context, constraints and unsolved issues

The pilot trials will take place at the ASM's headquarters. According to the general overview given in Figure 24, ASM district can be considered as a living lab because of the presence of advanced technologies already tested and in operation for the purpose of validating the main pillars of a smart grid (e.g. AMI, DR, Storage, EV integration, V2G).

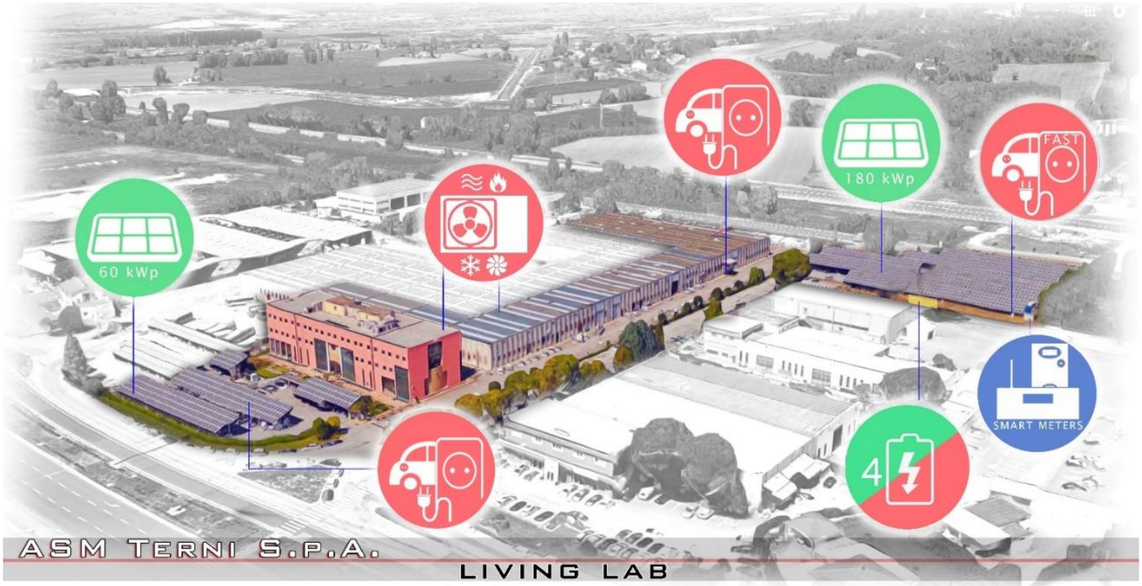


Figure 24. ASM headquarters

In particular, the district consists of the following block of energy units:

- Two PV arrays (180 kWp and 60 kWp), connected to the LV network;
- 72 kWh 2nd life Li-ion battery energy storage is the Block of Energy Unit (BoEU) providing the electric power storage and supply services. It is the BoEU that plays an important role in providing the district with the flexibility necessary to implement different services, especially ancillary services like Primary reserve, Dynamic reactive Power control and Reactive Power Compensation;
- ASM Terni buildings comprising a 4,050 m2 three-storey office building, a 2,790 m2 single-storey building consisting of technical offices, a computer centre and an operation control centre and a 1,350 m2 warehouse; usually the base load varies between 50 kW and 90 kW and peak load is between 120 kW and 170 kW, depending on seasonal factors. A daily load profile is shown in Figure 25 as an example.

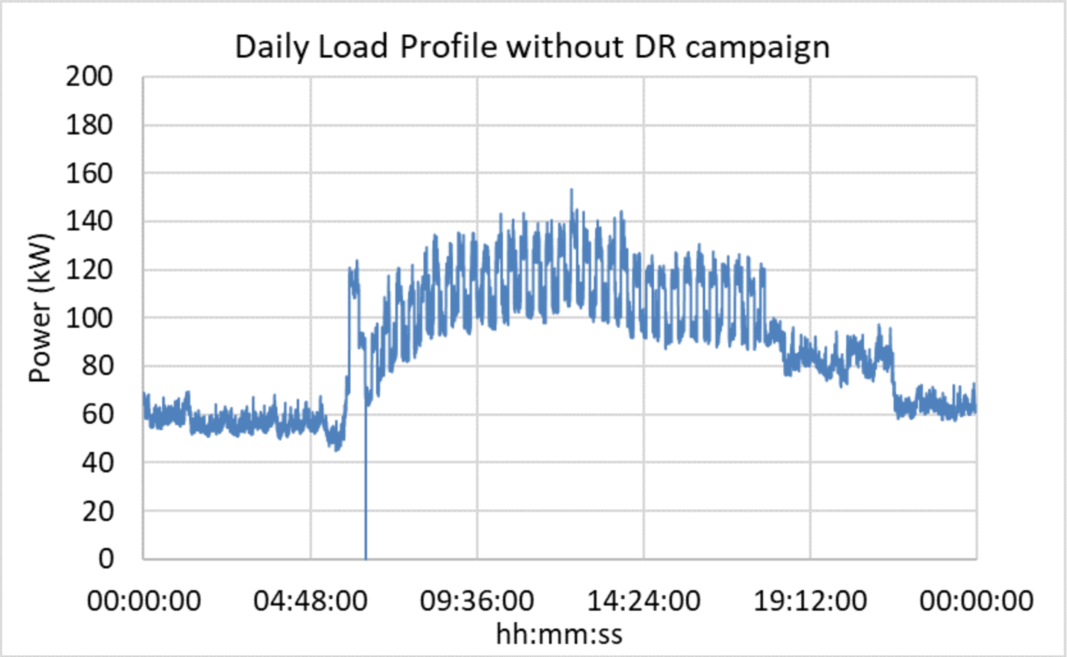


Figure 25. Daily load profile of ASM buildings

- Three smart charging stations (two SpotLink EVO and one Efacec QC45) and six electric vehicles (four Renault ZOE and two Nissan LEAF) will be part of the Terni pilot site. The 22 kWh/40 kWh lithium-ion batteries of Renault Zoe are charged at 22 kW SpotLink EVO charging station while a Efacec QC45 charging station, supplying up to 50 kW DC, is used to charge the 22 kWh lithium-ion batteries of Nissan LEAF.

This energy infrastructure is enhanced by an innovative metering system, made up of new generation smart meters (NORM[17]) able to collect real time data from the most crucial points of the district and transmit the data through the mobile network (GSM). Figure 26 shows the single line diagram of the ASM's living lab in which the optimized Demand Response campaigns will be tested and validated.

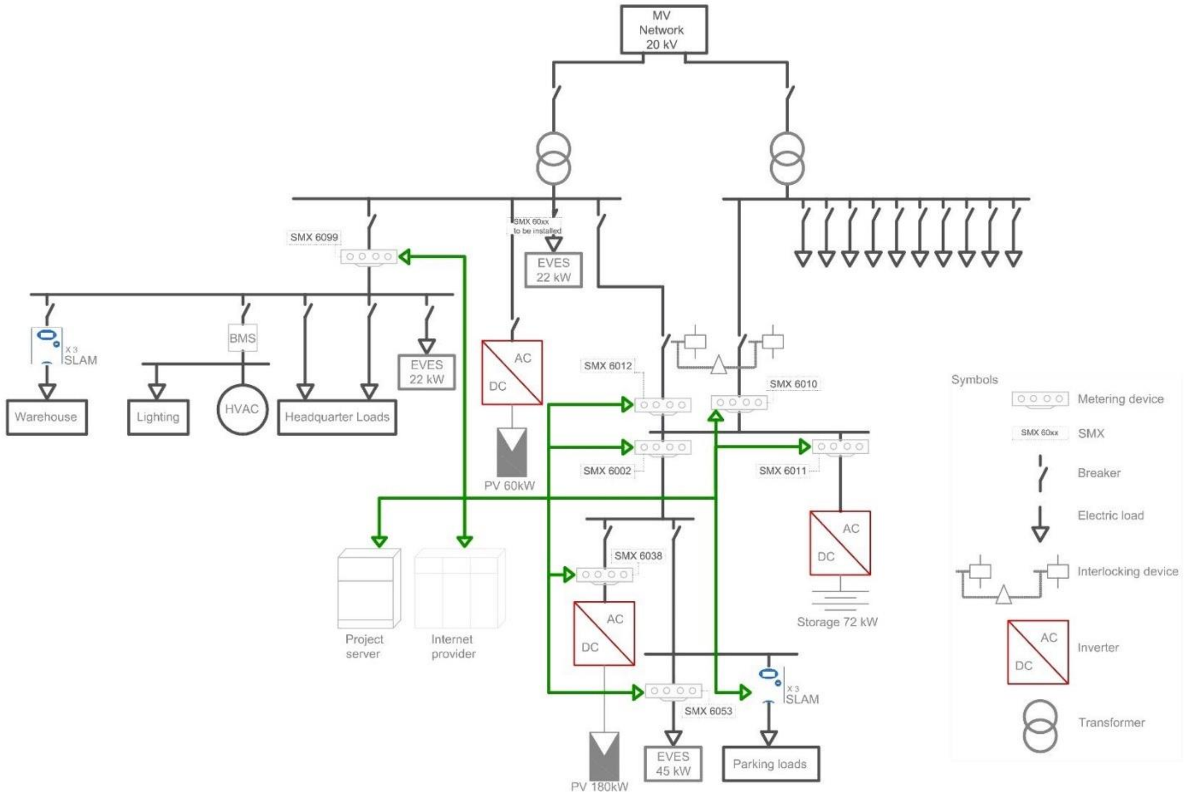


Figure 26. Terni pilot site configuration

It is worth pointing out that the aforementioned living lab is a part of the distribution power network of the city of Terni, which is owned and managed by ASM Terni. This power network is connected to the HV grid through three substations. There are also six MV/MV substations and more than six hundred MV/LV substations, supplying about sixty-five thousand energy customers. Nowadays, the ASM network shows relevant features of the "smart grid" since 99% of customers have a smart meter managed remotely by an Advanced Metering Infrastructure (AMI). Moreover, the high penetration of Renewable Energy Sources (RES) has dramatically changed the paradigm of the network management, creating a reverse flow and congestions where production and consumption are not balanced.

In 2017, the energy consumption reached 350 GWh, while the distributed production units connected to the MV/LV network (DER) generated 182 GWh. Thus, about 50% of the total consumption was covered by RES. In 2017 the local power network received renewable energy from 1228 power generation plants (1217 PV arrays, 7 hydro plants, 4 biomass/waste

[17] https://success-energy.eu/

to energy) using renewable sources, such as sunlight, water and biomass. In 2017 the total electric power generated from RES was as follows, considering only the main contribution (i.e. these data do not consider small plants that produce from other sources a negligible amount of energy):

- 36 GWh from solar energy;
- 68 GWh from hydropower;
- 78 GWh from waste material.

The energy production from hydropower, solar, and waste for each month in 2017 is shown in Figure 27.

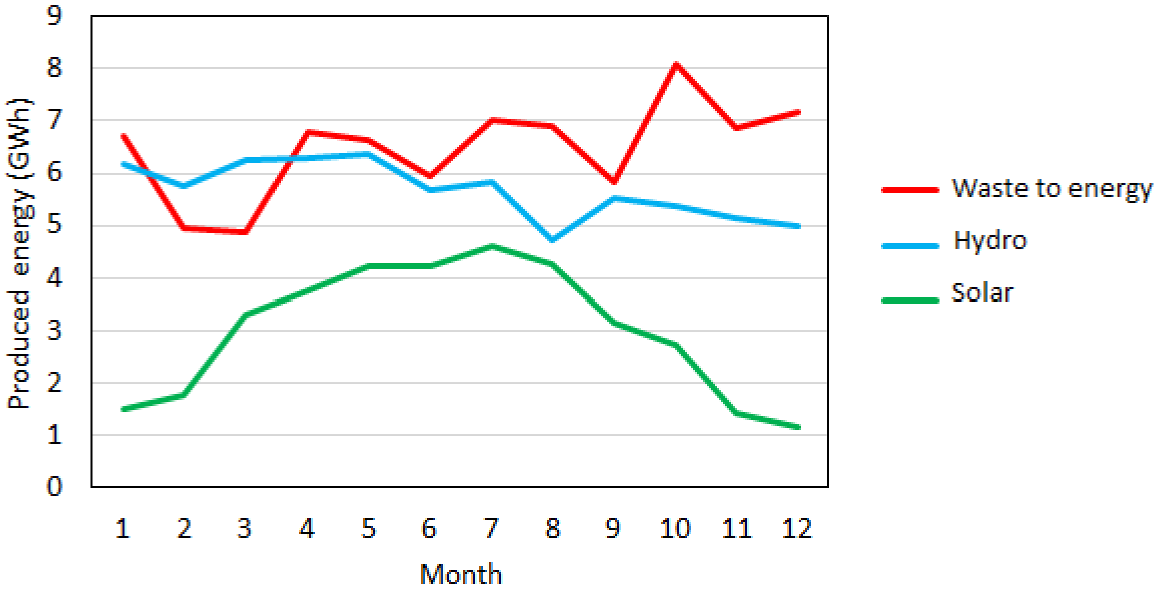


Figure 27. Monthly embedded generation in Terni (2017)

The energy transition requires enabling new services at capillary level by means of real time exchanges of both technical and economic information between new and old actors (e.g. DSOs, Aggregators, microgrid managers, EV fleet managers) and each single user and/or prosumer.

Considering the present status of the distribution network which connects an increasing amount of distributed generators, the DSO is going to assume more responsibility as coordinator of distributed local resources, notably, it is going to acquire observability of the network, in order to provide real-time data to various stakeholders (e.g., Transmission System Operators, market actors, customers). A reliable and secure observability can enable market participation of distributed generators[18] , as well as allow the implementation of flexibility market and peer-to-peer transactions foreseen by Directive[19]. The observability improves by itself the reliability of the Transmission Services that are already based on the load forecasting and production plan carried out by the programmable plants. In addition, real time measurements are a pillar for the deployment of a real time management of the distributed resources carried out by the DSO or other stakeholders (e.g., Aggregator, RESCO

[18] Italian Authority fo Electric Energy (ARERA), 298/2016/R/eel PRIMA FASE DELLA RIFORMA DEL MERCATO PER IL SERVIZIO DI DISPACCIAMENTO: APERTURA ALLA DOMANDA, ALLE FONTI RINNOVABILI NON PROGRAMMABILI E ALLA GENERAZIONE DISTRIBUITA, 2016

[19] DIRECTIVE (EU) 2018/2001 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 11 December 2018 on the promotion of the use of energy from renewable sources

companies). In order to provide these services, the DSO should be capable to manage real-time data as well as information exchange between distributed devices.

5.1.2 SOFIE added value

The interoperability between the IoT platforms of the Fleet Manager and the DSO obtained thanks to SOFIE Federation Framework paves the way for the business model implemented in the decentralized energy flexibility marketplace energy pilot. The rapid user-friendly mechanism to negotiate micro-contracts enabled by the SOFIE blockchain solution allows to perform powerful DR campaigns, not based only on forecast data, but performed in real-time when the DSO identifies the need for flexibility as well.

Thanks to the technology agnostic SOFIE framework, DSO and Fleet Manager can interoperate in the same decentralized marketplace, keeping intact their own internal siloed IoT platforms. The blockchain-enabled marketplace will grant security, transparency and auditability. The Pilot KPIs are defined in Table 18.

Table 18. KPIs of decentralized energy flexibility marketplace pilot

Technology Level	
Enabling hardware deployment	● Devices deployment effort and configuration time ● Compatibility with existing infrastructure
Federating platforms	● Degree of interoperability ● Federate ≥ 3 IoT platforms ● Transaction over ≥ 3 DLTs
Business Level	
DSO metrics	● RPF reduction ● Power losses reduction ● Voltage under the limits ● Green energy consumption
EV fleet manager metrics	● Monetary savings ● Reduction of number of charges monthly carried out ● Optimization of EV routing

5.1.3 Pilot status and timeplan

Currently, the pilot is under the development phase and close to finalizing its first release. At this stage, its status and plan do not deviate from that shown in Table 3. In particular, the two dashboards for the fleet manager and the DSO are being developed using an iterative approach and new functionalities are being added based on feedbacks from the actors involved. The marketplace smart contract and the related APIs have been developed extending the *"offer-marketplace"* business template implementation. The development of the architecture for the real time access to the smart meter data is being developed and inputs from this development activity will help the definition of the SOFIE IoT adapters. In particular, the timeplan for the upcoming activities includes in the next releases:

- the integration of the marketplace APIs with the fleet manager dashboard (ongoing)
- the integration of an external "oracle" to release the token payment after the vehicle charge (ongoing)
- the addition of a new user role, to simulate the retailer role
- full on-site validation

The actual actors involved in the real operational conditions have already been engaged and are part of the pilot environment.

5.2 Pilot scenarios, requirements and system architecture

5.2.1 Scenarios and use cases

The use cases and scenarios of the pilot have been presented in D5.1 “Baseline System and Measurements”. In this section, only the updates with respect to that version are presented.

5.2.1.1 Decentralized energy flexibility marketplace pilot scenarios

The pilot considers two scenarios, namely “Pull Offers” and “Push Offers”. Both of them have been presented in detail in §3.3.3 in D5.1 “Baseline System and Measurements”.

5.2.1.2 Decentralized energy flexibility marketplace pilot use cases

The UML use case diagram in Figure 28 illustrates how the various actors interact with the decentralized energy flexibility marketplace established by the pilot.

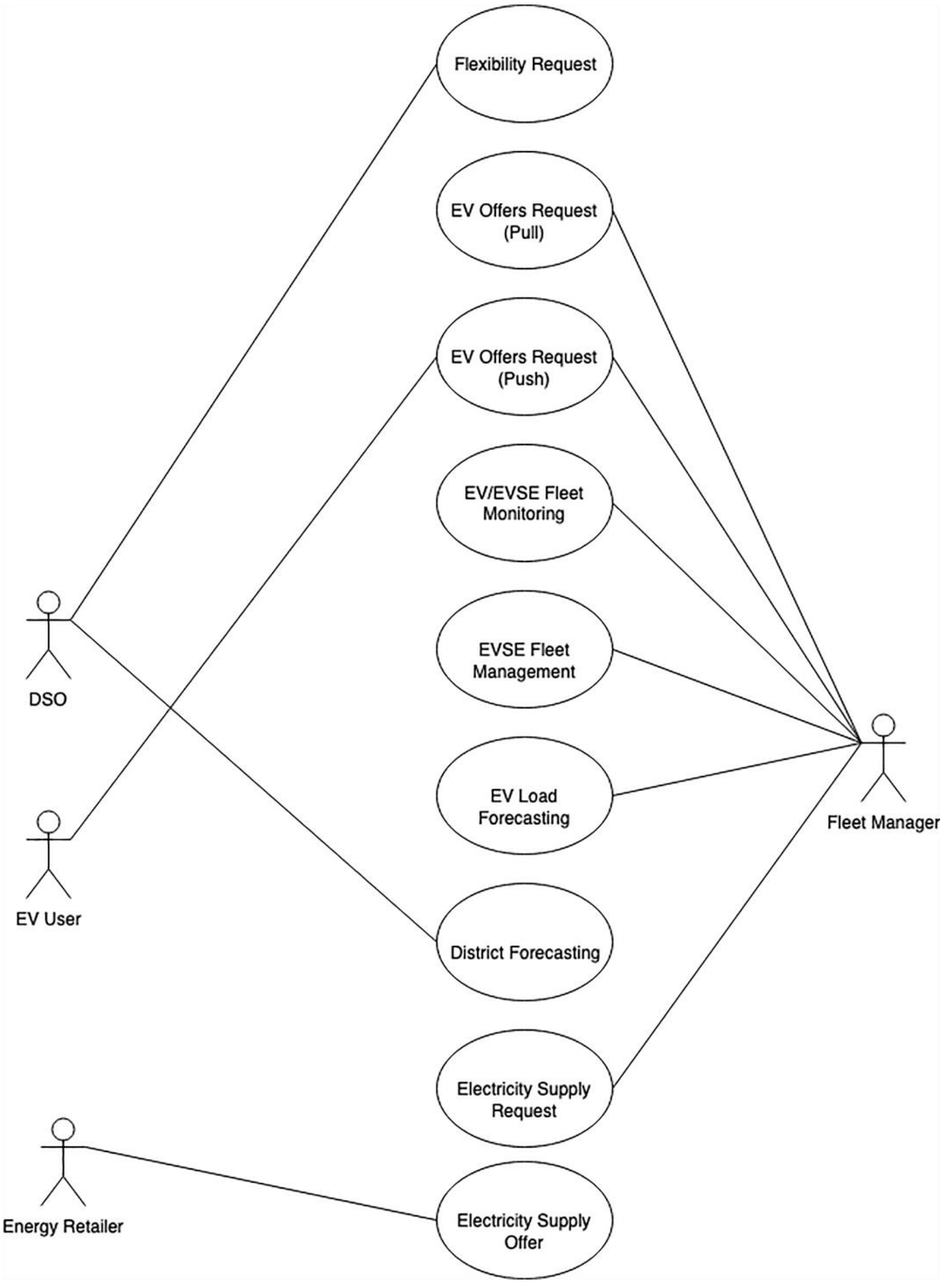


Figure 28. Actors’ interaction with the decentralized energy flexibility marketplace

Most of the use cases in Figure 28 have been already presented in §3.3.4 of “D5.1 – Baseline System and Measurements”. Table 19 summarizes them and provides the description of DEFM_UC9 which is added with respect to those introduced in D5.1.

Table 19: Use cases in the decentralized energy flexibility marketplace

ID	Name	Actors
DEFM_UC1	Flexibility Request	DSO
DEFM _UC2	EV Offers Request (Pull)	Fleet Manager
DEFM _UC3	EV Offers Request (Push)	Fleet Manager, EV User
DEFM _UC4	EV/EVSE Fleet Monitoring	Fleet Manager
DEFM _UC5	EVSE Fleet Management	Fleet Manager
DEFM _UC6	EV Load Forecasting	Fleet Manager
DEFM _UC7	District Forecasting	DSO
DEFM _UC8	Electricity Supply Request	Fleet Manager
DEFM _UC9	Electricity Supply Offer	Energy Retailer
USE CASE Description		
ID	DEFM_UC9	
Name	Electricity Supply Offer	
Actors	Energy Retailer	
Storyline	When the Fleet Manager accepts the flexibility requests available in the flexibility marketplace, he will request an electricity supply to energy retailers. The energy retailer that offers the electricity supply at the lowest price signs a micro contract with the Fleet Manager.	
Trigger events	Fleet Manager is requesting an electricity supply to satisfy DR campaign needs	
Preconditions	The Energy Retailer system is connected with the Fleet Manager system	
Postcoditions	The economic transaction from Fleet Manager to Energy Retailer is performed	
Related scenarios	Energy_Pilot_Scenario_1, Energy_Pilot_Scenario_2	

5.2.2 Requirements and software architecture

5.2.2.1 Domain requirements

In Table 20, the pilot domain and end-users requirements are summarized as those are identified by collecting feedback from engaged stakeholders.

Table 20. List of requirements in the decentralized energy flexibility marketplace pilot

ID	Name	Description	Priority	Related use case
REQ_ DEFM1.1	DR strategies assessment	DSO shall be able to forecast of electricity production/consumption	MUST	DEFM_UC1

REQ_ DEFM1.2	Checking load and production forecast	DSO shall be able to check the load and production forecasting of the whole distribution grid	MUST	DEFM_UC1
REQ_ DEFM1.3	Grid System flexibility DR services	DSO shall be able to forecast of electricity production / consumption at the grid level	MUST	DEFM_UC1
REQ_ DEFM1.4	DSO foresees and provides flexibility	DSO shall be able to shave picks of energy produced locally the day after so that instability of the system, overvoltage on the feeder, protection discoordination, increased fault currents, and incorrect operation of equipment could be avoided	MUST	DEFM_UC1
REQ_ DEFM1.5	Flexibility estimation	DSO shall be able to estimate the energy flexibility availability; Assess flexibility availability by using available historical data.	MUST	DEFM_UC1
REQ_ DEFM1.6	Flexibility Request	DSO shall be able to forecast system indicates a potential reverse powerflow to be mitigated and DSO system is connected with the flexibility marketplace. The DSO system is connected with the flexibility marketplace	MUST	DEFM_UC1
REQ_ DEFM2.1	DSO/Fleet Manager Micro-Contract	When the Fleet Manager obtains the responsibility to provide the flexibility required by the DSO, a micro contract between the the Fleet Manager and the DSO is executed	MUST	DEFM _UC2 DEFM _UC3
REQ_ DEFM2.2	Fleet Manager/ EV User Micro-Contract	When the Fleet Manager obtains the responsibility to provide the flexibility required by the DSO and EV users not belonging to the fleet manager EV fleet are involved in the DR campaign, a micro contract between the Fleet Manager and the EV user is executed	MUST	DEFM _UC2 DEFM _UC3
REQ_ DEFM4.1	EV/EVSE Systems Interoperability	With the objective of performing Demand Response (DR) campaigns, it is necessary that the management systems of electric vehicles and charging stations communicate with each other, so that it is possible to verify in real time the interaction between the two systems.	MUST	DEFM _UC4
REQ_ DEFM4.2	EV/EVSE Data Collection	To provide DSO flexibility in an efficient way, the data of electric vehicles and charging stations must be collected in real time (or very close to real time). Data coming from EVSEs and the EVs	MUST	DEFM _UC4 DEFM _UC6

		should be consistent, reliable, transparent and accessible to the partners. Furthermore, to perform optimized DR campaign it is necessary to constantly calculate EV load forecasting to estimate the amount of energy that electric vehicles could consume to meet the DSO's flexibility demand.		
REQ_ DEFM4.3	EV/EVSE Data Storage	It is necessary that the data of electric vehicles and charging stations are stored so that they can then be reprocessed, giving fruit to charts that show the effectiveness for the purposes of the DSO of DR campaigns performed during the trial.	MUST	DEFM _UC4
REQ_ DEFM4.4	EVSE Unique Identifier	As there will be more than one charging station on the pilot site, each individual charging station must have its own unique identifier.	MUST	DEFM _UC4
REQ_ DEFM4.5	EV Unique Identifier	As there will be more than one electric vehicle on the pilot site, each individual electric vehicle must have its own unique identifier.	MUST	DEFM _UC4
REQ_ DEFM4.6	EV/EVSE Web Platform	To allow the EV user to realize the available charging stations and the fees associated with them, a web platform is required.	MUST	DEFM _UC4
REQ_ DEFM4.7	EV/EVSE Connectivity	Both charging stations and electric vehicles must be connected to the internet in order to send data.	MUST	DEFM _UC4
REQ_ DEFM5.1	EVSE Remote Control	The charging station must be remotely controlled to start/stop charging sessions and to modulate the power output.	MUST	DEFM _UC5
REQ_ DEFM7.1	District forecasting	DSO shall be able to have to constantly calculate building consumption forecasting, PV production forecasting and manage batteries to estimate the amount of energy demand at ASM substation. Forecasting will be calculated periodically (every day). Need to reduce undesired reverse power flows	MUST	DEFM _UC7
REQ_ DEFM8.1	Fleet Manager/ Retailer Micro-Contract	When the Fleet Manager obtains the responsibility to provide the flexibility required by the DSO, a micro contract between the the Fleet Manager and the Retailer is executed for the energy supply to	MUST	DEFM _UC8 DEFM _UC9

		charge electric vehicles		

5.2.2.2 Pilot security and privacy specifications

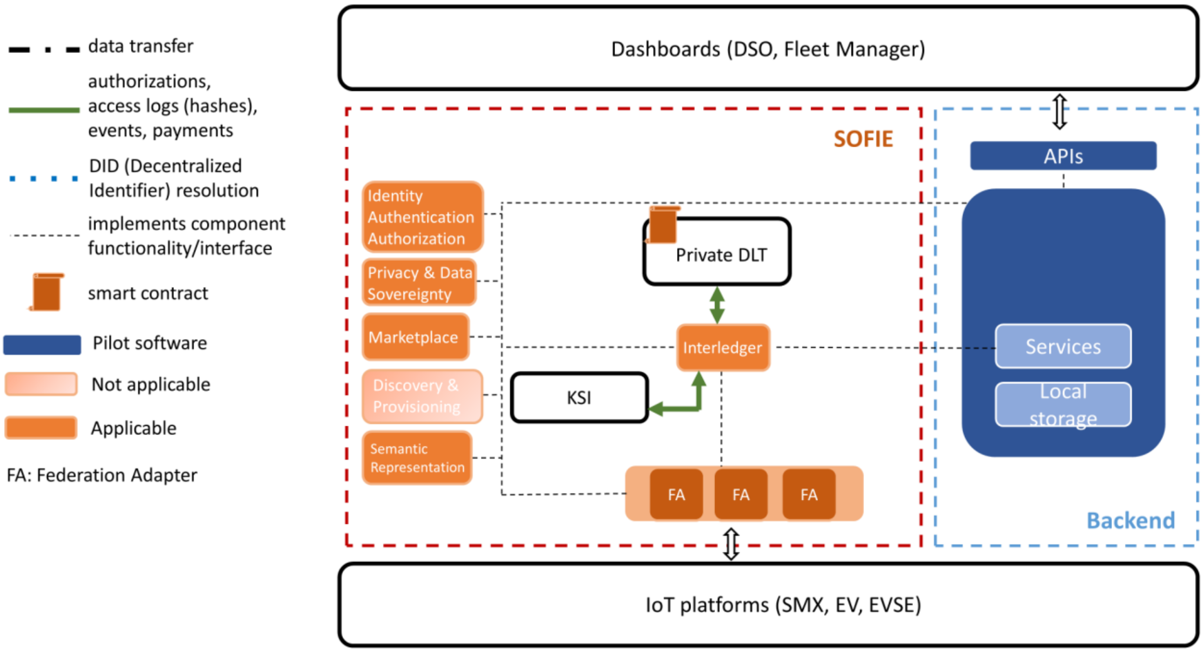
Table 21 presents how the Decentralized energy flexibility marketplace pilot aims to address the three important security and privacy challenges that have been identified in SOFIE.

Table 21. Security and privacy specifications in the decentralized energy flexibility marketplace pilot

Support for transactions, where only authorised entities can participate
Blockchains like Ethereum use asymmetric cryptography to ensure that only the user in possess of its own private key can make transactions on its own behalf. In addition to this, the smart contract governing the decentralized marketplace, defines specific roles (like *owner* or *administrator*) authorised to interact with specific functions.
Transactions must be authentic and verifiable
The decentralized marketplace is built on top of Ethereum blockchain. One of the key features provided by blockchains is the non-reputability of transactions. This, together with the asymmetric cryptography behind blockchain transactions, will grant that each marketplace action can be verified and its authenticity can be proved. In addition, the parameters of each marketplace actions are stored in the smart contract for further transparency and auditability.
Privacy issues and business secrets must be considered carefully when deciding what data (including authentication/authorization information, logs etc.) is collected, stored or exchanged between parties
If the private key of the smart contract owner becomes public for some reason, the smart contract is compromised. For example, the requests flexibility can be closed or evaluated before the established deadline. Also, the marketplace users private key must be kept secret as the key can be used to make unauthorized request.

5.2.2.3 Pilot system architecture

Figure 29 illustrates the system architecture of the decentralized energy flexibility marketplace pilot. The core functionality of the marketplace will be granted by the marketplace smart contract, running on a private Ethereum blockchain, while the Interledger component will be in charge of synchronizing the status of the internal blockchain with KSI ledger.



data transfer
authorizations, access logs (hashes), events, payments
DID (Decentralized Identifier) resolution
implements component functionality/interface
smart contract
Pilot software
Not applicable
Applicable
FA: Federation Adapter
Dashboards (DSO, Fleet Manager)
SOFIE
Identity Authentication Authorization
Privacy & Data Sovereignty
Marketplace
Discovery & Provisioning
Semantic Representation
Private DLT
Interledger
KSI
FA
FA
FA
APIs
Services
Local storage
Backend
IoT platforms (SMX, EV, EVSE)

Figure 29. Overview of the system architecture for the decentralized flexibility energy marketplace pilot

5.2.2.3.1 DSO dashboard

The functional overview of the DSO Dashboard is shown in Figure 30.

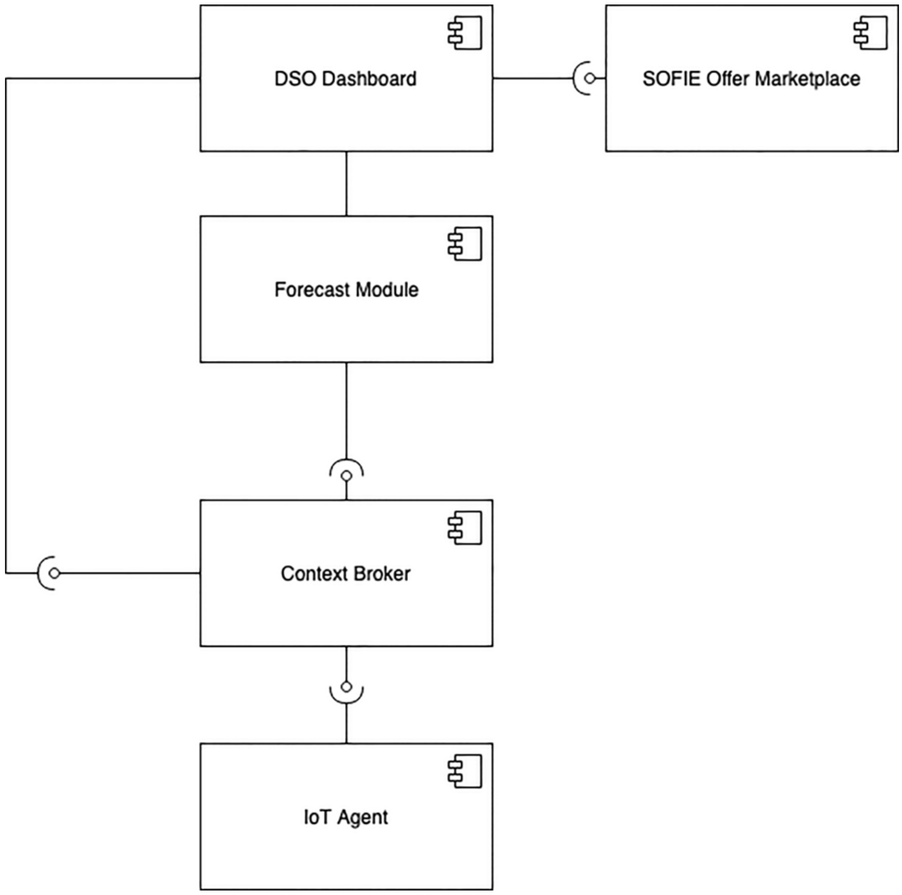


Figure 30. Overview of the DSO dashboard

The role of each structural component is as follows:

DSO Dashboard: This is the interface that the end user (DSO operator) will utilize, to analyze the need for flexibility requests and place the market offers accordingly. It graphically shows the current near real-time data from smart meters against the forecast for the next day. It also includes a request form for placing the flexibility requests in the marketplace.

Forecast Module: This component is used to calculate the forecast chart using the historical data collected by the smart meters.

Context Broker: This holds the context data of the IoT subsystem. It receives NGSI requests. It enables subscriptions from other components such as persistence layers.

IoT Agent: This acts as a bridge between JSON over MQTT messaging and NGSI.

Sofie Offer Marketplace: APIs for the actual SOFIE offer marketplace, enabling the DSO to place marketplace requests using the marketplace smart contract.

5.2.2.3.2 Fleet manager dashboard

The functional overview of the DSO Dashboard is shown in Figure 31.

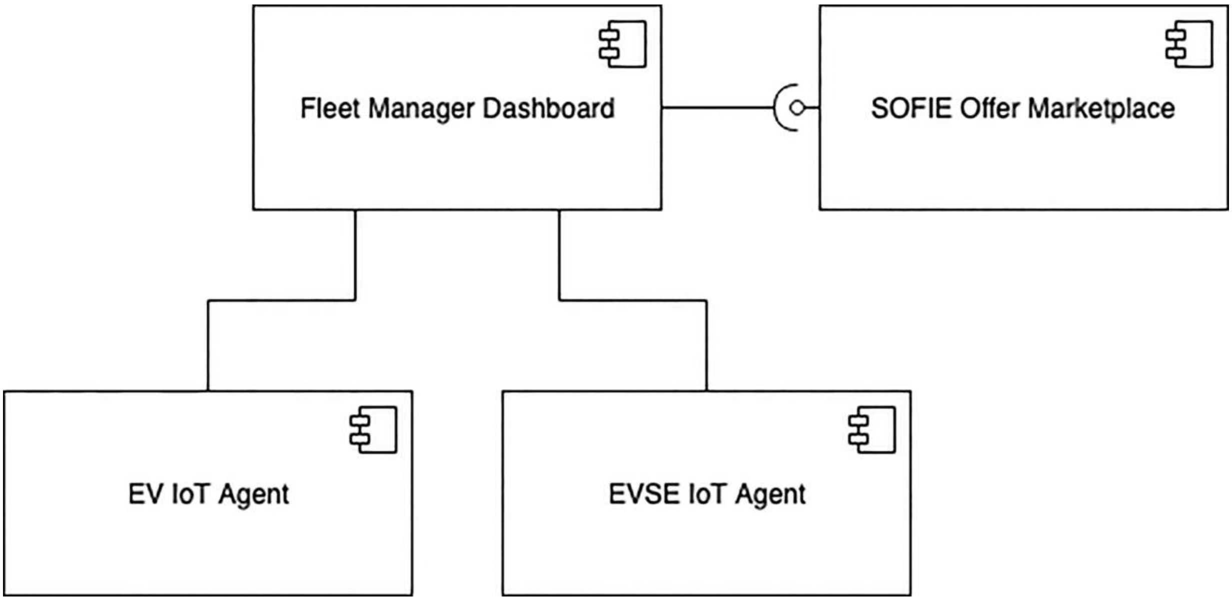


Figure 31. Overview of the fleet manager dashboard

The role of each structural component is as follows:

Fleet Manager Dashboard: This is the interface that the end user (Fleet Manager operator) will utilize, to list the active flexibility requests and place its marketplace offers according to the fleet needs and current status. It shows the geographical position of the charging stations and vehicles, and the current status for both.

EV IoT Agent: Enables the interaction with the Electric Vehicle IoT devices

EVSE IoT Agent: Enables the interaction with the Electric Vehicle Supply Equipment devices

Sofie Offer Marketplace: APIs for the actual SOFIE offer marketplace, enabling the Fleet Manager to list the active marketplace requests and to place marketplace offers using the marketplace smart contract.

5.2.2.3.3 Sequence of interaction for pilot use cases

The two sequence diagrams that follow show the interaction between architectural components and actors with respect to marketplace requests and marketplace offers.

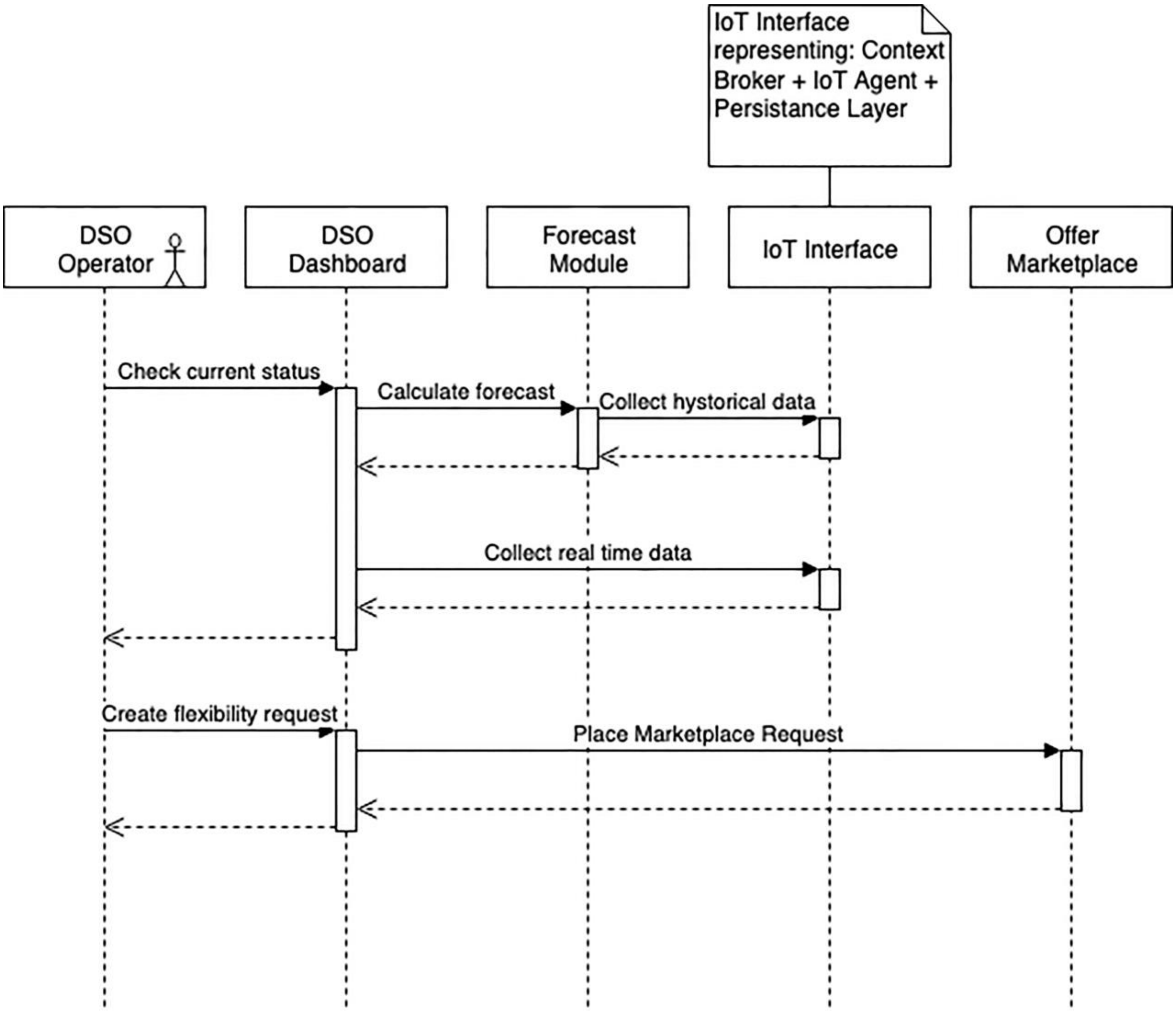


Figure 32. Create marketplace requests

Figure 32 shows the necessary steps to create a new market request. In particular, the DSO Operator, using the dedicated dashboard, checks the current status of the grid and the day ahead forecast. Context information in real time is provided by the context broker, while the forecast module uses a dedicated database to persist historical data used to calculate the forecast. The dashboard provides the forecast and the real time data interrogating the API of the forecast module and IoT subsystem. If the forecast shows the need for a rebalance in consumption for a specific grid, then the DSO operator, using the same dashboard, can place a market request on the decentralized marketplace.

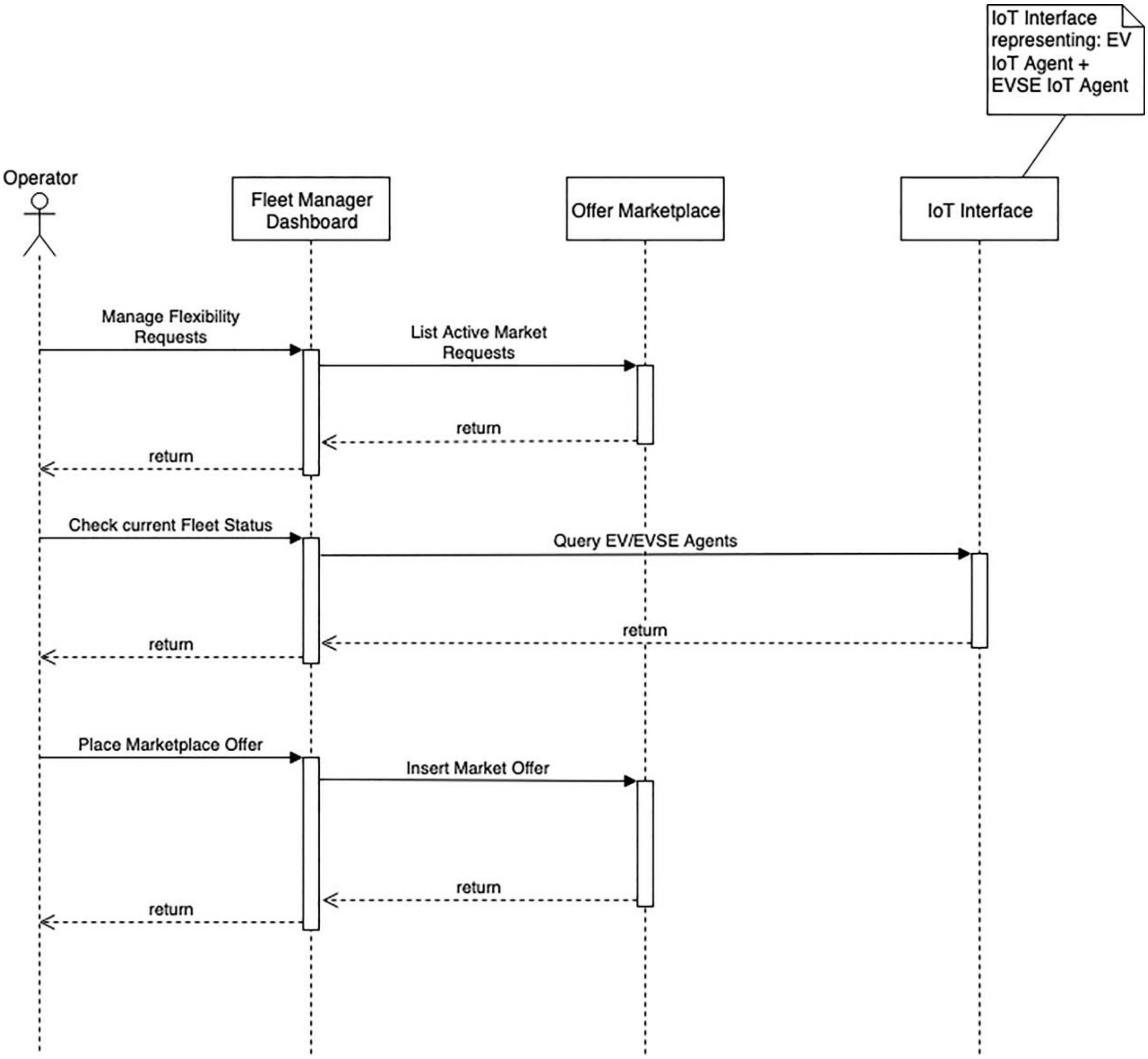


Figure 33. Seeking marketplace offers

As shown in Figure 33, in the case of a marketplace offer, the fleet manager interrogates the decentralized marketplace, using the dedicated dashboard, looking for active market requests. When a new request is received, the fleet manager according to the current status of the fleet (EVs position, charging status, etc.) decides to participate in the campaign or not. If the fleet manager participates, then he can place a market offer matching the open request.

5.3 Initial SOFIE validation

5.3.1 Installed infrastructure

Renault ZOE, acronym of ZerO Emissions, is a five-door supermini electric car produced by the French manufacturer Renault, Figure 34. The ZOE is powered by a 22 kWh lithium-ion battery pack weighing 275 kg, driving a 65 kW (87 bhp; 88 PS) synchronous electric motor supplied by Continental (the Q210). Maximum torque is 220 N·m (162 lb-ft) with a top speed of 135 km/h (84 mph). The NEDC cycle range is 210 km (130 mi). Renault estimates that in suburban use, the ZOE can achieve around 100 km (62 mi) in cold weather and 150 km (93 mi) in temperate conditions. The car features a charging system called "Caméléon" (Chameleon) charger that allows the ZOE to be charged at any level of power, taking between 30 minutes and nine hours (RENAULT, 2018).

Figure 34. Renault ZOE (left) and NISSAN Leaf (right) customized by EMOT

The Nissan LEAF, acronym of Leading, Environmentally Friendly, Affordable, Family car, is an electric propulsion car introduced by Nissan on the markets in December 2010. It is equipped with an 80 kW (109 hp) synchronous AC electric motor. The first version equipped a lithium-ion battery, consisting of 48 modules and each of them contains 4 cells for a total of 192, with a capacity of 24 kWh with an autonomy of 199 km NEDC cycle. Since 2016, a 30 kWh Lithium-ion battery with an operating cycle of 250 km NEDC is available as an accessory. Nissan LEAF recharges in alternating current or in direct current. In AC, LEAF uses an on-board charger with a maximum 7.4 kW (32A maximum current, 230V, single-phase) with the Type 2 socket. In DC it uses the CHAdeMO standard up to 50 kW of power. Charging times vary from 5/6 hours to about 7 kW up to 1 hour with direct current charging (NISSAN, 2018).

Three smart charging stations will be involved in the decentralized energy flexibility marketplace, two 22 kW charging stations (SpotLink EVO) and one 50 kW charging station (Efacec QC45), Figure 35. The SpotLink EVO charging station with one or two type 2 sockets recharges up to 32 A single-phase or three-phase for each socket; it is equipped with a 7” touch screen panel to manage recharges and offers many features, such as the location and navigation to the charging station (through the dedicated App) and the recharge with coupon code or credit card directly from the App. It allows the operator to remotely monitor and manage all the functions of the charging station such as recharge powers, recharge prices, statistics and reporting. SpotLink EVO works with plug Type 2 or Type 3A, its nominal voltage is 230VAC (AC Voltage) in mono phase or 400 VAC in three phases, its nominal current is 32A and its nominal frequency is 50 Hz. SpotLink EVO protection grade is IP54[20], the impact resistance is IK08[21] and the protection system is differential type A and type B, with an automatic unlocking of the connector in case of power failure. It equips a single-board computer, a certified Measuring Instruments Directive energy meter and a RFID reader. SpotLink EVO connectivity is through RJ45 port (LAN) or 3G modem.

The QC45 quick charging station provides a rapid battery charge and supports two EVs simultaneous AC and DC charging with multiple power output options. The QC45 is a flexible and open charging station able to charge in a standalone mode or integrated in any network with any central system. The QC45 has DC output with power up to 50 kW and optional AC output with power up to 43 kVA. The battery charging status is displayed in a TFT color screen. The QC45 has high quality and robust enclosure with corrosion protection, equivalent to stainless steel, to ensure extended equipment lifetime[22].

[20] meaning protection from dust and splashing water according to the International Protection marking.
[21] meaning resistance to impacts of 5 joule of kinetic energy.
[22] EFACEC 2016.

Figure 35. Emotion SpotLink EVO charging stations (left) and Efacec QC45 (right)

EMOT will use one server machine for the SOFIE demonstration, following the server details:

- CPU: 2 core 3.1 GHz;
- HDD: 50 GB;
- RAM: 8 GB;
- S.O.: Ubuntu 17.10.

EMOT charging stations will exchange data through their single-board computer, a Raspberry Pi 3, with a CPU of quad-core ARM Cortex A53 1.2 GHz, a SD of 16 GB, a RAM of 1 GB and a Raspbian Stretch 4.14 S.O.; EVSE data collected will be mainly the energy data (power, voltage, current) and the number of plugs in use (0/1/2). Regarding EV monitoring, EMOT will use an OBD device to retrieve data from the EV; OBD is an IoT component that utilize a TCP/IP communication to a TCP/IP server. The network connectivity of the OBD device is via data SIM (UMTS) and the server is a python software, which queries the EV each 5 seconds. The OBD connects to the diagnostic interface from which it is able to extract the information from the electric vehicle control unit using the CAN-bus protocol. The output data format of the OBD is an ASCII string; when the data is sent to the server, it is reorganized into a wrapper, thus obtaining a grouping of the data in JSON format. From the EV will be retrieved the following data: battery state-of-charge, geolocation, doors car state and engine car state.

The two PV arrays shown in Figure 36 will be part of the SOFIE demonstration. The two arrays, of the production pattern which is also shown in Figure 36 (180Kwp and 60KWp, respectively), are connected to the LV network of the Terni city

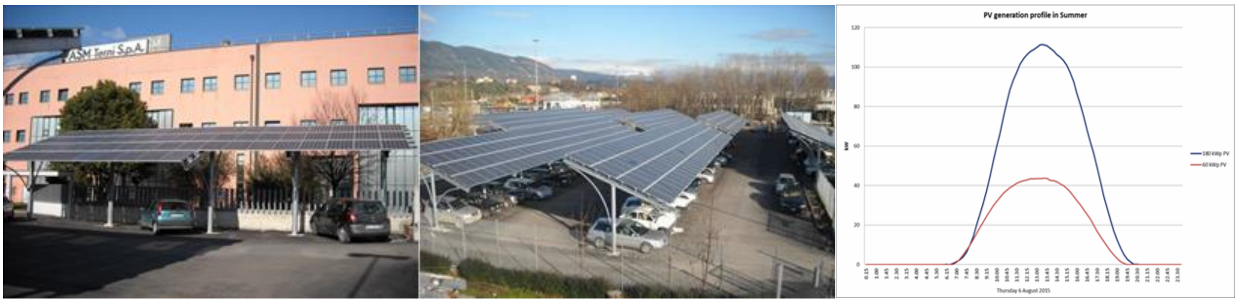


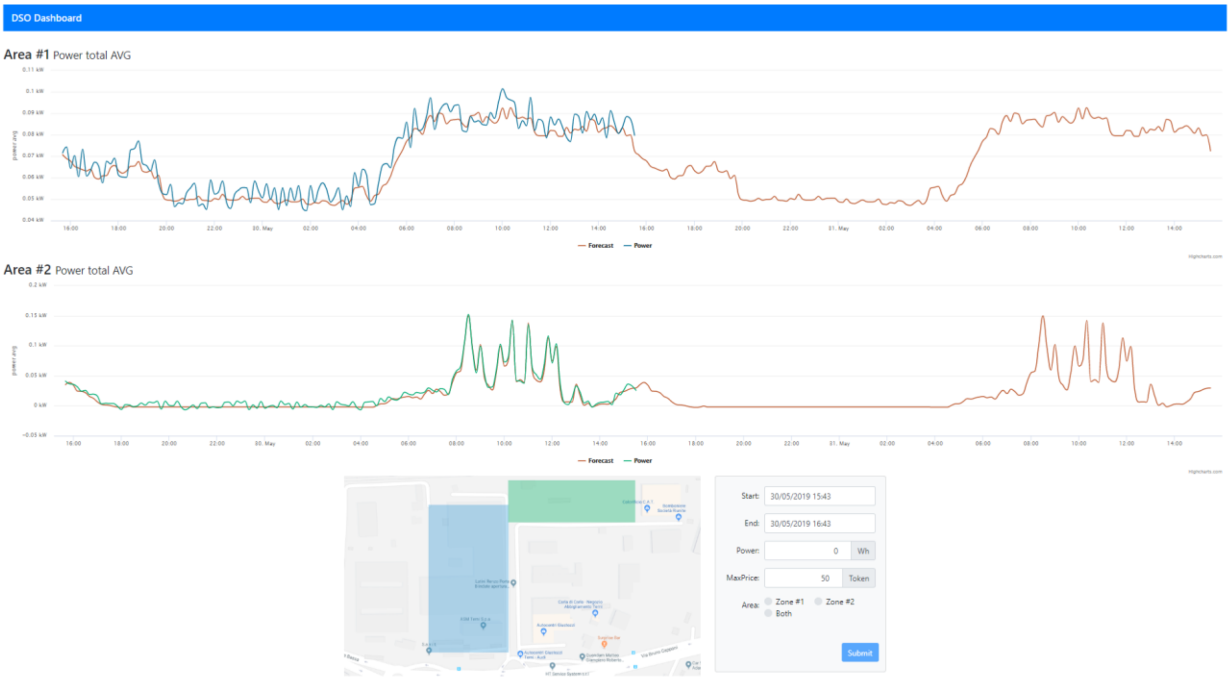
Figure 36. PV plants at the Terni trial site (left) and their production pattern (right)

5.3.2 Proof of concept prototype

The Proof of Concept (PoC) prototype will validate the decentralized marketplace implementation, more specifically in the context of an energy flexibility marketplace and the usage of the SOFIE platform to federate different siloed IoT platforms. It consists of two separate dashboards, connected to the decentralized marketplace. DSO and Fleet Manager operators can place requests and offers in the marketplace using them.

5.3.2.1 DSO Dashboard

The DSO dashboard, Figure 37, shows the current data from the smart meters together with the forecast for the next day. In the same page, a form enables the operator to interact with the marketplace, placing its flexibility requests.



Start: 30/05/2019 15:43

End: 30/05/2019 16:43

Power: 0 Wh

MaxPrice: 50 Token

Area: ○ Zone #1 ○ Zone #2 ○ Both

Figure 37. DSO dashboard showing forecast, smart meters data and the request creation form.

The same dashboard also presents a list of the DR campaigns created using the marketplace. For each campaign, the list shows:

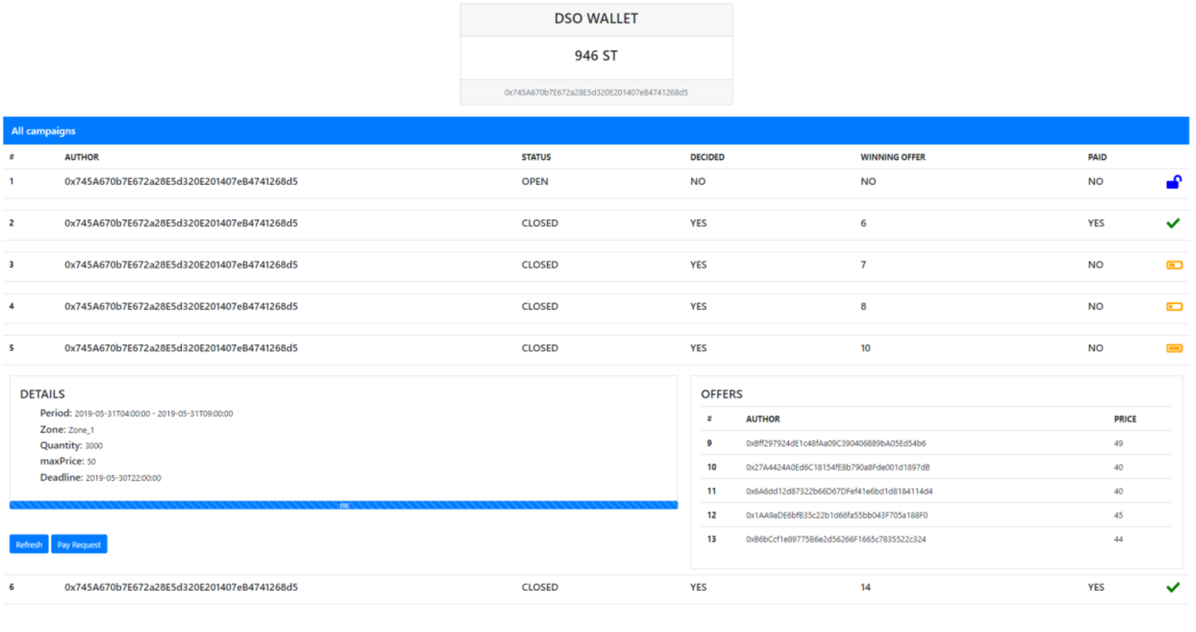
- Status: a boolean (OPEN/CLOSED) indicator, showing if the marketplace request is still open (i.e. it's possible to insert new offers).
- Decided: a boolean (YES/NO) indicator showing if, after the request was closed, the winning offer was already selected.
- Winning Offer: the identifier of the winning offer.

- Paid: a boolean (YES/NO) indicator showing if the token payment was already unlocked.
- A graphical representation, summarizing all of the above

Selecting one item from the list, a detailed view shows:

- request details: a summary of the main attributes of the request,
- offers: the list of participating offers, showing the author and the price,
- a progress indicator: showing (during a *charging* event) the quantity of energy being recharged vs the agreed quantity

The same interface shows the current token balance for the DSO, as shown in Figure 38.



DSO WALLET

946 ST

0x745A670b7E672a28E5d320E201407eB4741268d5

All campaigns

#	AUTHOR	STATUS	DECIDED	WINNING OFFER	PAID	
1	0x745A670b7E672a28E5d320E201407eB4741268d5	OPEN	NO	NO	NO	🔓
2	0x745A670b7E672a28E5d320E201407eB4741268d5	CLOSED	YES	6	YES	✓
3	0x745A670b7E672a28E5d320E201407eB4741268d5	CLOSED	YES	7	NO	▭
4	0x745A670b7E672a28E5d320E201407eB4741268d5	CLOSED	YES	8	NO	▭
5	0x745A670b7E672a28E5d320E201407eB4741268d5	CLOSED	YES	10	NO	▭
6	0x745A670b7E672a28E5d320E201407eB4741268d5	CLOSED	YES	14	YES	✓

DETAILS

Period: 2019-05-31T04:00:00 - 2019-05-31T09:00:00
Zone: Zone_1
Quantity: 3000
maxPrice: 50
Deadline: 2019-05-30T22:00:00

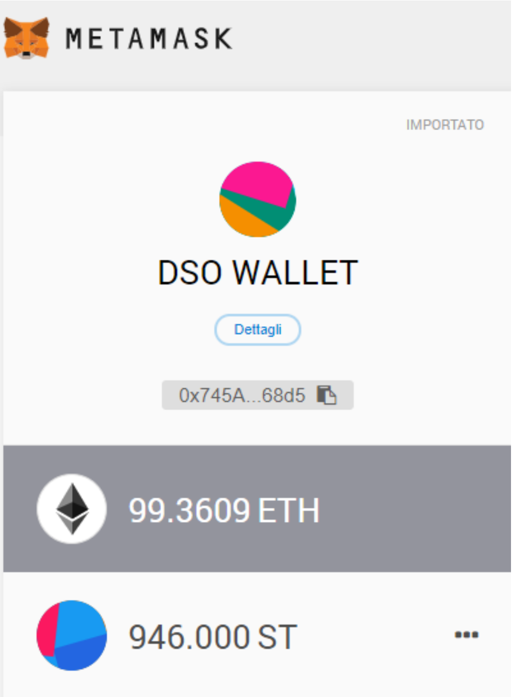
OFFERS

#	AUTHOR	PRICE
9	0x8ff297924dE1c48fAa09C390406889bA05Ed54b6	49
10	0x27A4424A0Ed6C18154fE8b790a8Fde001d1897dB	40
11	0x6A6dd12d87322b66D67DFef41e6bd1d8184114d4	40
12	0x1AA9aDE6bfB35c22b1d66fa55bb043F705a188F0	45
13	0x86bCcf1e89775B6e2d56266F1665c7835522c324	44

Figure 38. DSO dashboard showing the DR campaigns list and token balance

The smart contract managing the decentralized marketplace, implements a standard ERC20 contract interface[23]. This means that the "*Tokens*" which are used as incentives for the flexibility can be managed using a standard Ethereum wallet. Figure 39 shows for example the connection of a third-party wallet (Metamask).

[23] https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md



0x745A...68d5

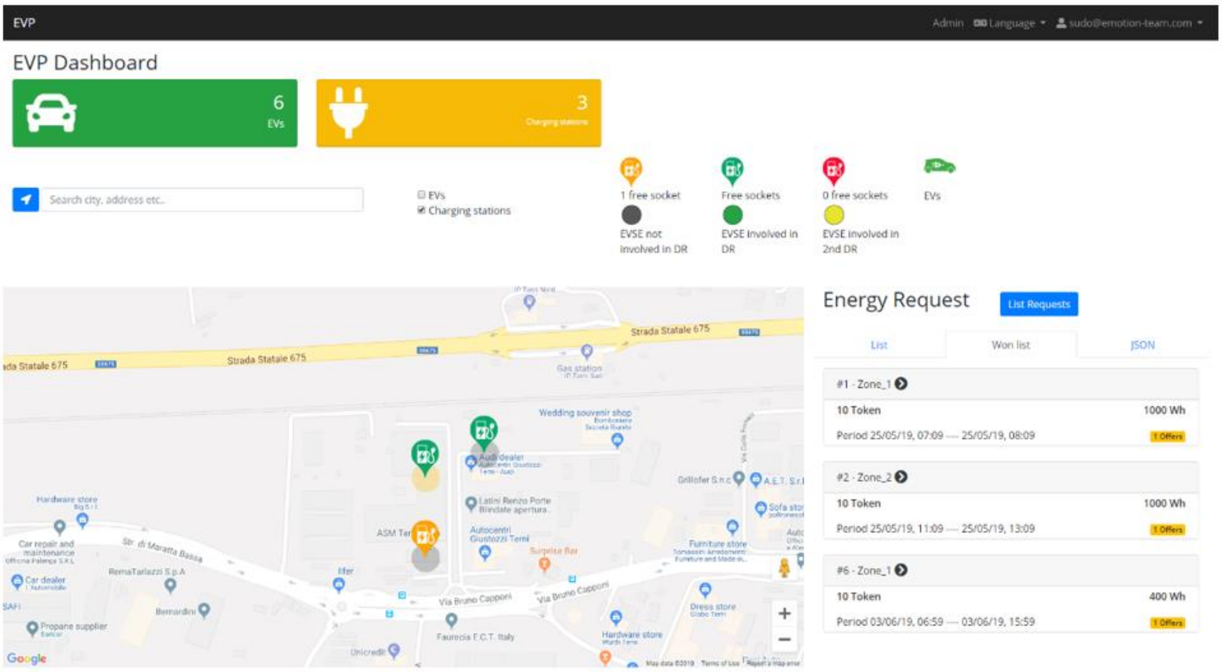
99.3609 ETH

946.000 ST

Figure 39. Metamask showing the token balance of the DSO.

5.3.2.2 Fleet Manager Dashboard

The Fleet Manager dashboard, presented below, shows the real-time data collected from the electric vehicles and charging stations; in the same page, DSO energy requests are shown and a form enables the operator to interact with the marketplace, clicking on a request and placing its flexibility offers, as shown in Figure 40.



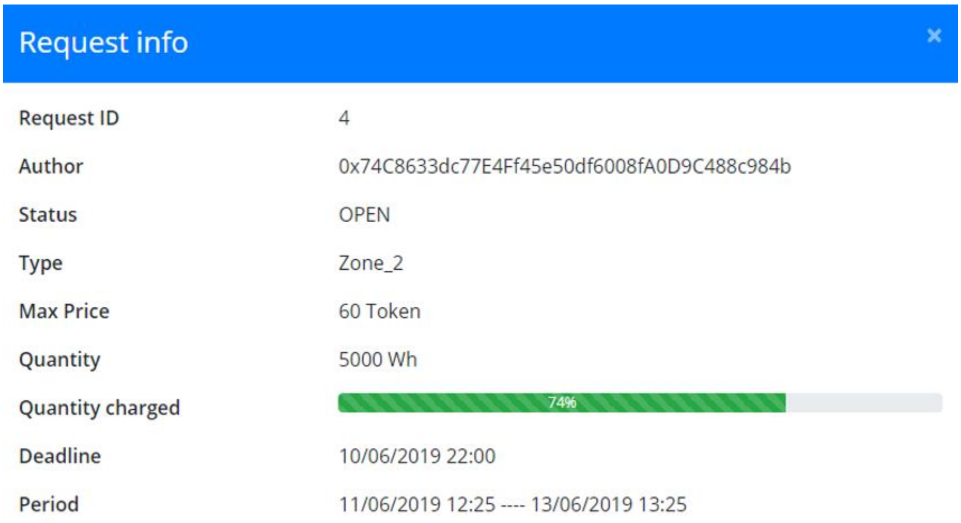
#1 - Zone_1
10 Token 1000 Wh
Period 25/05/19, 07:09 — 25/05/19, 08:09 1 Offers

#2 - Zone_2
10 Token 1000 Wh
Period 25/05/19, 11:09 — 25/05/19, 13:09 1 Offers

#6 - Zone_1
10 Token 400 Wh
Period 03/06/19, 06:59 — 03/06/19, 15:59 1 Offers

Figure 40. Flexibility Offer API

Once the flexibility request has been granted, the Fleet Manager can monitor the progress of the DR campaign directly from the dashboard, as shown in Figure 41.



Request info

Request ID	4
Author	0x74C8633dc77E4Ff45e50df6008fA0D9C488c984b
Status	OPEN
Type	Zone_2
Max Price	60 Token
Quantity	5000 Wh
Quantity charged	74%
Deadline	10/06/2019 22:00
Period	11/06/2019 12:25 ---- 13/06/2019 13:25

Figure 41. Details of DR campaign

When the energy request has been fulfilled, the transaction is enabled and the reward is transferred to the Fleet Manager wallet, as shown in Figure 42.

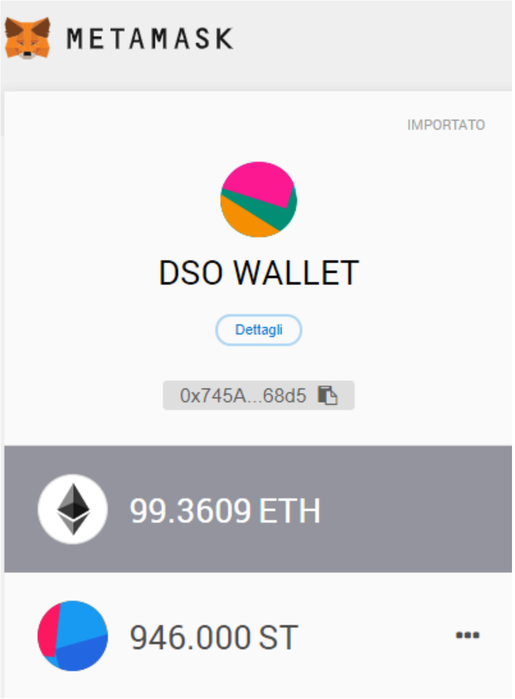


Figure 42. Metamask showing the token balance of the DSO

5.3.3 Validation results

For the initial validation of pilot scenarios, the following data have been used and collected:

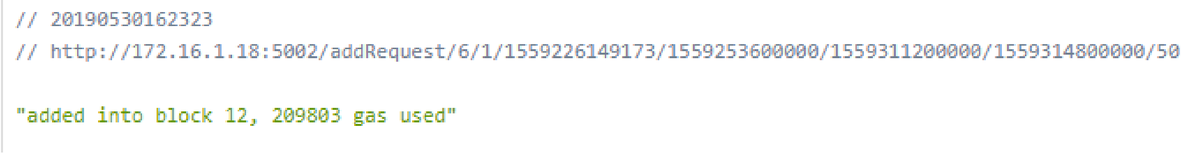
- PV data (production);
- Building data (consumption);
- EV data (state of charge and geolocation);
- Charging Station data (consumption).

The following table shows the use cases of the pilot which have been evaluated by the PoC prototype. For each of the validated use cases, the following subsections give an example of the tested services and related data.

UC ID	Name	Status
DEFM_UC1	Flexibility Request	Tested
DEFM_UC2	EV Offers Request (Pull)	Tested
DEFM_UC3	EV Offers Request (Push)	Not Tested Yet
DEFM_UC4	EV/EVSE Fleet Monitoring	Tested
DEFM_UC5	EVSE Fleet Management	Tested
DEFM_UC6	EV Load Forecasting	Not Tested Yet
DEFM_UC7	District Forecasting	Tested
DEFM_UC8	Electricity Supply Request	Not Tested Yet
DEFM_UC9	Electricity Supply Offer	Not Tested Yet

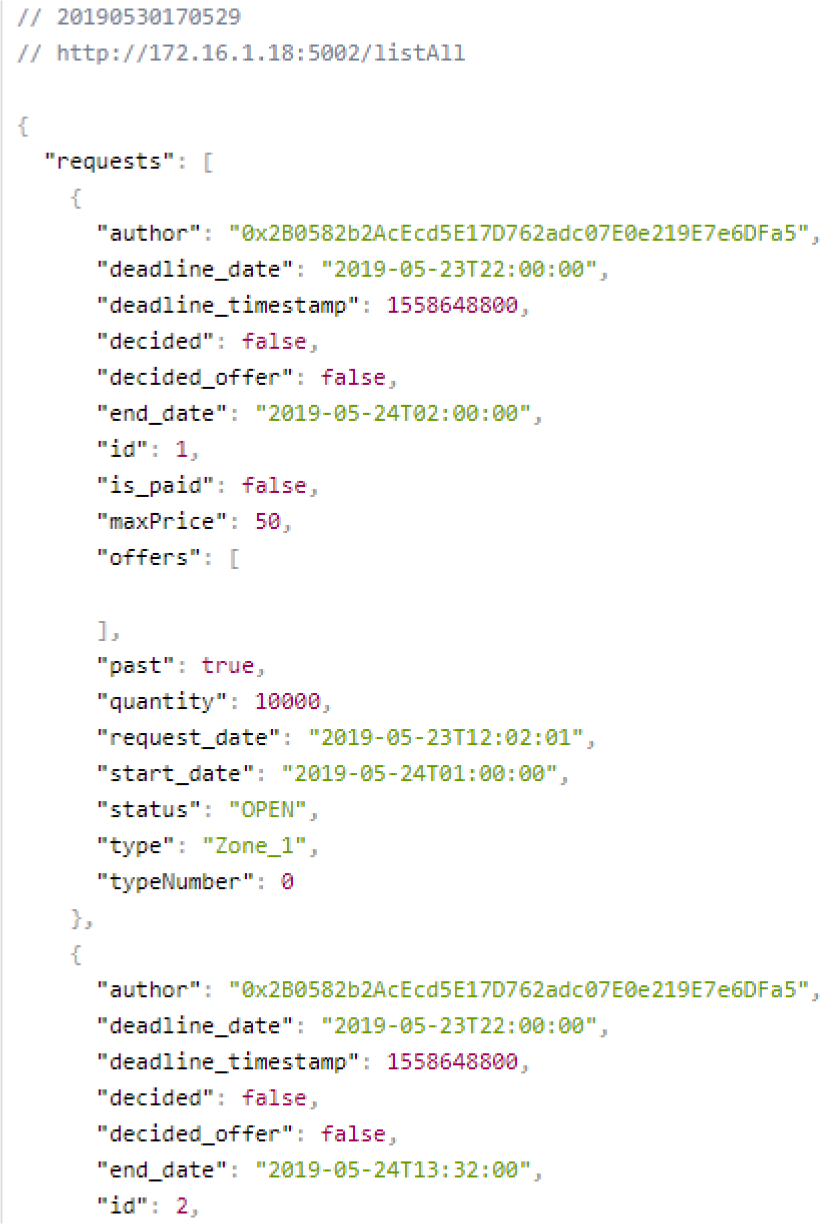
5.3.3.1 DEFM_UC1: Flexibility request

Figure 43, shows the results of a new marketplace request creation on the system, while Figure 44 shows the list of created requests.



```
// 20190530162323
// http://172.16.1.18:5002/addRequest/6/1/1559226149173/1559253600000/1559311200000/1559314800000/50

"added into block 12, 209803 gas used"
```

Figure 43. Request creation

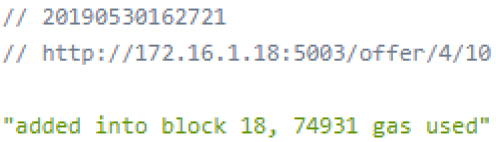


```
// 20190530170529
// http://172.16.1.18:5002/listAll

{
  "requests": [
    {
      "author": "0x2B0582b2AcEcd5E17D762adc07E0e219E7e6DFa5",
      "deadline_date": "2019-05-23T22:00:00",
      "deadline_timestamp": 1558648800,
      "decided": false,
      "decided_offer": false,
      "end_date": "2019-05-24T02:00:00",
      "id": 1,
      "is_paid": false,
      "maxPrice": 50,
      "offers": [

      ],
      "past": true,
      "quantity": 10000,
      "request_date": "2019-05-23T12:02:01",
      "start_date": "2019-05-24T01:00:00",
      "status": "OPEN",
      "type": "Zone_1",
      "typeNumber": 0
    },
    {
      "author": "0x2B0582b2AcEcd5E17D762adc07E0e219E7e6DFa5",
      "deadline_date": "2019-05-23T22:00:00",
      "deadline_timestamp": 1558648800,
      "decided": false,
      "decided_offer": false,
      "end_date": "2019-05-24T13:32:00",
      "id": 2,
```

Figure 44. Requests list

5.3.3.2 DEFM_UC2: EV offers request (pull)

Figure 45 shows the creation of a new offer within an open marketplace request.

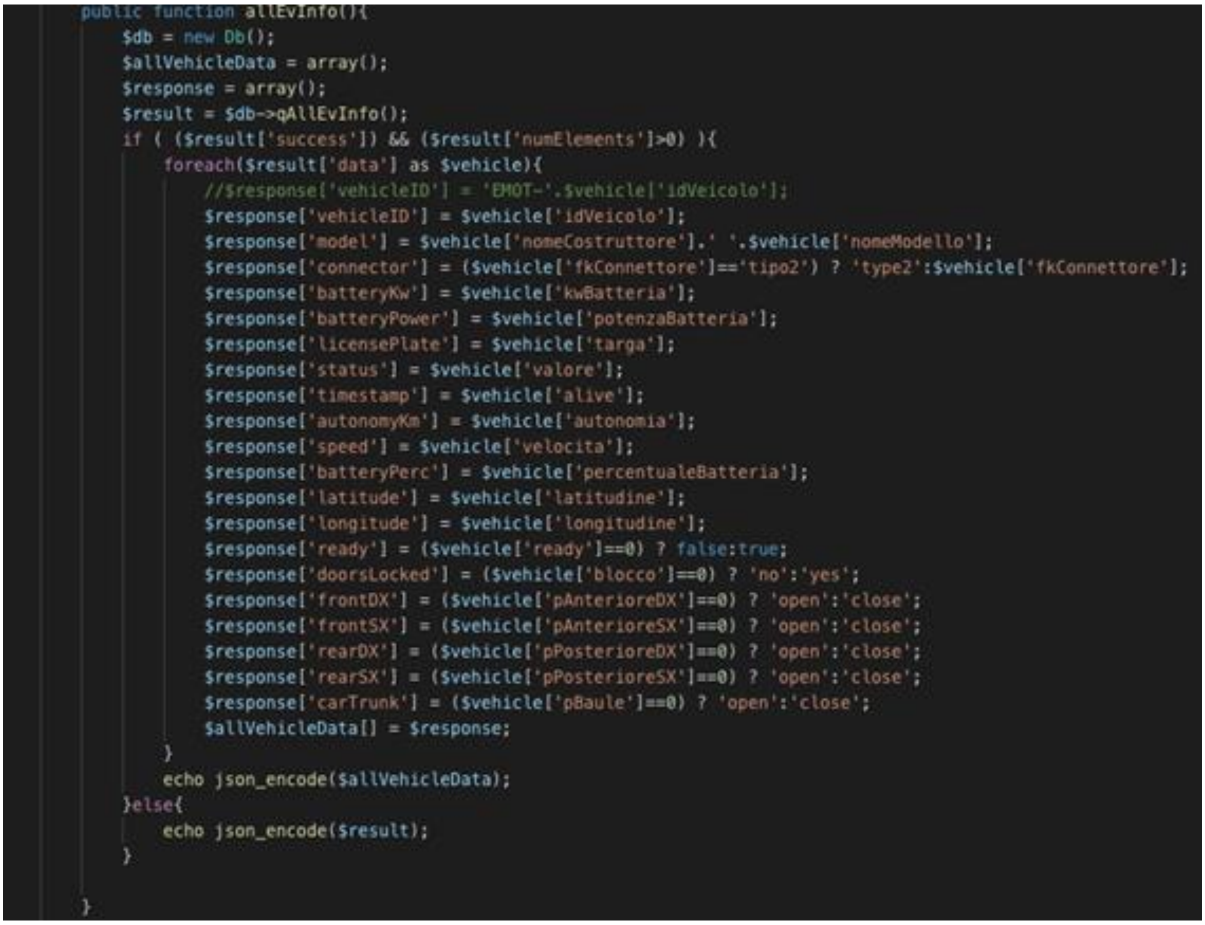


```
// 20190530162721
// http://172.16.1.18:5003/offer/4/10

"added into block 18, 74931 gas used"
```

Figure 45. Offer creation

5.3.3.3 DEFM_UC4: EV/EVSE fleet monitoring

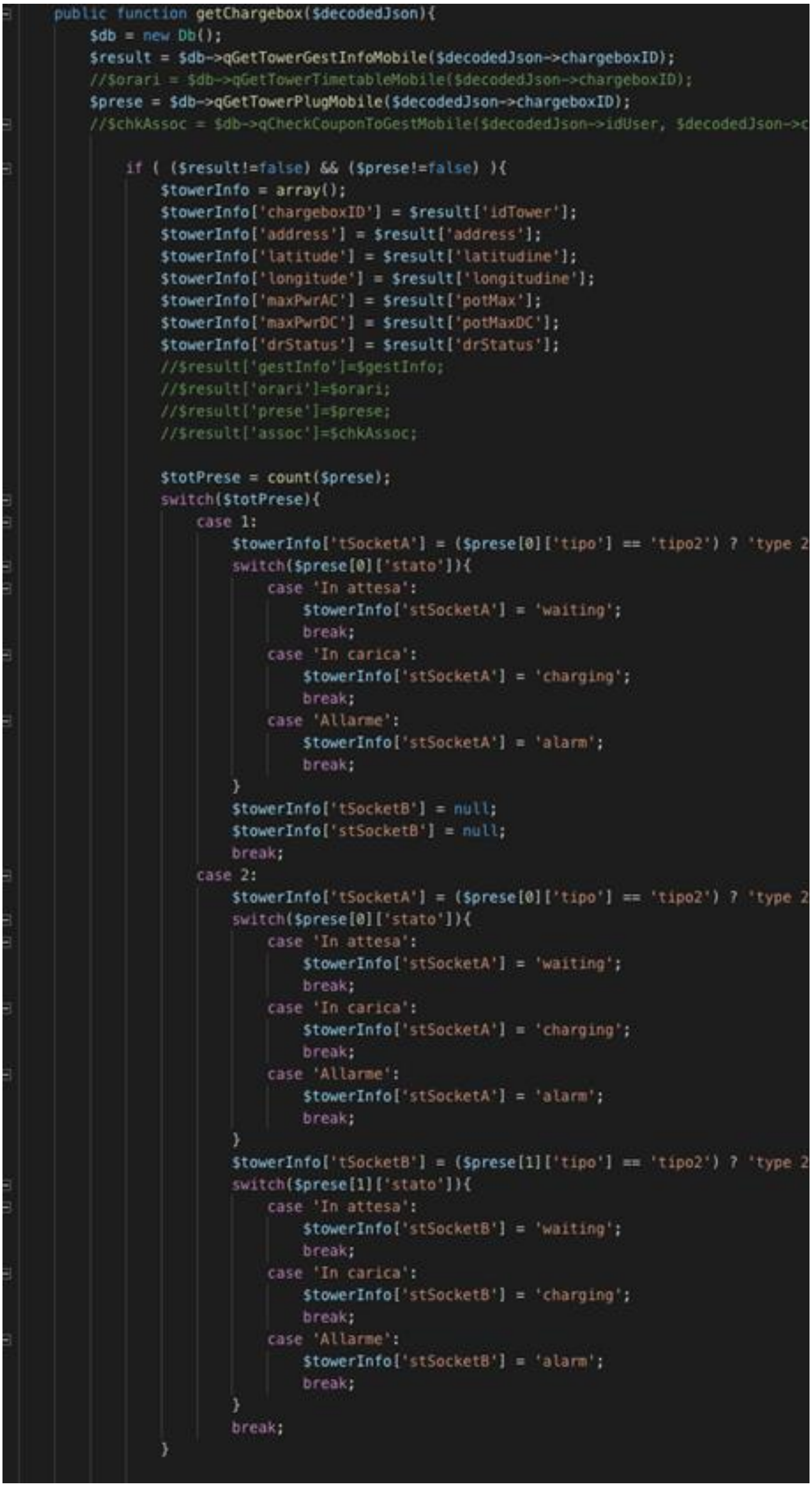
Figure 46 shows the collection request of EV data.



```
public function allEvInfo(){
    $db = new Db();
    $allVehicleData = array();
    $response = array();
    $result = $db->qAllEvInfo();
    if ( ($result['success']) && ($result['numElements']>0) ){
        foreach($result['data'] as $vehicle){
            //$response['vehicleID'] = 'EMOT-'.$vehicle['idVeicolo'];
            $response['vehicleID'] = $vehicle['idVeicolo'];
            $response['model'] = $vehicle['nomeCostruttore'].' '.$vehicle['nomeModello'];
            $response['connector'] = ($vehicle['fkConnettore']=='tipo2') ? 'type2':$vehicle['fkConnettore'];
            $response['batteryKw'] = $vehicle['kwBatteria'];
            $response['batteryPower'] = $vehicle['potenzaBatteria'];
            $response['licensePlate'] = $vehicle['targa'];
            $response['status'] = $vehicle['valore'];
            $response['timestamp'] = $vehicle['alive'];
            $response['autonomyKm'] = $vehicle['autonomia'];
            $response['speed'] = $vehicle['velocita'];
            $response['batteryPerc'] = $vehicle['percentualeBatteria'];
            $response['latitude'] = $vehicle['latitudine'];
            $response['longitude'] = $vehicle['longitudine'];
            $response['ready'] = ($vehicle['ready']==0) ? false:true;
            $response['doorsLocked'] = ($vehicle['blocco']==0) ? 'no':'yes';
            $response['frontDX'] = ($vehicle['pAnterioreDX']==0) ? 'open':'close';
            $response['frontSX'] = ($vehicle['pAnterioreSX']==0) ? 'open':'close';
            $response['rearDX'] = ($vehicle['pPosterioreDX']==0) ? 'open':'close';
            $response['rearSX'] = ($vehicle['pPosterioreSX']==0) ? 'open':'close';
            $response['carTrunk'] = ($vehicle['pBaule']==0) ? 'open':'close';
            $allVehicleData[] = $response;
        }
        echo json_encode($allVehicleData);
    }else{
        echo json_encode($result);
    }

}
```

Figure 46. EV remote monitoring

Figure 47 shows the collection request of charging station data.

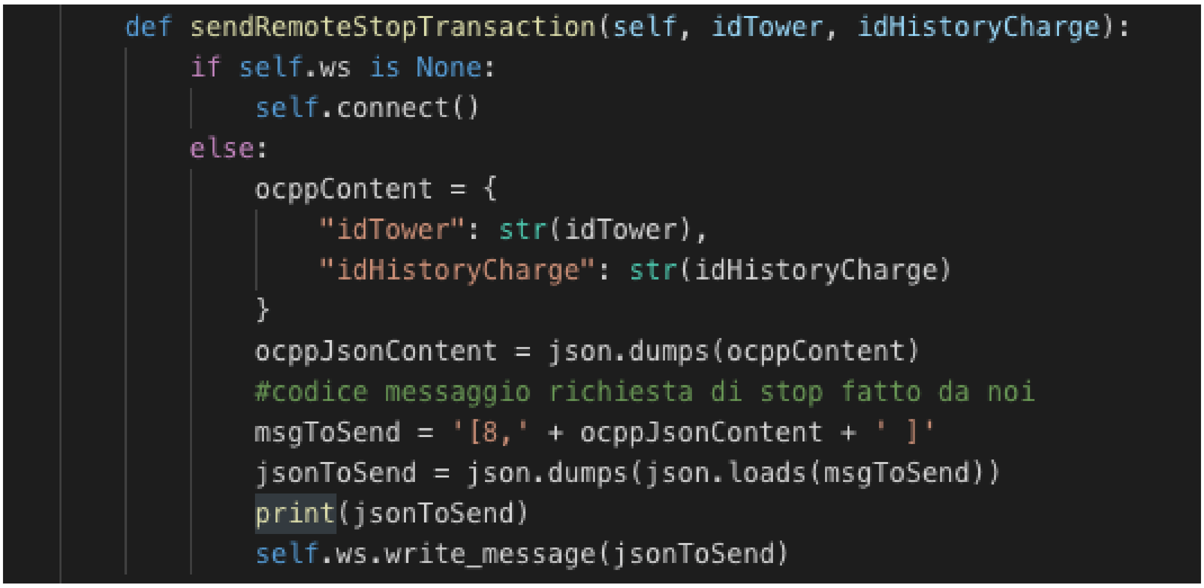


```
public function getChargebox($decodedJson){
    $db = new Db();
    $result = $db->qGetTowerGestInfoMobile($decodedJson->chargeboxID);
    //$orari = $db->qGetTowerTimetableMobile($decodedJson->chargeboxID);
    $prese = $db->qGetTowerPlugMobile($decodedJson->chargeboxID);
    //$chkAssoc = $db->qCheckCouponToGestMobile($decodedJson->idUser, $decodedJson->c

        if ( ($result!=false) && ($prese!=false) ){
            $towerInfo = array();
            $towerInfo['chargeboxID'] = $result['idTower'];
            $towerInfo['address'] = $result['address'];
            $towerInfo['latitude'] = $result['latitudine'];
            $towerInfo['longitude'] = $result['longitudine'];
            $towerInfo['maxPwrAC'] = $result['potMax'];
            $towerInfo['maxPwrDC'] = $result['potMaxDC'];
            $towerInfo['drStatus'] = $result['drStatus'];
            //$result['gestInfo']=$gestInfo;
            //$result['orari']=$orari;
            //$result['prese']=$prese;
            //$result['assoc']=$chkAssoc;

            $totPrese = count($prese);
            switch($totPrese){
                case 1:
                    $towerInfo['tSocketA'] = ($prese[0]['tipo'] == 'tipo2') ? 'type 2
                    switch($prese[0]['stato']){
                        case 'In attesa':
                            $towerInfo['stSocketA'] = 'waiting';
                            break;
                        case 'In carica':
                            $towerInfo['stSocketA'] = 'charging';
                            break;
                        case 'Allarme':
                            $towerInfo['stSocketA'] = 'alarm';
                            break;
                    }
                    $towerInfo['tSocketB'] = null;
                    $towerInfo['stSocketB'] = null;
                    break;
                case 2:
                    $towerInfo['tSocketA'] = ($prese[0]['tipo'] == 'tipo2') ? 'type 2
                    switch($prese[0]['stato']){
                        case 'In attesa':
                            $towerInfo['stSocketA'] = 'waiting';
                            break;
                        case 'In carica':
                            $towerInfo['stSocketA'] = 'charging';
                            break;
                        case 'Allarme':
                            $towerInfo['stSocketA'] = 'alarm';
                            break;
                    }
                    $towerInfo['tSocketB'] = ($prese[1]['tipo'] == 'tipo2') ? 'type 2
                    switch($prese[1]['stato']){
                        case 'In attesa':
                            $towerInfo['stSocketB'] = 'waiting';
                            break;
                        case 'In carica':
                            $towerInfo['stSocketB'] = 'charging';
                            break;
                        case 'Allarme':
                            $towerInfo['stSocketB'] = 'alarm';
                            break;
                    }
                    break;
            }
```

Figure 47. Charging station remote monitoring

5.3.3.4 DEFM_UC5: EVSE fleet management

Figure 48 shows the stop charging session command request.

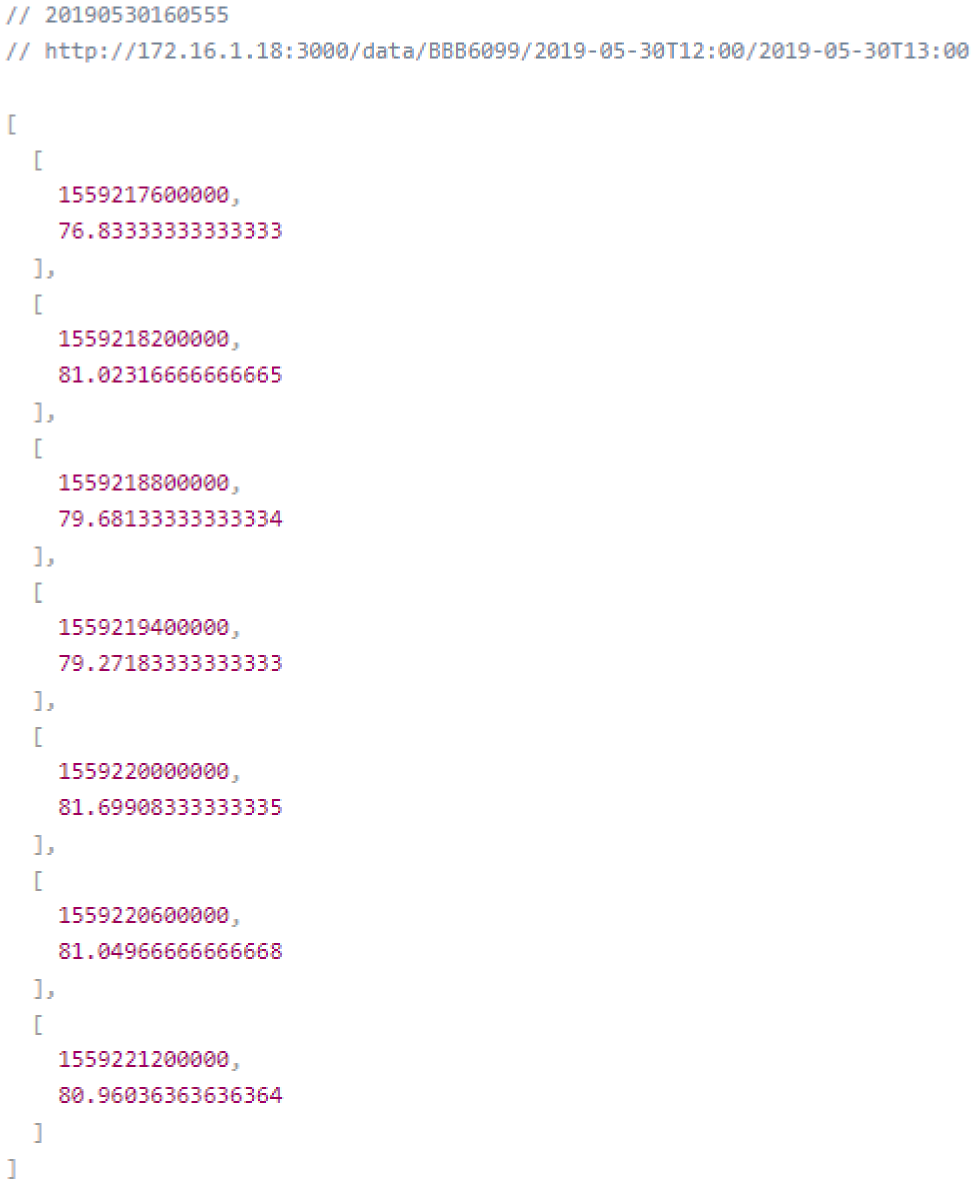


```
    def sendRemoteStopTransaction(self, idTower, idHistoryCharge):
        if self.ws is None:
            self.connect()
        else:
            ocppContent = {
                "idTower": str(idTower),
                "idHistoryCharge": str(idHistoryCharge)
            }
            ocppJsonContent = json.dumps(ocppContent)
            #codice messaggio richiesta di stop fatto da noi
            msgToSend = '[8,' + ocppJsonContent + ' ]'
            jsonToSend = json.dumps(json.loads(msgToSend))
            print(jsonToSend)
            self.ws.write_message(jsonToSend)
```

Figure 48. Charging station remote management

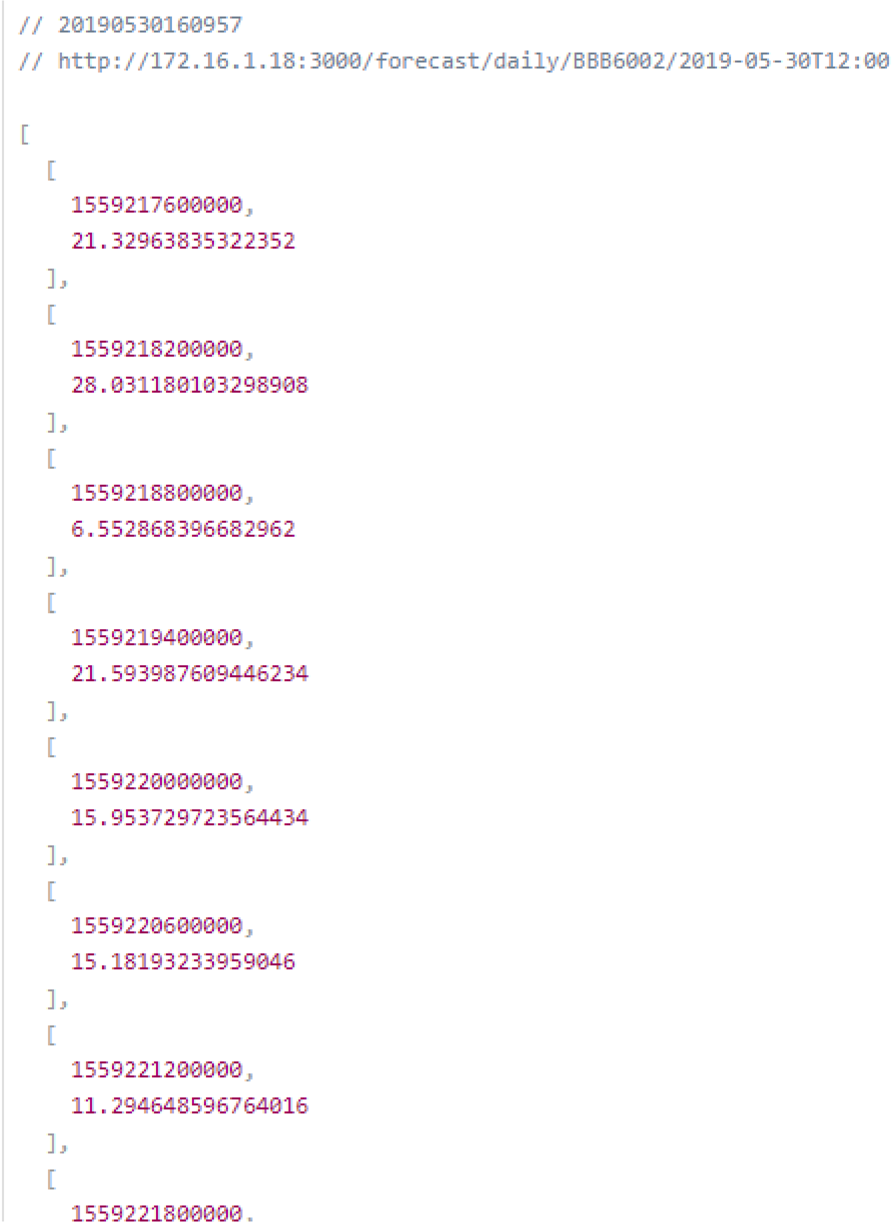
5.3.3.5 DEFM_UC7 District Forecasting

Figure 49, shows the current consumption of the district. Each entry is characterized by timestamp and power. Figure 50 and Figure 51 show two different statistical forecasting methods, based on the historical data gathered by smart meters. The first one is more appropriate when the measure is independent from the specific day of the week (e.g. PV production), while the second one performs better for measures depending also from the day of the week (e.g. a building consumption is different on weekdays or holidays).



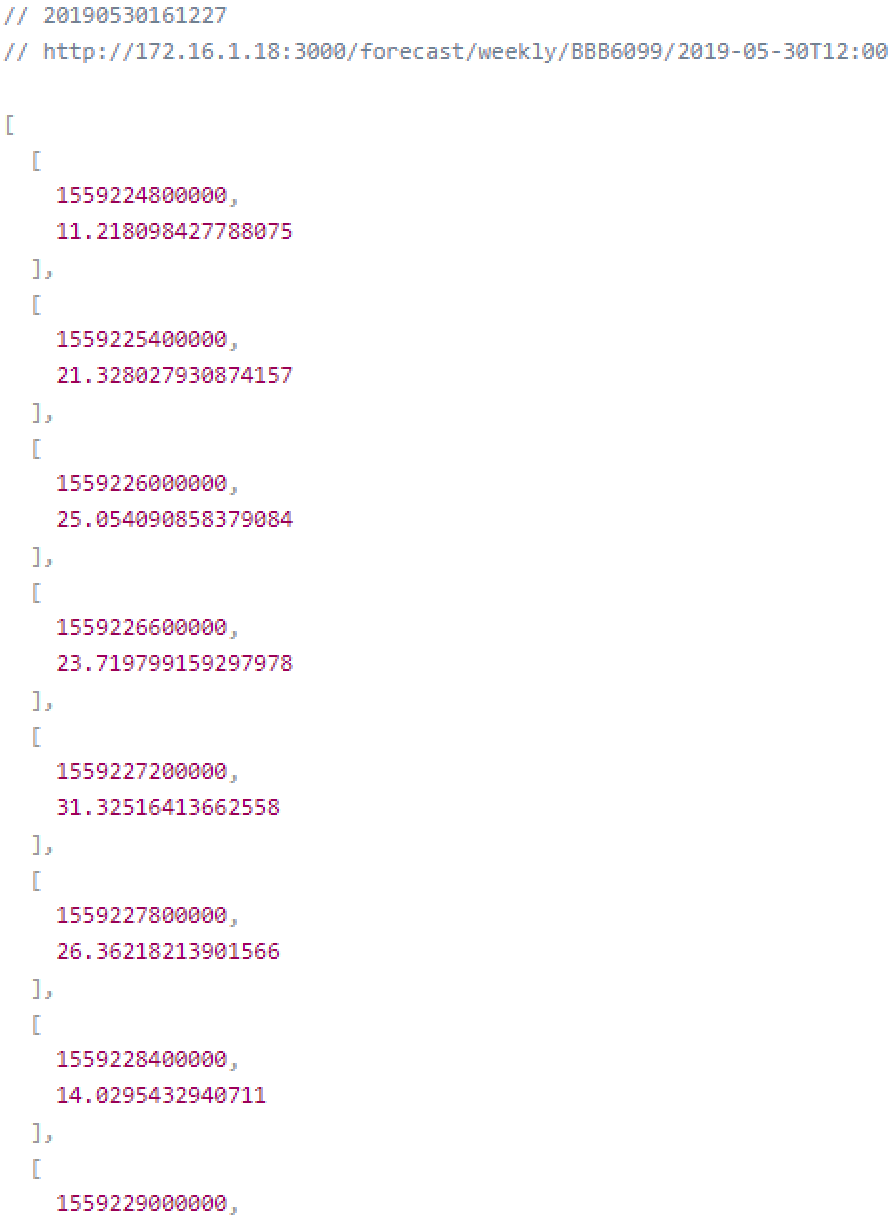
```
// 20190530160555
// http://172.16.1.18:3000/data/BBB6099/2019-05-30T12:00/2019-05-30T13:00

[
  [
    1559217600000,
    76.83333333333333
  ],
  [
    1559218200000,
    81.02316666666665
  ],
  [
    1559218800000,
    79.68133333333334
  ],
  [
    1559219400000,
    79.27183333333333
  ],
  [
    1559220000000,
    81.69908333333335
  ],
  [
    1559220600000,
    81.04966666666668
  ],
  [
    1559221200000,
    80.96036363636364
  ]
]
```

Figure 49. Current status (smart meters reading with timestamps)



```
// 20190530160957
// http://172.16.1.18:3000/forecast/daily/BBB6002/2019-05-30T12:00

[
  [
    1559217600000,
    21.32963835322352
  ],
  [
    1559218200000,
    28.031180103298908
  ],
  [
    1559218800000,
    6.552868396682962
  ],
  [
    1559219400000,
    21.593987609446234
  ],
  [
    1559220000000,
    15.953729723564434
  ],
  [
    1559220600000,
    15.18193233959046
  ],
  [
    1559221200000,
    11.294648596764016
  ],
  [
    1559221800000.
```

Figure 50. District forecast (generated considering the hour but not the day of the week)



```
// 20190530161227
// http://172.16.1.18:3000/forecast/weekly/BBB6099/2019-05-30T12:00

[
  [
    1559224800000,
    11.218098427788075
  ],
  [
    1559225400000,
    21.328027930874157
  ],
  [
    1559226000000,
    25.054090858379084
  ],
  [
    1559226600000,
    23.719799159297978
  ],
  [
    1559227200000,
    31.32516413662558
  ],
  [
    1559227800000,
    26.36218213901566
  ],
  [
    1559228400000,
    14.0295432940711
  ],
  [
    1559229000000,
```

Figure 51. District forecast (generated considering the hour and the day of the week)

5.3.4 Future validation tests

The following validation tests are planned for the next update of the pilot system prototype:

- Finalization of DSO and Fleet manager dashboards and their mutual interaction;
- PUSH scenario test;
- Energy retailer actor integration and testing of the interactions associated with it, including micro contract and micro transaction.

6 Mixed Reality Mobile Gaming Pilot

6.1 Pilot overview

6.1.1 Application context, constraints and unsolved issues

The focus of this pilot is to explore how DLTs can be used to provide new gaming features for players alongside with validating the potential of location-based IoT use cases. The pilot seeks to overcome known technical issues about the ability of DLTs to scale, so as to cost-effectively support millions of active users per day with thousands of transactions per second. It will also seek to investigate business issues, namely:

- Game discovery: how to reach mass market, given mobile app ecosystem (Apple/Google) access restriction on cryptocurrencies.
- IoT device scale: how devices (so that players can engage in their locality) fit for gaming use cases that can easily provide global scale.

In the first use case, we prototyped a game that enables players to collect and trade in-game content swapping or buying with other players (e.g. characters, weapons, equipment, parts) leveraging DLTs to provide player ownership of the asset, transparency and consistency of asset attributes and transactions. Attributes 'DNA' of the in-game assets were published on the blockchain.

In this use case, we plan to prototype the scavenger hunt location-based game using IoT beacons and an ecosystem backed by DLT. The players need to solve the riddles using the clues they receive on their smartphones. Solving the riddle will reveal the location of the IoT beacon and the player has to physically visit those areas to collect the points. The competition is to find the beacon locations and solve the tasks until they reach the last beacon. On the other hand, blockchain will be used to manage the relationship i.e. player check-in, points collection or rewards.

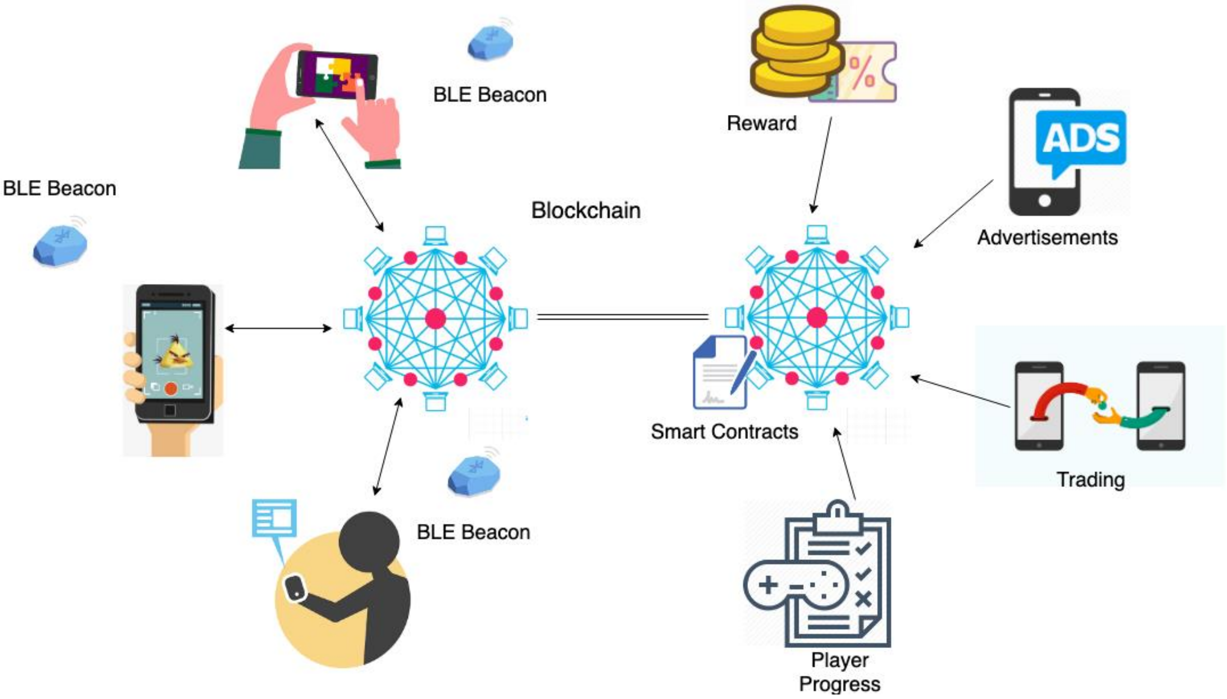


Figure 52. Mixed reality mobile gaming pilot overview

IoT beacons will be used to provide the proximity location of the players when they visit the point of Interests (PoI). These beacons will communicate with the smartphone using the Bluetooth Low Energy (BLE), acting as transmitters. Upon arriving, the mobile application detects the BLE beacon and notifies the player with the relevant task. The player performs the task i.e. collect items, take a picture or solve the puzzle, to reveal the location of next POI. After performing a series of tasks and visiting multiple locations, the points will be rewarded to players that can be later redeemed for rewards.

6.1.2 SOFIE added value

The focus of the mixed reality mobile gaming pilot will be on leveraging on the SOFIE platform to provide new gaming features for players. In the first use-case, the pilot will use the SOFIE DLT platform to provide player ownership of the asset, transparency, and consistency of asset attributes and transactions. The SOFIE marketplace will be used for trading the gaming assets along with providing security and traceability. In the second use-case, the pilot will use the SOFIE platform to establish a hybrid data organization, where some data is stored locally and some of it will be shared.

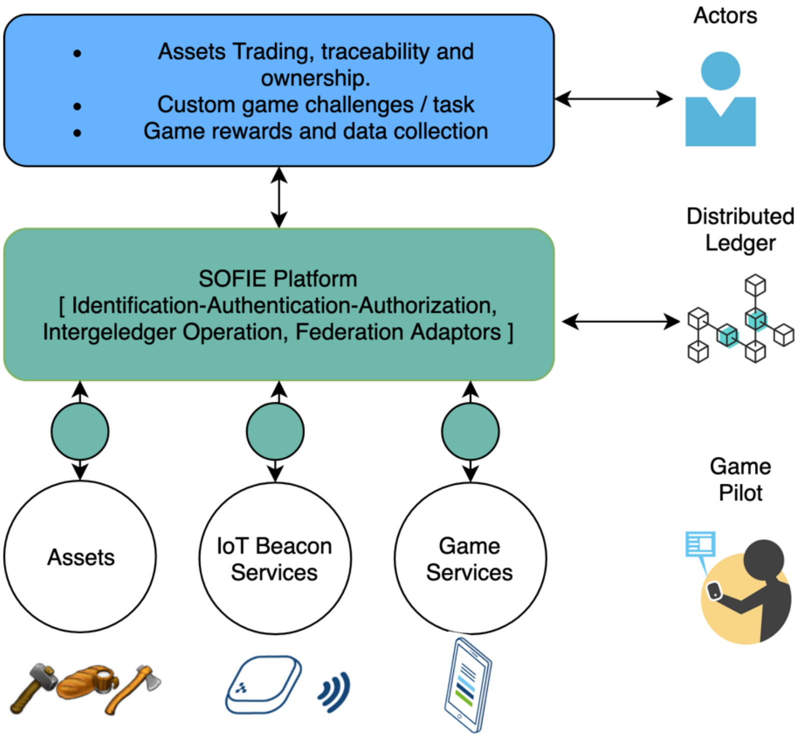

Assets Trading, traceability and ownership.
Custom game challenges / task
Game rewards and data collection
Actors
SOFIE Platform
[Identification-Authentication-Authorization, Intergeledger Operation, Federation Adaptors]
Distributed Ledger
Game Pilot
Assets
IoT Beacon Services
Game Services

Figure 53. The concept of mixed reality mobile gaming pilot

The pilot will also use the SOFIE identity and authentication module to secure access and interledger module for end-to-end security for data transactions. Pilot KPIs are presented in Table 22.

Table 22. KPIs in mixed reality mobile gaming pilot

Technology Level	
Enabling hardware deployment	• Devices deployment effort and configuration time • Compatibility with existing infrastructure
Enabling software and development of applications	• Deployment effort for 3rd parties' application • Dynamic and intuitive interface

Federating platforms	• Degree of interoperability • Transaction over ≥ 2 DLTs • 1 million plus active users per day supporting
Business Level	
Gaming player metrics	• Gaming experience is fun and valued by the player
Customer metrics	• Customer satisfaction • Time and effort for finding resources • Percentage of resolved disputes
Revenue opportunity	• Means to generate €100k+ per day

6.1.3 Pilot status and timeplan

Currently, the pilot is under the development phase. At this stage, the pilot status and plan do not deviate from that shown in Table 3.

6.2 Pilot scenarios, requirements and system architecture

The different people in the mixed reality mobile gaming pilot are:

- the game administrator,
- the game player,
- the assert/challenge designer,
- the ads administrator,
- the PoI employee.

The game administrator/developer has full access to the game and its data. He can view and edit all the challenges, player profiles, and related information. All the accounts need to be approved by the game administrator before accessing the system. Game developers are also responsible for setting up and administering the IoT beacons.

Players can access the game by the mobile application. They can view their profile and reward data through the app. They can join any challenge by scanning the QR code. They need a user account to access the data and play the game.

The game designer can create new challenges or asserts using the Web-enabled browser. They can create new tasks and puzzles for the existing beacons. They have access to the location of all the IoT beacons. They need to have developers account to access the restricted data and submit new challenges. The assert designer can also list their creation for the trade on the SOFIE platform.

The Ads administrator is responsible for the advertisement related tasks. He monitors and approves the advertisements shown in the application. He accesses the system and its ad managing account using the web browser.

The PoI employee can access the system using the web browser. He can view limited data about the challenges added by the PoI company. He can monitor the offered rewards and also add new ones. Last, he needs to have an employee account and administrator approval to access the system.

6.2.1 Scenarios and use cases

6.2.1.1 Mixed reality mobile gaming scenarios

SCENARIO	**Play game / challenges**
History	v0.2
Key Actors	Players
Assumptions / Dependencies	1) Players have installed the application and registered their accounts. 2) Players have internet connectivity and BLE activated. 3) IoT beacons are installed and active.
Objective(s)	Players can join the challenges and compete for the reward.
Description	The player after installing the application, joins the challenges using the QR code or challenge ID. After joining, the timer starts, and the player receives the location of the first point of Interest. As the player reaches the POI, the application detects the IoT beacon and downloads the clue from the server. Solving the clue reveals the location of the next point of interest. The player has to physically visit all the location to get the clue and collect points. The user can skip any task using the In-App tokens. Tokens can be either bought from the fiat currency, rewarded to the user by viewing advertisements, winning challenges or traded on the marketplace. After solving the last clue and visiting multiple locations, the timer is stopped, and the points are calculated based on the time taken by the player to reach the end. After the end of the challenge, the points are automatically added to the players' account, which can be later redeemed for prizes.
Services	• Challenges IDs record. • Marketplace. • Rewards record.
Metrics	• Check whether the player can play challenges and compete for the rewards.
SCENARIO	**Develop new challenges**
History	v0.2
Key Actors	Developer, Player, PoI
Assumptions / Dependencies	1) Users have registered their accounts as the challenge developer 2) IoT beacons are installed and active. 3) User is connected to the blockchain platform.
Objective(s)	Create custom challenges with custom clues using the beacons installed.
Description	Anyone with developer account can create the custom challenge using the web application. After logging in, the user is shown the list of active IoT beacons and the location where they are installed. The user can select any of those beacons and create custom clues for it. User can also add custom rewards for the challenges. After creating the whole challenge with multiple tasks, the user submits it for the approval from the game company. A challenge ID is created and shared on the blockchain for the players to join and compete for the reward.
Services	• Challenges IDs record. • Rewards record.
Metrics	• Check if the challenge contains all the required information and it is possible to complete the task.
SCENARIO	**In-App Advertisement**
History	V0.2
Scenario Name	In-App Advertisement
Key Actors	Developer, Ads Manager
Assumptions / Dependencies	1) Users have registered their accounts as the advertisers. 2) User is connected to the blockchain platform.

Objective(s)	Create custom In-App advertisement with rewards.
Description	Anyone with advertiser account can publish custom In-App advertisements. After logging in, the user can upload the advertisement video and input information regarding it. User can also add custom rewards for the viewership of the advertisement. After submitting it, a smart contract is generated and deployed on blockchain. Every interaction with the advertisement is tracked using the blockchain.
Services	• Advertisement smart contracts. • Traceability and viewability data.
Metrics	• Advertisements are shown to players and rewards are given at the end. • Interactions with the advertisement are recorded on blockchain.

6.2.1.2 Mixed reality mobile gaming pilot use cases

Figure 54 shows how the various actors interact with the mixed reality mobile gaming system.

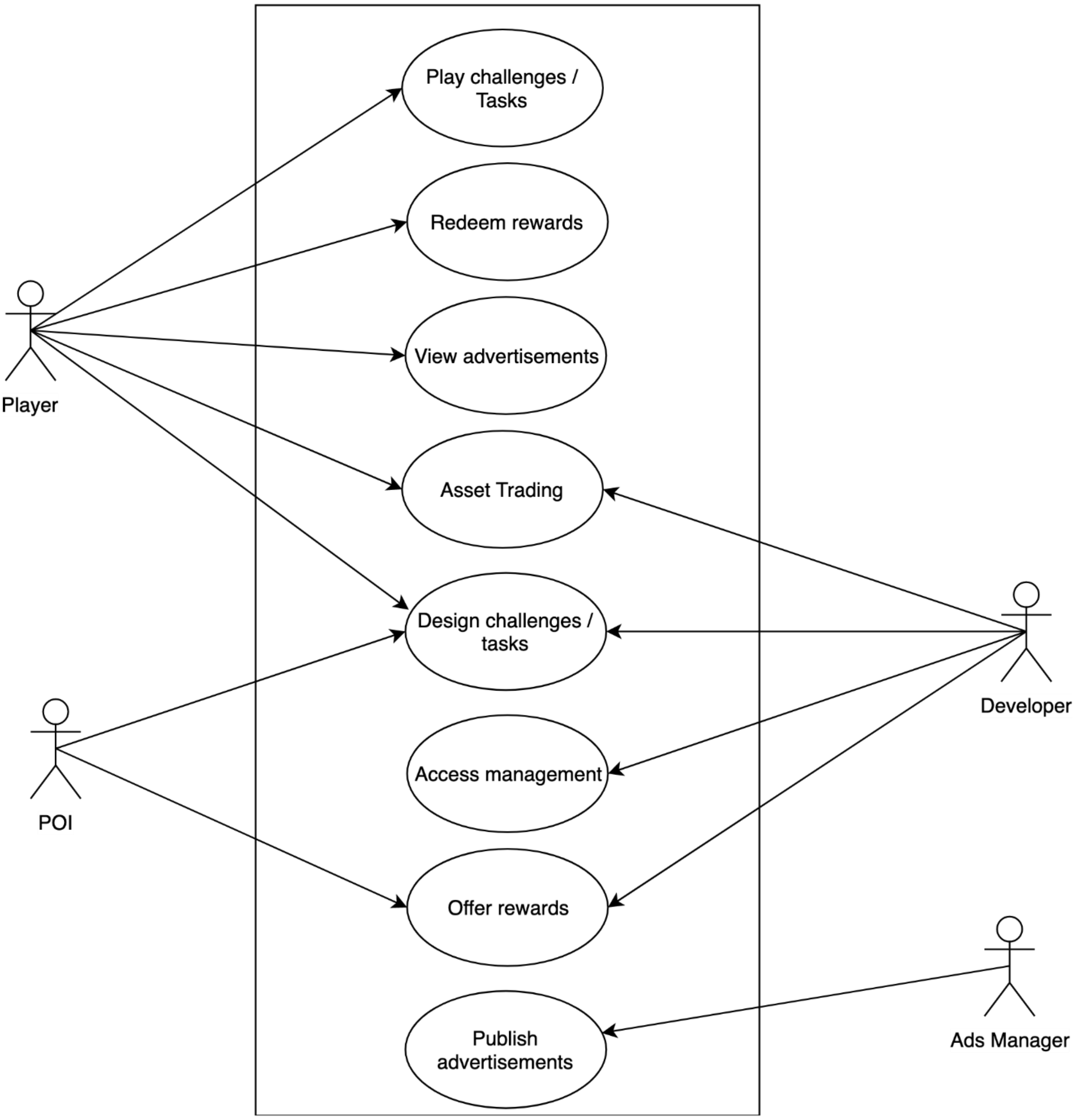


Figure 54. Actors' interaction with the mixed reality mobile gaming system

Pilot use cases are analyzed in Table 23.

Table 23. Use cases in the mixed reality mobile gaming pilot

ID	Name	Actors
MRMG_UC1	Play challenges / tasks	Player
MRMG _UC2	Redeem rewards	Player
MRMG _UC3	View In-App Advertisements	Player
MRMG _UC4	Assert trading	Player, Developers
MRMG _UC5	Design new challenges	Player, Developers, PoI
MRMG _UC6	Access management	Developers
MRMG _UC7	Offer rewards	Developers, PoI
MRMG _UC8	Publish new advertisements	Ads manager

USE CASE Description	
ID	MRMG_UC1
Name	Play challenges / tasks
Actors	Player
Storyline	The player can join any challenge, receives the clues and can compete for the reward.
Trigger events	Entering challenge ID or scanning QR code.
Preconditions	Player is registered and logged in.
Postconditions	Players receive points for the tasks completed.
Related scenarios	Play game / challenge

USE CASE Description	
ID	MRMG_UC2
Name	Redeem rewards
Actors	Player
Storyline	After completing the challenge, points are calculated for each player and the winner receives the reward (Coupons, Tokens, etc.)
Trigger events	Time for the challenge ends.
Preconditions	Player should join the challenge.

Postconditions	Token rewards are added to player account.
Related scenarios	Play game / challenge
USE CASE Description	
ID	MRMG_UC3
Name	View Advertisements
Actors	Player
Storyline	During the challenges, players will be given the option to view advertisements.
Trigger events	Player selects to view the advertisement video.
Preconditions	Players should be playing the game
Postconditions	Token rewards are added to player account.
Related scenarios	Play game / challenge
USE CASE Description	
ID	MRMG_UC4
Name	Assert trading
Actors	Player, Developer
Storyline	Players can trade coupons and tokens on the marketplace.
Trigger events	Player buys or sell their assets on the blockchain
Preconditions	Player has valid account and blockchain wallet.
Postconditions	Assert trading transactions are recorded on blockchain
Related scenarios	Play game / challenge
USE CASE Description	
ID	MRMG_UC5
Name	Design new challenges

Actors	Player, Developer, POI
Storyline	New challenges are created using the installed beacons. Custom clues can be added for each beacon.
Trigger events	Users go to web application to create the challenge.
Preconditions	User should have developers account.
Postconditions	New challenge is submitted for the review
Related scenarios	Design new challenges
USE CASE Description	
ID	MRMG_UC6
Name	Access management
Actors	Game company
Storyline	New accounts for developer, POI employees, ads manager needs to be approved by the game company.
Trigger events	After creating a new account, it is submitted for approval.
Preconditions	User must provide required information.
Postconditions	User account are ready to use.
Related scenarios	Play game / challenge Design new challenges In-App Advertisement
USE CASE Description	
ID	MRMG_UC7
Name	Offer rewards
Actors	Game company, POI, Ads Manager
Storyline	Rewards can be offered for challenges or ads.
Trigger events	Creating a new challenge Publishing a new advertisement

Preconditions	User offering rewards should be added in the blockchain.
Postconditions	Rewards can be redeem by the players
Related scenarios	Design new challenges In-App advertisement
USE CASE Description	
ID	MRMG_UC8
Name	Publish advertisement
Actors	Ads manager
Storyline	New ads for In-App advertisement can be published using the smart contract.
Trigger events	Ads manager uploads the advertisement video on the web application.
Preconditions	Ads and their related information are approved by the game company.
Postconditions	Ads are shown during the challenges.
Related scenarios	In-App advertisement

6.2.2 Requirements and software architecture

6.2.2.1 Domain requirements

In Table 24, the domain and end-user requirements of the mixed reality mobile gaming pilot are listed.

Table 24. List of requirements in the mixed reality mobile gaming pilot

ID	Name	Description	Priority	Related use case
REQ_ MRMG0.1	Unique identifiers for every actor	Each personal interacting with the game or web application should have a unique identifier.	MUST	ALL
REQ_ MRMG1.1	Game Mobile application	Game challenges are accessible using the Android application	MUST	MRMG_UC1
REQ_ MRMG1.2	Joining any game challenge	Players can join any challenge by scanning the QR code or manually entering challenge ID.	MUST	MRMG_UC1
REQ_ MRMG1.3	Unique identifier for challenges	Each challenge should have a unique identifier	MUST	MRMG_UC1

REQ_ MRMG1.4	Record the time taken to complete a challenge.	Time should be recorded for each player, Starting after joining the challenge till the player completes it.	MUST	MRMG_UC1
REQ_ MRMG1.5	Receive Clues / tasks	Players should receive unique clues / task when near the IoT beacons based on their challenge.	MUST	MRMG_UC1
REQ_ MRMG1.6	Skip any task	Players should be able to skip any task and receive location of next IoT beacon using the In-App tokens.	MUST	MRMG_UC1
REQ_ MRMG1.7	Purchase In-App tokens	Players can buy an unlimited amount of In-App token using the fiat currency	MUST	MRMG_UC1
REQ_ MRMG1.8	Points calculation	System should automatically calculate the points based on the time taken to complete any challenge	MUST	MRMG_UC1
REQ_ MRMG2.2	Rewards distribution	System should automatically add the rewards to the players account after the challenge ends.	MUST	MRMG_UC2
REQ_ MRMG3.1	In-App Advertisement video	Player should be given the option to view advertisements while playing a challenge.	MUST	MRMG_UC3 MRMG_UC8
REQ_ MRMG3.2	Advertisement reward	Player should receive tokens for viewing the advertisement.	MUST	MRMG_UC3 MRMG_UC7 MRMG_UC8
REQ_ MRMG3.3	Advertising viewability data	Every ad viewability data should be recorded as a transaction on the blockchain.	MUST	MRMG_UC3 MRMG_UC7
REQ_ MRMG4.1	Assert marketplace	Players can buy and sell In-App asserts on the blockchain	MUST	MRMG_UC4
REQ_ MRMG4.2	Assert trading data	Every asset traded on the platform should be recorded as a transaction on the blockchain.	MUST	MRMG_UC4
REQ_ MRMG5.1	Web Application	Web application for designing new challenges and uploading advertisement.	MUST	MRMG_UC5 MRMG_UC8
REQ_ MRMG5.2	Access control to the web services	Access control to the web services based on the role of the user.	MUST	MRMG_UC5 MRMG_UC6 MRMG_UC7 MRMG_UC8
REQ_ MRMG7.1	Offer rewards	Rewards can offered to the players through challenges and	MUST	MRMG_UC7

		advertisement videos.		
REQ_ MRMG7.2	Rewards data	Rewards should be added and recorded on the blockchain.	MUST	MRMG_UC7
REQ_ MRMG8.1	Publish new advertisements	Ads manager should publish any ad video using the web application	MUST	MRMG_UC8

6.2.2.2 Pilot security and privacy specifications

In this subsection, it is discussed how the SOFIE main security and privacy requirements are addressed in pilot.

Table 25. Security and privacy specifications in the mixed reality mobile gaming pilot

Support for transactions, where only authorised entities can participate
The pilot will use permissioned DLT i.e. Hyperledger Fabric to store the transactions. Primarily, smart contracts are used to generate transactions which are subsequently distributed to every peer node in the network where they are immutably recorded on their copy of the ledger. For an identity to be verifiable, it must come from a trusted authority. A membership service provider (MSP) is how this is achieved in Fabric. More specifically, an MSP is a component that defines the rules that govern valid identities. We will use traditional Public Key Infrastructure (PKI) to achieve this through MSPs.
Transactions must be authentic and verifiable
Only registered entities will be able to do the transaction in the consortium ledger. Apart from this, the web application will enable a role-based access control using the user ID and password. Each user will have different access levels according to their affiliation e.g. game developer or POI employee. Session service will be used for authenticating players when accessing the game application services. An access token is used to authenticate the user in server API calls. The access token will expire after a specific period of time and a new one will be automatically created.
Privacy issues and business secrets must be considered carefully when deciding what data (including authentication/authorization information, logs etc.) is collected, stored or exchanged between parties
In the Hyperledger Fabric, channels will be used as the primary communications mechanism by which the members of a consortium can communicate each other. There can be multiple channels in a network. These channels will provide private communications between different entities. Access to these channels can only be done by verifying identities.

6.2.2.3 Pilot system architecture

The overview of the pilot system architecture is shown in Figure 55. The focus of the mixed reality mobile gaming pilot will be on leveraging of the SOFIE platform to provide new gaming features for players. The pilot will use the private DLT to provide player ownership of the asset, transparency, and consistency of asset attributes and transactions. The public DLT will be used for trading of the gaming assets along with providing security. The pilot will also use the SOFIE inter-ledger module for end-to-end security for data transactions.

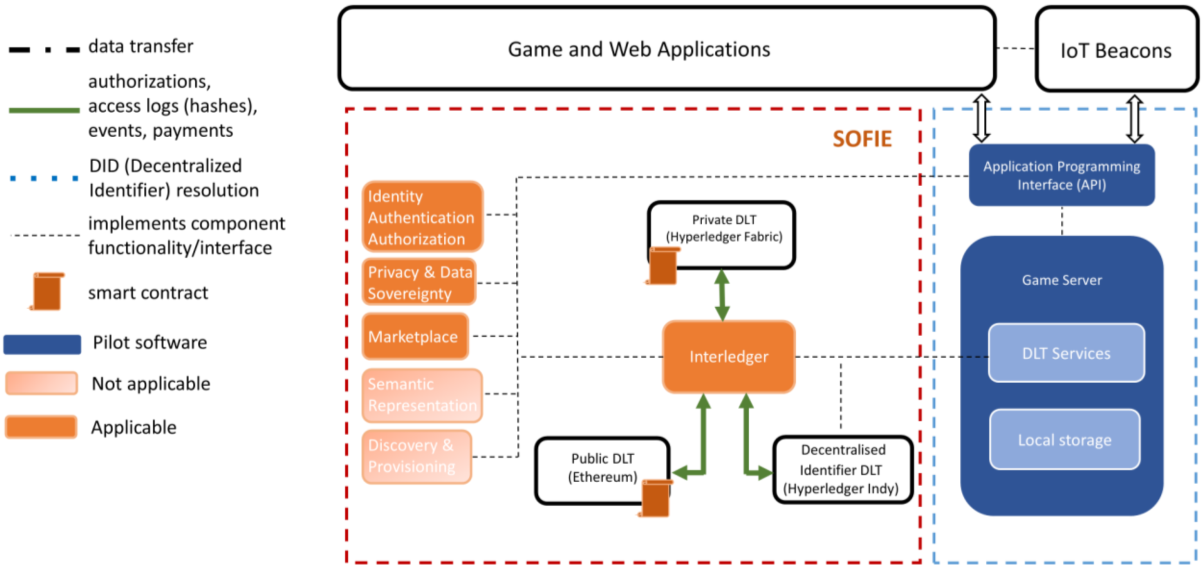

data transfer
authorizations, access logs (hashes), events, payments
DID (Decentralized Identifier) resolution
implements component functionality/interface
smart contract
Pilot software
Not applicable
Applicable
Game and Web Applications
IoT Beacons
SOFIE
Identity Authentication Authorization
Privacy & Data Sovereignty
Marketplace
Semantic Representation
Discovery & Provisioning
Private DLT (Hyperledger Fabric)
Interledger
Public DLT (Ethereum)
Decentralised Identifier DLT (Hyperledger Indy)
Application Programming Interface (API)
Game Server
DLT Services
Local storage

Figure 55. Overview of the system architecture for the mixed reality mobile gaming pilot

Figure 56 shows in detail the architecture of the game application for playing the challenges. The introduced modules are as following:

Mobile Application: A mobile application with a graphical user interface running on the Android platform. Players will install the application to play the challenges, to redeem rewards and to trade assets on the SOFIE marketplace. This application communicates with the game server using REST APIs.

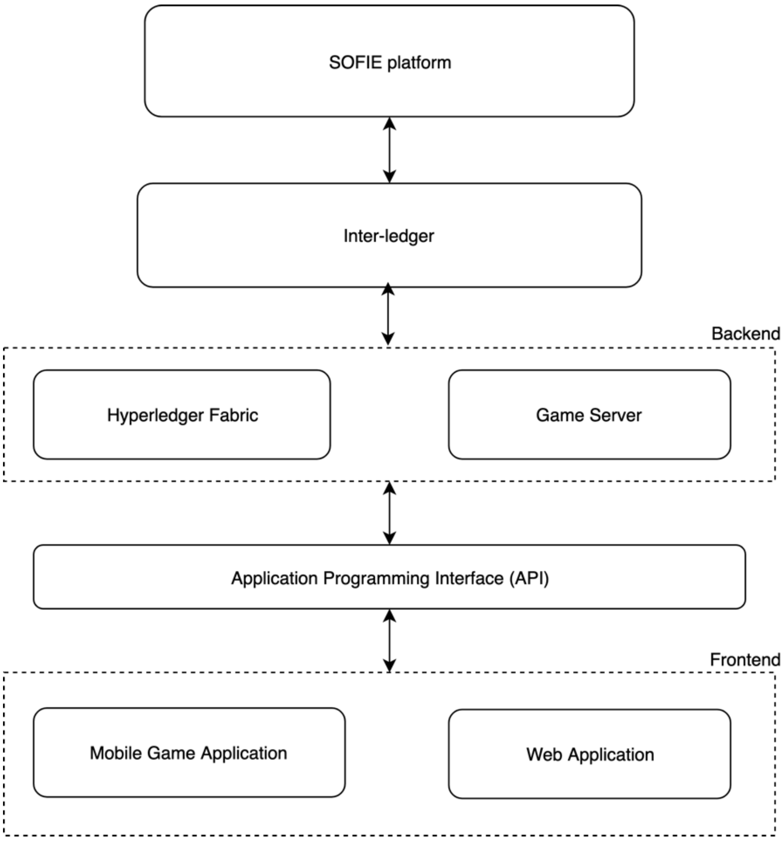


Figure 56. Software architecture for mixed reality gaming pilot application

Web Application: A web interface for services related to the game. It is only accessible by the game company, PoIs or developers account. It can be used to configure game related services, access the beacon-related information and also provide a GUI to do transactions with blockchain. A PoI can use the web application to create custom challenges and also provide rewards. The developers can view all the locations of the IoT beacons and add custom clues / tasks for them. They can also publish rewards through this application

Game Server: A server that provides services to the game and also acts as middleware for communicating with the SOFIE platform. It can be accessed through the REST APIs. It will also be connected to a private database to store the information related to the game and players.

Inter-Ledger: A SOFIE architecture component enabling communication with the SOFIE platform.

Hyperledger Fabric: A permissioned blockchain to store data from the game. Smart contracts will be coded and used to generate transactions that will be recorded on ledger.

6.2.2.3.1 Sequence of interaction for pilot use cases

The following two sequence diagrams show how the system accomplishes two important actions in the mixed reality mobile gaming pilot, e.g. how a player can access and play the game and how a developer can design and upload a new game challenge.

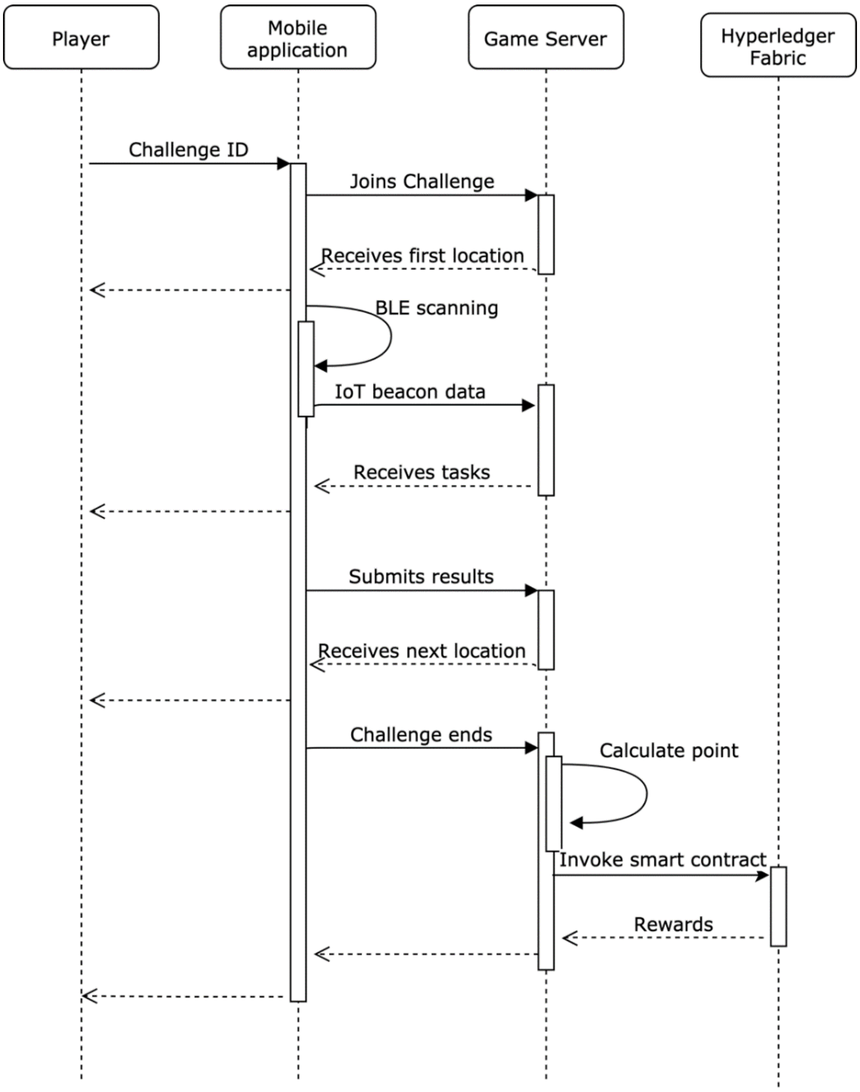


Figure 57. Sequence diagram about how a player can access and play the game

In Figure 57, the player joins the challenges using the QR code or challenge ID. After joining, the timer starts, and the player receives the location of the first point of Interest. As the player reaches the POI, the application detects the IoT beacon and downloads the clue from the server. Solving the clue reveals the location of the next point of interest. After solving the last clue and visiting multiple locations, the timer is stopped, and the points are calculated based on the time taken by the player to reach the end. After the end of the challenge, the points are automatically added to the players' account, which can be later redeemed for prizes.

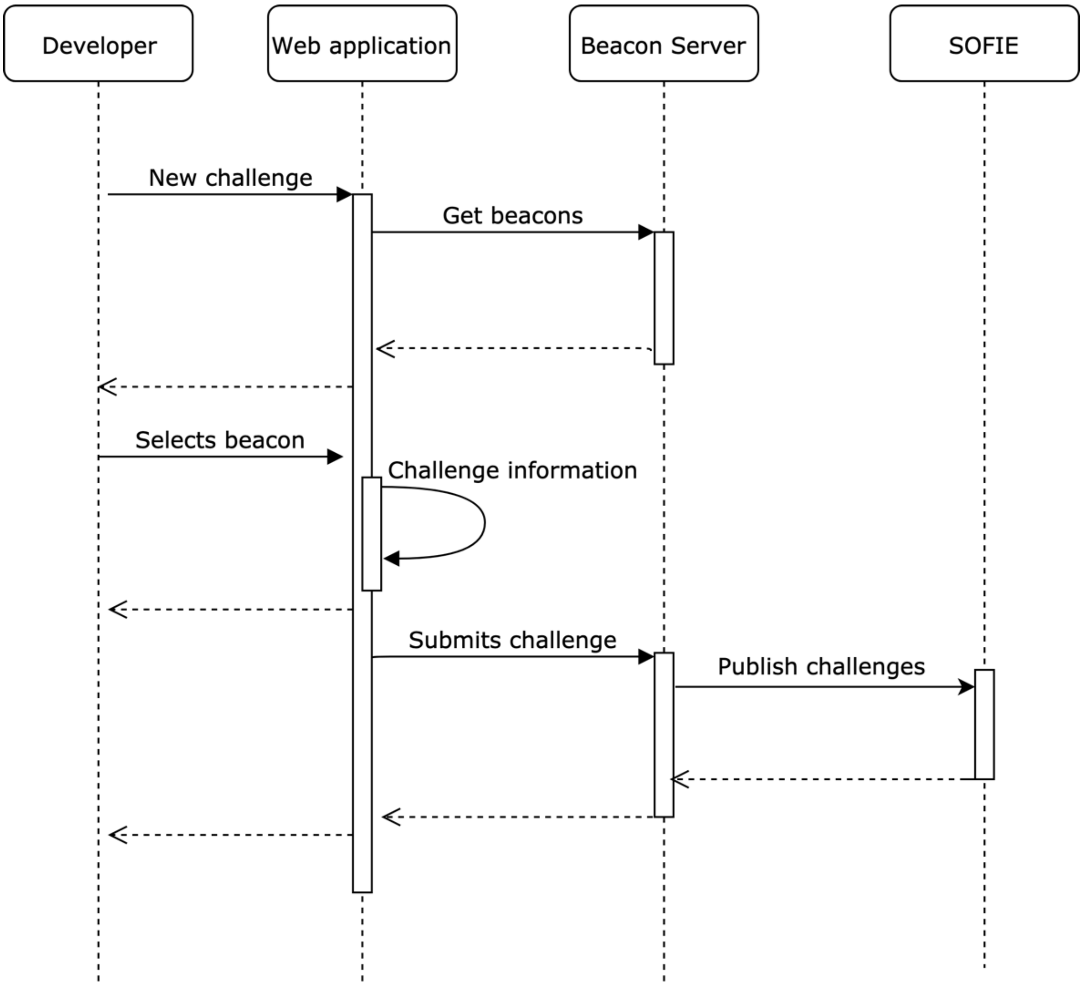


Figure 58. Sequence diagram about how the developer can upload a new game challenge

As shown in Figure 58, anyone with developer account can create the custom challenge using the web application. After logging in, the user is shown the list of active IoT beacons and the location where they are installed. The user can select any of those beacons and create custom clues for it. He can also add custom rewards for the challenges. After creating the whole challenge with multiple tasks, the user submits it for the approval from the game company. A challenge ID is created and shared on the blockchain for the players to join and compete for the reward.

6.3 Initial SOFIE validation

6.3.1 Installed infrastructure

We used combination of iBKS and iBKS plus beacons to provide proximity location for the Scavenger hunt game. iBKS, shown in Figure 59, is a Bluetooth Low Energy (BLE) beacon based on Nordic Semiconductors nrf51822 chipset that uses a CR2477 coin cell battery. These beacons are compatible with iBeacon and Eddystone (UID, URL, TLM, EID) at the same time. These beacons have 30-40 months of battery lifetime depending on the Tx power at 1s interval. IBKS beacons have a range of approximately 70 meters using full power.

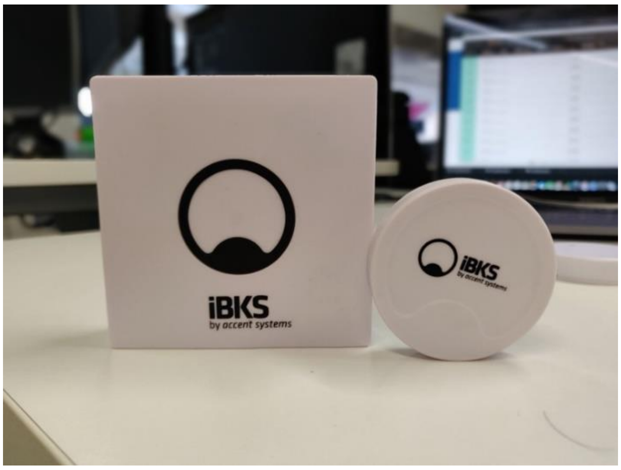


Figure 59. iBKS beacons

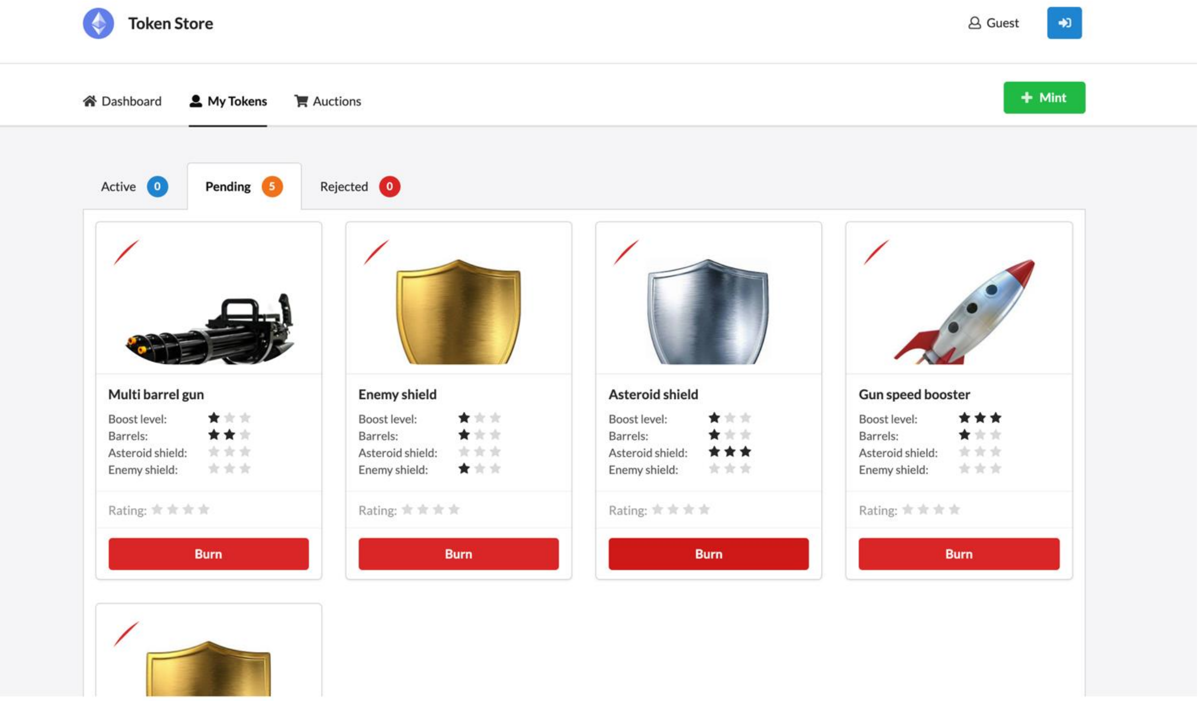
Before an IoT beacon can be used, we provisioned it to set the frame type, broadcast intervals and power levels. We used Eddystone-EID to control who can access the beacons and only services that share an encryption key with an Eddystone-EID beacon can resolve message data from that beacon.

6.3.2 Proof of concept prototype

The initial prototype consists of a web dashboard connected to the Ethereum client for the real ownership and trading of the asset on the decentralized marketplace. We also configured multiple IoT beacons with the Eddystone Ephemeral Identifier (EID) firmware to provide clues and tasks in the game and a dashboard for backend services for the IoT beacons.

6.3.2.1 Asset dashboard

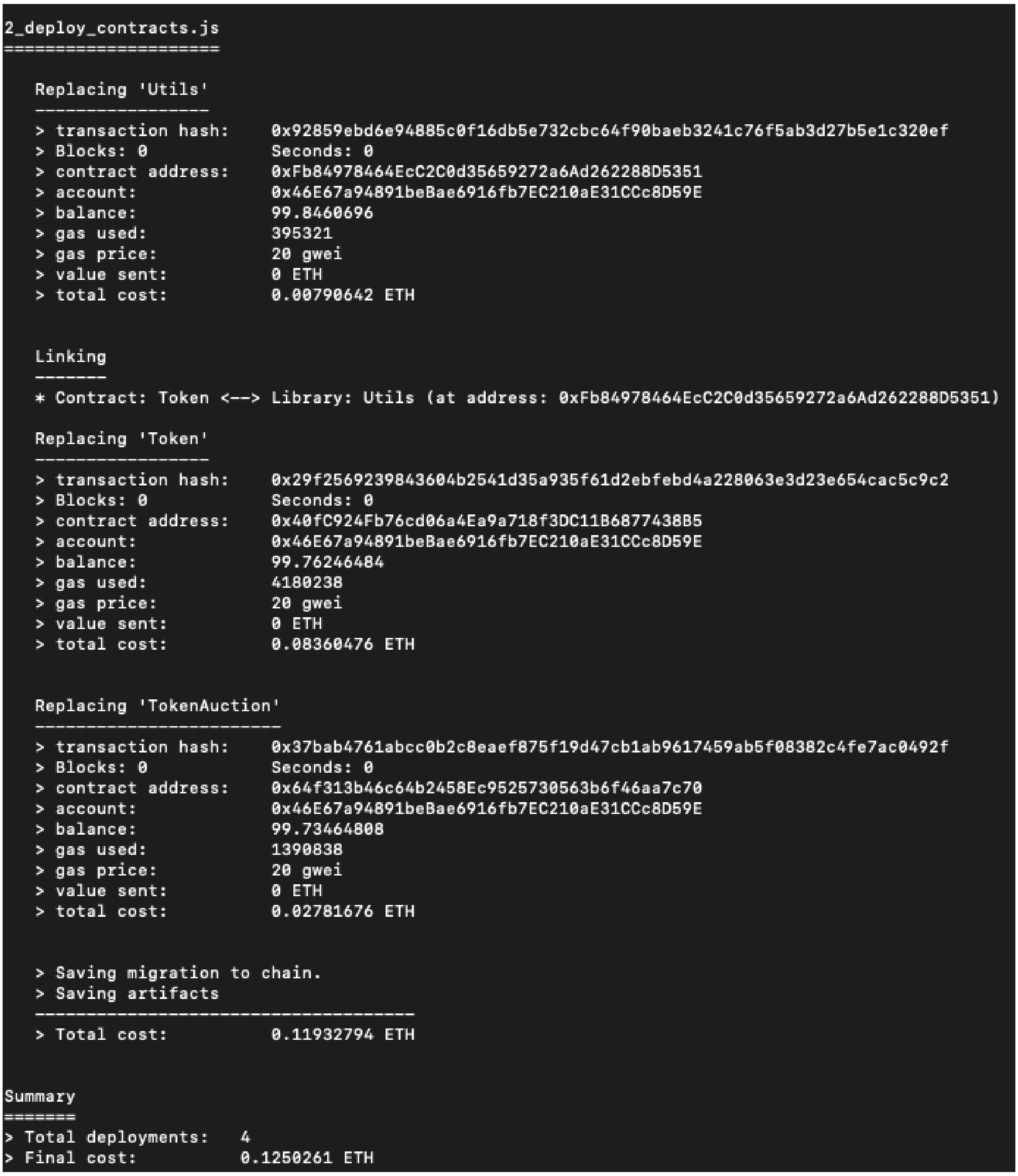
The asset dashboard, Figure 60, shows the assets from the game that are tokenized onto the Ethereum blockchain. The purpose of this dashboard is to demonstrate how blockchain can be used to give players real ownership of in-game content and later be used for trading.



	Multi barrel gun	Enemy shield	Asteroid shield	Gun speed booster
Boost level:	★☆☆	★☆☆	★☆☆	★★★
Barrels:	★★☆	★☆☆	★☆☆	★☆☆
Asteroid shield:	☆☆☆	☆☆☆	★★★	☆☆☆
Enemy shield:	☆☆☆	★☆☆	☆☆☆	☆☆☆
Rating:	☆☆☆☆	☆☆☆☆	☆☆☆☆	☆☆☆☆

Figure 60. Asset dashboard

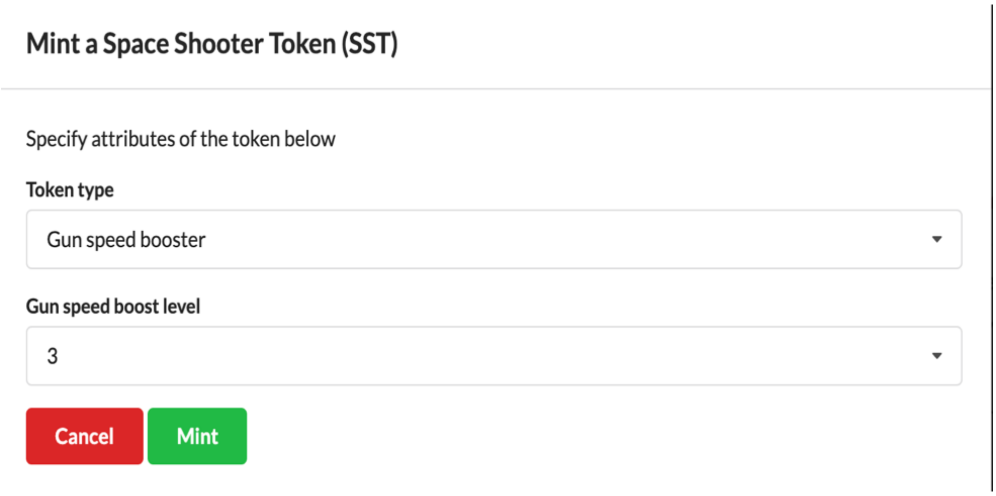
The smart contract managing the assets, implements an ERC-721 contract interface. This means that the assets can be tokenized and saved onto the blockchain. To start with, ERC-721 contract is extended to include custom asset attributes and the Truffle framework is used to compile and deploy the smart contract. Ganache client is used to work as a personal blockchain for Ethereum.



```
2_deploy_contracts.js
=====================

   Replacing 'Utils'
   -----------------
   > transaction hash:    0x92859ebd6e94885c0f16db5e732cbc64f90baeb3241c76f5ab3d27b5e1c320ef
   > Blocks: 0            Seconds: 0
   > contract address:    0xFb84978464EcC2C0d35659272a6Ad262288D5351
   > account:             0x46E67a94891beBae6916fb7EC210aE31CCc8D59E
   > balance:             99.8460696
   > gas used:            395321
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00790642 ETH


   Linking
   -------
   * Contract: Token <--> Library: Utils (at address: 0xFb84978464EcC2C0d35659272a6Ad262288D5351)

   Replacing 'Token'
   -----------------
   > transaction hash:    0x29f2569239843604b2541d35a935f61d2ebfebd4a228063e3d23e654cac5c9c2
   > Blocks: 0            Seconds: 0
   > contract address:    0x40fC924Fb76cd06a4Ea9a718f3DC11B6877438B5
   > account:             0x46E67a94891beBae6916fb7EC210aE31CCc8D59E
   > balance:             99.76246484
   > gas used:            4180238
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.08360476 ETH


   Replacing 'TokenAuction'
   ------------------------
   > transaction hash:    0x37bab4761abcc0b2c8eaef875f19d47cb1ab9617459ab5f08382c4fe7ac0492f
   > Blocks: 0            Seconds: 0
   > contract address:    0x64f313b46c64b2458Ec9525730563b6f46aa7c70
   > account:             0x46E67a94891beBae6916fb7EC210aE31CCc8D59E
   > balance:             99.73464808
   > gas used:            1390838
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.02781676 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.11932794 ETH


Summary
=======
> Total deployments:   4
> Final cost:          0.1250261 ETH
```

Figure 61. Deploying ERC-721 smart contracts

Once the smart contracts are deployed, a custom web application (asset dashboard) is created and linked to the Ethereum blockchain using JavaScript. New assets can be added to the platform using the “Mint” button and based on the asset type, the user can specify attributes related to that asset. Once the required information is given, the user can call the smart contract function to tokenize the asset. We also connected the web application to third-party wallet i.e. Metamask, for the confirmation of the smart contract transaction.



Mint a Space Shooter Token (SST)

Specify attributes of the token below

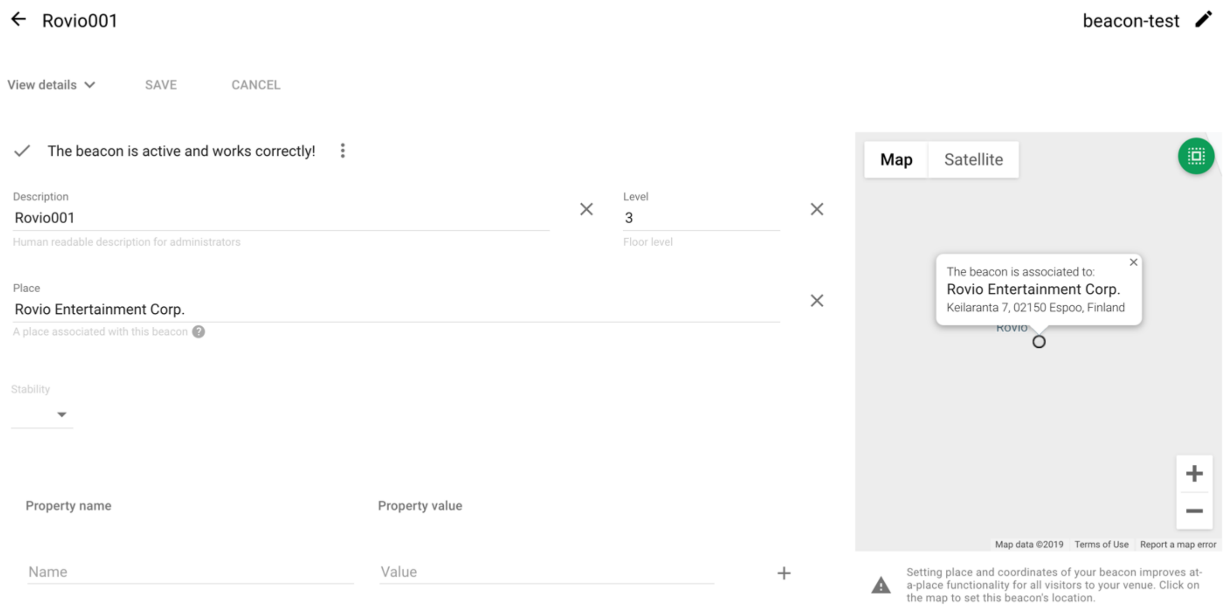
Token type: Gun speed booster

Gun speed boost level: 3

Figure 62. Tokenizing new assets

6.3.2.2 Beacon dashboard

The gaming application is built with the Nearby Messages API, that help to detect deployed IoT beacons and retrieve messages that have been associated with them. When using the Nearby API with beacons, the messages take the form of beacon attachments i.e. use of two fields namespaced Type and data. The IoT beacons dashboard, shown in Figure 63, is used to register the beacons, add attachment to them and access those attachments as clues, with player's game application using nearby messages. These messages are stored in the cloud and can be updated as often as we like without the need to update the beacons themselves.



Rovio001

beacon-test

✓ The beacon is active and works correctly!

Description: Rovio001
Human readable description for administrators

Level: 3
Floor level

Place: Rovio Entertainment Corp.
A place associated with this beacon

Stability

Property name: Name

Property value: Value

The beacon is associated to:
Rovio Entertainment Corp.
Keilaranta 7, 02150 Espoo, Finland

Setting place and coordinates of your beacon improves at-a-place functionality for all visitors to your venue. Click on the map to set this beacon's location.

Figure 63. Registration of IoT Beacon

The proximity Beacon API from Google is used to register the beacons and share the encryption key with the platform. The same API is also used to add attachments to the beacons also. These attachments are stored as blobs in the scalable cloud. The attachment type is an arbitrary string that can be used to separate different categories of attachments. The attachments can be up to 1024 bytes long and any string that has some meaning for the application can be bound with.

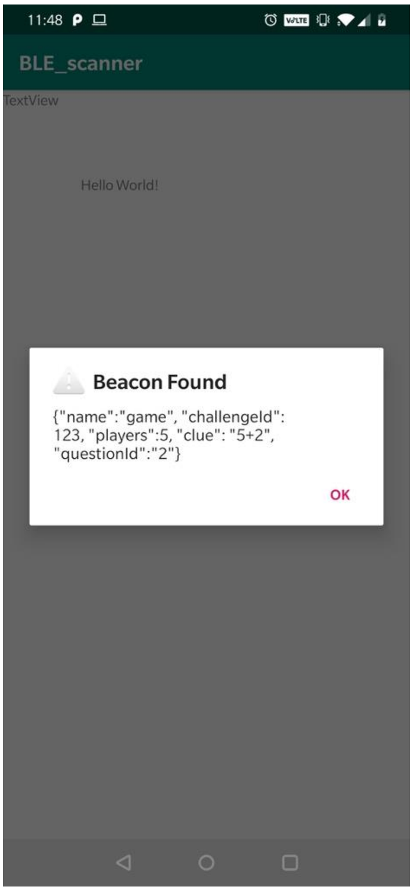
6.3.3 Validation results

As the Scavenger Hunt game is still in the development phase, we tested few of the use cases individually. The follow table shows the use cases that were evaluated in the initial validation

ID	Name	Status
MRMG_UC1	Play challenges	Partial, just getting clues and tasks
MRMG _UC4	Asset trading	Completed
MRMG _UC5	Design new challenges	Completed

6.3.3.1 Play Challenges

As the mobile gets in the range of Bluetooth of the IoT beacons, it downloads the attachment automatically from the cloud as shown in Figure 64.



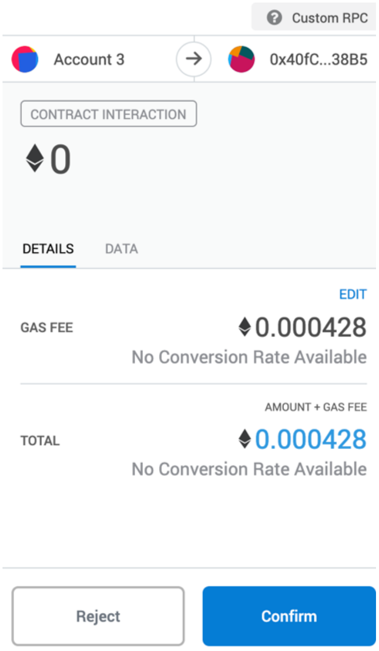
Beacon Found

{"name":"game", "challengeId": 123, "players":5, "clue": "5+2", "questionId":"2"}

Figure 64. Notification for the challenge / task

6.3.3.2 Asset trading

After the tokenization of the new assets, we just transfer the ownership of the asset. Figure 65 shows the transaction confirmation of such transfers.



♦0

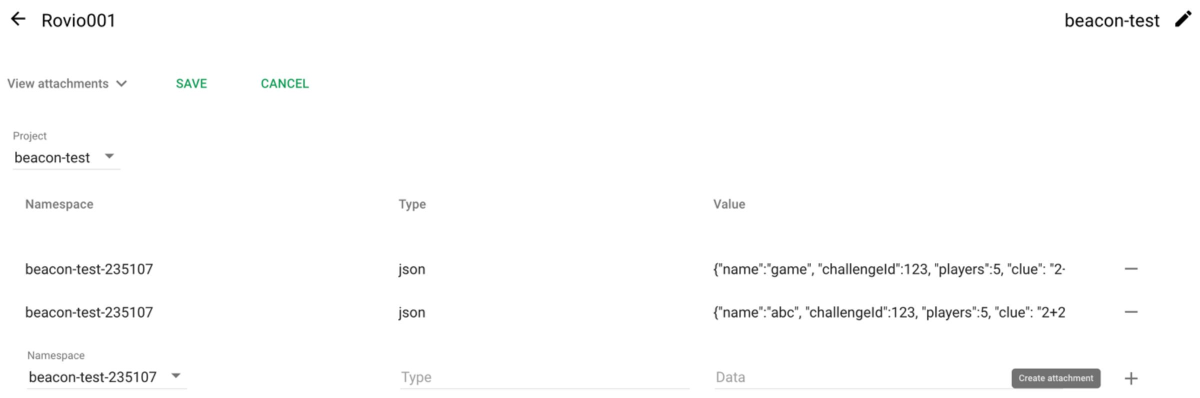
GAS FEE ♦0.000428
No Conversion Rate Available

AMOUNT + GAS FEE
TOTAL ♦0.000428
No Conversion Rate Available

Figure 65. Confirmation of transaction

6.3.3.3 Design new challenges

Figure 66 shows the web application for adding the custom task to the already deployed IoT beacons, combination of multiple tasks makes a single challenge in the game pilot. Customs tasks can also be attached using the custom API developed.



Project
beacon-test

Namespace	Type	Value
beacon-test-235107	json	{"name":"game", "challengeId":123, "players":5, "clue": "2-
beacon-test-235107	json	{"name":"abc", "challengeId":123, "players":5, "clue": "2+2

Namespace
beacon-test-235107
Type
Data

Figure 66. Attachments of clues in IoT beacons

6.3.4 Future validation tests

6.3.4.1 Implementation plan

The Table 26 shows the use cases that will be evaluated in the next phase.

Table 26. Status of use-cases to be evaluated in the mixed reality mobile gaming pilot

ID	Name	Status
MRMG_UC 1	Play challenges / tasks	In development - Game server, DLT and Mobile application are in

		implementation phase
MRMG _UC2	Redeem rewards	In development – Smart contracts are being coded.
MRMG _UC3	View In-App Advertisements	Implementation scheduled for Winter'19
MRMG _UC6	Access management	In development – Game server in implementation phase
MRMG _UC7	Offer rewards	In development – Web application and smart contracts in implementation phase
MRMG _UC8	Publish new advertisements	Implementation Scheduled for Winter'19

The implementation plan includes the following main steps:

- Implementation of all the components of Scavenge hunt game by September'19
- Unit and system tests for the game pilot
- Implementation of SOFIE components in the game pilot in Fall '19.
- Onboarding of the game pilot for Continuous Integration / development to the SOFIE platform.
- Validation tests in the end.

6.3.4.2 Validation Tests

The validation tests categories that have been planned by the end of the project are summarized in Table 27.

Table 27. List of future validation tests

Validation	Target	Measure
Player value	Gaming experience is fun and valued by the player	Play tests, questionnaires
Scale	Mass consumer reach 1 million plus active users per day supporting.	System performance, scalability, cost
Revenue opportunity	Means to generate revenue	Go-to-market channel access, business case
Technical fit	Blockchain and federation is superior solution to alternative solutions	Architecture Review

Player value:

- Internal play tests as well as later outside tests, for example on the premises of Aalto university campus.
- In the Rovio office, we can qualitatively track real-time re-actions and do retrospective testing.
- In both cases post-test questionnaires will be given and generate analytics from them.

Scale and Revenue measurement:

- Scalability is not a problem for example with Amazon Web Server and with Rovio's resources given the size of our previous games.
- We must also ensure that the gameplay and Distributed Ledger Technology would support upto 1 million users.

Business impact:

- We will calculate a business case with potential revenues from advertisers and Point of Interest partners such as Starbucks and McDonalds.
- We will also evaluate qualitatively taking into account the big marketing powers of Rovio and big POI partners.

Technical fit:

- Qualitative architecture review and an analysis of the benefits and drawbacks of DLT & IoT in such a game.
- Benefits of back-end technologies such as trading of inter-industry assets on the SOFIE platform.
- Benefits that are directly visible selling points for users such as anonymity and non-fungibility in DLT or spoof-ability of IoT beacons.

7 Conclusions

WP5 aims at setting up the four pilots of the SOFIE project and validating its federation architecture in real operating conditions. This deliverable has presented the application context, the status and the roadmap of each pilot based on an assessment of both business and technical requirements. A detailed architectural design and system specifications have been determined and a proof of concept prototype has been engineered that proves the feasibility of core SOFIE research and innovation, so that more extended SOFIE validation activities can be effectively be carried out as the pilot environment progress. The results provided in this document are meant to be a reference point for the following technical system implementation, prototype integration and overall pilot evaluation which will be reported in the following deliverables of WP5.