



SOFIE - Secure Open Federation for Internet Everywhere

779984

DELIVERABLE D5.4

Final Validation & Replication Guidelines

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	7.5.2021
Author(s)	Ioannis Oikonomidis (SYN), Filippo Vimini (LMF), Vasilios Siris (AUEB), Francesco Bellesini, Davide Minetti (EMOT), Giuseppe Raveduto (ENG), Tommaso Bragatto (ASM), Priit Anton, Mait Mardin, Margus Haavala (GT), Ahsan Manzoor, Max Samarin (ROV), Yki Kortensniemi (AALTO)
Responsible person	Ioannis Oikonomidis (SYN), oikonomidis@synelixis.com
Target Dissemination Level	Public
Status of the Document	Completed
Version	1.10
Project website	https://www.sofie-iot.eu





Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Summary of changes compared to previous version

Version	Major changes
1.10	Added “components involved” row in all pilot validation Test Cases. Added more details (s/w setup and other testing details) in pilot KPIs Evaluation sections. Added additional explanation in all pilot TRL sections. Added specific deployment details in pilot Replication sections. Added clarification in SMAUG section.



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Table of Contents

1. Introduction.....	9
1.1 Scope of this document.....	9
1.2 Structure of the deliverable	9
2. SOFIE Architecture.....	10
3. Food Supply Chain Pilot	12
3.1 Pilot overview	12
3.2 Validation.....	14
3.2.1 Final end-to-end on-site validation.....	14
3.2.2 Data collected and published.....	25
3.2.2.1 Sensor data.....	25
3.2.2.2 User feedback.....	25
3.3 Evaluation.....	28
3.3.1 Pilot performance assessment and KPIs evaluation.....	28
3.3.2 Evaluation of the Pilot's Competitive Advantage.....	31
3.3.3 TRL.....	32
3.4 Lessons learned and replication guidelines	33
3.4.1 Replication guidelines.....	33
4. Decentralized Energy Data Exchange Pilot	40
4.1 Pilot overview	40
4.2 Validation.....	40
4.2.1 Final end-to-end on-site validation.....	40
4.2.2 Data collected and published.....	45
4.3 Evaluation.....	45
4.3.1 Pilot performance assessment and KPIs evaluation.....	45
4.3.2 Evaluation of the Pilot's Competitive Advantage.....	47
4.3.3 TRL.....	48
4.4 Lessons learned and replication guidelines	48
4.4.1 Replication guidelines.....	49
5. Decentralized Energy Flexibility Marketplace Pilot.....	54
5.1 Pilot overview	54
5.2 Validation.....	55
5.3 Evaluation.....	69
5.3.1 Pilot performance assessment, KPIs evaluation, and benefits	69
5.3.2 TRL.....	74
5.4 Lessons learned and replication guidelines	74
5.4.1 Replication guidelines.....	74
6. Context-Aware Mobile Gaming Pilot	80
6.1 Pilot overview	80
6.2 Validation.....	82
6.3 Evaluation.....	85
6.3.1 Pilot performance assessment and KPIs evaluation.....	85
6.3.2 TRL.....	86
6.3.3 Business Opportunities	86



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

6.4 Replication guidelines	87
6.4.1 Replication guidelines	87
6.4.2 Limitations	89
7. SMAUG	90
7.1 Overview	90
7.2 Validation	90
7.3 Evaluation	90
7.3.1 TRL	90
7.4 Replication Guidelines	90
8. Cross pilot scenarios and testing plan	91
8.1 Cross pilot data exchange	91
8.1.1 Overview	91
8.1.2 Technical Description	91
8.1.2.1 Architecture and implementation	91
8.1.3 Validation	93
8.2 Cross pilot reward exchange	95
8.2.1 Overview	95
8.2.2 Technical Description	96
8.2.2.1 Architecture	96
8.2.2.2 Implementation	98
9. Conclusions	100
10. References	101
11. Appendix I: Food Supply Chain user questionnaire	102
12. Appendix II: Pilot Validation Matrix	104
12.1 Food Supply Chain	104
12.2 Decentralized Energy Data Exchange Pilot	110
12.3 Decentralized Energy Flexibility Marketplace	112
12.4 Context-Aware Mobile Gaming Pilot	116



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

List of Figures

Figure 1: The SOFIE Architecture.	10
Figure 2: High-level FSC pilot architecture.	12
Figure 3: FSC pilot platform deployment view.	13
Figure 4: FSC pilot's final architecture and SOFIE components' relation.	14
Figure 5: FSC Questionnaire question categories.	26
Figure 6: FSC Questionnaire results – Total.	27
Figure 7: FSC Questionnaire results – per actor type.	27
Figure 8: DEDE architecture overview.	40
Figure 9: High-level architecture of the DEFM pilot.	54
Figure 10: Deployment view of the DEFM pilot.	55
Figure 11: DEFM integration tests executed on development environment.	56
Figure 12: History of DEFM pilot's integration tests results on CI/CD environment.	56
Figure 13: Charging station locations - DEFM.	72
Figure 14: Reverse power flow (EV absence) - DEFM.	72
Figure 15: Reverse power flow (EV presence) - DEFM.	73
Figure 16: Components role in DEFM pilot's deployment.	75
Figure 17: The high-level architecture of the CAMG pilot.	80
Figure 18: The Scavenger Hunt game prototype. Starting, playing, and ending a hunt on a mobile client - CAMG.	81
Figure 19: Viewing and equipping items in Blockmoji - CAMG.	82
Figure 20: Deployment diagram of the cross-pilot scenario using the FA from the DEDE pilot.	92
Figure 21: List of the DEDE pilot services as seen in the configuration UI of the FA. Provided by the single smart meter (VcG...) and Estfeed (BRJ...).	93
Figure 22: Test client discovering the services of the DEFM pilot (ABf...).	93
Figure 23: Cross pilot reward exchange scenario.	95
Figure 24: Class Diagram.	97
Figure 25: Sequence Diagram.	97



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

List of Tables

Table 1: Validation of FSC Test Cases.....	15
Table 2: Data collected and published.....	25
Table 3: Performance and Business KPIs.	28
Table 4: SOFIE's added value compared to the existing traceability systems in FSC.....	31
Table 5: TRL levels of the FSC and its assets.....	32
Table 6: Synfield Thing Description schema.....	33
Table 7: Transportation Thing Description schema.....	34
Table 8: Aberon Thing Description schema.....	36
Table 9: Validation of DEDE Test Cases.....	41
Table 10: Performance and Business KPIs.	45
Table 11: SOFIE added value for DEDE.	48
Table 12: TRL levels of the DEDE pilot and its assets.....	48
Table 13: Services to be offered by each energy metering data source.....	50
Table 14: Functional tests.	57
Table 15: DEFM pilot collected data.....	67
Table 16: Decentralized Energy Flexibility Market KPIs.....	69
Table 17: kWh DEFM	71
Table 18: Marketplace pilot KPIs.....	73
Table 19: TRL levels of the DEFM pilot and assets	74
Table 20: DEFM pilot interfaces.	76
Table 21: Requirement validation - CAMG.	83
Table 22: CAMG performance KPIs.	85
Table 23: TRL levels of the CAMG pilot and assets.....	86
Table 24: Measured latency overhead of the FA. Mean value over 400 requests.....	94
Table 25: ERC-20 software interface.....	96



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

List of Acronyms

AMI	Advanced Metering Infrastructure
AWS	Amazon Web Services
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
FA	Federation Adapter
FSC	Food Supply Chain
GSM	Global System for Mobile communication
GUI	Graphical User Interface
GW	Gateway
HTLC	Hashed Time-Lock Contracts
IAA	Identification, Authentication and Authorization
IoT	Internet of Things
MPO	MarketPlace Owners
CAMG	Context-aware Mobile Gaming
NORM	Next generation Open Real time smart Meter
PoC	Proof of Concept
Pol	Point of Interest
PV	PhotoVoltaic
QR	Quick Response
RBAC	Role Based Access Control
RES	Renewable Energy Sources
RPF	Reverse Power Flow
SLO	Smart Locker Owners
SLR	Smart Locker Renters
SM	Supermarket



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

SSEV	Self-Sufficiency for EV
SWS	Supervisor Web Server
TPS	Transactions Per Second
TSO	Transmission System Operator
TR	Transportation
TRL	Technology Readiness Level
V2G	Vehicle to Grid
WH	Warehouse



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

1. Introduction

1.1 Scope of this document

This deliverable summarizes the final system software platform architecture and final end-to-end validation results of the four SOFIE pilots. For each pilot, the final validation results (from stakeholders' perspective) of the pilot platforms with the latest versions of the SOFIE components are reported. Also, the evaluation of the KPIs defined in the Deliverable D5.1 "Baseline System and Measurements", is included, along with a business-focused evaluation summary and the Technology Readiness Level achieved. Then, the replication guidelines for external parties wishing to join a pilot platform or develop a SOFIE-compliant component are provided for each pilot. In addition to the pilots, an update on SOFIE's reference application, Secure Marketplace for Access to Ubiquitous Goods (SMAUG), is included. Finally, two cross-pilot cases are described, one that focuses on the technical aspects of data exchange between pilot platforms, and one that focuses on the business value from combining the pilot platforms.

1.2 Structure of the deliverable

This document first provides an overview of the SOFIE Architecture used to enable secure federation in the pilots. Then, an individual section is devoted to each of the four SOFIE pilots, containing an overview (including the pilot platform architecture updates), the end-to-end (integration) validation results, the evaluation results, and finally the replication guidelines. These sections are structured in the same way by using the following four subsections:

- Subsection X.1 presents an overview of the pilot, summarising its application context and any updates from the previously reported version, any updates on the architecture of the pilot platform (including architecture diagrams, e.g., high-level overview and deployment diagrams)
- Subsection X.2 presents the integrated, end-to-end validation results of the final pilot platform versions that were deployed on-site (where applicable)
- Subsection X.3 presents the evaluation results, based on the KPIs table that was defined in Deliverable D5.2 Initial validation results. Also, the TRL reached by the pilot and its assets is summarized.
- Subsection X.4 reports and presents replication guidelines for external parties that wish to join the corresponding pilot platform and/or develop their own SOFIE-compliant platform

Next, updates on the reference application (namely SMAUG) that utilizes and demonstrates all SOFIE framework components are reported.

A section then covers the updated descriptions of cross-pilot cases that were developed in the context of the SOFIE pilots.

The final chapter concludes the deliverable by summarising some key contents of this document.



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

2. SOFIE Architecture

The *SOFIE Architecture* enables interoperability between IoT systems in an open and secure manner by *federating the actions between the different IoT systems using interledger technologies*. The Architecture consists of 6 components that enable key functionalities for federation scenarios, plus federation adapters used to connect existing IoT systems to the architecture *without requiring changes to the IoT systems*. The architecture can be extended to support different use cases and the individual components can be implemented using technologies that best suit the context. The Architecture has been described in SOFIE Deliverable D2.6 [D2.6].

The *SOFIE Framework* is an example implementation of the Architecture designed to support the SOFIE pilots (Deliverable D5.2 “Initial validation results”) and to fulfill the requirements set in D2.6. The Framework has been described in SOFIE Deliverable D2.7 [D2.7], and the complete Framework, detailed technical documentation, and multiple examples of how the components and adapters can be utilised are all available as open-source software in the GitHub [Framework].

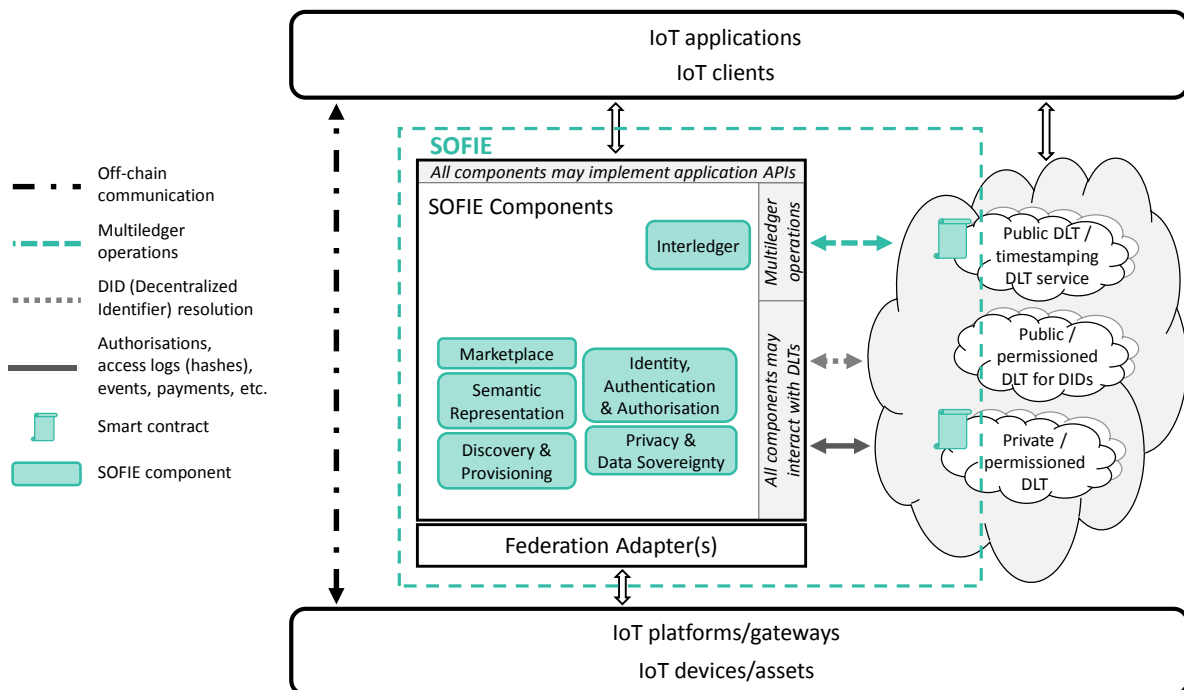


Figure 1: The SOFIE Architecture.

A key element of the SOFIE Architecture, depicted in Figure 1, is that it is a *framework architecture* that defines the types of functionalities provided by the components and adapters, but not an exhaustive list of supported functions. This is due to the fact that SOFIE is intended to support IoT federation in many application areas, therefore it is infeasible to define a set of functions that would encompass all the needs (including future needs) of the different application areas. Instead, the Architecture defines key functionalities for federation and provides example implementations of each component and adapter in the SOFIE Framework. The provided examples are based on the pilots in the SOFIE project and they can be freely adapted and expanded to suit the needs of other applications.

Next, we go through the level of the architecture from the bottom level to the top-most layer. The lowest level of the architecture contains all the IoT systems. This includes the *IoT assets* (or resources), e.g., IoT sensors for sensing the physical environment, actuators for acting on



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

the physical environment, and boxes with RFID tags that are used to transport products which can be connected to or be integrated in actual devices. *IoT platforms* include platforms with data stores, where e.g., the measurements from sensors are collected and made available to third parties, as well as servers providing IoT services.

The *federation adapter(s)* are used to interface the IoT systems with the Architecture. This allows the IoT systems to interact with SOFIE without requiring any changes to the IoT systems themselves. Different scenarios and pilots can utilise different types of federation adapters, which expose only the required parts of the SOFIE functionality to the IoT platform.

Of the six components, the architecture emphasises the *interledger component* responsible for interconnecting the different types of *Decentralised Ledger Technologies* (DLTs), which can have quite different features and functionality. Public (or permissionless) DLTs offer wide-scale decentralised trust and immutability but incur a higher cost and latency. On the other hand, permissioned or consortium DLTs have a lower, or even zero, transaction cost and low latency, but their trustworthiness is determined by the peers in the set of permissioned nodes that participate in the DLT's consensus mechanism. Moreover, the level of privacy afforded also differs: the transactions and data on public/permissionless blockchains are completely open to everyone (public), while private/permissioned DLTs can arrange for their records to be visible only to permissioned nodes (private), or make them readable by anyone (public), but writing is always limited to the permissioned nodes. Finally, DLTs can also differ in the functionality they provide: a DLT can focus, e.g., on cryptocurrency payments, recording of IoT events, access authorisation, or providing resolution of *Decentralised Identifiers* (DIDs). Utilising multiple ledgers that are interconnected through interledger functionality, instead of a single DLT, provides the flexibility to exploit these trade-offs.

The other SOFIE framework components are: *Identity, Authentication, and Authorisation*, which provides identity management and supports multiple authentication and authorisation techniques; *Privacy and data sovereignty*, which provides mechanisms that enable data sharing in a controlled and privacy preserving way and supports privacy preserving surveys using differential privacy; *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 decentralised way; and *Provisioning & Discovery*, which provides functionality for the management and discovery of services.

Finally, all the components can expose *application APIs*, which provide the interfaces for IoT clients and applications to interact with the SOFIE components. Also, the framework adapters and IoT applications can communicate directly either through the DLTs or using off-chain channels. In Figure 1, the multiledger operations are positioned next to the Interledger component as it is mostly using that functionality, but any of the other components can also utilise multiledger operations when required. The figure also does not show the interactions between the components – these are described in more detail in Deliverable D2.7.

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

3. Food Supply Chain Pilot

3.1 Pilot overview

The focus of the *Food Supply Chain* (FSC) pilot is to demonstrate the use of the SOFIE architecture and framework components in the food supply chain and validate a provenance *Business Platform* (BP) that offers two main important services, 1) a traceability service used by the consumers to access the full history of grapes from the field to the supermarket shelf, and 2) an audit service used by the supermarket company to verify the integrity of data which is collected as grapes are transferred over the supply chain as well as relevant business rules (driven by this data) which have been agreed with the suppliers.

In this section, the final version of the *Food Supply Chain* (FSC) pilot platform is described. It has not changed from the version described and presented in D5.3, “End-to-end Platform Validation”, so, we provide a brief summary of its architecture and its deployment view to highlight the platform’s deployment on-site. In addition, a figure which depicts the link between the pilot’s architecture and the SOFIE framework components that have been used in the FSC pilot is included.

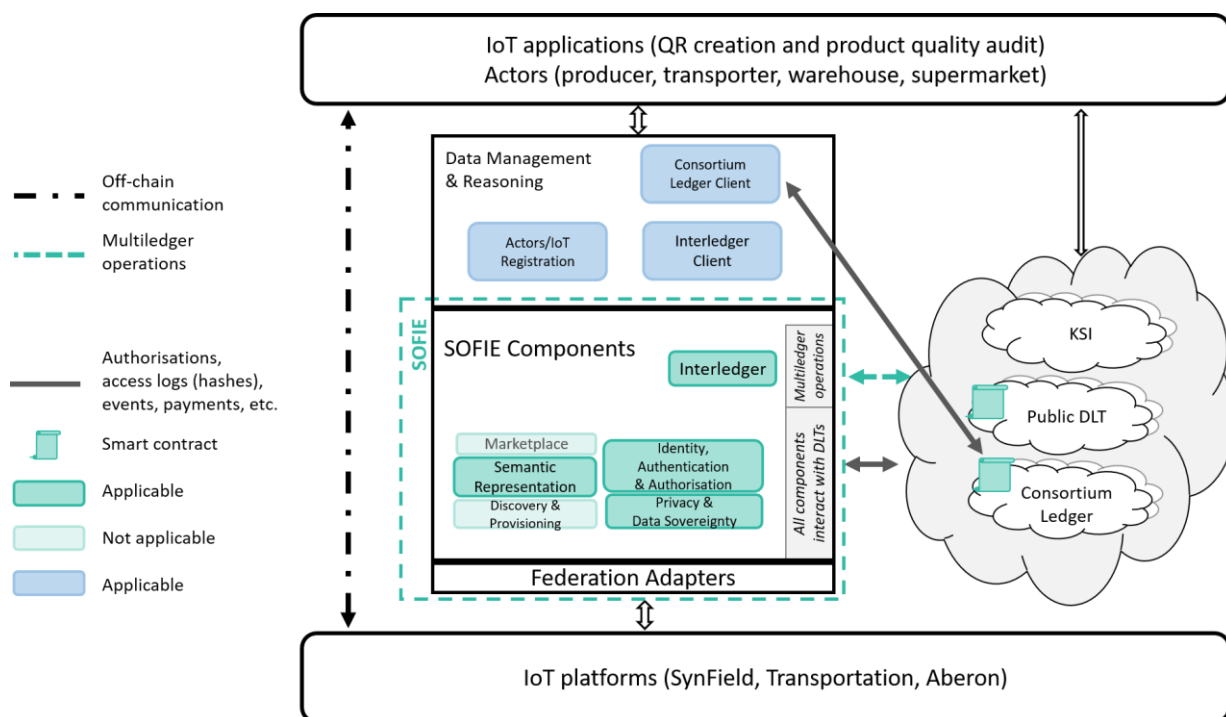


Figure 2: High-level FSC pilot architecture.

Figure 2 depicts the platform’s high-level architecture. The SOFIE components that have been utilized are shown in this architecture view and are listed below:

- Federation Adapters (FA) (one for each IoT platform)
- Identity, Authentication, Authorisation (IAA)
- Privacy and Data Sovereignty (PDS)
- Semantic Representation (SR)
- Interledger (IL)

The pilot-specific software components are also shown in this figure, including the *Supervisor Web Server* (SWS) component, which offers a public API for the internal services provided by

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

the Data Management and the Reasoning sub-components (i.e., Actors/IoT registration, Interledger client, Consortium Ledger Client).

This architecture offers two main applications (bundled in a web application), namely the usage of QR codes to encode *product history* from the field to the market shelf, and product *quality audits and resolution of disputes* in product quality degradation events. Both services, as well as other simple services, are provided to the pilot actors through an FSC web application.

Figure 3 presents the Deployment view of the FSC Pilot platform. Starting from the bottom layer, the Federation Adapter components are deployed on-site, at the IoT platform premises, i.e., at the farm, in the transportation vehicle, and in the warehouse. The adaptation follows the Federation paradigm: it does not require any modifications on the IoT platform side, but it adds the functionality required by each IoT platform to connect to the Supervisor Web Server component on top, in a separate component named Federation Adapter (FA). The Supervisor Web Server component along with the SOFIE components offer an API to the user's application, deployed on the pilot's cloud infrastructure that has been setup for this purpose. This is also where the Consortium ledger resides. As expected, the Public ledger is an external entity to the pilot.

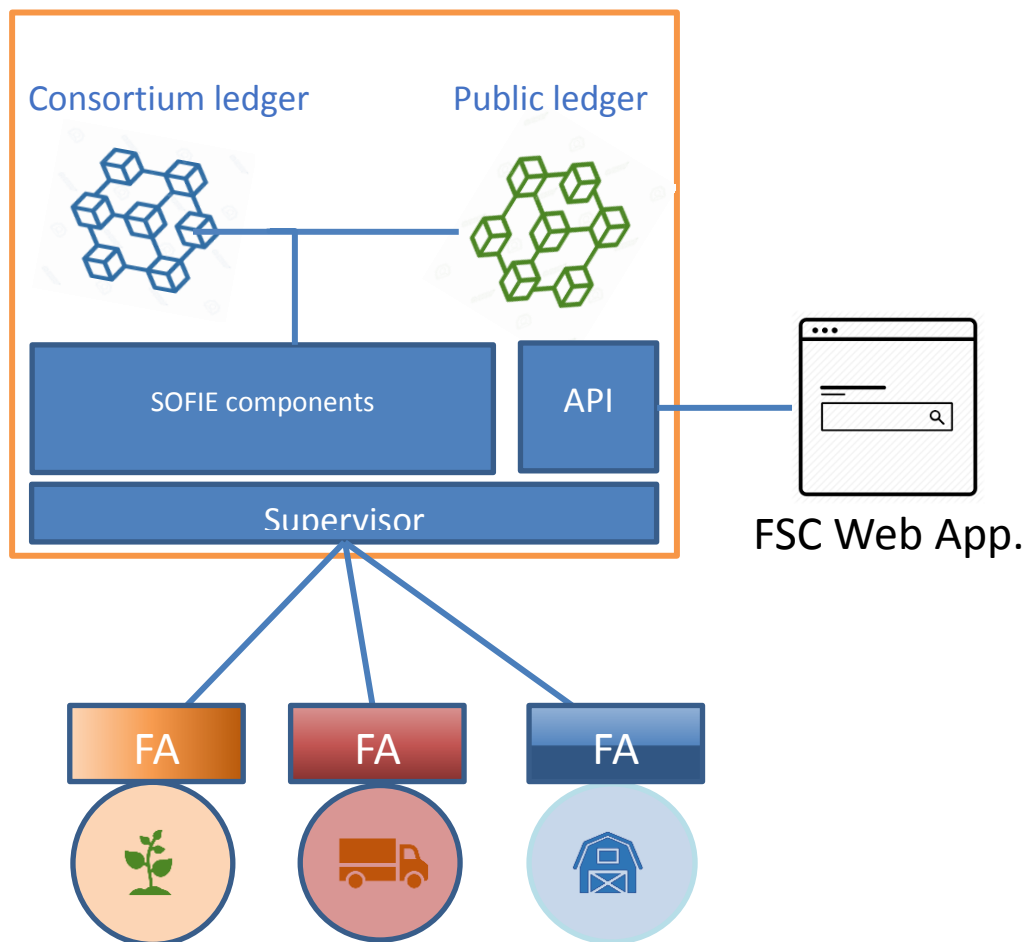


Figure 3: FSC pilot platform deployment view.

The role/usage of the SOFIE components in the FSC pilot is shown in Figure 4. As shown at the bottom of this figure, three IoT platforms are federated:

1. the SynField IoT platform that collects measurements about the growing conditions in the field,

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

2. a Transportation IoT platform that collects measurements about the produce as they are transferred from one site to another, and
3. the Aberon IoT platform that is responsible for collecting measurements related to the storage conditions of produce in the warehouse.

A Federation adapter has been developed for each of these IoT platforms and has been applied on top of the northbound API of each IoT environment to adapt the corresponding data and metadata using the SOFIE Semantic Representation component and, also, to support authentication and Interledger procedures.

Further updates on the pilot platform and its functionalities will be considered in the future, depending on the needs of potential customers and users of the platform. For the time being, this is considered as the final version of the pilot platform.

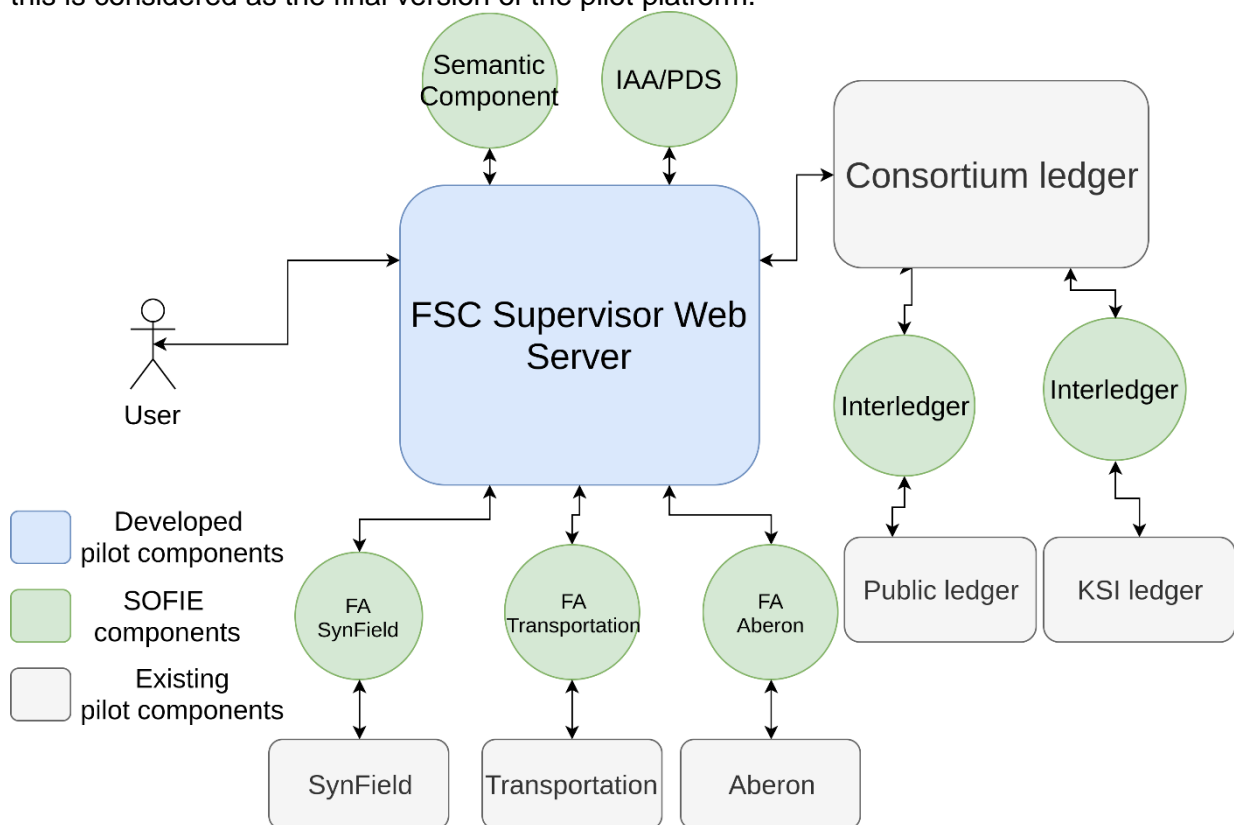


Figure 4: FSC pilot's final architecture and SOFIE components' relation.

3.2 Validation

3.2.1 Final end-to-end on-site validation

The final part of the validation of the Food Supply Chain pilot platform was performed on-site, i.e., the end-users tested the pilot platform while the IoT components (sensors, gateway, mobile devices) were deployed at the field, in the transportation van, and in the warehouse. The goal of this validation differs from the previous validation activities; whereas the goal of the previous validation (reported in D5.3, "End-to-End platform validation") was to validate and verify the proper functionality of all the services offered by the pilot platform in the context of the test cases that were described in D5.1, "Baseline system and measurements", the final validation part reported in this deliverable aims to go through the same test cases but this time by performing the actions (steps) on-site with real users and real-world conditions. Table 1 contains the updated validation results, also including screenshots from the validation results taken from the web application and also during the trials in the pilot trials in Kiato, Greece in September 2020





Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

by Synelixis (producers, transporters, warehouse and supermarket employees were involved). All validation tests were successfully passed; hence, the pilot platform fulfilled the goals that have been set in the beginning of the project. In addition, the pilot included integration tests allowed the pilot to reach maturity level “5” in the project CI/CD pipeline (D3.3 [D3.3] *Business Platforms, Pilot Release*): they are executed automatically every time a new version of the platform is developed, and the build process ends successfully only if all the tests pass on the CD test deployment.

Table 1: Validation of FSC Test Cases.


Test ID	FSC_TC01																						
Test description	Measurements from each deployed sensing device are collected by the corresponding IoT platform and they are properly stored in its database system.																						
Test location	IoT platforms used to monitor field, transportation and warehouse sites																						
Related use cases	FSC_UC2, FSC_UC4, FSC_UC6, FSC_UC7, FSC_UC8, FSC_UC9																						
Related requirements	REQ_FSC0.4, REQ_FSC 7.1, REQ_FSC 9.1, REQ_FSC 9.2, REQ_FSC 14.1																						
Feature(s) under test	Metering & data collection																						
Components involved	SynField IoT platform, Aberon IoT platform, Transportation IoT platform																						
Test environment	Devices are placed on site and the IoT platforms are operational. The IoT platform provides an endpoint to retrieve (past) data which has been collected from an integrated sensing device.																						
Dependencies	N/A																						
Steps	<ol style="list-style-type: none"> 1. Sensing devices are deployed on site and they are properly configured to communicate and send data to the corresponding IoT platform. 2. Collect data for a given period of time (e.g., few days) 3. Use IoT platform API to retrieve data from each integrated sensing devices within a specific time period. 																						
Pass criteria	All relevant measurement values are properly retrieved.																						
Result	<div> <div>LogHandoverWarehouseTransport</div> <div> <div>2138899</div> <div>0xcd3264d25c9e9406ccf40d6beadcc1da323c7805fd1b2125b0eb777983a695a</div> </div> <table> <tr> <td>Source actor</td><td>df658f09-c471-4c63-85ba-7197fa28b7ca</td></tr> <tr> <td>Source platform</td><td>0x99245a929029D8b5F6C12b7d80158f71fAC19198</td></tr> <tr> <td>Destination actor</td><td>1f6b5af2-d297-41cf-936b-cef365863c60</td></tr> <tr> <td>Destination platform</td><td>0x35A69278FEA8d80d9490B64cD52915575149A898</td></tr> <tr> <td>Minimum temperature</td><td>-1 C</td></tr> <tr> <td>Average temperature</td><td>-0.5 C</td></tr> <tr> <td>Maximum temperature</td><td>0 C</td></tr> <tr> <td>Minimum humidity</td><td>84 %</td></tr> <tr> <td>Average humidity</td><td>84.5 %</td></tr> <tr> <td>Maximum humidity</td><td>85 %</td></tr> <tr> <td>Transport</td><td>0x46ff9921da43e9b388eaa5fc7b1166abc9965641a82d188da2cdf782ef6058e6</td></tr> </table> </div>	Source actor	df658f09-c471-4c63-85ba-7197fa28b7ca	Source platform	0x99245a929029D8b5F6C12b7d80158f71fAC19198	Destination actor	1f6b5af2-d297-41cf-936b-cef365863c60	Destination platform	0x35A69278FEA8d80d9490B64cD52915575149A898	Minimum temperature	-1 C	Average temperature	-0.5 C	Maximum temperature	0 C	Minimum humidity	84 %	Average humidity	84.5 %	Maximum humidity	85 %	Transport	0x46ff9921da43e9b388eaa5fc7b1166abc9965641a82d188da2cdf782ef6058e6
Source actor	df658f09-c471-4c63-85ba-7197fa28b7ca																						
Source platform	0x99245a929029D8b5F6C12b7d80158f71fAC19198																						
Destination actor	1f6b5af2-d297-41cf-936b-cef365863c60																						
Destination platform	0x35A69278FEA8d80d9490B64cD52915575149A898																						
Minimum temperature	-1 C																						
Average temperature	-0.5 C																						
Maximum temperature	0 C																						
Minimum humidity	84 %																						
Average humidity	84.5 %																						
Maximum humidity	85 %																						
Transport	0x46ff9921da43e9b388eaa5fc7b1166abc9965641a82d188da2cdf782ef6058e6																						


Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

	 
Test ID	FSC_TC02
Test description	Each registered actor of any type (e.g., producer, transporter, warehouse, or supermarket employee) can access and utilise all the services provided by the FSC web application based on their role.
Test location	Field/warehouse/supermarket location
Related use cases	FSC_UC1-FSC_UC12, FSC_UC14
Related requirements	REQ_FSC0.1, REQ_FSC0.2, REQ_FSC0.3
Feature(s) under test	AAA
Components involved	Supervisor Web Server, FSC Web Application
Test environment	The actor has already registered in the pilot platform. SOFIE platform has been deployed in the production environment. The tablet used by the actor must have internet connection.
Dependencies	N/A
Steps	<ol style="list-style-type: none"> 1. The actor initiates an HTTPS session to the FSC web application login page. 2. The HTTPS traffic is intercepted, and the authorization is initiated by the Authentication Server (AS) of the SOFIE platform. The login page is sent to the actor. 3. The actor enters a username and password, which are sent to the AS of the SOFIE platform. 4. The OAuth2.0 server authenticates the actor and creates a unique token that is used to enable role-based access to FSC web application resources.
Pass criteria	Actor's access policy is activated. The actor is able to access FSC web application resources.
Result	Below are some pictures of the results from performing the steps




Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines				
Security:	Public	Date:	7.5.2021	Status:	Completed
		Version:	1.10		




 **SOFIE Login**

Access the SOFIE dashboard with your **keycloak** credentials

Username




Password



ENTER

Food Supply Chain Dashboard

LOGOUT



Home

Actions

Register Box

Register Box Session

Handover from producer

Handover to warehouse

Handover from wareho...

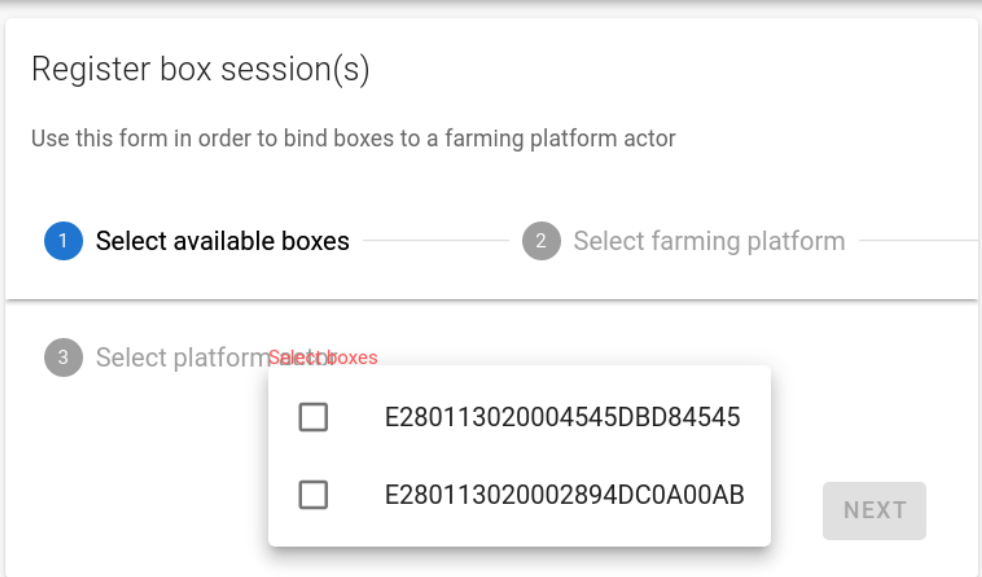
Handover to supermark...

Welcome transporter. Use the menu on the left to perform actions on the Food Supply Chain.


SOFIE H2020 project © Synelxis Solutions S.A. 2019

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10



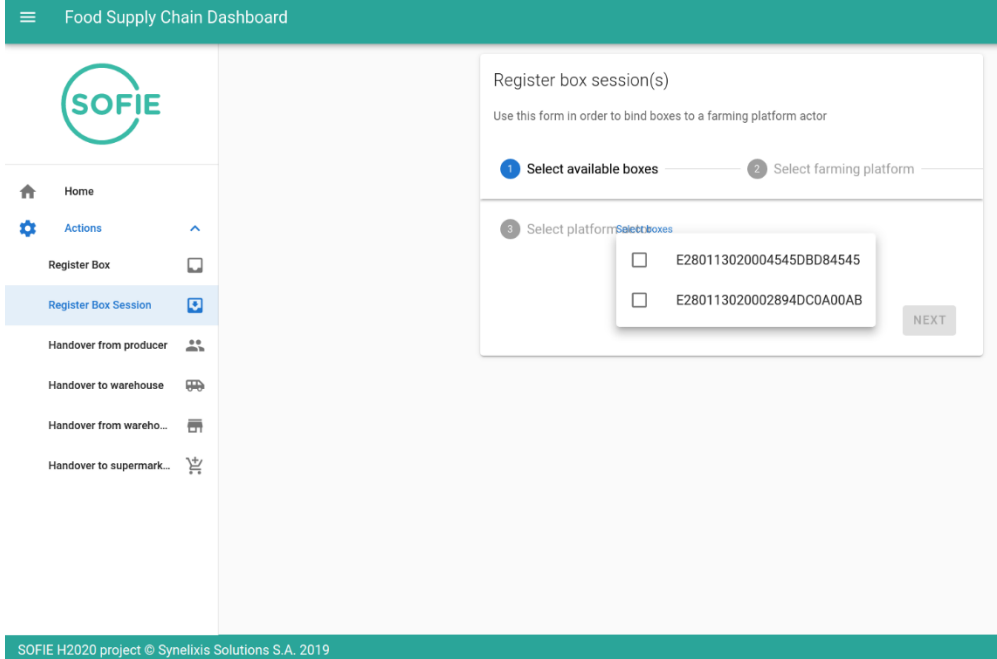

Test ID	FSC_TC03
Test description	Test that box reuse is possible (after its release) and that registration of a box with an ID that is already used by another box is impossible (box unique identifier).
Test location	On site by using the FSC web application.
Related use cases	FSC_UC5, FSC_UC12
Related requirements	REQ_FSC 5.1, REQ_FSC 5.2, REQ_FSC 12.1
Feature(s) under test	Asset management
Components involved	Supervisor Web Server, FSC Web Application, Transportation IoT platform, SR, IAA, PDS, Transportation FA
Test environment	An actor uses the FSC web application to perform the action under test.
Dependencies	FSC_TC02 is successful.
Steps	<ol style="list-style-type: none"> 1. An actor (transporter) enters its profile in the FSC web applications and activates register box action. 2. The actor provides as input to the action a box ID which has been already registered in the used DLT. 3. The actor provides as input the ID of a released box.
Pass criteria	Registration of a box with an already used ID (by another box) is prohibited. Reuse of a released box is possible.
Result	<p>Below are some pictures of the results from performing the steps</p> 

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines				
Security:	Public	Date:	7.5.2021	Status:	Completed
				Version:	1.10

	
Test ID	FSC_TC04
Test description	Presence of a group of boxes (RFID tags) is detected as they are placed/removed inside the truck.
Test location	Inside the truck
Related use cases	FSC_UC3, FSC_UC4, FSC_UC7, FSC_UC8
Related requirements	REQ_FSC 5.2
Feature(s) under test	Metering & data collection, Asset management
Components involved	Transportation IoT platform, Transportation FA, SR, IAA, PDS, Supervisor Web Server, FSC Web Application
Test environment	GW and sensors of the transportation IoT platform have been deployed inside the truck. RFID reader has been calibrated to scan certain area of the truck. Tags are attached to the boxes.
Dependencies	N/A
Steps	<ol style="list-style-type: none"> 1. A number of boxes are placed inside the truck at a certain time instant. 2. Some of the boxes are removed from the truck at another time instant. 3. The boxes which were removed in step 2 are placed again inside the truck at a third time instant but in a different location (inside the RFID range)
Pass criteria	The presence of all the boxes inside the truck is properly detected by the transportation IoT platform at all times (taking also into account the delay in collecting measurements)
Result	Below are some pictures of the results from performing the steps

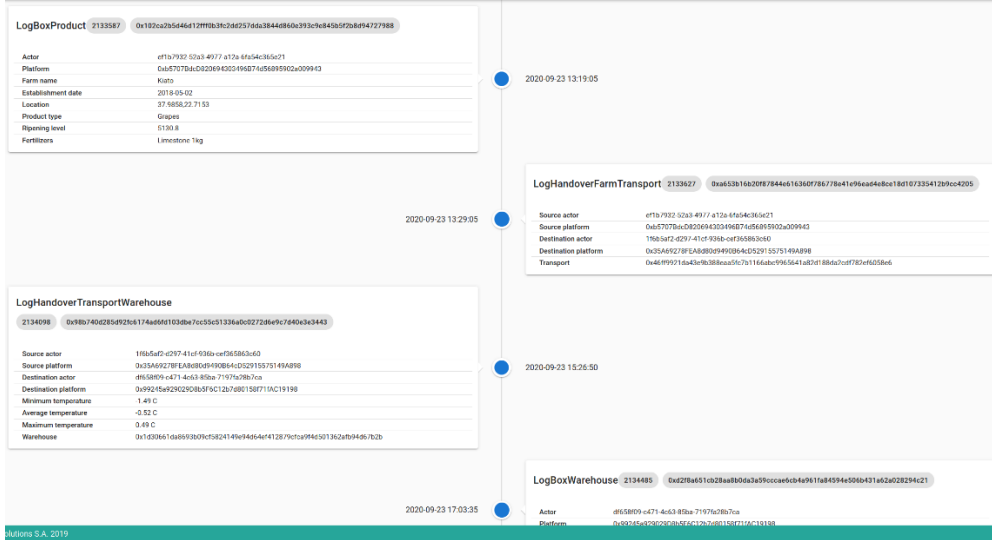



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10


		
		
Test ID	FSC_TC05	
Test description	The SOFIE platform receives data from the transportation GW deployed in the truck both when the vehicle is moving and when the vehicle has been switched off.	
Test location	As the vehicle moves and makes stops from one place to another (on the road)	
Related use cases	FSC_UC7	
Related requirements	REQ_FSC 6.1	
Feature(s) under test	Metering and data collection services	
Components involved	Transportation IoT platform, Transportaion FA, SR, IAA, PDS, Supervisor Web Server, FSC Web Application	
Test environment	GW and sensors of the transportation IoT platform have been deployed inside the truck. Tags are attached to the boxes. The truck is moved and 3G/4G coverage exists in the followed route.	
Dependencies	FSC_TC04 is successful.	
Steps	1. A group of boxes is placed inside the truck. 2. At a certain time the truck starts to move from site A to site B.	



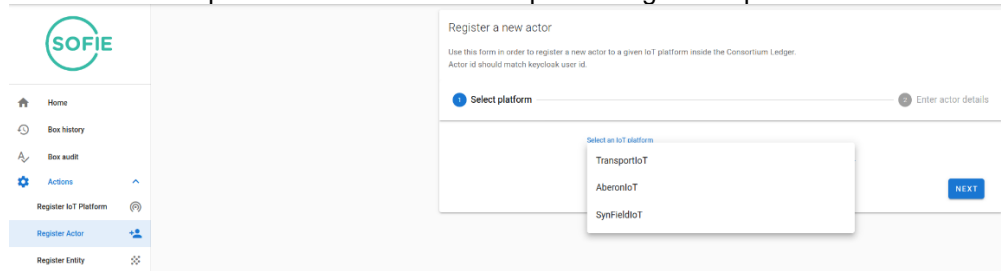
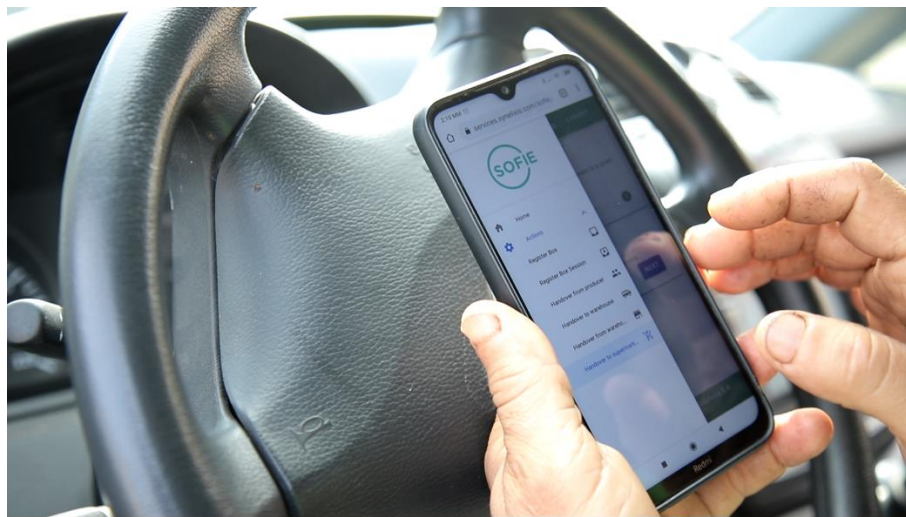
Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

	<p>2. Before reaching its destination, the truck stops for a certain period of time and its engine is turned off for a certain period of time (few minutes).</p> <p>3. The engine is turned on and the truck moves to reach site B.</p>
Pass criteria	The presence of the boxes inside the truck is continually detected (given the used time resolution in collecting data from the truck) from A to B.
Result	<p>Below some pictures from the results when performing the steps</p>  
Test ID	FSC_TC06
Test description	Data and metadata provided by the actors through the FSC web application are recorded in DLTs. The payload of any transaction is verified.
Test location	On site, depending on the type of activity
Related use cases	FSC_UC1-FSC_UC12, FSC_UC14
Related requirements	REQ_FSC0.5, REQ_FSC1.1, REQ_FSC3.1, REQ_FSC 5.3, REQ_FSC 5.4, REQ_FSC 6.2, REQ_FSC 14.12, Security Challenge #1
Feature(s) under test	Asset management, User interaction
Components involved	SynField IoT platform, Aberon IoT platform, Transportation IoT platform, SynField FA, Aberon FA, Transportation FA, SR, IAA, PDS, Supervisor Web Server
Test environment	An actor uses the FSC web application to perform the action under test.
Dependencies	FSC_TC02 is successful.
Steps	1. An actor accesses the FSC web application and activates an action.

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10


	<div><div>2. The actor performs any operation in the physical world needed to complete the action (e.g., onboarding the boxes in the truck) and inputs the necessary (meta)data.</div><div>3. The actor completes the action (thus data is recorded in the DLTs)</div><div>4. The actor accesses the logs of the performed operation and verifies that information recorded in the DLTs is correct.</div></div>																																																																																		
Pass criteria	Data of the transaction which is stored in the DLTs matches the relative activity and metadata.																																																																																		
Result	<div><div>Below are some pictures of the results from performing the steps</div><div><div><div>LogBoxProduct</div><div><div>1649770</div><div>0xe872184f16761bd79a84243c589e10720095dcc9505866fd469e3e05c90e5dcd</div></div><div><table><tr><td>Actor</td><td>ef1b7932-52a3-4977-a12a-6fa54c365e21</td></tr><tr><td>Platform</td><td>0xb5707BdcD820694303496B74d56895902a009943</td></tr><tr><td>Farm name</td><td>Kiato</td></tr><tr><td>Establishment date</td><td>2018-05-02</td></tr><tr><td>Location</td><td>κ. Μουλκιου, Μούλκι, Δήμος Σικυωνίων, Corinthia Regional Unit, Peloponnese Region, Peloponnese, West Greece and Ionian Sea, 20200, Greece</td></tr><tr><td>Product type</td><td>Grapes</td></tr><tr><td>Ripening level</td><td>3675.5</td></tr><tr><td>Fertilizers</td><td>Limestone 1kg</td></tr></table></div></div></div><div><div><table><tr><th>Name</th><th>Block #</th><th>Log Index</th><th>Tx Index</th><th>Tx Hash</th><th>Args</th></tr><tr><td>LogPlatformRegistered</td><td>39</td><td>0</td><td>0</td><td>0x1c79a89a0d7c4e0b94773e00479ba500ac324a0c72e0ab70e072177</td><td>Q</td></tr><tr><td>LogPlatformRegistered</td><td>39</td><td>0</td><td>0</td><td>0xb6273931c449f832d1b1c1d1721182c807e0b9f49c39bc7213d7748393</td><td>Q</td></tr><tr><td>LogActorRegistered</td><td>61</td><td>0</td><td>0</td><td>0xd7c793a4f9e063307a0d73ed7e733049a0b05a118453280c0ee338e09</td><td>Q</td></tr><tr><td>LogActorRegistered</td><td>62</td><td>0</td><td>0</td><td></td><td>Q</td></tr><tr><td>LogBoxSessionStarted</td><td>72</td><td>0</td><td>0</td><td></td><td>Q</td></tr><tr><td>LogBoxSessionStarted</td><td>81</td><td>0</td><td>0</td><td></td><td>Q</td></tr><tr><td>LogBoxSessionStarted</td><td>82</td><td>0</td><td>0</td><td></td><td>Q</td></tr><tr><td>LogDataCropBlobAdded</td><td>110</td><td>0</td><td>0</td><td></td><td>Q</td></tr><tr><td>LogDataCropBlobAdded</td><td>112</td><td>0</td><td>0</td><td></td><td>Q</td></tr><tr><td>LogDataCropBlobAdded</td><td>112</td><td>1</td><td>1</td><td></td><td>Q</td></tr></table><div><div>Event Arguments</div><div><div>id</div><div>1b5b45e8-2919-4785-b723-222a26571a37</div></div><div><div>role</div><div>0</div></div><div><div>platform</div><div>0xb5707BdcD820694303496B74d56895902a009943</div></div><div><div>timestamp</div><div>1560860258</div></div><div>CLOSE</div></div></div></div><div></div></div>	Actor	ef1b7932-52a3-4977-a12a-6fa54c365e21	Platform	0xb5707BdcD820694303496B74d56895902a009943	Farm name	Kiato	Establishment date	2018-05-02	Location	κ. Μουλκιου, Μούλκι, Δήμος Σικυωνίων, Corinthia Regional Unit, Peloponnese Region, Peloponnese, West Greece and Ionian Sea, 20200, Greece	Product type	Grapes	Ripening level	3675.5	Fertilizers	Limestone 1kg	Name	Block #	Log Index	Tx Index	Tx Hash	Args	LogPlatformRegistered	39	0	0	0x1c79a89a0d7c4e0b94773e00479ba500ac324a0c72e0ab70e072177	Q	LogPlatformRegistered	39	0	0	0xb6273931c449f832d1b1c1d1721182c807e0b9f49c39bc7213d7748393	Q	LogActorRegistered	61	0	0	0xd7c793a4f9e063307a0d73ed7e733049a0b05a118453280c0ee338e09	Q	LogActorRegistered	62	0	0		Q	LogBoxSessionStarted	72	0	0		Q	LogBoxSessionStarted	81	0	0		Q	LogBoxSessionStarted	82	0	0		Q	LogDataCropBlobAdded	110	0	0		Q	LogDataCropBlobAdded	112	0	0		Q	LogDataCropBlobAdded	112	1	1		Q
Actor	ef1b7932-52a3-4977-a12a-6fa54c365e21																																																																																		
Platform	0xb5707BdcD820694303496B74d56895902a009943																																																																																		
Farm name	Kiato																																																																																		
Establishment date	2018-05-02																																																																																		
Location	κ. Μουλκιου, Μούλκι, Δήμος Σικυωνίων, Corinthia Regional Unit, Peloponnese Region, Peloponnese, West Greece and Ionian Sea, 20200, Greece																																																																																		
Product type	Grapes																																																																																		
Ripening level	3675.5																																																																																		
Fertilizers	Limestone 1kg																																																																																		
Name	Block #	Log Index	Tx Index	Tx Hash	Args																																																																														
LogPlatformRegistered	39	0	0	0x1c79a89a0d7c4e0b94773e00479ba500ac324a0c72e0ab70e072177	Q																																																																														
LogPlatformRegistered	39	0	0	0xb6273931c449f832d1b1c1d1721182c807e0b9f49c39bc7213d7748393	Q																																																																														
LogActorRegistered	61	0	0	0xd7c793a4f9e063307a0d73ed7e733049a0b05a118453280c0ee338e09	Q																																																																														
LogActorRegistered	62	0	0		Q																																																																														
LogBoxSessionStarted	72	0	0		Q																																																																														
LogBoxSessionStarted	81	0	0		Q																																																																														
LogBoxSessionStarted	82	0	0		Q																																																																														
LogDataCropBlobAdded	110	0	0		Q																																																																														
LogDataCropBlobAdded	112	0	0		Q																																																																														
LogDataCropBlobAdded	112	1	1		Q																																																																														
Test ID	FSC_TC07																																																																																		
Test description	Metadata related to an actor's activity (in the FSC app.) is accessible by that actor at any time and is invisible to any other actor.																																																																																		

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Test location	On site by using the FSC web application
Related use cases	FSC_UC1-FSC_UC12, FSC_UC14
Related requirements	REQ_FSC14.3, REQ_FSC14.4
Feature(s) under test	AAA, User interaction
Components involved	Supervisor Web Server, FSC Web Application, Transportation IoT platform, Transportation FA, SR, IAA, PDS
Test environment	An actor uses the FSC web application to perform the action under test.
Dependencies	FSC_TC02 is successful.
Steps	<ol style="list-style-type: none"> 1. An actor logs in using his profile in the FSC web application. 2. The actor performs a number of actions 3. The actor confirms that he can access the logs of all performed actions and that recorded information per (trans)action is correct. 4. The actor tries to access a view/endpoint for which he does not have the authority (based on his role).
Pass criteria	Access of each actor to its own resources is allowed, while access to other resources is prohibited.
Result	<p>Below are some pictures of the results from performing the steps</p>  
Test ID	FSC_TC08
Test description	A QR code which is created by the supermarket employee using the FSC web application can be read offline by using different smartphones devices. Readability of all included information is confirmed.
Test location	On site by using a smartphone.
Related use cases	FSC_UC13
Related requirements	REQ_FSC2.1, REQ_FSC 11.1, REQ_FSC 11.4, REQ_FSC 11.5, REQ_FSC 13.1



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Feature(s) under test	User interaction.
Components involved	IL, Supervisor Web Server, FSC Web Application
Test environment	A QR code has been attached to a product package. The smartphones used have a QR reading application.
Dependencies	N/A
Steps	1. The customer uses a smartphone to read information encoded in the QR code of the package. 2. The action is repeated by using five different smartphone devices/QR reading applications.
Pass criteria	The revealed information includes (at least) the following information: farm location, type of product, harvesting date, used fertilizers, packetizing date, ID of used box and session ID.
Result	Below is a picture of the results from performing the steps 
Test ID	FSC_TC9
Test description	Test that the audit service can access/process data streams containing relevant information and discard requests containing irrelevant information, e.g., improper box ID and session ID.
Test location	On site by using the FSC web application
Related use cases	FSC_UC14
Related requirements	REQ_FSC 14.1
Feature(s) under test	User interaction
Components involved	IL, Supervisor Web Server, FSC Web Application
Test environment	A quality issue is detected on a product package. A QR code has been attached to the package.
Dependencies	FSC_TC08 is successful.
Steps	1. The supermarket employee scans the QR code attached to the product. 2. The supermarket employee requests an audit by accessing the corresponding service in the FSC web application and providing box ID and session ID values.
Pass criteria	Audit services are properly executed once relevant data is provided, whereas they are aborted in cases of irrelevant data.
Result	Below are some pictures of the results from performing the steps



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Transaction hash

0x0c0c0f0f77b0b0e40b7d5d05738f889075133b0a5c0246c0f0a39a0143ad

Event timestamp

2020-09-23 15:26:30

Calculated signature

0x9271e55a0f0f54e433e7ef57b0dc0ef0a51ab4f08830b017950ab0f1531016d1

2020-09-23 17:03:35

LogPacketizeProduct

Product ready

Start pending

Block number

2134193

Transaction hash

0x0c0c0f0f77b0b0e40b7d5d05738f889075133b0a5c0246c0f0a39a0143ad

Event timestamp

2020-09-23 17:05:35

Calculated signature

0xb2e48e77548bf21bc30e0d0d85f0b1007000b0f1301a6514ac53408ba0ca

2020-09-24 11:27:09

LogHandoverWarehouseTransport

Product ready

Start pending

Block number

2138899

Transaction hash

0x0d0544205c0e940ec040f0b0a0c16a323c78019181b212060b777983a095a

Event timestamp

2020-09-24 11:27:09

Temperature

76.00

Calculated signature

0x0d543657863d0f0a5d0f0a9296c0897afca784027082f0c738378cb75a0bb43

2020-09-24 11:39:24

LogHandoverTransportSupermarket

Product ready

Start pending

Block number

2138948

Transaction hash

0xb2a7f536e47179c01ac0b0802a00c1609206cd77653678c97260e4f0a0199

Event timestamp

2020-09-24 11:39:24

Calculated signature

0x4902660f164b5fca0f0c0d0f0d055c793664b48071ee795153a4d533790eb

During the trials a video has been compiled for demo purposes. It is planned to be presented during the final review but also for demo purposes in stakeholders.

3.2.2 Data collected and published

3.2.2.1 Sensor data

During the on-site validation, the Food Supply Chain pilot platform collected data from all the IoT platforms along the field-to-fork path. Sensor data from the SynField, the Transportation, and the Aberon IoT platform were generated and collected. This data was processed (anonymized, product information was removed) and were uploaded to Zenodo as Open Data for testing and repeatability purposes. For more details on data related to the Food Supply Chain pilot, see Deliverable “D6.5 – Data Management Plan” and its updates. The published data contain the information shown in Table 2.

Table 2: Data collected and published.

Category	Data type	Unit	Frequency	Size
Field sensor measurements (per SynField node)	Growing Degree -day (GDD)	Natural number	Once per day	<2KB
Transportation sensor measurements (per Truck gateway)	Box (Tag) presence detection	Binary	20 sec	<1MB
	Temperature	Celsius degrees	20 sec	
Warehouse sensor measurements (per monitored storage room)	Temperature	Celsius degrees	5min	< 400KB
	Humidity	%	5min	

3.2.2.2 User feedback

Regarding feedback collection we designed anonymous questionnaires that the users filled in, either on-site or offline (one of the pandemic’s effects on our pilot). The questionnaires include a short number of questions addressed to the users (producer, transporter, warehouse employee, supermarket employee, supermarket customer), aiming to identify the following:

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

- The usefulness of the product information included and whether it helps him/her towards deciding on purchasing the product.
- The user- “friendliness” of the web application of the pilot platform, the displayed information, how easy it was to understand the information presented and how well it was presented.
- What other information might be useful for the user.

In general, using the questionnaire we tried to identify how useful, usable, applicable, performant, innovative, and effective the platform is (as perceived by the users). The percentage of questions in each category is depicted in Figure 5 below:

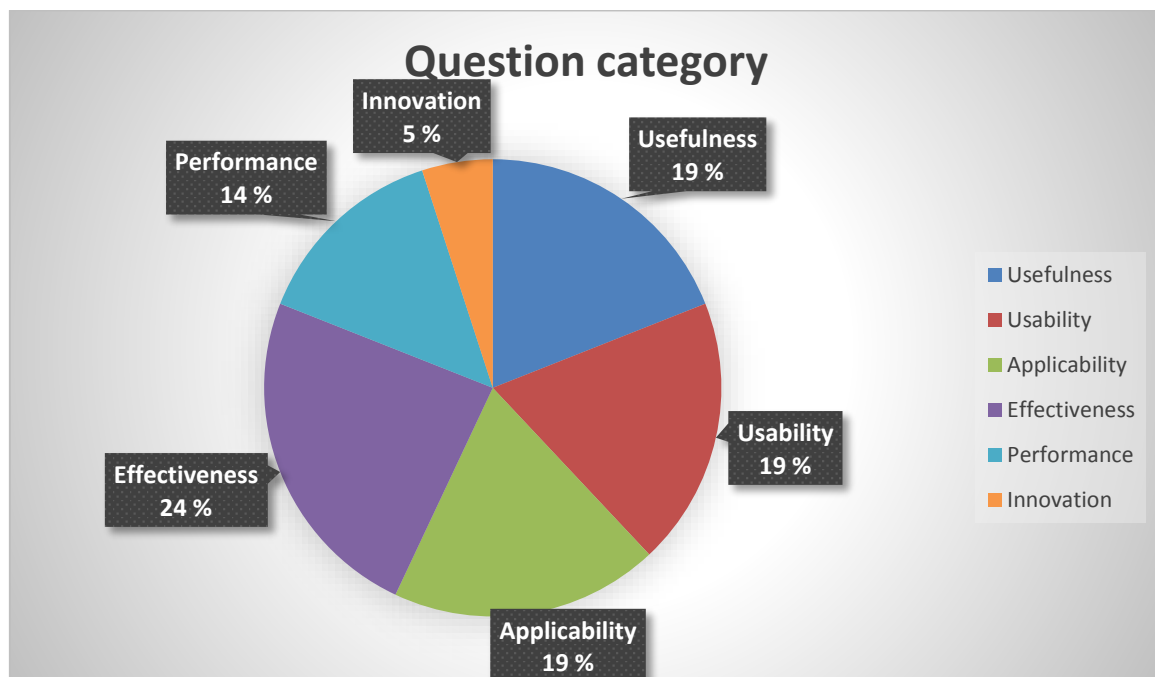


Figure 5: FSC Questionnaire question categories.

The questionnaire template can be found in Appendix I. Some of the questionnaires were filled-in during the on-site deployment (customers) and some of them were filled-in later (by the IoT-platform related actors).

Below, we present the results from the questionnaires collected:

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

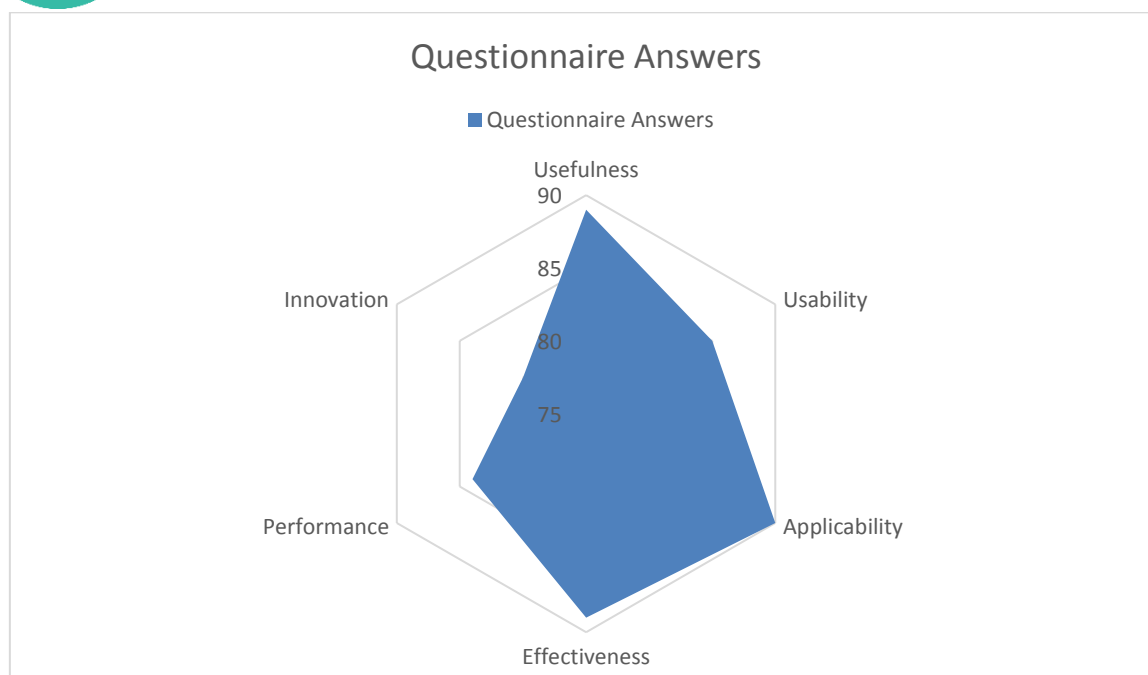


Figure 6: FSC Questionnaire results – Total.

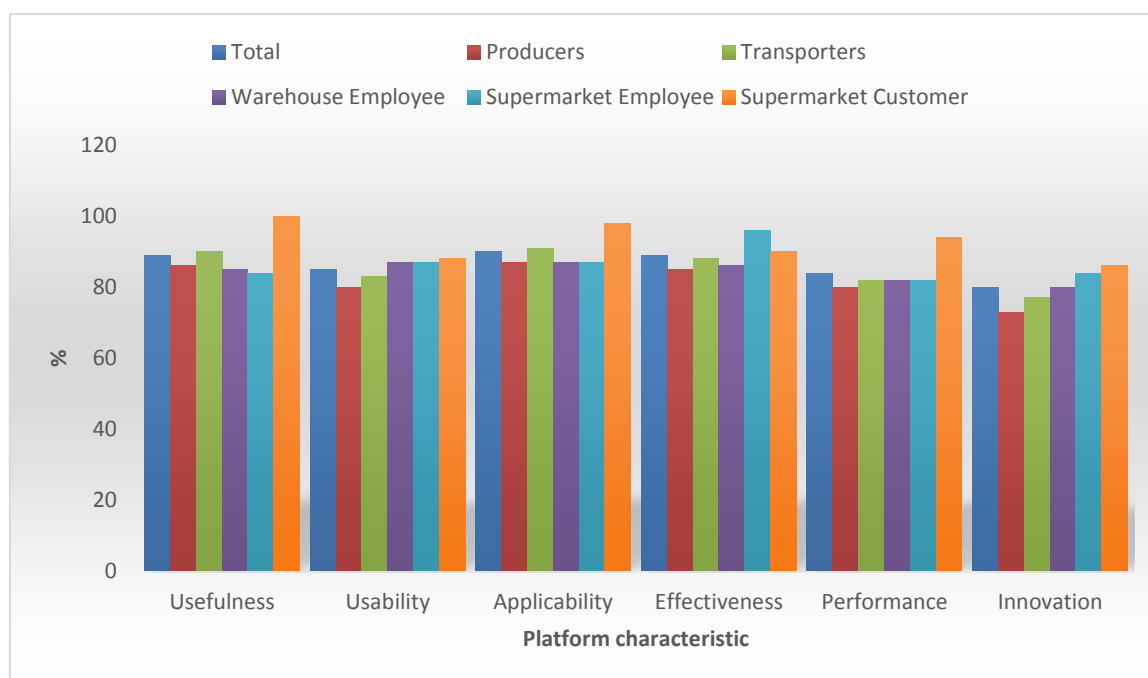


Figure 7: FSC Questionnaire results – per actor type.

Figure 6 shows that the strongest characteristics of the platform are its applicability, its usefulness, and its effectiveness. The users of the platform recognize that it offers something that fit their needs and they find it very useful. They also regard the way it tries to achieve its offering very effective. While they do not notice any performance issues, they do not see the platform as particularly innovative. However, this might seem understandable given that the major innovation of the platform is “hidden” under the hood, where the blockchain technology lies. This will improve as people get more acquainted with blockchain technology and its general characteristics and benefits (e.g., due to the bitcoin spread in various economic sectors). A conclusion that could be drawn from Figure 7, is that the end consumers seem to be in general



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

more enthusiastic about the platform; this could be explained due to the fact that they just see the result of the platform (without caring about the internals) which seems of great importance to them: the history of the product they purchase. On the other hand, the platform seems to be less attractive to the transporters group, which could be explained by the fact that this is the group that needs to put more effort; they participate in two handovers in every transportation. The results of the questionnaire will be further analysed to identify points for improvement while we continue further with this type of validation and evaluation in the context of exploiting the pilot platform. In particular, it might be important to focus on explaining blockchain technology to the users in an informative and user-friendly way.

3.3 Evaluation

3.3.1 Pilot performance assessment and KPIs evaluation

In this section, we provide an evaluation of the pilot platform (more details on this can be found in D4.5 "Final Architecture, System, and Pilots Evaluation Report"). The evaluation was performed using the KPI tables included in D5.1 "Baseline System and Measurements" and was based on the measurements performed during the validation of the pilot platform, as well as on the questionnaires that were filled in by users of the platform. KPIs are divided in system performance related KPIs and business-related KPIs. The former type KPIs were evaluated technically on the pilot platform while the latter were evaluated based on the questionnaire responses. The FSC pilot managed to successfully achieve all KPIs. The KPIs are presented in Table 3 below along with their evaluation.

Table 3: Performance and Business KPIs.

KPI	Name	Description	Metric	Method of measurement	Target	Result
System performance						
KPI_FSC_1	Ledger execution cost in public ledger	Cost for executing operations on a ledger	Ledger execution cost (e.g., gas in Ethereum)	Measure the total execution cost per box	As low as possible	0.420€ / Box
KPI_FSC_2	Handover time	Time to register data to blockchain during a handover between two stages	Time unit (e.g., seconds)	Measure the total time required for blockchain-related operations during a handover of a box between two stages	<1min	15 seconds
KPI_FSC_3	Internal state transition time	Time to register data to blockchain during a box' state transition occurring internally within a single stage	Time units (e.g., seconds)	Measure the total time required for blockchain-related operations during a state transition of a box within a single stage	<30sec	15 seconds
KPI_FSC_4	Throughput	Number of boxes that can be processed per time unit in any possible handover or internal state transition	Number of boxes per time unit	Measure the handover and state transition delays	>6000 boxes per day	30 boxes / minute => 43200 boxes /day

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

KPI_FSC_5	Time Scalability	Blockchain registration time for a handover or internal state transition, as a function of the number of boxes involved	Derivative of the blockchain registration time with respect to the number of boxes involved	Measure handover and state transition blockchain registration time as a function of the number of boxes involved	linear or sublinear	Linear
KPI_FSC_6	Cost Scalability	Public blockchain costs associated with box handovers or internal state transitions, as a function of the number of boxes involved	Derivative of public ledger cost with respect to the number of boxes involved	Measure public blockchain cost for handovers and state transitions as a function of the number of boxes involved	linear or sublinear	Linear
KPI_FSC_7	Response time for audit requests	The time it takes to respond to an audit request, by pulling out all data related to the box in question	Time units (e.g., minutes)	Measure the time it takes to pull out all records related to a given box, and to cross check them to identify potential issues	<1min	0.5 seconds
Business goals						
KPI_FSC_8	Customer confidence improvement	Confidence of consumers on brands and product authenticity, safety, and quality	Mean opinion score	Use of questionnaires to measure customer feedback	≥4.0/5.0	Achieved (4.3/5.0 in the responses about this)
KPI_FSC_9	Time for QR creation	The time it takes for the supermarket employee to create the QR code which is attached on a packet and includes all the history of the product.	Time units (e.g., seconds)	Measure QR creation delay in the backend platform	≤1sec	0.5 seconds
KPI_FSC_10	Increase in the effectiveness of targeted product withdrawals	Stored traceability data is used to perform targeted withdrawals of products, thereby minimising disruption to trade	% reduction in the scope of the product recall	Use of questionnaires to get feedback from the supermarket company and compare with current practices based on paper records.	≥50% reduction	Achieved (4.7/5.0 in the responses about this)
KPI_FSC_11	Percentage of product defects which are resolved	Number of defects or anomalies detected on a product delivered by the customer which are explained through deficiencies in the supply chain	% per total instances	Use of questionnaires to measure the success of resolved product defects based on the business rules	≥90%	Achieved (4.8/5.0 in the responses about this)



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

		operation. It excludes defects that are not due to the involved organizations.		that have been set by the involved organizations		
--	--	--	--	--	--	--

Within System Performance KPIs, KPI_FSC_1-3 can be regarded efficiency-related KPIs whereas KPI_FSC_4-7 can be regarded as scalability-related KPIs. Below, we describe how the System performance KPI results were reached:

KPI_FSC_1:

Estimate public ledger cost primarily in gas spent on the public Ethereum. For conversion to euros, please assume 10 nanoETH per gas unit, and €200 per ETH, i.e., $€2 \times 10^{-6}$ per gas per gas unit.

The generated public transaction hash of the above example is in the Ropsten public network is the following:

0x2ad9c500364eb85b11938f5c7c3be2cf2b5d3dc7e6a87f4160183b00adaff132

It can be viewed here:

<https://ropsten.etherscan.io/tx/0x2ad9c500364eb85b11938f5c7c3be2cf2b5d3dc7e6a87f4160183b00adaff132>

In the scope of the public Ethereum testnet (Ropsten) the costs can be considered as zero as a faucet can be used to acquire Ether. For the mainnet (Public) network, the average cost for a transaction is:

$207220 \text{ gas} \times 10 \times 10^{-9} \text{ ETH/gas} = 0.002072 \text{ ETH}$
 $0.002072 \text{ ETH} \times 200 \text{ € / ETH} = 0.4144 \text{ €} \approx 0.42 \text{ € / Box}$

Therefore, the cost is linear with regards to the number of boxes.

KPI_FSC_2, KPI_FSC_3:

The handover time and the internal state transition time both translate to the block generation & propagation time for the consortium blockchain. These parameters are set inside the **genesis.json** (configuration options used to generate the genesis block of the blockchain) and they are set at 15 seconds. So, on average, the time is 15 seconds.

KPI_FSC_4, KPI_FSC_5:

The throughput and the time scalability was measured by a python script, emulating (fake) handovers between actors. The target blockchain nodes (3 + 1) used for the testing were under a load balancer to achieve better performance. The output number does not take into consideration the cold start effect.

In the scope of the consortium blockchain, the supervisor ethereum node can process about 15 ethereum transactions per second (removing the node cold start effect). Meaning that each generated block will contain, on average 15 transactions. Keeping in mind that one handover can **contain multiple boxes**, the best way to measure consortium throughput is by mapping each handover to a transaction. Our pilot can therefore process roughly 15 different handovers per block. Assuming each transport containing 10 boxes and having a fleet of 600 transport trucks (total 6000 boxes). To process 600 different handovers simultaneously, and assuming 10 seconds per block generation (defined in consortium blockchain genesis state) it would take:

$1 \text{ block} / 15 \text{ handovers} \times 600 \text{ handovers} = 40 \text{ blocks}$
 $40 \text{ blocks} \times 10 \text{ sec/block} = 400 \text{ seconds}$

For the public blockchain, every transaction is bound to a box. The Interledger component can, at the moment, process one transaction at a time. Assuming mainnet/ropsten 20 second block generation and that each box must be confirmed via a tx receipt (meaning that one block will contain one box session), we have:

$6000 \text{ boxes} \times 20 \text{ sec / box} = 12000 \text{ sec} \approx 3.3 \text{ hours} (=0.5 \text{ boxes/sec})$



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

So the total throughput is 30 box / minute

KPI FSC 6:

The only public blockchain cost is at the end of the supply chain, when the box reaches the supermarket, then, via the **INTERLEDGER COMPONENT**, a transaction is created with the master hash of each of the chain steps. One box arriving at the supermarket maps to exactly one transaction to the public ledger.

KPI FSC 7:

The audit is performed via metamask in an in-browser environment. A simple **call** request is made to the smart contract and no transaction is performed. The time was measured via invoking a **node-js javascript script**, using **metamask** to contact the **ROPSTEN** test-network multiple times. The average response time was taken into consideration.

3.3.2 Evaluation of the Pilot's Competitive Advantage

In addition to what has been reported in D5.2 (section 3.1.2) regarding SOFIE's added value to the Food Supply Chain pilot, we describe an advantage that was made more prominent during the last months due to the COVID-19 pandemic and the impact it had to the Food Supply Chain, when many companies and production facilities were forced to pause their operations. When an entity that participates in the Food Supply Chain pauses its operations, the dependent participants are affected and will, thus, try to find alternate entities to cooperate with, in order to continue their operation. This, however, even if they manage to find alternative partners, implies that quality, certification, and trust issues are -again- to be resolved; something that is quite time consuming. The whole process of "integrating" new partners in the Food Supply Chain causes delays in the whole chain which, obviously, affect the end consumers. The speed at which companies can replace a broken link could make the difference between staying in business or shutting down. Therefore, adopting a platform that simplifies and accelerates the approval process could be a critical factor in a business' success. The SOFIE-enabled Food Supply Chain pilot provides, among others, the ability to replace any partner that has been part of the supply chain but, for any reason, has paused operations, with another partner that is already registered in the pilot platform that has a proven record of offering high quality services.

Based both on D5.2 and on the above, Table 4 summarizes the added value of SOFIE for the FSC pilot:

Table 4: SOFIE's added value compared to the existing traceability systems in FSC.

Legacy FSC systems	SOFIE added value
Centralized data management (without using any DLTs)	<ul style="list-style-type: none">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 corresponding 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.



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Systems that make use of DLT and IoT technology + IoT ^{1,2}	<ul style="list-style-type: none"> Introduces a transparent data adaptation layer for IoT platforms. Proposes an easy to use and non-disruptive solution to federating 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, checks (sample-based) quality (sample-based)	<ul style="list-style-type: none"> Improves safety and quality of the delivered products by enabling more effective proactive audits (via automation) for product quality assurance and improves reactive solutions in the cases of detected issues. Increases transparency and trust in the auditing processes in cases of safety/quality issues and disputes among the companies of the food provenance BP.
Replacement of a participant of the Food Supply Chain with another trusted participant is a time-consuming process	<ul style="list-style-type: none"> A participant that already participates in the platform and has a proven record of offering high-quality services can be selected to replace a participant that for any reason cannot operated anymore.

3.3.3 TRL

The FSC pilot platform and the related assets have managed to reach TRL-7 by the end of the project. The pilot platform has been deployed, demonstrated, and validated in an operational environment. The TRL level of the pilot and assets have been summarized in Table 5:

Table 5: TRL levels of the FSC and its assets

TRL	Justification
7	The pilot prototype was demonstrated in operational environment. The pilot platform was deployed and demonstrated on real operational environment in 7Grapes Pegasos premises in Kiato region, Greece where the last validation of the platform took place (as described in 3.2.1). The IoT platforms were deployed on a grape field (owned by local producers that are cooperating with the company, where SynField installation is already deployed), in a transportation vehicle (transportation van with refrigerated storing space, owned by the company, where the Transportation IoT platform was deployed for the project), and in a warehouse (two rooms owned by the company, one of them refrigerated, where the Aberon IoT platform was deployed for the project), while the platform was operated by producers, transporters, warehouse and supermarket employees with the assistance and guidance of pilot partners during September 2020. A part of the deployment and operation at the pilot site (Kiato) have also been demonstrated through a video presented during the review. In addition, it was also deployed in Agrinion Union for demo purposes.
7	The SynField Federation Adapter is actively used by the Food Supply Chain pilot to expose a RESTful API to provide data and things services to the Supervisor Web Server (SWS) and also uses the SynField platform's PKI to digitally sign every data object that is sent to the SWS. It is deployed on top of the SynField IoT platform
7	The Aberon Federation Adapter is actively used by the Food Supply Chain pilot by federating the Aberon IoT platform to the SWS. It is deployed on top of the Aberon IoT platform
7	The Transportation Federation Adapter is actively used by the Food Supply Chain pilot as it implements syntactic and semantic interoperability between the Transportation IoT platform and the pilot information model. It is deployed on top of the Transportation IoT platform

¹ <https://diamonds.everledger.io/>

² <https://www.provenance.org/>



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

7	The SynField platform for traceability and audit services are used by the Food Supply Chain pilot
---	--

The assessment of the TRL of the components described above was based on Horizon H2020 annex on TRL³.

3.4 Lessons learned and replication guidelines

During the on-site deployment of the Food Supply Chain platform, there was a need to train the participants of the trial in the use of the web application and, in general, on how to use the pilot platform. We found that the web application was self-intuitive, and no significant issues were encountered. The main effort was on explaining the steps that each actor (producer, transporter, warehouse employee, supermarket employee) needed to perform. An important factor in the success of the training process was that the participants were familiar with the technology and web applications in particular. We are aware that this might not always be the case, especially in the case of older participants that may be more reluctant to technological changes.

3.4.1 Replication guidelines

In the case of the FSC pilot, the main problem that was addressed was the different heterogeneous, and siloed IoT platforms that comprised the supply chain. The way in which this problem was addressed by SOFIE was through the Federation Adapters (FA), one of the core elements of the Food Supply Chain pilot. The FA implements syntactic and semantic interoperability between the corresponding IoT platform and the pilot information model. It exposes a RESTful API to provide data and things services to the Supervisor Web Server (SWS) and also uses the platform's PKI to digitally sign every data object that is sent to the SWS.

Therefore, a developer that would like to connect his/her IoT platform to the FSC platform needs mainly to implement the FA for the IoT platform by defining the Thing Description schema and exposing it via an FA endpoint. After that, registration to the FSC platform can be achieved through FSC graphical user interface (GUI) and the regular registration steps on the Login page of the platform⁴.

Table 6, Table 7, and Table 8 below contain the Thing Description schemas used in the FAs of the three IoT platforms that were federated in the FSC pilot.

Table 6: Synfield Thing Description schema.

```
{
  "@context": "https://www.w3.org/2019/wot/td/v1",
  "title": "SynFieldThing",
  "id": "urn:dev:wot:com:sofie:fcg:adapter:synfield",
  "description": "SynField Federation Adapter Thing Description model for the Food Supply Chain pilot",
  "securityDefinitions": {
    "nosec_sc": {
      "scheme": "nosec"
    }
  },
  "security": "nosec_sc",
  "properties": {
    "crops": {
      "type": "array",
      "readOnly": true,

```

³ https://ec.europa.eu/research/participants/data/ref/h2020/other/wp/2016_2017/annexes/h2020-wp1617-annex-g-trl_en.pdf

⁴ <https://services.synelixis.com/sofie/#/login>



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

<pre>"description": "The array of crop ids served from the underlying SynField IoT platform", "items": { "type": "string", "description": "The crop id in hashed/encrypted form" }, "forms": [{ "op": "readproperty", "href": "https://192.168.1.167/synfield/api/crops", "contentType": "application/json" }] }, "crop_details": { "type": "object", "readOnly": true, "description": "The details of the specific crop", "properties": { "name": { "type": "string" }, "product_type": { "type": "string" }, "location": { "type": "string" }, "ripening_level": { "type": "number" } }, "uriVariables": { "cropId": { "type": "string", "description": "The hashed/encrypted id of the crop" } }, "forms": [{ "op": "readproperty", "href": "https://192.168.1.167/synfield/api/crop/{cropId}/details", "contentType": "application/json" }] } }</pre>

Table 7: Transportation Thing Description schema.

<pre>{ "@context": "https://www.w3.org/2019/wot/td/v1", "title": "TransportationThing", "id": "urn:dev:wot:com:sofie:fc:adapter:transportation", "description": "Transportation Federation Adapter Thing Description model for the Food Supply Chain pilot", "securityDefinitions": { "nosec_sc": { "scheme": "nosec" } } }</pre>



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

```
}
},
"security": "nosec_sc",
"properties": {
  "transports": {
    "type": "array",
    "readOnly": true,
    "description": "The array of transport ids served from the underlying Transportation IoT platform",
    "items": {
      "type": "string",
      "description": "The transport id in hashed/encrypted form"
    },
  },
  "forms": [
    {
      "op": "readproperty",
      "href": "https://192.168.1.167/transportation/api/transports",
      "contentType": "application/json"
    }
  ]
},
"transport_boxes": {
  "type": "array",
  "readOnly": true,
  "description": "The boxes that are part (inside) the given transport at that time",
  "items": {
    "type": "string",
    "description": "The box id"
  },
  "uriVariables": {
    "transportId": {
      "type": "string",
      "description": "The hashed/encrypted id of the transport"
    }
  },
  "forms": [
    {
      "op": "readproperty",
      "href": "https://192.168.1.167/transportation/api/transport/{transportId}/boxes",
      "contentType": "application/json"
    }
  ]
},
"transport_readings": {
  "type": "object",
  "readOnly": true,
  "description": "The readings of the specific transport",
  "properties": {
    "min_temperature": {
      "type": "number"
    },
    "avg_temperature": {
      "type": "number"
    },
    "max_temperature": {
      "type": "number"
    }
  },
  "uriVariables": {
    "transportId": {
```

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

```

    "type": "string",
    "description": "The hashed/encrypted id of the transport"
  },
  "start": {
    "type": "string",
    "description": "The start datetime of the readings"
  },
  "end": {
    "type": "string",
    "description": "The end datetime of the readings"
  }
},
"forms": [
  {
    "op": "readproperty",
    "href": "https://192.168.1.167/transportation/api/transport/{transportId}/readings{?start,end}",
    "contentType": "application/json"
  }
]
}
}
}

```

Table 8: Aberon Thing Description schema.

```

{
  "@context": "https://www.w3.org/2019/wot/td/v1",
  "title": "AberonThing",
  "id": "urn:dev:wot:com:sofie:fcg:adapter:aberon",
  "description": "Aberon Federation Adapter Thing Description model for the Food Supply Chain pilot",
  "securityDefinitions": {
    "nosec_sc": {
      "scheme": "nosec"
    }
  },
  "security": "nosec_sc",
  "properties": {
    "warehouses": {
      "type": "array",
      "readOnly": true,
      "description": "The array of warehouse ids served from the underlying Aberon IoT platform",
      "items": {
        "type": "string",
        "description": "The warehouse id in hashed/encrypted form"
      }
    },
    "forms": [
      {
        "op": "readproperty",
        "href": "https://192.168.1.167/aberon/api/warehouses",
        "contentType": "application/json"
      }
    ]
  },
  "warehouse_rooms": {
    "type": "array",
    "readOnly": true,
    "description": "The rooms/sections of the specific warehouse",
  }
}

```



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

```
"items": {
  "type": "string",
  "description": "The room id"
},
"uriVariables": {
  "warehouseId": {
    "type": "string",
    "description": "The hashed/encrypted id of the warehouse"
  }
},
"forms": [
  {
    "op": "readproperty",
    "href": "https://192.168.1.167/aberon/api/warehouse/{warehouseId}/rooms",
    "contentType": "application/json"
  }
],
"room_readings": {
  "type": "object",
  "readOnly": true,
  "description": "The readings of the specific warehouse room",
  "properties": {
    "min_humidity": {
      "type": "number"
    },
    "avg_humidity": {
      "type": "number"
    },
    "max_humidity": {
      "type": "number"
    },
    "min_temperature": {
      "type": "number"
    },
    "avg_temperature": {
      "type": "number"
    },
    "max_temperature": {
      "type": "number"
    }
  },
  "uriVariables": {
    "warehouseId": {
      "type": "string",
      "description": "The hashed/encrypted id of the warehouse"
    },
    "roomId": {
      "type": "string",
      "description": "The id of the room"
    },
    "start": {
      "type": "string",
      "description": "The start datetime of the readings"
    },
    "end": {
      "type": "string",
      "description": "The end datetime of the readings"
    }
  }
}
```



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

```
},
  "forms": [
    {
      "op": "readproperty",
      "href":
        "https://192.168.1.167/aberon/api/warehouse/{warehouseId}/room/{roomId}/readings{?start,end}",
      "contentType": "application/json"
    }
  ]
}
```

The implementation of one of these Federation Adapters (Transportation Federation Adapter) is publicly available on Github⁵ and can be used as a reference (source files included) from an external party to implement their own Federation Adapter for their own IoT platform. Below, we summarize the steps needed from a developer to develop and deploy the Transportation Federation Adapter:

Required Software

The following are required for development and deployment:

- Python 3.6.2+⁶
- Django framework (2.2.x)⁷
- Redis⁸
- Docker engine⁹
- docker-compose¹⁰

Installation

To install, in a new python virtualenv, execute:

```
pip3 install -r requirements.txt
```

Running/deploying the client locally (for development purposes)

File *transportation_adapter.settings.dev* holds the settings to be used in development. Please, refer to the source file on SOFIE's Github for more details.

Make sure that access to a redis server & the mongodb of the underlying KAA IoT platform is present, then run:

```
export DJANGO_SETTINGS_MODULE=transportation_adapter.settings.dev
python3 manage.py migrate

python3 manage.py runserver 0:8000
```

⁵ <https://github.com/SOFIE-project/fsc-transportation-federation-adapter>

⁶ <https://www.python.org/downloads/release/python-362/>

⁷ <https://www.djangoproject.com/>

⁸ <https://redis.io/>

⁹ <https://docs.docker.com/engine/>

¹⁰ <https://docs.docker.com/compose/>



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

The django development server is now running at *localhost:8000*

Running the unit tests

To run the unit tests, execute:

```
python3 manage.py test --settings=transportation_adapter.settings.test
```

Deploying with Docker (for production releases)

File *transportation_adapter.settings.prod* holds the settings to be used in production. Please, refer to the source file on SOFIE's Github for more details.

To build & run the containers, execute:

```
cd config  
docker-compose up -d --build
```

API Swagger

In addition, the F.A. also exposes a swagger page under *api/swagger/*, which is merely the online documentation that includes the exposed endpoints of the API.

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

4. Decentralized Energy Data Exchange Pilot

4.1 Pilot overview

The core idea of the *Decentralized Energy Data Exchange* (DEDE) pilot is 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 develops and use the capabilities of the SOFIE federated platform and Energy grid adapters to deliver the required functionality to stakeholders.

The focus of the final stage of the pilot has been on the deployment of the DEDE adapters. The input for deployment in the production environment has mainly been received from the TSO/DSO stakeholder group. The DEDE adapters, the main integration point for any third party using the pilot's solution, are deployed in the Elering Estfeed live environment and in the DSO (Elektrilevi) test site, in single smart meter units. The data exchange and interaction between Energinet and Wind Farm IoT was done in an emulated environment. The stakeholders provided necessary input for the test environment and basic business logic for the data exchange process.

The deployment of the DEDE adapters provide a secure data exchange and agreements to data access rights between smart meter data and infrastructure owners and energy service providers (intermediaries, distributors, and brokers). The pilot developed and used the capabilities of the SOFIE federated platform and Energy grid adapters to deliver the required functionality to stakeholders.

A general overview of the pilot can be seen in Figure 8:

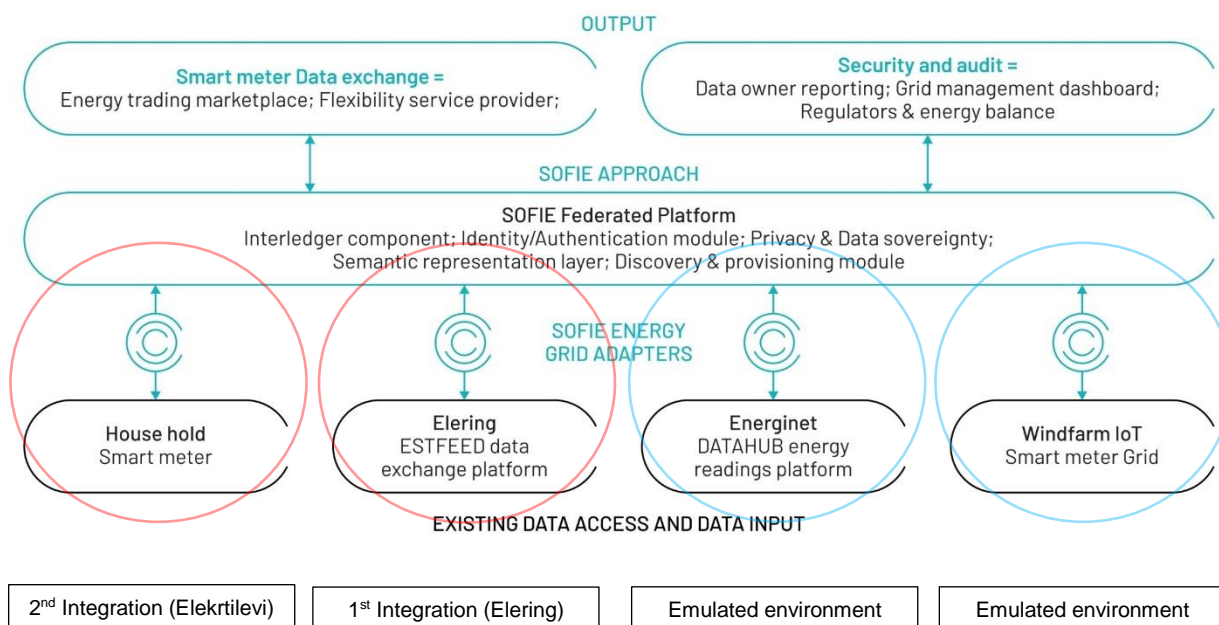


Figure 8: DEDE architecture overview

4.2 Validation

4.2.1 Final end-to-end on-site validation

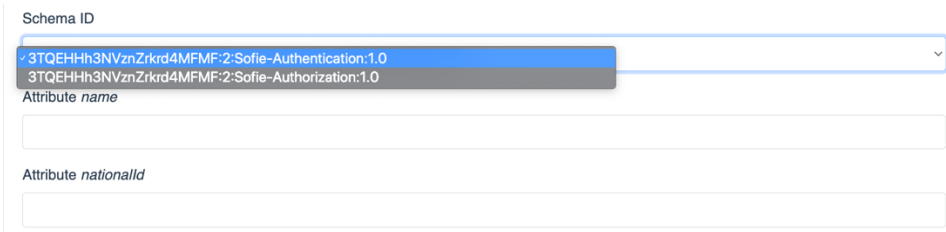
The final validation was carried out in multiple stages. Real integration with Estfeed was finalised earlier, in addition to that different emulated scenarios were added, and multiple additional deployments were done. The validation has multiple approaches: technical, integration capabilities and business-related. Compared to the validation done in D5.3 section “End-to-End



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

platform validation” the current solution is feature-complete, and it was possible to validate the flows end-to-end. The pilot successfully passed all the validation tests which is summarised in Table 9. The pilot has also developed integration tests that allowed the pilot platform to reach level “5” in the project CI/CD pipeline (D3.3 [D3.3]).

Table 9: Validation of DEDE Test Cases.

Test ID	EDE_TC01
Test description	Request for electricity consumption data from data hub by data owner.
Test location	Admin interface of federation adapter
Related use cases	DEDE_UC1, DEDE_UC2
Related requirements	REQ_DEDE1.1, REQ_DEDE1.2, REQ_DEDE2.2, REQ_DEDE2.3
Feature(s) under test	Data sharing
Components involved	Federation adapter, Estonian data hub (Estfeed) adapter
Test environment	Federation adapter has been installed and configured. Credential has been provided to the data owner with attributes important for resolving identity.
Dependencies	N/A
Steps	<ol style="list-style-type: none"> 1. Credential scheme is defined for authentication (<i>Sofie-Authentication:1.0</i>) 2. Credential is issued to target DID 3. Credential is used when accessing API service
Pass criteria	Credentials can be issued and provided as proof when accessing the service.
Result	<p>Credential schema selection:</p>  <p>Issued credential view:</p>



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Federation Adapter

Identifiers Credentials Services Permissions Discover Log

Credentials

Issued To	Credential Definition ID	Attributes	Valid
4uJo1f5Xouu32FzP8Pdw7	VcGwzAh30p1vceyaXEeJJc3:CL:727:458261583	nationalId: 12345 name: Mait	yes

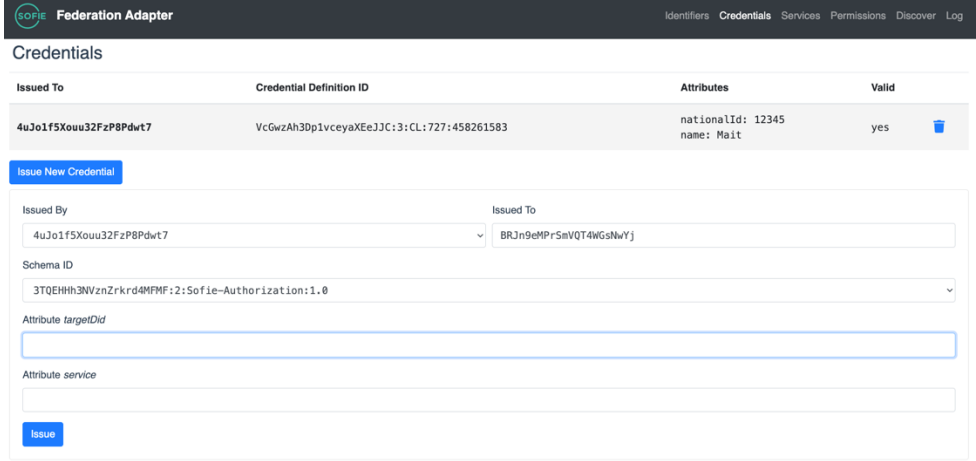
Issue New Credential

Metering data API response payload:

```
[
  {
    "person": "PNOEE-38502136521",
    "meteringPoint": "38Z121212123-U",
    "from": 1567332000000,
    "to": 1571616000000,
    "readings": [
      {
        "from": 1567332000000,
        "to": 1567335600000,
        "consumption": 123456.789,
        "production": 0.0,
        "unit": "kWh"
      },
      {
        "from": 1567335600000,
        "to": 1567339200000,
        "consumption": 123456.789,
        "production": 0.0,
        "unit": "kWh"
      },
      {
        "from": 1567339200000,
        "to": 1567342800000,
        "consumption": 123456.789,
        "production": 0.0,
        "unit": "kWh"
      }
    ]
  }
]
```



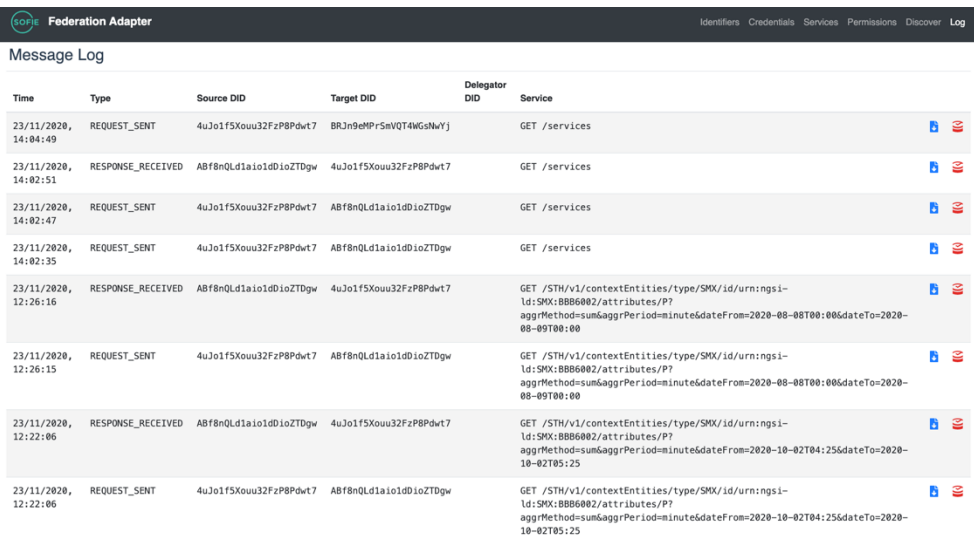
Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Test ID	EDE_TC02
Test description	Delegation of access rights to electricity consumption data from data owner to a third party.
Test location	Admin interface of federation adapter
Related use cases	DED_UC3, DED_UC4
Related requirements	REQ_DEDE2.1, REQ_DEDE2.4, REQ_DEDE2.5
Feature(s) under test	Access rights
Components involved	Federation adapter, IAA
Test environment	The federation adapter has been installed and configured by a third party. The third party is known to the data hub federation adapter and delegation through credentials can be added.
Dependencies	N/A
Steps	<ol style="list-style-type: none"> 1. Data owner selects a third party to delegate access to 2. Data owner specifies the data hub that is the target of delegated access 3. Third party requests data from the data hub on behalf of data owner
Pass criteria	Delegation can be performed
Result	<p>Here is the visual presentation of the delegation process which passed successfully:</p> 

Test ID	EDE_TC03
---------	----------



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Test description	Request for audit log concerning data owner from the data hub.
Test location	Admin interface of federation adapter
Related use cases	DED_UC5, DED_UC6
Related requirements	REQ_DEDE2.2, REQ_DEDE2.3, REQ_DEDE2.7, REQ_DEDE5.1, REQ_DEDE5.2
Feature(s) under test	Auditability
Components involved	Federation adapter
Test environment	Both data owner and delegated third party have performed requests to the data hub. Therefore, the audit log is not empty at the data hub.
Dependencies	N/A
Steps	1. Data owner selects a data hub to request audit log from
Pass criteria	Admin interface of the federation adapter shows the audit log containing all the interactions.
Result	<p>Every interaction in the system produces a signed audit log.</p> 

Test ID	EDE_TC04
Test description	Performance of the federation adapter, measured by the number of requests handled per second.
Test location	Admin interface of federation adapter



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Related cases	DEDE_UC2
Related requirements	REQ_DEDE1.1
Feature(s) under test	Scalability
Components involved	Federation adapter
Test environment	Federation adapter has been installed and configured by the data hub and a test client. Automated test scripts perform HTTP requests.
Dependencies	N/A
Steps	1. Test script is executed to start sending a preconfigured amount of requests to the data hub 2. Total time of completing all the requests is measured and throughput in requests per second is calculated
Pass criteria	Federation adapter is able to service at least 100 requests per second.
Result	Federation adapter is horizontally scalable through load balancing. Application uses non-blocking architecture and is not thread-based, which provides much better resource utilization. It is capable to process more than 100 parallel requests.

4.2.2 Data collected and published

The pilot makes decentralized data exchange possible without storing the metering data itself. It is designed to provide secure and flexible connections between different parties. As stated in the delivery “D6.5 - Data Management Plan” private datasets are not meant for sharing. The public datasets require more wide-spread deployment of the federation adapters in real-life, which has not happened yet. Currently, no open datasets are published.

4.3 Evaluation

4.3.1 Pilot performance assessment and KPIs evaluation

In this section the evaluation of the pilot solution is presented. The evaluation criteria were defined in D5.1 “Baseline System and Measurements”. The following system performance KPIs are efficiency-related: KPI_DEDE_1, KPI_DEDE_2, KPI_DEDE_3 and KPI_DEDE_4. And other system performance KPIs are scalability focused. All KPIs were successfully reached and are presented in Table 10 along with their evaluation:

Table 10: Performance and Business KPIs.

KPI	Name	Description	Metric	Method of measurement	Target	Result
System performance						
KPI_DEDE_1	Response time for DID operations	Time for performing read/write operations on the identity	Time units (e.g., seconds)	Measure time between instant system receives a request until	<5 sec	0.7 seconds



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

		ledger (Hyperledger Indy)		the instant that the system responds		
KPI_DEDE_2	Response time for KSI Blockchain signatures	Response time for KSI Blockchain signatures	Time units (e.g., seconds)	Measure time between instant system receives a request until the instant that the system responds	<2 sec	1.5 seconds
KPI_DEDE_3	Processing time (overhead) of requests in adapter	Overhead for processing incoming requests - includes audit log entry, verifying credentials, setting up secure channel	Time units (e.g., seconds)	Measure time between instant system receives a request until the instant that the system responds	<5 sec	0.1 seconds
KPI_DEDE_4	Response time for audit logs	Time for the system to respond to audit log requests	Time units (e.g., seconds)	Measure time between instant system receives a request until the instant that the system responds	<15 sec	0.3 seconds
KPI_DEDE_5	Scalability – cost	Increase of cost as load (number of DID operators or retrieving KSI Blockchain signatures per time unit) increases	Ratio of delta cost over delta of load (number of DID operators or retrieving KSI Blockchain)	Measure cost for different loads	linear or sublinear	Not applicable. No cost associated with ledgers used
KPI_DEDE_6	Scalability – time	Increase of response time as load (e.g., number of transactions per time unit, number of nodes) increases	Ratio of delta time over delta of load (number of transactions/nodes)	Measure response time for different loads	Linear or sublinear	Linear

Business goals	
-----------------------	--



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

KPI_DEDE_7	SOFIE adapter overhead for DSO/TSO	Sysadmin of DSO/TSO must be content that the operation of the new 3rd party (SOFIE adapter integrator) code does not affect the existing data access to service providers) The use of SOFIE adapters should not delay the access to data (expectations to grant access in a matter of seconds)	Time units (e.g., seconds) Impact measured in DSO system performance change	Measure the time it takes to get the service provider to get access to the data or get response that the access has been revoked by data owner. Measure the impact of SOFIE adapter to DSO/TSO existing data access services	Time: seconds Impact to system: 0% (only push requests to be serviced from SOFIE adapter)	Achieved – no overhead to internal processes
KPI_DEDE_9	SOFIE adapter functionality delivered geographically to data owner	SOFIE adapter usage must be measured in cross country situations. the functionality and the proof of using adapters must be provided.	YES / NO number of participants	Measure how many different (geographically) data owners can get access to the functionality that SOFIE adapter offers	Access to new service: YES/NO Number of countries = 3	Achieved – 3 countries covered

During the pilot's development phase on the last year, the business KPI DEDE_8 was discarded. The interaction with the stakeholders (DSO/TSO level) revealed that the number of verified logs was not affecting the business process. Enabling the verification of logs was a pre-requirement to use the SOFIE adapters, so no metrics was needed for this evaluation.

4.3.2 Evaluation of the Pilot's Competitive Advantage

The initial concept of innovation in decentralized energy data exchange was reported in D5.2 [D5.2] (section 4.1.2). The need for transparent and secure data exchange for energy data is even more important now. Compared to the initial concept, the role of the data owner has become more prominent and through the use of self-sovereign identities there is a way to put the person at the center with full control.



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Table 11 summarizes the added value of SOFIE, and the decentralized energy data exchange pilot, compared to existing solutions.

Table 11: SOFIE added value for DEDE.

Current status	SOFIE added value
Data owner is not in control, centralized systems have full control instead	The novel integration of self-sovereign identity enables user-centric data access providing verifiable claims and privacy
Data exchanges are centralized, complicated onboarding to closed systems	By enabling decentralized data exchange possibilities completely new forms of interactions can be created, which could accelerate speed in the energy domain.
Auditing closed systems lacks transparency and accessibility	Increases trust and transparency in the auditing process, enables better automation

4.3.3 TRL

The pilot and its assets reached TRL-7 after successful deployment in real operational environments. Based on the Horizon H2020 TRL evaluation guideline¹¹ the DEDE pilot and its assets should be demonstrated in the operational environment to be assigned the TRL 7. The TRL level that the pilot and the outcome assets that have derived from it reached, is summarized in Table 12.

Table 12: TRL levels of the DEDE pilot and its assets

TRL	Justification
7	<p>The pilot prototype was demonstrated in operational environment. The main end-user system, where pilot is implemented is Estonian Transmission System operator (TSO), Elering, Data exchange platform called Estfeed. Estfeed is used in production for Estonian smart meter data distribution. A pre-live environment of Estfeed was used to deploy DEDE prototype and its functionality. The Pilot demonstrates how a live smart meter can be accessed and a revocation mechanism applied to block the access to the data. Estfeed platform uses 700 000 smart meters, as the main data source to provide energy consumption data to relevant parties in the Energy market.</p> <p><i>Pilot is using the DEDE adapter as the core asset from SOFIE. The adapter enables the functionality of Interledger, IAA and PDS Components in the pilot.</i></p>
7	<p>The Decentralised Data Exchange Adapter is used by Decentralized Energy Data Exchange pilot. The adapters were deployed in Estfeed environment and played a key role of executing the DEDE scenarios and demonstrating the functionality that each participant is using.</p>

4.4 Lessons learned and replication guidelines

One of the main design goals of the DEDE pilot was to do federation in a decentralized way. From the very beginning, it was clear that we cannot have a central entity running SOFIE components for everyone and have everything routed through it. Different entities would need to connect directly to one another and would need to agree on a communication protocol to do so. Thus, every SOFIE component that we were to use, would have to be run by individual nodes with no special privileges.

Given the decentralized nature of our pilot, we started by specifying a communication protocol for the nodes. The communication protocol enables each node to provide arbitrary services and

¹¹ https://ec.europa.eu/research/participants/data/ref/h2020/other/wp/2016_2017/annexes/h2020-wp1617-annex-g-trl_en.pdf



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

also to consume services from others. All communication is secured by default. Most importantly, this protocol had to solve the problem of identifying, authenticating and authorizing nodes that provide and consume services. Hyperledger Indy was chosen as the trust anchor for node identity, as it also offers an authentication and authorization mechanism by utilizing verifiable credentials. Everything that was not essential to secure data exchange was left out from this base layer.

4.4.1 Replication guidelines

We implemented the protocol as a federation adapter. All nodes either must run this adapter or implement their own. After interaction with relevant stakeholders, Guardtime created a adapter docker image for users. In order to follow the SOFIE exploitation plan and business ambition set by Guardtime Management board, there was a need to follow the replication guideline under Guardtime's supervision and consultancy. This led to a decision that the adapter source code is not available as open source. For partner organizations involved in the platform Guardtime will provide non-public Docker image, which includes fully functioning federation adapter solution. The following SOFIE components can be used in the implementation:

- **Privacy and Data Sovereignty** - It is up to the nodes that provide services, to say which attributes they want their clients to prove about themselves. By default, the protocol expects the attributes to be proven with verifiable credentials. But the PDS component can offer a simpler alternative here, by issuing a JSON Web Token (JWT) that proves certain attributes about the holder of the token. In either case, it is up to the service provider node to decide which issuers it trusts. If the trusted issuer is specified just by a node identifier (and not by a credential definition id), it is expected that the client sends a JWT issued by the trusted node together with the service request.
- **Identity, Authentication, Authorization** - This is the counterpart component for the PDS component, if JWT based proved attributes are used. If the client sends a JWT together with a service request, the IAA component can verify this token. In this case, the authorization decision can be made without an extra round-trip to the client requesting proof of credentials.
- **Interledger** - Each node can increase their trust for the Hyperledger Indy instance by periodically recording its state in the KSI blockchain. The interledger component takes care of this. It is not strictly required for the protocol to work but can serve as an additional tamper-proofing mechanism for private Hyperledger Indy deployments.

Other SOFIE components might be useful outside the base layer:

- **Semantic representation** - the DEDE pilot fixes the service description format to be OpenAPI 3.0. All nodes that want to provide services need to describe their services in this format. The semantic representation component might help in translating from other service description formats (like the WoT Thing Description) to OpenAPI 3.0. But this is an implementation decision that is left for the service provider.
- **Marketplace** - can be implemented as another service once the data consumers and data providers are federated. Also, there can be more than one marketplace.
- **Provisioning and Discovery** - Before a client node can connect to a service providing node, it somehow has to learn its identifier (DID). Once the client knows the identifier of the service provider, it can look up the rest of the important details from the ledger. In the DEDE pilot, we view the process of discovering partners as an out-of-band activity. It can have many different forms and we do not prescribe one.

Once the protocol has been implemented or the official federation adapter installed, each energy metering data source should provide the services described in Table 13.

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Table 13: Services to be offered by each energy metering data source.

```
{
  openapi: '3.0.1',
  info: {
    version: '1.0.0',
    title: 'SOFIE Energy Metering Data Source',
    description: 'Provides access to metering points and their energy consumption data'
  },
  paths: {
    '/meteringpoints': {
      get: {
        tags: ['Metering points'],
        description: 'Get metering points',
        operationId: 'getMeteringPoints',
        parameters: [
          {
            name: 'person',
            in: 'query',
            schema: {
              type: 'string'
            },
            required: true,
            description: 'ETSI person ID'
          }
        ],
        responses: {
          '200': {
            description: 'A list of metering points',
            content: {
              'application/json': {
                schema: {
                  $ref: '#/components/schemas/ArrayOfMeteringPoints'
                }
              }
            }
          }
        }
      },
      '/consumption': {
        get: {
          tags: ['Consumption data'],
          description: 'Get consumption data',
          operationId: 'getConsumptionData',
          parameters: [
            {
              name: 'person',
              in: 'query',
              schema: {
                type: 'string'
              },
              required: true,
              description: 'ETSI person ID'
            },
            {
              name: 'meteringPoint',
              in: 'query',

```



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

```
    schema: {
      type: 'string'
    },
    required: true,
    description: 'EIC code of metering point'
  },
  {
    name: 'from',
    in: 'query',
    schema: {
      type: 'string',
      format: 'date-time',
    },
    required: false,
    description: 'Period start datetime in ISO 8601 format
(default: current time - 24h)'
  },
  {
    name: 'to',
    in: 'query',
    schema: {
      type: 'string',
      format: 'date-time',
    },
    required: false,
    description: 'Period end datetime in ISO 8601 format (default:
current time)'
  }
],
responses: {
  '200': {
    description: 'Consumption data fetched',
    content: {
      'application/json': {
        schema: {
          $ref: '#/components/schemas/ConsumptionData'
        }
      }
    }
  }
}
},
components: {
  schemas: {
    timestamp: {
      type: 'string',
      format: 'date-time',
      description: 'Consumption timestamp',
      example: '2019-10-09T03:48:50.562Z'
    },
    consumption: {
      type: 'number',
      example: 1.99
    },
    production: {
      type: 'number',
      example: 0.0
    }
  }
}
```

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

```

    },
    unit: {
      type: 'string',
      example: 'kWh'
    },
    meterReading: {
      type: 'object',
      properties: {
        timestamp: {
          $ref: '#/components/schemas/timestamp'
        },
        consumption: {
          $ref: '#/components/schemas/consumption'
        },
        production: {
          $ref: '#/components/schemas/production'
        },
        unit: {
          $ref: '#/components/schemas/unit'
        }
      }
    },
    ConsumptionData: {
      type: 'object',
      properties: {
        person: {
          type: 'string',
          description: 'ETSI ID of the person owning the metering point'
        },
        meteringPoint: {
          type: 'string',
          description: 'Metering point EIC code'
        },
        meterReadings: {
          type: 'array',
          items: {
            $ref: '#/components/schemas/meterReading'
          }
        }
      }
    },
    location: {
      type: 'object',
      properties: {
        country: {
          type: 'string'
        },
        county: {
          type: 'string'
        },
        municipality: {
          type: 'string'
        },
        locality: {
          type: 'string'
        },
        streetAddress: {
          type: 'string'
        }
      }
    },

```



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

```
        postCode: {
          type: 'string'
        }
      },
      meteringPoint: {
        type: 'object',
        properties: {
          code: {
            type: 'string',
            description: 'EIC code of the meter'
          },
          location: {
            $ref: '#/components/schemas/location'
          }
        }
      },
      arrayOfMeteringPoints: {
        type: 'array',
        items: {
          $ref: '#/components/schemas/meteringPoint'
        }
      }
    }
  }
}
```

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

5. Decentralized Energy Flexibility Marketplace Pilot

5.1 Pilot overview

The objective of the Decentralized Energy Flexibility Marketplace (DEFM) pilot is to demonstrate the use of the SOFIE architecture and its components to support the implementation of a decentralized energy flexibility marketplace in the context of an energy district/electricity grid with a high penetration of distributed generation from renewable energy sources.

The platform's high-level architecture and deployment view presented in D5.3 *End-to-End Platform Validation* are still relevant to describe the final release of the platform. In this section, both the diagrams are briefly presented for reference. The role of each module in relation with the different layers of a multi-tier web application is illustrated in Section 5.4, below, together with the replication guidelines.

Figure 9 illustrates the high-level architecture of the *Decentralized Energy Flexibility Marketplace* (DEFM) pilot. The pilot platform utilizes the following modules:

- Interledger (IL)
- Marketplace
- Semantic Representation (SR)
- Federation Adapter (FA)

together with pilot-specific software and the pilot IoT platforms, and in connection with the private and public blockchains running the decentralized smart contracts.

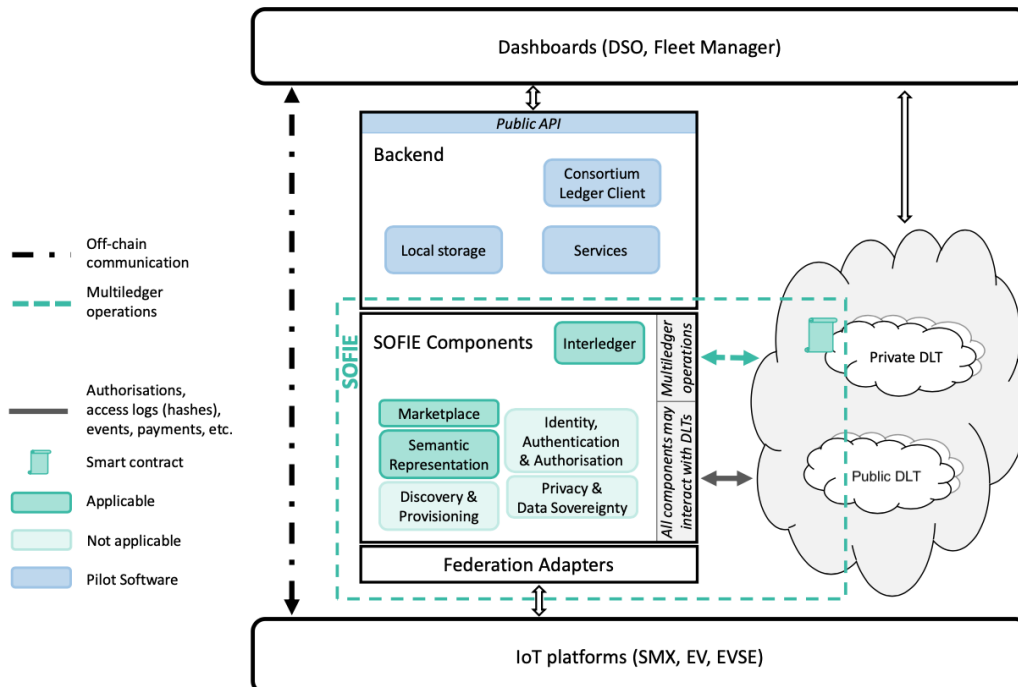


Figure 9: High-level architecture of the DEFM pilot.

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

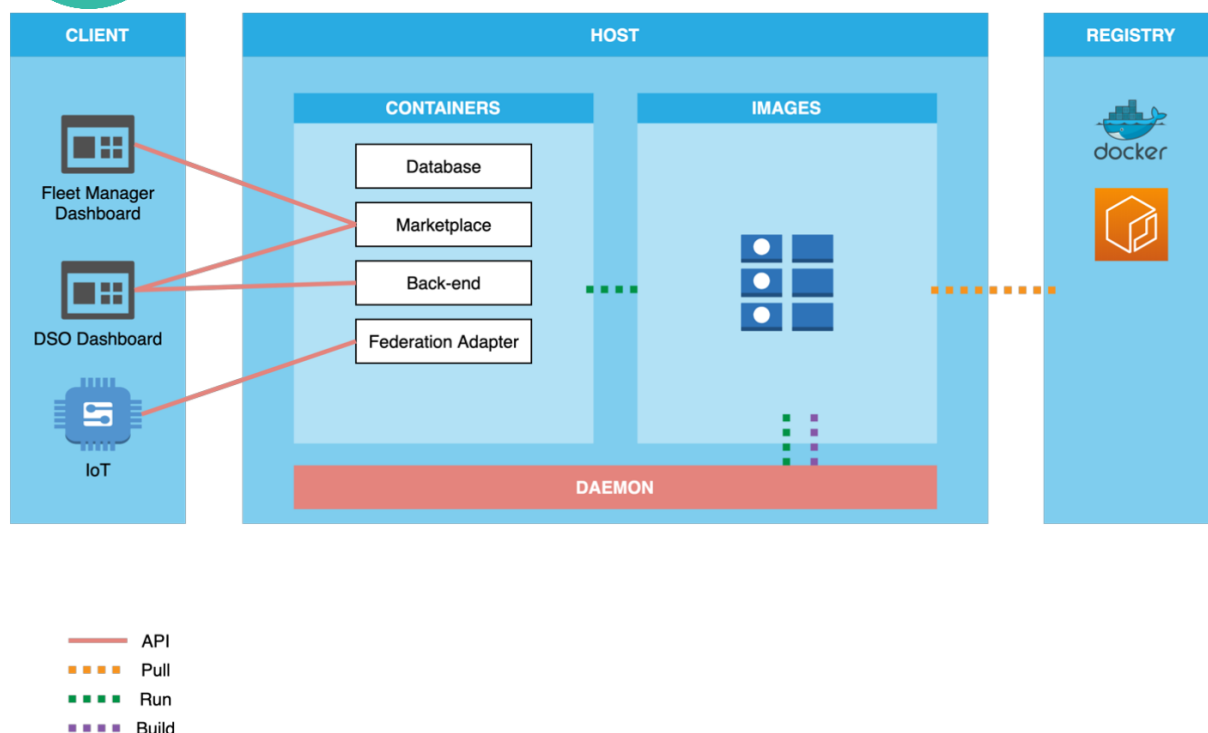


Figure 10: Deployment view of the DEFM pilot.

Figure 10 presents the deployment view of the pilot platform. The host system runs each component as a software container. Every time a new feature is added to the application, a new image is built and pushed on a container registry. The host system then fetches the updated images and runs them.

5.2 Validation

The validation of the Decentralized Energy Flexibility Marketplace pilot was performed on site.

The platform was deployed on the pilot site, where the IoT smart meters and the relevant assets (PV panels, EVs and EVSEs) were already present. The initial validation activities were focused on the services offered by the platform (D5.1 *Baseline system and measurements*) and on the technical feasibility (D5.3 *End-to-End platform validation*). The potential end users of the platform were on board since the beginning of the project, ensuring that the use cases and scenarios defined are relevant for their needs and that the requirements are clearly defined.

The test cases defined during the previous stages of the project and collected in deliverables D5.1 and D5.3, were defined taking into account the possibility to automate them and, therefore, they were used as a basis for the development of the pilot's integration tests.

The integration tests are the main requirement to reach maturity level "5" in the project CI/CD pipeline (D3.3 *Business Platforms, Pilot Release*): they are executed automatically every time a new version of the platform is developed, and the build process ends successfully only if all the tests pass on the CD test deployment.

Their automatic execution not only contributes to high quality between platform updates, but also ensures the availability of all services necessary for platform operation (e.g., access to real time data from smart meters or availability of the load forecasting service) even in the event of edge cases (e.g., availability of an API route under different auth conditions).

Figure 11 and Figure 12 show the result of a test run on a development environment and the results of the different test run automatically executed and passed on the CI/CD environment.



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

```
wardialer@hawking:~/workspace/sofie/integration/pilots_deployments/italian-energy-pilot|master$
./test.sh

platform darwin -- Python 3.7.1, pytest-6.1.1, py-1.9.0, pluggy-0.13.1 -- /Users/wardialer/.pyenv/versions/3.7.1/bin/python3.7
cachedir: .pytest_cache
rootdir: /Users/wardialer/workspace/sofie/integration/pilots_deployments/italian-energy-pilot, configfile: pytest.ini
plugins: tavern-1.6.0
collected 15 items

tests/test_dso.tavern.yaml::Charging station info PASSED [ 6%]
tests/test_dso.tavern.yaml::Charging sessions PASSED [ 13%]
tests/test_dso.tavern.yaml::EV info PASSED [ 20%]
tests/test_dso.tavern.yaml::Test FM dashboard frontend PASSED [ 26%]
tests/test_dso.tavern.yaml::Test DSO dashboard frontend PASSED [ 33%]
tests/test_dso.tavern.yaml::DSO backend no authentication PASSED [ 40%]
tests/test_dso.tavern.yaml::Login with invalid username and password PASSED [ 46%]
tests/test_dso.tavern.yaml::Attempt DSO login with wrong password PASSED [ 53%]
tests/test_dso.tavern.yaml::Log in to DSO with valid user PASSED [ 60%]
tests/test_dso.tavern.yaml::DSO backend with authentication (live data) PASSED [ 66%]
tests/test_dso.tavern.yaml::DSO backend with authentication (forecast) PASSED [ 73%]
tests/test_dso.tavern.yaml::Marketplace requests list PASSED [ 80%]
tests/test_dso.tavern.yaml::Marketplace open requests PASSED [ 86%]
tests/test_dso.tavern.yaml::Marketplace request creation without username and password PASSED [ 93%]
tests/test_dso.tavern.yaml::Marketplace request creation with username and password PASSED [100%]

===== warnings summary =====
tests/test_dso.tavern.yaml: 15 warnings
/Users/wardialer/.pyenv/versions/3.7.1/lib/python3.7/site-packages/tavern/testutils/pytestutils/item.py:63: PytestDeprecationWarning: The '_fillfuncargs' function is deprecated,
use function_request._fillfixtures() instead if you cannot avoid reaching into internals.
  fixtures.fillfixtures(self)

-- Docs: https://docs.pytest.org/en/stable/warnings.html
generated xml file: /Users/wardialer/workspace/sofie/integration/pilots_deployments/italian-energy-pilot/testresults.xml
===== 15 passed, 15 warnings in 25.59s =====
wardialer@hawking:~/workspace/sofie/integration/pilots_deployments/italian-energy-pilot|master$
```

Figure 11: DEFM integration tests executed on development environment.

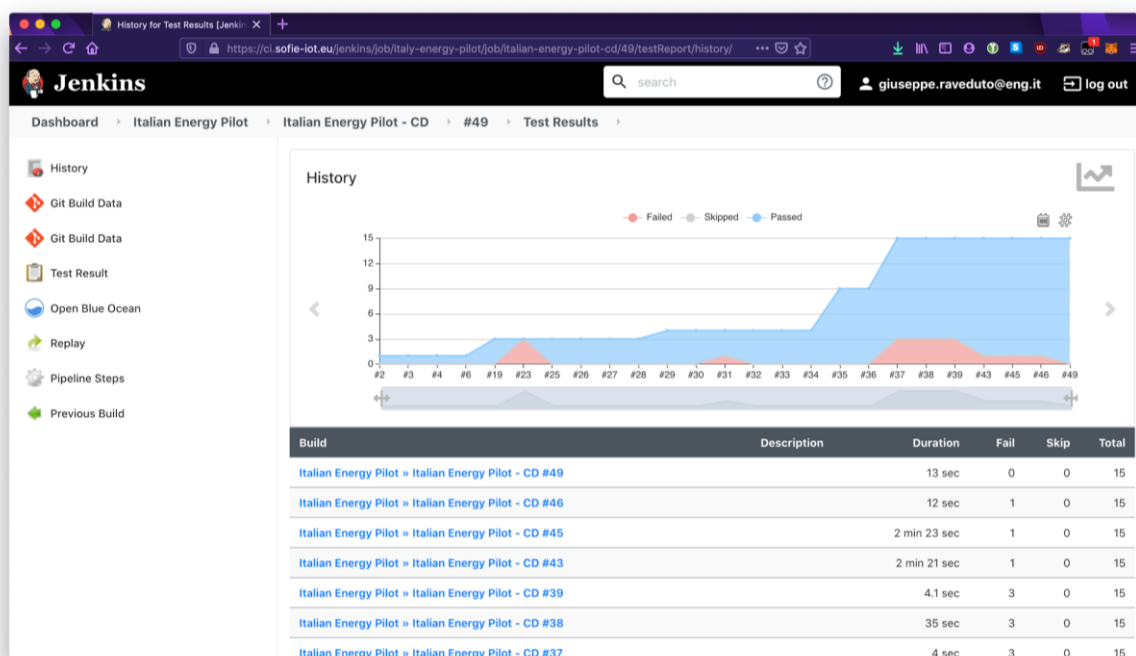


Figure 12: History of DEFM pilot's integration tests results on CI/CD environment.

Considering the platform macro-functionalities and the test cases already defined, the integration tests that were developed can be divided into three different groups:

- DSO Platform tests
- Fleet Manager Platform tests



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

- Marketplace Platform tests

ensuring that the platform, once deployed, can interface with the pilot's infrastructure to provide the services required by the DSO (e.g., availability of the DSO dashboard, access to smart meter live data, availability of the production and load forecasts), the Fleet Manager (e.g., access to EVSEs live data, availability of the FM dashboard, access to EVs live data), or both (e.g., creation of new requests and offers, availability of existing requests and offers).

The functional tests that were used to validate the pilot platform were all successful. Table 14 below, describes in detail the results of the functional tests complementing the automated tests.

Table 14: Functional tests.

Test ID	DEFM_TC01
Test description	Measurements from each deployed smart meter device are collected by the corresponding IoT platform and they are properly stored in its database system.
Test location	IoT platform used to monitor electricity grid
Related use cases	DEFM_UC1, DEFM_UC7
Related requirements	REQ_DEFM1.1, REQ_DEFM1.2, REQ_DEFM1.3, REQ_DEFM1.4, REQ_DEFM1.5, REQ_DEFM1.6, REQ_DEFM7.1
Feature(s) under test	Metering & data collection
Components involved	Advanced Metering Infrastructure (AMI) platform, FA, Forecasting System, Application back-end, web application front-end
Test environment	Smart meters are placed on pilot site and IoT platform is operational
Dependencies	N/A
Steps	1. Smart meter devices are deployed on site and are properly configured to communicate and send data to the corresponding IoT platform. 2. Collect data from a given period of time (e.g. a few days) 3. Use IoT platform API to retrieve data from each integrated smart meter devices within a specific time period.
Pass criteria	Historical and real-time data provided by the smart meters are properly retrieved



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines				
Security:	Public	Date:	7.5.2021	Status:	Completed
		Version:	1.10		

Result	<p>Request: http://[Host]:[Port]/data/[MeterId]/[DatetimeStart]/[DatetimeEnd]</p> <p>Response (truncated) of the passed test:</p> <pre>{"result":[[1608223860000,80.02783333333333],[1608223920000,83.21233333333333],[1608223980000,82.96275000000001],[1608224040000,82.61133333333333],[1608224100000,85.32533333333333],[1608224160000,84.15541666666668],[1608224220000,82.18725],[1608224280000,84.64308333333334],[1608224340000,83.99558333333333],[1608224400000,82.33816666666668],[1608224460000,85.01291666666667],[1608224520000,85.22725000000001],[1608224580000,84.41],[1608224640000,84.42833333333333],[1608224700000,86.40974999999999],[1608224760000,87.96466666666667],[1608224820000,85.67908333333334],[1608224880000,81.70325000000001],[1608224940000,80.91558333333334],[1608225000000,83.55233333333334],[1608225060000,83.52816666666668],[1608225120000,83.13008333333333],[1608225180000,81.90333333333334],[1608225240000,83.25466666666667],[1608225300000,82.26083333333334],[1608225360000,82.45191666666668],[1608225420000,81.68683333333333],[1608225480000,81.06600000000002],[1608225540000,80.24133333333334],[1608225600000,81.69783333333334],[1608225660000,82.92016666666667],[1608225720000,80.12541666666668],[1608225780000,79.79275],[1608225840000,77.61449999999999],[1608225900000,78.67116666666666],[1608225960000,78.28866666666667],[1608226020000,77.48575000000001],[1608226080000,79.39766666666667],[1608226140000,78.47458333333334],[1608226200000,79.39966666666666],[1608226260000,80.38800000000002],[1608226320000,82.76933333333334],[1608226380000,82.71583333333332],[1608226440000,81.52],[1608226500000,79.27208333333334],[1608226560000,77.21983333333334],[1608226620000,77.02708333333335],[1608226680000,82.23966666666668],[1608226740000,81.09558333333335],[1608226800000,79.64066666666668],[1608226860000,78.70775000000002],[1608226920000,80.57308333333333],[1608226980000,81.38725000000001],[1608227040000,80.18183333333334],[1608227100000,74.29608333333336],[1608227160000,76.92683333333333],[1608227220000,77.82383333333335],(...TRUNCATED)],"message":"Meter BBB6099, validation succeeded"}</pre>
--------	--



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Test ID	DEFM_TC02
Test description	Measurements from each deployed charging station are collected by the corresponding IoT platform and they are properly stored in its database system.
Test location	IoT platform used to monitor charging stations
Related use cases	DEFM_UC2, DEFM_UC3, DEFM_UC3, DEFM_UC4, DEFM_UC5
Related requirements	REQ_DEFM2.1, REQ_DEFM2.2, REQ_DEFM4.1, REQ_DEFM4.2, REQ_DEFM4.3, REQ_DEFM4.4, REQ_DEFM4.6, REQ_DEFM4.7, REQ_DEFM5.1
Feature(s) under test	Metering & data collection
Components involved	E-Mobility platform, FA, Forecasting System, Application back-end, web application front-end
Test environment	Charging stations are placed on pilot site and IoT platform is operational
Dependencies	N/A
Steps	<ol style="list-style-type: none"> 1. Charging stations are deployed on site and are properly configured to communicate and send data to the corresponding IoT platform. 2. Collect data from a given period of time (e.g. few days) 3. Use IoT platform API to retrieve data from each integrated charging station within a specific time period.
Pass criteria	Historical and real-time data provided by the charging stations are properly retrieved
Result	<p>Request #1 (charging station info & real-time status): <a chargeboxid\":\"24\"}"="" href="https://panel.spot-link.it/public/api/chargeboxes/{\">https://panel.spot-link.it/public/api/chargeboxes/{\"chargeboxID\":\"24\"} Method: GET Response #1:</p> <pre>{ \"chargeboxID\": \"24\", \"address\": \"ASM Terni, Strada di Maratta Bassa, TR - Parcheggio\", \"latitude\": \"42.5673558\", \"longitude\": \"12.6070454\", \"maxPwrAC\": \"64\", \"maxPwrDC\": \"0\", \"drStatus\": \"1\", \"idSocketA\": \"37\", \"tSocketA\": \"type 2\", \"stSocketA\": \"charging\", \"idSocketB\": \"38\", \"tSocketB\": \"type 2\", \"stSocketB\": \"waiting\" }</pre> <p>Request #2 (charging sessions data): <a chargeboxid\":\"24\"}"="" href="https://panel.spot-link.it/public/api/historyCharges/{\">https://panel.spot-link.it/public/api/historyCharges/{\"chargeboxID\":\"24\"} Method: GET Response #2 (truncated):</p> <pre>{ \"numCharge\": \"686\", \"recharges\": [</pre>



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

	<pre> { "chargeID": "12523", "dataStart": "2020-12-18 15:19:03", "dataStop": "notEnded", "kWh": 0, "importoTot": "0", "address": "ASM Terni, Strada di Maratta Bassa, TR - Parcheggio", "idUser": "1403", "socketID": "37", "chargeboxID": "24", "nomePresa": "presa A" }, { "chargeID": "12477", "dataStart": "2020-12-16 11:30:21", "dataStop": "2020-12-16 13:32:04", "kWh": 23, "importoTot": "8.28", "address": "ASM Terni, Strada di Maratta Bassa, TR - Parcheggio", "idUser": "1403", "socketID": "37", "chargeboxID": "24", "nomePresa": "presa A" } } </pre>
Test ID	DEFM_TC03
Test description	Measurements from each deployed electric vehicle are collected by the corresponding IoT platform and they are properly stored in its database system.
Test location	IoT platform used to monitor electric vehicles
Related use cases	DEFM_UC2, DEFM_UC3, DEFM_UC4, DEFM_UC6
Related requirements	REQ_DEFM4.1, REQ_DEFM4.2, REQ_DEFM4.3, REQ_DEFM4.5, REQ_DEFM4.6, REQ_DEFM4.7
Feature(s) under test	Metering & data collection
Components involved	E-Mobility platform, FA, Forecasting System, Application back-end, web application front-end
Test environment	Electric vehicles are placed on pilot site and IoT platform is operational
Dependencies	N/A
Steps	<ol style="list-style-type: none"> 1. Electric vehicles are deployed on pilot site and are properly configured to communicate and send data to the corresponding IoT platform. 2. Collect data from a given period of time (e.g., a few days) 3. Use IoT platform API to retrieve data from each integrated electric vehicle within a specific time period.



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

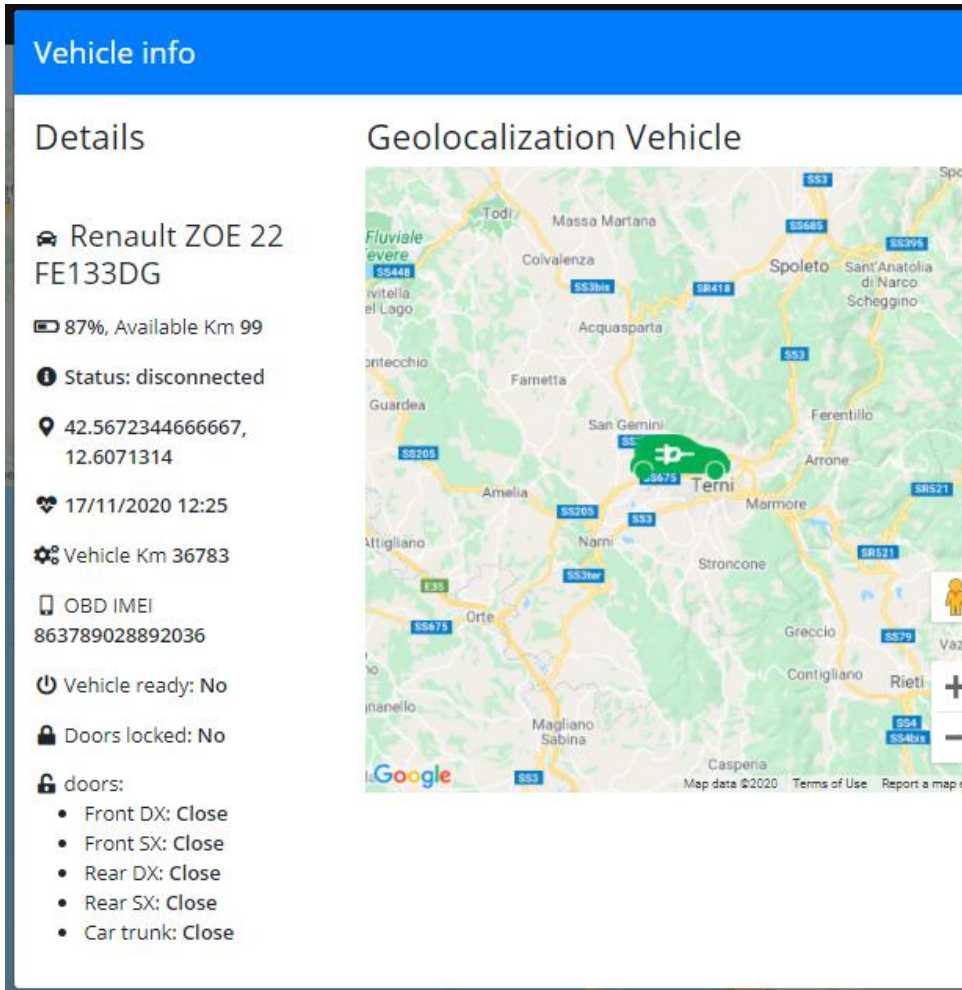
Pass criteria	Historical and real-time data provided by the electric vehicles are properly retrieved
Result	Request: https://panel.spot-link.it/public/api/ev{"vehicleID":"10"} Method: GET Response: [{"characteristic":{"vehicleID":"10","model":"Renault ZOE","connector":"type2","batteryKw":"41","batteryPower":"22","licensePlate":"FE132DG"},"status":{"status":"disconnected","timestamp":"2021-11-16T10:07:06","autonomyKm":"261","speed":"0","batteryPerc":"100","latitude":"43.0720388833333","longitude":"12.3419610333333","ready":false,"doorsLocked":"yes","frontDX":"close","frontSX":"close","rearDX":"close","rearSX":"close","carTrunk":"close"}}]
Test ID	DEFM_TC04
Test description	Commands from IoT platform are received from each deployed charging station and are properly implemented.
Test location	IoT platform used to control charging stations
Related use cases	DEFM_UC2, DEFM_UC3, DEFM_UC5
Related requirements	REQ_DEFM2.1, REQ_DEFM2.2, REQ_DEFM5.1
Feature(s) under test	Remote control
Components involved	E-Mobility platform, FA, Application back-end, web application front-end
Test environment	Charging stations are placed on pilot site and IoT platform is operational
Dependencies	N/A
Steps	1. Charging stations are deployed on site and are properly configured to communicate to the corresponding IoT platform. 2. Use IoT platform API to send commands to the integrated charging station
Pass criteria	Start&Stop and power output modulation remote commands are performed by deployed charging stations
Result	Request #1 (Start command): <a chargeboxid\":\"24\",\"socketid\":\"37\"}"="" href="https://panel.spot-link.it/public/api/startChargebox/{\">https://panel.spot-link.it/public/api/startChargebox/{\"chargeboxID\":\"24\",\"socketID\":\"37\"} Method: GET Response #1: "START_OK" Request #2 (Stop command): <a chargeboxid\":\"24\",\"socketid\":\"37\"}"="" href="https://panel.spot-link.it/public/api/stopChargebox/{\">https://panel.spot-link.it/public/api/stopChargebox/{\"chargeboxID\":\"24\",\"socketID\":\"37\"} Method: GET Response #2: "STOP_OK" Request #3 (Power output modulation command):



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

	<a \"socketid\":\"37\"}"="" chargeboxid\":\"24\",="" href="https://panel.spot-link.it/public/api/stopChargebox/{\">https://panel.spot-link.it/public/api/stopChargebox/{\"chargeboxID\":\"24\", \"socketID\":\"37\"} Method: GET Response #3: "SET_OK"
Test ID	DEFM_TC05
Test description	The marketplace platform correctly exposes the smart contract's functionalities. Users can operate the platform.
Test location	Marketplace platform, deployed at pilot site
Related use cases	DEFM_UC1
Related requirements	REQ_DEFM1.1-REQ_DEFM1.6
Feature(s) under test	Decentralized marketplace management
Components involved	SOFIE Marketplace, DSO/FM Applications back-end, web applications front-end
Test environment	The smart contract is developed and deployed on site. The marketplace module provides endpoints to map smart contract methods with APIs
Dependencies	N/A
Steps	1. The marketplace software module is accessible 2. The DSO and the fleet manager can access the list of requests and filter the requests by status 3. The DSO can create new requests 4. It is not possible to create new requests without proper authentication
Pass criteria	All the marketplace functionalities are working as expected, tokens are transferred after a successful transaction
Result	Integration tests: 1. Marketplace requests list 2. Marketplace open requests 3. Marketplace request creation with username and password 4. Marketplace request creation without username and password
Test ID	DEFM_TC06
Test description	Fleet Manager can access and perform all the services provided by the eMobility platform
Test location	On site by using the eMobility web application
Related use cases	DEFM_UC2, DEFM_UC3, DEFM_UC4, DEFM_UC5, DEFM_UC6, DEFM_UC8
Related requirements	REQ_DEFM2.1, REQ_DEFM2.2, REQ_DEFM4.1, REQ_DEFM4.2, REQ_DEFM4.3, REQ_DEFM4.4, REQ_DEFM4.5, REQ_DEFM4.6, REQ_DEFM4.7, REQ_DEFM5.1, REQ_DEFM8.1

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

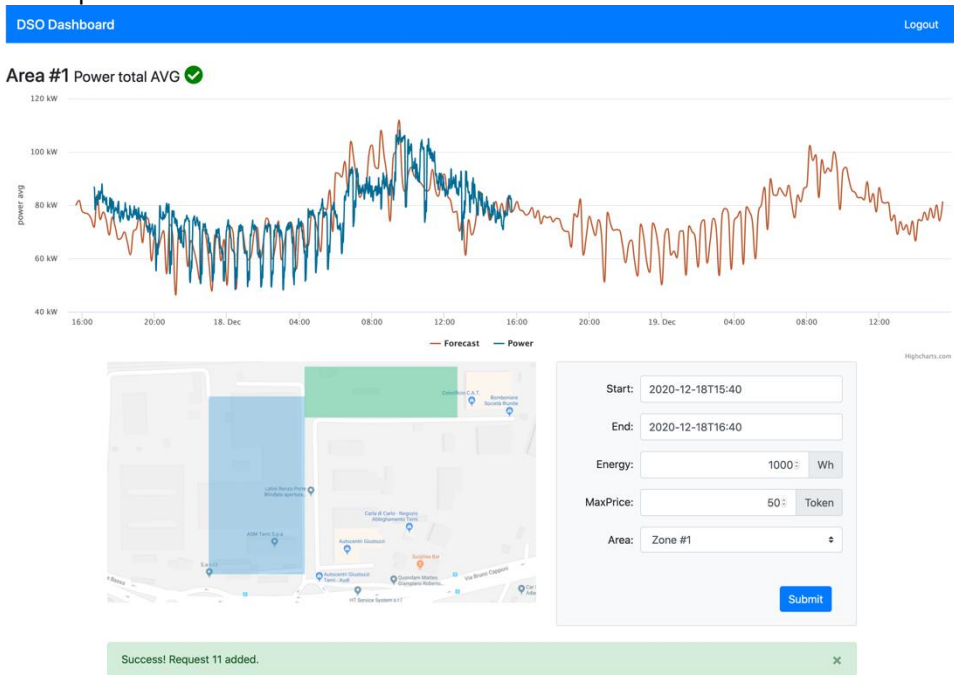
Feature(s) under test	Electric mobility IoT web platform
Components involved	E-Mobility platform, FA, Application back-end, web application front-end
Test environment	An actor uses the eMobility web application to perform the action under test
Dependencies	DEFM_TC02, DEFM_TC03, DEFM_TC04 are successful
Steps	<ol style="list-style-type: none"> 1. An actor (Fleet Manager) registers its profile in the eMobility web applications. 2. The actor integrates its electric vehicles. 3. The actor is enabled to participate in the Decentralized Energy Flexibility Marketplace.
Pass criteria	Registration of an electric vehicle with an already used OBD is prohibited.
Result	
Test ID	DEFM_TC07
Test description	The DSO operator can access the platform and obtain load data and forecast from the IoT smart meters
Test location	On site by using the DSO web application



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Related use cases	DEFM_UC1, DEFM_UC6
Related requirements	REQ_DEFM1.1-REQ_DEFM1.6; REQ_DEFM2.2
Feature(s) under test	DSO IoT web platform
Test environment	Smart meters are placed on site and IoT platforms are operational. The IoT platform provides endpoints to retrieve historical and real-time data provided by the smart meters
Dependencies	N/A
Steps	<ol style="list-style-type: none"> The operator accesses the DSO dashboard <ol style="list-style-type: none"> it is not possible to log in using the wrong credentials The dashboard queries the backend for real data and load forecast <ol style="list-style-type: none"> The dashboard shows the smart meter data for the last 24 hours The dashboard shows the load forecast for the next 24 hours It is not possible to query the backend without proper authentication
Pass criteria	The operator can log in to the platform, the dashboard shows the collected data.
Result	Integration tests: <ol style="list-style-type: none"> Test DSO dashboard frontend Login with invalid username and password Attempt DSO login with wrong password DSO backend with authentication (live data) DSO backend with authentication (forecast) DSO backend no authentication
Test ID	DEFM_TC08
Test description	DSO creates a Demand Response (DR) campaign and Fleet Managers participate to provide flexibility
Test location	On site by using the DSO and eMobility web applications
Related use cases	DEFM_UC1, DEFM_UC2, DEFM_UC3, DEFM_UC4, DEFM_UC5, DEFM_UC6, DEFM_UC7, DEFM_UC8, DEFM_UC9
Related requirements	REQ_DEFM1.1, REQ_DEFM1.2, REQ_DEFM1.3, REQ_DEFM1.4, REQ_DEFM1.5, REQ_DEFM1.6, REQ_DEFM2.1, REQ_DEFM2.2, REQ_DEFM4.1, REQ_DEFM4.2, REQ_DEFM4.3, REQ_DEFM4.4, REQ_DEFM4.5, REQ_DEFM4.6, REQ_DEFM4.7, REQ_DEFM5.1, REQ_DEFM7.1, REQ_DEFM8.1
Feature(s) under test	DR campaign
Components involved	SOFIE Marketplace, DSO/FM Applications back-end, web applications front-end
Test environment	Actors use DSO and eMobility web applications to perform the action under test
Dependencies	DEFM_TC01, DEFM_TC02, DEFM_TC03, DEFM_TC04, DEFM_TC05, DEFM_TC06, DEFM_TC07 are successful

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines				
Security:	Public	Date:	7.5.2021	Status:	Completed
		Version:	1.10		

Steps	<ol style="list-style-type: none"> 1. DSO identifies flexibility need via District Forecasting. 2. DSO creates a DR campaign in the Decentralized Energy Flexibility Marketplace. 3. Fleet Managers in the Decentralized Energy Flexibility Marketplace, provide their flexibility offers. 4. Fleet Manager with the best offer wins the auction 5. Smart contract between Fleet Manager and DSO is signed 6. Smart contract between Fleet Manager and Energy Retailer is signed 7. Fleet Manager sends the electric vehicles to the predetermined location point at the predetermined time and starts the charge (s) 8. Flexibility is provided, the parties have received their due and DR campaign is concluded
Pass criteria	Decentralized Energy Flexibility Marketplace subscription
Result	<p>1. Request Creation</p>  <p>7. Offers received</p>



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

DSO WALLET

1000 ST

0x5A7604dbe012C19F1131d10A5E16923048E67017

INTERLEDGER

There are not label.

#	Author	Status	Winning Offer	Paid	Info
1	dso: 0x5A7604dbe012C19F1131d10A5E16923048E67017	open	# false		

Details

Delivery Interval: 23/11/2020 11:52 - 23/11/2020 12:52
Zone: Zone_1
Quantity: 1000 Wh
Max Price: 50
Deadline: 2020-11-23 10:52:00

[Get Winning Offer](#) [Unlock Payment](#)

Offers

#	Author	Price
1	fm: 0xBAFAB60f35FcBB82C806026D4E0D03eb3c9e5FB2	45
2	fm: 0xBAFAB60f35FcBB82C806026D4E0D03eb3c9e5FB2	34
3	fm: 0xBAFAB60f35FcBB82C806026D4E0D03eb3c9e5FB2	40

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

1. Winning offer selection

#	Author	Status	Winning Offer	Paid	Info
1	dso: 0x5A7604dbe012C19F1131d10A5E16923048E67017	decided	# 2		

Details

Delivery Interval: 23/11/2020 11:52 - 23/11/2020 12:52
Zone: Zone_1
Quantity: 1000 Wh
Max Price: 50
Deadline: 2020-11-23 10:52:00

[Get Winning Offer](#) [Unlock Payment](#)

Offers

#	Author	Price
1	fm: 0xBAFAB60f35FcBB82C806026D4E0D03eb3c9e5FB2	45
2	fm: 0xBAFAB60f35FcBB82C806026D4E0D03eb3c9e5FB2	34
3	fm: 0xBAFAB60f35FcBB82C806026D4E0D03eb3c9e5FB2	40

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

2. Delivery

DSO WALLET

1000 ST

0x5A7604dbe012C19F1131d10A5E16923048E67017

INTERLEDGER

Request	Offer	Token	Status
1	2	34	

#	Author	Status	Winning Offer	Paid	Info
1	dso: 0x5A7604dbe012C19F1131d10A5E16923048E67017	decided	# 2		

Details

Delivery Interval: 23/11/2020 11:52 - 23/11/2020 12:52
Zone: Zone_1
Quantity: 1000 Wh
Max Price: 50
Deadline: 2020-11-23 10:52:00

100

[Get Winning Offer](#) [Unlock Payment](#)

Offers

#	Author	Price
1	fm: 0xBAFAB60f35FcBB82C806026D4E0D03eb3c9e5FB2	45
2	fm: 0xBAFAB60f35FcBB82C806026D4E0D03eb3c9e5FB2	34
3	fm: 0xBAFAB60f35FcBB82C806026D4E0D03eb3c9e5FB2	40

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

3. Payment

SOFIE

66(121)



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

DSO WALLET

966 ST

0x5A7604dbe012C19F1131d10A5E16923048E67017

INTERLEDGER

Request	Offer	Token	Status
1	2	34	\$

#	Author	Status	Winning Offer	Paid	Info
1	dso: 0x5A7604dbe012C19F1131d10A5E16923048E67017	decided	# 2	⊗	⊗

Details

Delivery Interval: 23/11/2020 11:52 - 23/11/2020 12:52
 Zone: Zone_1
 Quantity: 1000 Wh
 Max Price: 50
 Deadline: 2020-11-23 10:52:00

100

Get Winning Offer
 Unlock Payment

Offers

#	Author	Price
1	fm: 0xBAFAB60F35FcBB82C806026D4E0D03eb3c9e5FB2	45
2	fm: 0xBAFAB60F35FcBB82C806026D4E0D03eb3c9e5FB2	34
3	fm: 0xBAFAB60F35FcBB82C806026D4E0D03eb3c9e5FB2	40

Request 1 is paid now.

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

In Table 15, data collected during the Decentralized Energy Flexibility Marketplace validation are shown:

Table 15: DEFM pilot collected data

Category	Data type	Unit	Frequency	Size
Charging Stations	ID	Natural Number	1 second	<1MB
	Address	Text		
	Latitude	Rational Number		
	Longitude	Rational Number		
	Maximum Power AC	Natural Number (Ampere)		
	Maximum Power DC	Natural Number (Ampere)		
	DR Status	Natural Number		
	Socket ID	Natural Number		
	Socket Type	Text		
	Socket Status	Text		



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

	Charging Session ID	Natural Number		
	Start Time	Natural Number		
	End Time	Natural Number		
	Energy	Rational Number (Watt*Hour)		
	Cost	Rational Number (€)		
Electric Vehicles	ID	Natural Number	5 seconds	<1MB
	Model	Text		
	Connector Type	Text		
	Nominal Battery Power	Natural Number (Watt)		
	Nominal Battery Energy	Natural Number (Watt*Hour)		
	License Plate	Text		
	Plug Status	Text		
	Timestamp	Natural Number		
	Kilometers Autonomy	Natural Number (Km)		
	EV Speed	Natural Number (Km*Hour)		
	Battery Percentage	Natural Number (%)		
	Latitude	Rational Number		
	Longitude	Rational Number		
	Engine Status	Text		
	Doors Status	Text		
	Load	kW	5 seconds	KBs

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Smart Meters (time series, for each smart meter)	Forecast	kW	on demand	
--	----------	----	-----------	--

5.3 Evaluation

5.3.1 Pilot performance assessment, KPIs evaluation, and benefits

All business KPIs were successfully achieved and reported in Table 16, showing the benefits of the SOFIE approach.

Table 16: Decentralized Energy Flexibility Market KPIs.

KPI	Name	Description	Metric	Method of measurement	Target	Result
Business goals						
KPI_DEFM_6	RPF reduction	Amount of RPF reduced	kWh/day	Measure the energy that flows from the users to the network in the secondary /primary substations.	About 15 kWh/day	13.7 kWh/day
KPI_DEFM_7	Power losses reduction	Reduced flow in comparison with business-as-usual operation	kWh	Measure the reduced flows because of a better overlapping among consumptions	About 1 kWh saved in comparison with Business-as-usual operation	about 0.6 kWh/day
KPI_DEFM_8	Voltage under the limits	Voltage waveforms among the limits	Voltage limits %	Assess the voltage waveforms among the limits and according to network configuration	Voltage limits in +/- 1%	minimum: -2 % maximum: + 6 %
KPI_DEFM_9	Green energy consumption	increased share of consumption from green energy producers	kWh	Measure the reduction of RPF and therefore the increased share of consumptions drawn from green energy producers	About 15 kWh	13.7 kWh
KPI_DEFM_10	EV fleet manager metrics	Involvement in DR campaign provide advantageous energy price for EV	Monetary savings	Measure the money saved involving EV fleet in DR Campaigns:	money saved: 0.13 €/kWh	0.06 €/kWh for Fleet Managers in current



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

		Fleet Manager, due to DSO benefits and Retailers auction		energy cost in DR campaign vs energy cost in not- DR campaign		scenario, up to 0.13 €/kWh considering end user's self- consumption
--	--	---	--	--	--	---

Within the System Performance KPIs, KPI_DEFM 6-10 can be regarded efficiency-related. They have been calculated analysing of 1-year (2019) of consumption in the ASM. The data were collected and elaborated through a MATLAB script and they were referred to timestamps of 10 minutes.

Initially, the scenario where flexibility requests are not present was considered (ex-ante scenario). The local power generation comes from PV plants, whereas the consumption is due to the facilities and the EV charging sessions. The second scenario is characterized by a different scheduling of the charging sessions according to some constraints, reported in [D5.3].

The constraints introduce an important improvement in the realism of the model. In this scenario the new charging session scheduling and relevant KPIs are directly calculated from the model of the network.

- KPI_DEFM_6, RPF reduction Amount of RPF is on average 13.7 kWh/ day, applicable only if a charging session happen very close to the expected value reported in D5.1 (i.e., 15kWh/

$$KPI_DEFM_6 = ((RPF_{ex-ante} - RPF_{scenario})/RPF_{ex-ante}) * 100$$

- KPI_DEFM_7, Power losses reduction, considering that about 3 MWh could be consumed when it is locally produced, a beneficial effect is the reduction of that power in the Medium Voltage network and therefore power losses would be reduced up to 75% (i.e., a quarter of losses are produced in the LV part of the grid).

$$KPI_DEFM_7 = ((E_{lost,ex-ante} - E_{lost,SOFIE})/E_{lost,ex-ante}) * 100$$

- KPI_DEFM_8, Voltage under the limits Voltage waveforms, simulation results show that maximum and minimum voltages are 0.98 p.u. and 1.06 p.u., respectively. This KPI has been directly calculated from the model of the network and the measurement.
- KPI_DEFM_9, Green energy consumption, the increased share of consumption from green energy producers has been measured as the reduction of RPF and therefore the increased share of consumption drawn from green energy producers (i.e., about 13.7 kWh/day as in KPI _DEFM_6 RPF).

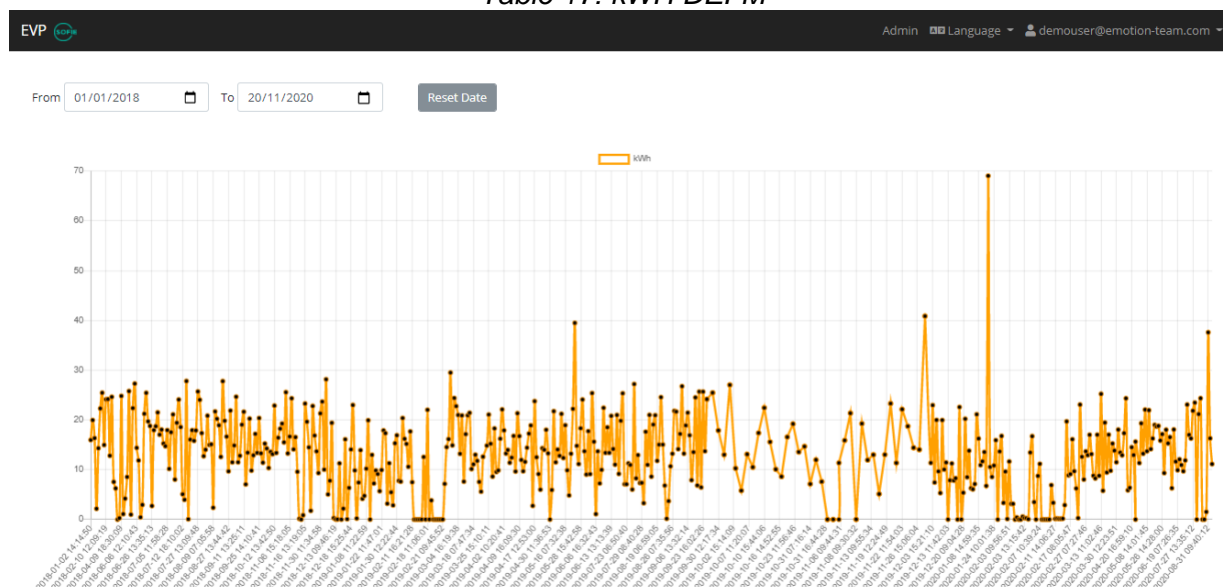
With respect to KPI_DEFM_10, During the SOFIE project, the maximum cost of 1 kWh of flexibility paid to the EV users has been calculated considering DSO money savings due to the Demand Response campaigns performed; it will be 0.06 €, resulting in about 2.6 € for a full recharge (considering a full recharge with a capacity of 44 kWh). Considering that the average cost per kWh for charging an electric vehicle in Italy is 0.45 €, this means that the incentive is equal to a 15% discount on the electric vehicle charge. For this reason, since during the three



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

years of SOFIE project the six electric vehicles deployed in the Italian pilot have withdrawn more than 7500 kWh from the electricity grid, we can assume that the economic saving is about 500 €.

Table 17: kWh DEFM



Furthermore, in addition to the economic benefit, the Demand Response campaigns have led to charging the electric vehicles with the energy surplus produced by the Italian pilot's photovoltaic systems, resulting in a significant benefit in terms of CO2 emissions: about 5000 kg of CO2 were not emitted into the atmosphere. Using renewable energy, as outlined by Italian Institute for Environmental Protection and Research (ISPRA), it is possible to avoid emitting 491 g CO2/kWh¹², an average amount of CO2 associated to the energy mix that is purchased by energy retailers.

The benefits derived from the SOFIE approach were also extended to a district of the distribution system of Terni. Three clusters of charging stations were virtually located in the district, as shown in Figure 13, identified by the red square. The total EVs consumption due to the charging sessions during an entire year were analyzed and distributed along the charging stations. The effect of the EVs penetration in the district in terms of reverse power flow was evaluated.

¹² ISPRA, "Fattori di emissione atmosferica di gas a effetto serra nel settore elettrico nazionale e nei principali Paesi Europei," 2019.

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10



Figure 13: Charging station locations - DEFM.

Figure 14 and Figure 15 show the reverse power flow at the primary substation related to the district, in case of EVs absence and presence. The maximum recorded without the EVs absorption is about 10 MW, while with EVs consumption about 6.5 MW.

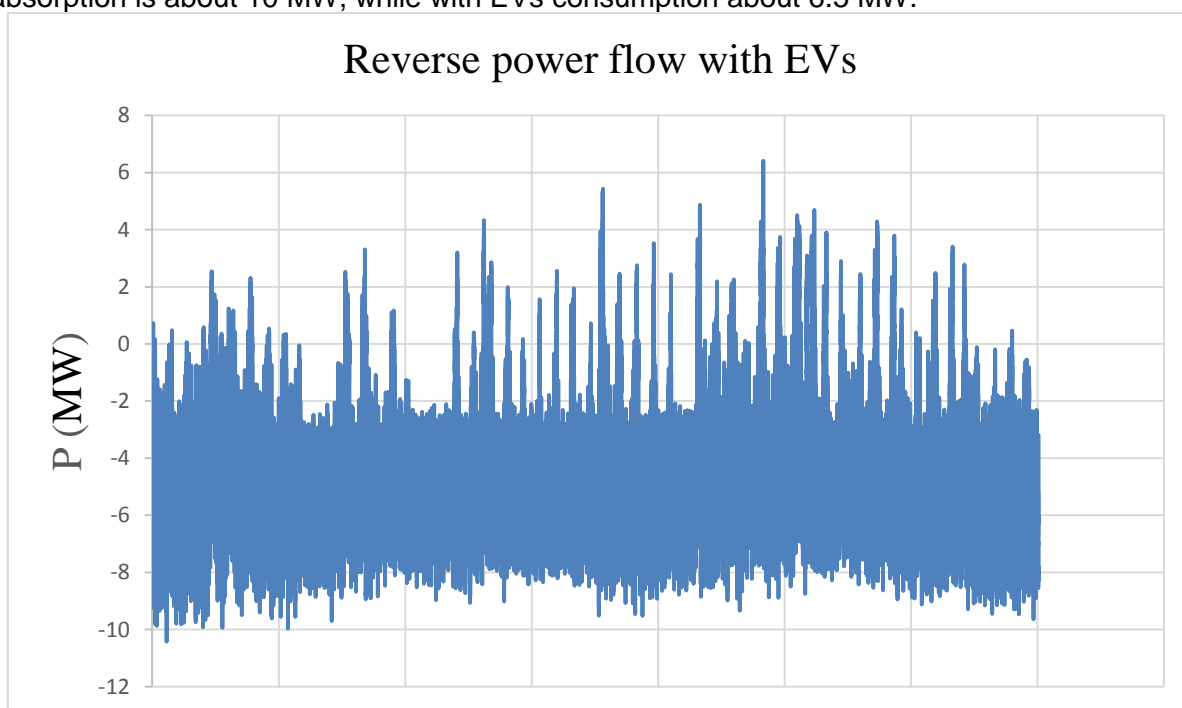


Figure 14: Reverse power flow (EV absence) - DEFM.

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

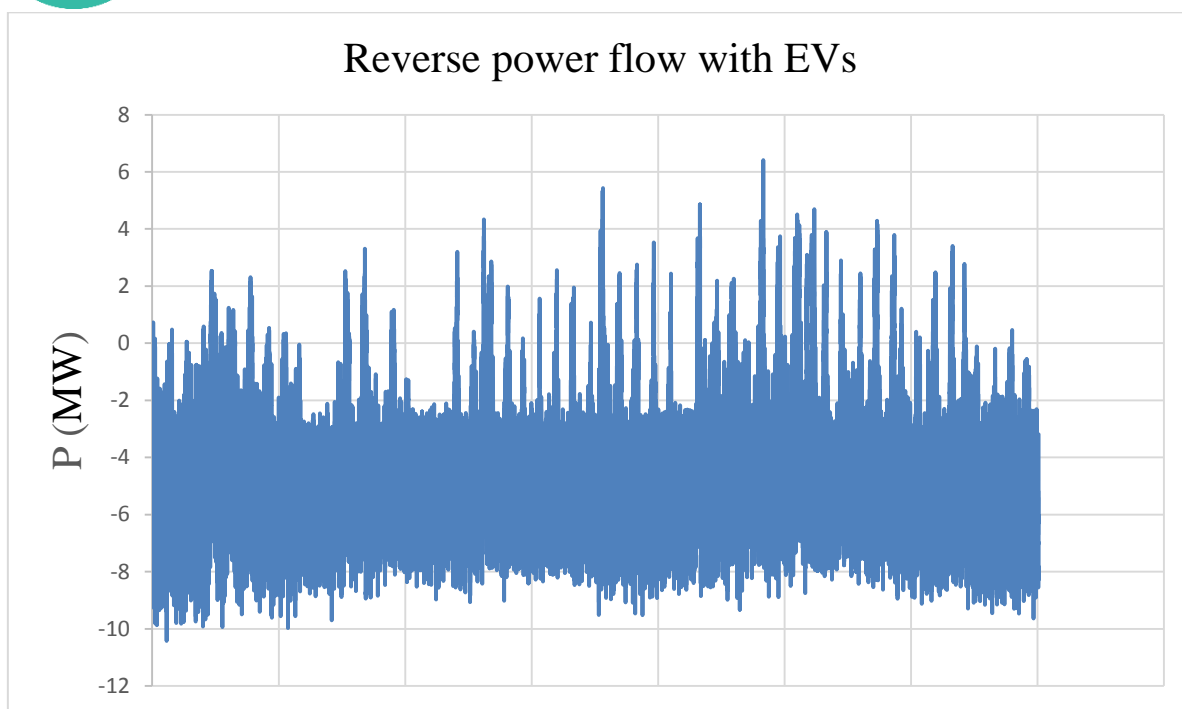


Figure 15: Reverse power flow (EV presence) - DEFM.

Marketplace Evaluation

For the marketplace component evaluation, some KPIs were already determined during the project. The evaluation reveals the results obtained in the pilot site. Table 18 shows the results considering a deployment over a public network. The latency time and the throughput can be improved by using a private network with an ad-hoc configuration.

Table 18: Marketplace pilot KPIs

ID	Name	Target	Result
KPI_DEFM_1	Ledger execution cost	As low as possible	~290k gas for <i>write</i> function
KPI_DEFM_2	Response time for requests and offers	< 5 min	< 13 s including block-time
KPI_DEFM_3	Response time for determining the winner of the auction	< 5 min	
KPI_DEFM_4	Response time for verifying the winning bid and compensating (or fining) the winner	< 5 min	
KPI_DEFM_5	Throughput	> 100 per hour	> 180 per Hour considering the whole lifecycle (request creation, offers creation, winner selection, finalisation)
KPI_DEFM_6	Scalability – time	Linear of sublinear	Constant



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

5.3.2 TRL

The pilot managed to achieve TRL-7 due to its demonstration the operational environment described in the previous sections. The TRL level that the pilot and the outcome assets that have derived from it reached, is summarized in Table 19:

Table 19: TRL levels of the DEFM pilot and assets

TRL	Justification
7	The DEFM pilot prototype was demonstrated in operational environment. The pilot platform backend components were deployed on a virtual machine that is part of ASM Terni ICT infrastructure. The web application dashboard was demonstrated using a workstation placed on the operation control centre and is accessible from ASM intranet. ASM Terni operates as DSO in the municipality of Terni, in Italy. The pilot trials took place at ASM's headquarters, a district including two PV arrays (180kWp and 60kWp) connected to the LV network, 72 kWh 2nd life Li-ion battery energy storage, ASM buildings with a base load varying between 50kW and 90kW and a peak load varying between 120kW and 170kW, three smart charging stations and a small fleet of electric vehicles. The Electric Vehicle Supply Equipment (EVSE) used in the demonstration includes two SPOTLINK - EVO smart charging stations connected to the e-mobility platform developed by Emotion and one EFAPOWER EV-QC45 charging station, with a total of 1590 charging sessions performed and 7902550 Wh supplied to electric vehicles. The EV fleet is composed by six EVs (Renault ZOE and Nissan LEAF), each of them equipped with an On-Board Diagnostic (OBD) device developed by Emotion, having more than 125,000 km traveled during SOFIE project lifetime.
7	The Decentralised Energy Flexibility Federation Adapter is used by Decentralized Energy Flexibility Marketplace pilot. It is deployed as part of the DEFM pilot prototype.
7	Decentralized Marketplace for Energy Flexibility Services are used by Decentralized Energy Flexibility Marketplace pilot. It is deployed as part of the DEFM pilot prototype.

5.4 Lessons learned and replication guidelines

In this section, we start with some lessons learned from the pilot platform deployment. The on-site deployment of the DEFM platform showed that the web interfaces are easy to use and this is also reflected in the fact that little training was needed for the operators. The concepts on which most attention has been placed concern the correlation between the flexibility requests and the actions on the marketplace (i.e., explaining the meaning of each field on the marketplace requests and offers and the different statuses of the requests).

The web interfaces developed have also proven effective in allowing the use of blockchain technology hiding the intrinsic complexity to the end user, who can use the functionality provided by the platform without worrying about knowing in detail the technical specifications of distributed networks and cryptography.

The training results may be affected by the fact that the participants were familiar with the platform since its design, and this might not be always the case especially in case of participants unfamiliar with technology or energy flexibility concepts.

5.4.1 Replication guidelines

The DEFM pilot is composed of SOFIE components, namely SOFIE Marketplace, Interledger, and Semantic Representation (SR) components, a Federation Adapter that is used to collect the input data from the IoT devices, and pilot-specific software components.

Figure 16, below, shows where the components are positioned with respect to a multi-tier architecture. It can be seen how the SOFIE components manage the business logic and the Federation Adapter acts as a gateway enabling the communication with the IoT devices.



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

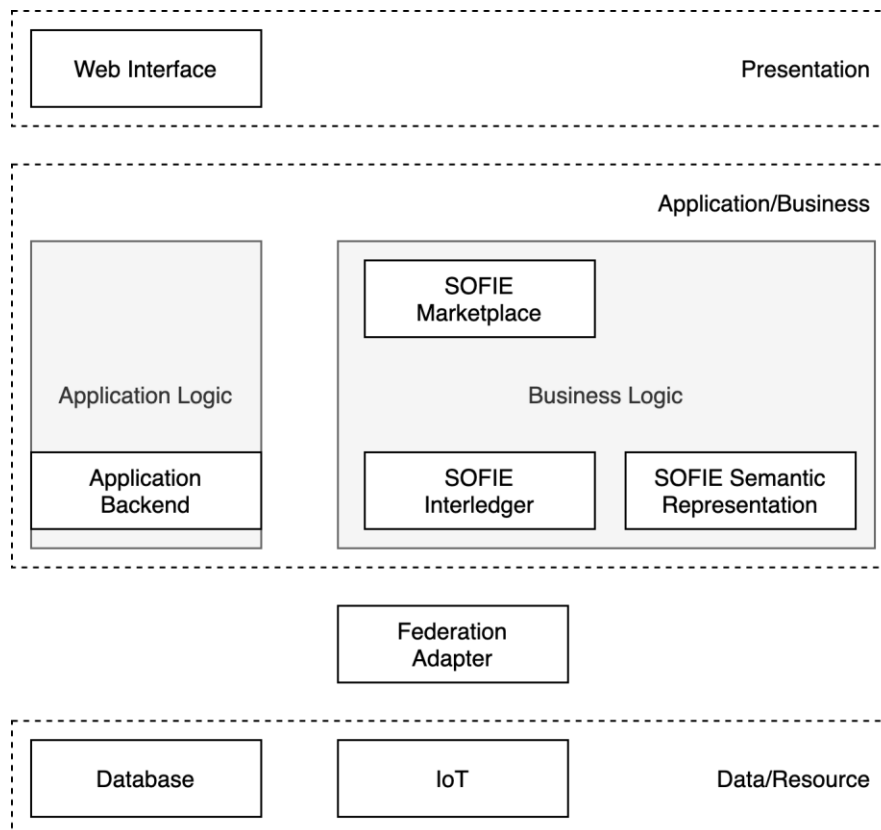


Figure 16: Components role in DEFM pilot's deployment.

Each of these components is part of the SOFIE software release and is publicly available for deployment and as a reference¹³¹⁴¹⁵¹⁶.

Each component is available as a containerized application. Containers do not require dependencies on the application infrastructure, reducing the operation complexity and being extremely suitable for automation services and tasks. Therefore, a developer that would like to replicate a decentralized marketplace needs to configure the SOFIE components and Federation Adapters following the provided documentation and examples and to implement the application logic and the interface for its end users.

The DEFM FA implementation is publicly available on GitHub¹⁷ and can be used as a reference for the replication on other IoT platforms. For convenience, below are summarized the steps needed for replicating the deployment:

Key Technologies:

- Docker with docker-compose
- Node.js with npm

Execution:

¹³ <https://github.com/SOFIE-project/efm-federation-adapter>

¹⁴ <https://github.com/SOFIE-project/Marketplace>

¹⁵ <https://github.com/SOFIE-project/Interledger>

¹⁶ <https://github.com/SOFIE-project/Semantic-Representation>

¹⁷ <https://github.com/SOFIE-project/efm-federation-adapter>



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

The pre-requisites are provided as a docker-compose configuration file (docker-compose.yml), which includes the services specifications together with the appropriate configuration parameters.

The pre-required services are:

- FIWARE IoT-Agent
- FIWARE Orion
- FIWARE STH-Comet
- Mosquitto MQTT Broker
- MongoDB NoSQL Database

and can be launched with:

docker-compose up -d

The repository includes as documentation a demonstrative CLI application. To run the application, it is necessary to install the following packages as dependencies using npm:

- axios
- mqtt
- colors

This can be done with the command:

npm install

after that, the application can be executed with the command:

npm start

The command line utility can be used to verify the correct operation of the system, checking the IoT-Agent service health, provisioning service groups and sensors, sending context updates via MQTT and retrieving context data.

Services:

When the components are running, the following services are available:

- http://localhost:27017/ mongo-db
- http://localhost:1883/ mosquitto
- http://localhost:9001/ mosquitto
- http://localhost:8666/ fiware-sth-comet
- http://localhost:1026/ fiware-orion
- http://localhost:4041/ fiware-iot-agent
- http://localhost:7896/ fiware-iot-agent

The source code includes a sample client that can be used as is for understanding the basic operating principles or as a reference for more complex ad-hoc clients.

As a reference, in Table 20 below reports the interfaces developed for the DEFM pilot backend. The list can be used to implement other similar applications leveraging on the advantages provided by a decentralized marketplace.

Table 20: DEFM pilot interfaces.

Endpoint	/marketplace/requests
----------	-----------------------



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Method	GET
Description	List of marketplace requests
Request Example	/marketplace/requests
Response Example	<pre>{ requests: [{ author: '0x5A7604dbe012C19F1131d10A5E16923048E67017', deadline_date: '2021-10-02 13:36:00', deadline_timestamp: 1633174560000, decided: null, decided_offer: false, end_date: 1633178160, id: 2, is_paid: false, maxPrice: 51, offers: [], past: false, quantity: 0, request_date: 1601638583, start_date: 1633174560, status: 'open', type: 'Zone_2', typeNumber: 1 }] }</pre>
Method	POST
Description	Creation of a new market request
Request Example	<pre>{ 'quantity': 0, 'zone': 2, 'deadline': 1633176900000, 'startDate': 1633176900000, 'endDate': 1633180500000, 'maxPrice': 50 }</pre>
Response Example	'Request 3 added.'
Endpoint	/marketplace/requests/:<id>
Method	GET
Description	Returns a specific request, by ID
Request Example	/marketplace/requests/3
Response Example	<pre>{ requests: [{ author: '0x5A7604dbe012C19F1131d10A5E16923048E67017', deadline_date: '2021-10-02 14:15:00',</pre>



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines				
Security:	Public	Date:	7.5.2021	Status:	Completed
		Version:	1.10		

	<pre>deadline_timestamp: 1633176900000, decided: null, decided_offer: false, end_date: 1633180500, id: 3, is_paid: false, maxPrice: 50, offers: [{'id': 4, 'author': '0xBAFAB60f35FcBB82C806026D4E0D03eb3c9e5FB2', 'price': 30}, {'id': 5, 'author': '0xBAFAB60f35FcBB82C806026D4E0D03eb3c9e5FB2', 'price': 25}, {'id': 6, 'author': '0xBAFAB60f35FcBB82C806026D4E0D03eb3c9e5FB2', 'price': 40},], past: false, quantity: 0, request_date: 1601640941, start_date: 1633176900, status: 'open', type: 'Both', typeNumber: 2 }]</pre>
Method	PUT
Description	Updates an existing request. Useful to decide an open request or finalize a payment.
Request Example	/marketplace/requests/3?action=decide
Response Example	{ message: 'Request 3 is decided.' }
Method	POST
Description	Adds a new offer to the marketplace request
Request Example	<pre>{ 'id': 2, 'price': 30 }</pre>
Response Example	'Offer 7 added.'
Endpoint	/marketplace/addresses/:<address>/tokens
Method	GET
Description	Returns the number of tokens owned by the specified address
Request Example	/marketplace/addresses/0x5A7604dbe012C19F1131d10A5E16923048E67017/tokens
Response Example	900



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

6. Context-Aware Mobile Gaming Pilot

6.1 Pilot overview

The focus of the *Context-aware Mobile Gaming*¹⁸ (CAMG) pilot is to explore how DLTs can be used to provide new gaming features for players, as well as to validate the potential of location based IoT gaming use cases. The pilot seeks to overcome the known technical issues of DLTs with respect to scale, in order to cost-effectively support millions of active users per day.

A first game prototype was developed, that enables players to collect and trade in-game content, swap or trade with other players (e.g., characters, weapons, equipment, parts), leveraging DLTs to provide player ownership of the asset as well as transparency and consistency of asset attributes and transactions. Attributes, or the “DNA” of the in-game assets were published on the blockchain.

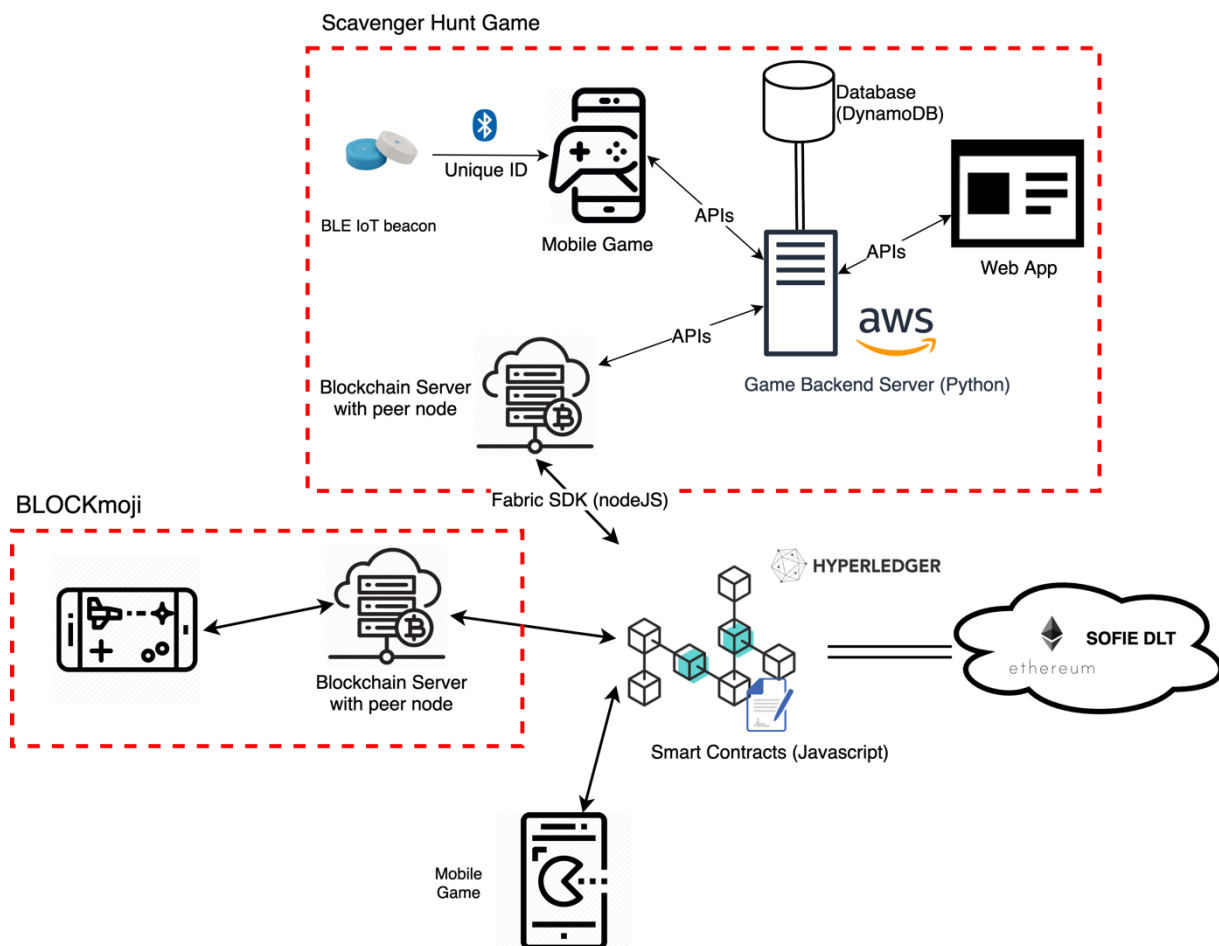


Figure 17: The high-level architecture of the CAMG pilot.

As our second use case, we developed a Scavenger Hunt game prototype in order to explore location based IoT gaming. In Figure 17: The high-level architecture of the CAMG pilot demonstrates the high-level architecture of our pilot, showing how the Scavenger Hunt game prototype connects with our Hyperledger Fabric platform. In the game, the player starts a hunt, which takes them on a journey of predetermined real-world locations. At each location, a *Bluetooth Low Energy* (BLE) beacon is deployed, either indoors or outdoors. When the mobile

¹⁸ The Context-Aware Mobile Gaming (CAMG) pilot was called Mixed Reality Mobile Gaming (MRMG) in previous deliverables.

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines				
Security:	Public	Date:	7.5.2021	Status:	Completed
		Version:	1.10		

game client detects the beacon, it means that the player has arrived at the correct location, and they receive a task in the form of a question. By observing their real-world surroundings, the player can learn the answer to the question, type it, and receive the clue on where the next correct location is. At the end of a hunt (a series of tasks and clues), the player receives rewards that can bring in-game advantages in the next hunts. The game steps are shown in Figure 18.

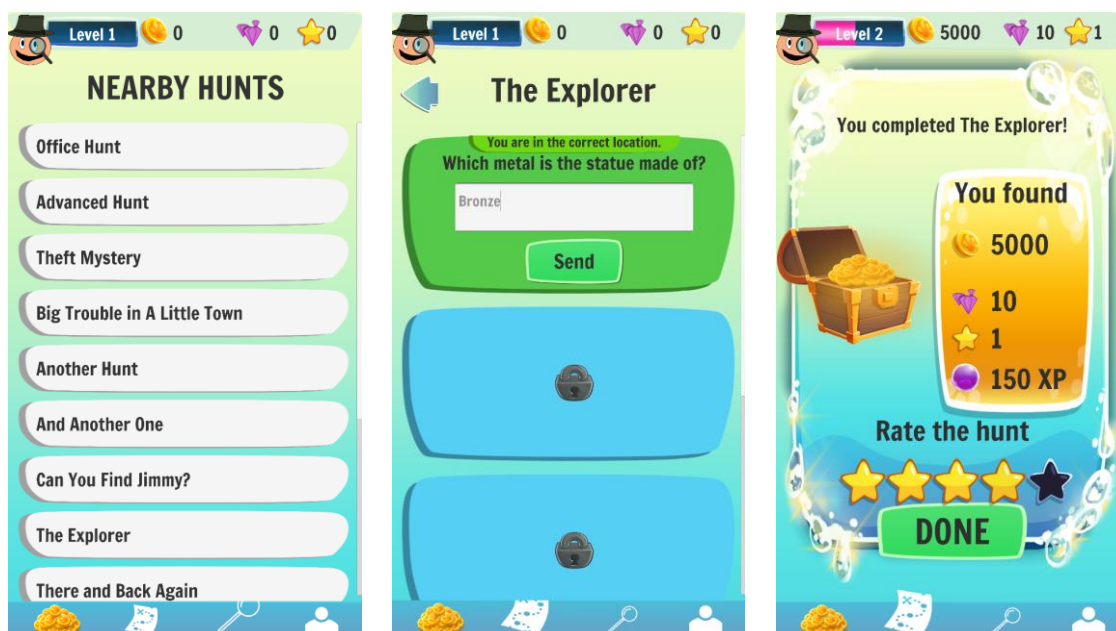


Figure 18: The Scavenger Hunt game prototype. Starting, playing, and ending a hunt on a mobile client - CAMG.

As additional rewards, the player receives items that are stored on a distributed ledger as non-fungible tokens. To browse and manage these items, a companion application was created: Blockmoji. In this mobile application, the player can see which items they own, and equip or unequip them on their virtual avatar (Figure 19). Shared items between Scavenger Hunt and Blockmoji demonstrate that it is possible to share the same items between multiple games, where it is up to the game designers to decide on how to interpret the attributes of the player's Blockmoji items and which in-game benefits they would bring. In our Scavenger Hunt game prototype, the Blockmoji items do not bring in-game benefits, but, instead, the game acts as a source of these items.

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

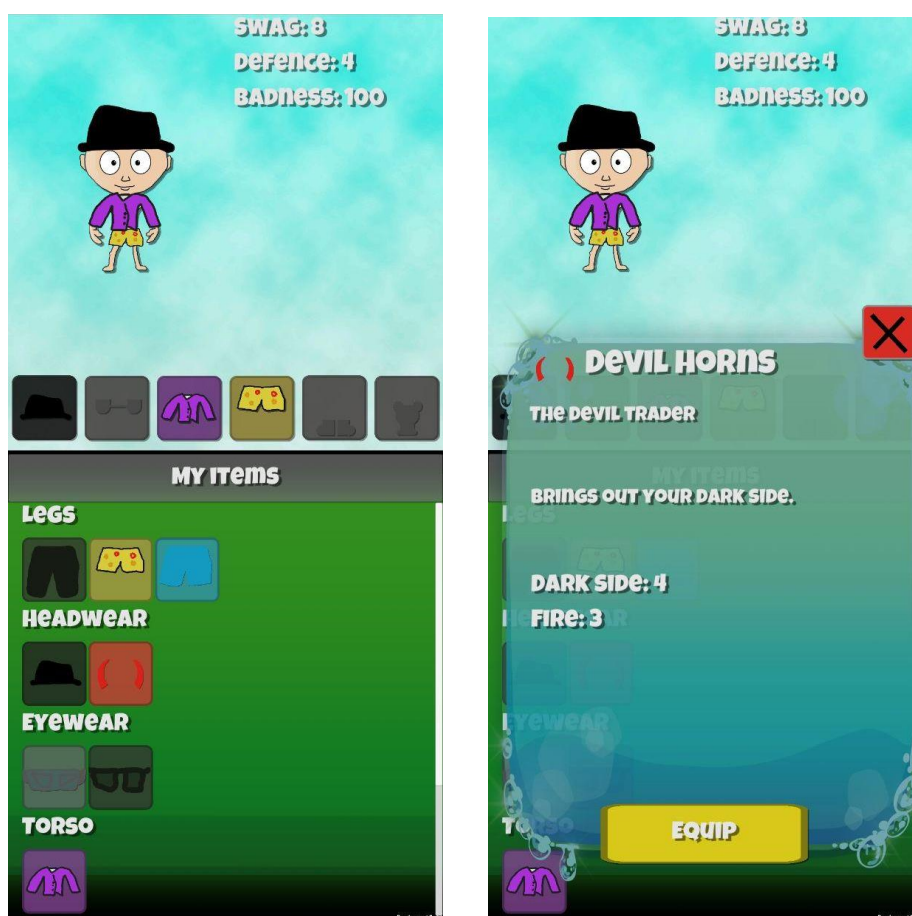


Figure 19: Viewing and equipping items in Blockmoji - CAMG.

In addition to these use cases, the Interledger and Marketplace components were integrated into our pilot. Furthermore, the Provisioning and Discovery component is being used to discover IoT beacons and add them to the database for a location-based game, such as for the Scavenger Hunt prototype.

The COVID-19 pandemic has not allowed us to playtest the Scavenger Hung game prototype. Regardless, all requirements that are listed in the validation matrix in 2020 can be validated, as these do not require physical presence.

Since D5.2 [D5.2] (July 2019), we have replaced multiple validation requirements of our pilot in order to better reflect which functionalities we expect from our use cases, as well as to better align with our planned mobile ads use case. The new requirements have IDs CAMG9.1-4, as can be seen in the updated validation matrix.

6.2 Validation

Due to the COVID-19 situation, physical engagement of end users in the form of internal playtesting has been interrupted. Therefore, we have instead shifted our focus to DLT and BLE beacon performance tests. There has been progress in validation results regarding the Scavenger Hunt and Blockmoji use cases.

To this date, we have validated six of our pilot requirements, as seen in Deliverable 5.3 [D5.3] Chapter 6.3. The remaining requirements (all except the ones relating to the decentralized identity use case) are successfully validated and described in Table 21. Proof of requirement fulfilment is given via screenshots in the table.

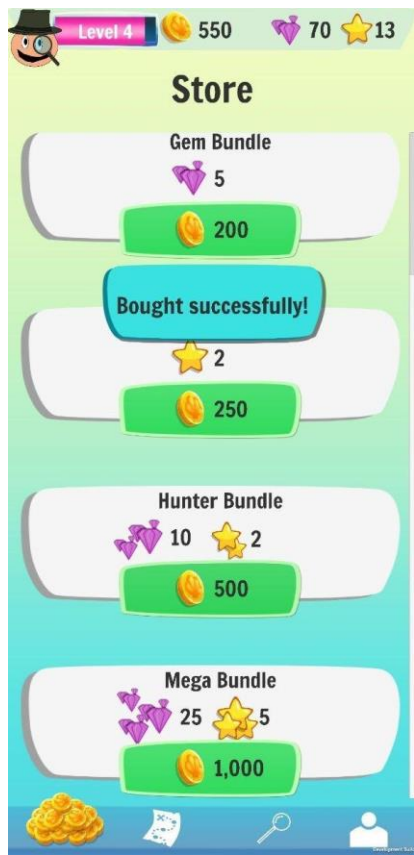
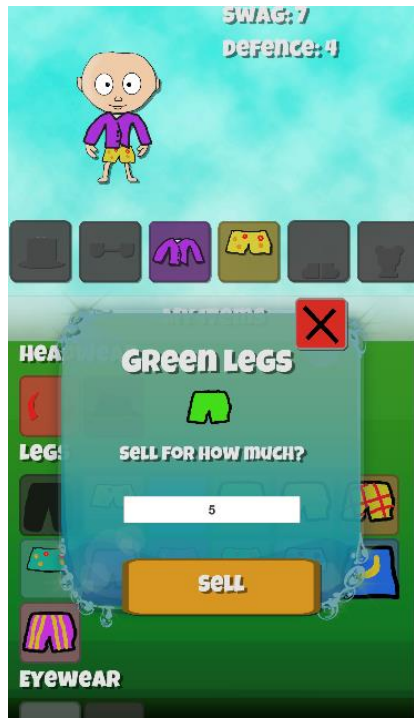


Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Table 21: Requirement validation - CAMG.

Requirement ID	Requirement Description	Test Description	Components Involved	Validation
REQ_CAMG0.1	Each person interacting with the game should have a unique identifier.	The test passes if all player IDs are different.	Game Web Server, Scavenger Hunt Application	<div>androidId ⓘ</div> <div>12f6ac22bee5e109ab66f55b7f2f820a</div> <div>85793297599356a8570b027fa2a59899</div> <div>RandomNewPlayer_8MOY5P02OYQ9VEB1T74K</div> <div>RandomNewPlayer_GJ5UL8JGM9D2V0F8P5J</div>
REQ_CAMG1.3	Each challenge should have a unique identifier	The test passes if IDs of all Scavenger Hunt challenges are different.	Game Web server	<div>huntID ⓘ</div> <div>0ddaf6ca-aa97-11ea-8ac9-ce6c8f318d7b</div> <div>540a035a-9c34-11ea-a186-f6dbc58721eb</div> <div>5ac05a28-9c34-11ea-a186-f6dbc58721eb</div> <div>75271596-9c34-11ea-afef-f671e4a98e74</div> <div>8bf0af6c-9c34-11ea-a186-f6dbc58721eb</div> <div>9f4f478-9c34-11ea-afef-f671e4a98e74</div> <div>9ff43fe2-83fc-11ea-8f1b-6aecb3a7b19</div> <div>a4f12104-9c34-11ea-afef-f671e4a98e74</div> <div>bb69bb30-9f4f-11ea-8e27-826137cc4fdc</div>
REQ_CAMG1.4	Time should be recorded for each player, starting after joining the challenge till the player completes it.	The requirement is met if, after a user plays the challenge, the completed challenge's start time is recorded	Game Web Server, Scavenger Hunt Application	<div>completed_hunts List [1]</div> <div> <div>0</div> <div>Map (5)</div> <div> <div>current_clue Number : 0</div> <div>end_time Null : true</div> <div>hasUsedStar Boolean : false</div> <div>huntID String : 8bf0af6c-9c34-11ea-a186-f6dbc58721eb</div> <div>start_time String : 2020-07-30 13:10:52.533619</div> </div> </div>

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

REQ_ CAMG1.7	Players can buy In-App tokens using in-game currency	The requirement is met if the player can spend in-game coins in the application to buy Gem and Star tokens - increasing Gem and Star amounts in possession and decreasing Coins in possession.	Game Web Server, Scavenger Hunt Application	
REQ_ CAMG4.1	Players can buy and sell Blockmoji assets on the blockchain	The requirement is met if players are able to buy and sell Blockmoji items on the blockchain.	Game Web Server, Scavenger Hunt Application, Interledger, Marketplace	

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

6.3 Evaluation

We evaluated how the new technologies, such as blockchains and IoT, perform in the mobile gaming ecosystems. The results of the evaluation are included in the Deliverable 5.3 [D5.3] Chapter 6.3.

6.3.1 Pilot performance assessment and KPIs evaluation

The pilot has successfully managed to achieve all KPIs that have been defined in D5.1 [D5.1]]. The evaluations performed in Deliverable 5.3 were based on the KPIs. The setup for the evaluation, tests performed, and results can be found in D5.3 Chapter 6.3. The system performance KPIs for the mobile gaming pilot are presented in Table 22.

Table 22: CAMG performance KPIs.

Name	Description	Metric	Method of measurement	Target	Result
Public ledger execution cost	Cost for executing operation on a public ledger	Ledger execution cost units (e.g., gas in Ethereum)	Measure the total execution cost for all operations that a transaction involves	As low as possible	0 (Permissioned Ledger)
Configuration time for a new challenge	Time for configuration to complete	Time units (e.g., seconds)	Measure time between start of configuration until completion of configuration	< 5 sec.	2.247 seconds (Transaction time)
Configuration time for a new advertisement or for In-App tokens	Time for configuration to complete	Time units (e.g., seconds)	Measure time between start of configuration until completion of configuration	< 15 sec.	N/A
Response time for getting points after completing a challenge	Time for the system to respond to the request or to execute a transaction	Time units (e.g., seconds)	Measure time between instant system receives a request or transaction until the instant that the system responds	< 4 sec.	2.247 seconds (Transaction time)
Response time for skipping a challenge or for getting In-App tokens or for redeeming rewards	Time for the system to respond to the request or to execute a transaction	Time units (e.g., seconds)	Measure time between instant system receives a request or transaction until the instant that the system responds	< 20 sec.	2.247 seconds (Transaction time)
Throughput	Maximum number of transactions per time unit that the system can support	Number of transactions per time unit	Measure transactions per time unit	As high as possible	307 TPS (Read) 128 TPS (Write)
Scalability - cost	Increase of cost as number of challenges or active users or ads increases	Ratio of delta cost over delta of challenges or active users or ads	Measure cost for different numbers of challenges or active users or ads	Linear or sublinear	Linear



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Scalability – time	Increase of response time as number of challenges or active users or ads increases	Ratio of delta time over delta of challenges or active users or ads	Measure response time for different numbers of challenges or active users or ads	Linear or sublinear	Linear
--------------------	--	---	--	---------------------	--------

6.3.2 TRL

The pilot including the mobile games that have been developed has been managed to achieve TRL-6. The TRL level that the pilot and the outcome assets that have derived from it reached, is summarized in Table 23.

Table 23: TRL levels of the CAMG pilot and assets

TRL	Justification
6	The pilot prototype was demonstrated in a relevant environment. It was hosted on the Amazon Web Servers and IoT beacons were deployed in the Rovio entertainment office. The Mobile application package was distributed through the internal platform installable on iOS and Android. An example hunt was created for the users, to give them a tour of the office. The user who played the hunt received the rewards as Non-Fungible Tokens (NFT) stored on the fabric blockchain and was also able to customize their characters using the Blockmoji application. A part of the deployment and operation have also been demonstrated through a video presented during the final review.
6	The Scavenger Hunt game is used by Context-Aware Mobile Gaming Pilot

6.3.3 Business Opportunities

Location based IoT gaming would likely involve many entities. A hypothesis is that such an ecosystem with different actors would gain from trust created through distributed ledger technologies. The potential actors and their potential gains are as follows:

Providers of IoT beacons: From navigation beacons to temperature and quality assurance meters, the world is filled with various IoT devices: both stationary and mobile. In 2018 there were 23.14 billion connected IoT devices worldwide. Holders of such devices have the opportunity to receive passive income as their beacons are used for location-based games. This process can be facilitated and automated with distributed ledger technologies and the Provisioning and Discovery component. In order to grow the domain of utilized IoT devices, mobile device users could scan for suitable beacons with the Discovery and Provisioning component. In the current implementation, this involves deliberate searching from the user. However, in practice, Discovery and Provisioning -like software could run as a background process, even within a location-based game. When an IoT device is detected and deemed to be suitable for the game, it would be configured and provisioned to the game infrastructure through a smart contract on the blockchain. This smart contract would ensure that the game developer pays micropayments for the device providers as their beacons are used.

Game Developers: Multiple game developers can utilize the same IoT beacons for different games. They can also partially share the same economy and in-game assets, such as currency and even virtual items. If they wish so, multiple game developers can make their games support inter-game non-fungible tokens. For instance, a reward item in one game could provide the player functional benefits in the Scavenger Hunt game, and a reward received from a hunt can be used as a cosmetic item in another game. The benefit of this approach, as opposed to traditional servers and databases, is that if any of these games were killed, players' virtual items would continue to exist and would potential even preserve some value thanks to decentralized interoperability. DLT allows for such shared items to exist on a blockchain outside of the games themselves. In the gaming pilot, the Blockmoji companion application was developed as a proof of concept for an inter-game item management application.



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Players: Player play the game(s) and earn rewards from them. The awarded items can be used as cosmetics in the games or as items that provide benefits in a game. Players can trade these rewards with each other on a marketplace of non-fungible tokens and earn money.

Advertisers: In a complex network of mobile game advertising, blockchain technologies could potentially help combat fraud by offering more transparency in attribution metrics. In addition, distributed identity technologies could facilitate the anonymity of users and let them have more control over their data. By resetting decentralized identifiers, users can revoke the access to their ad profile from AdTech companies. A framework like this would facilitate AdTech companies' compliance with regulations such as GDPR.

Points of Interest: Businesses such as restaurants, cafes, museums, and malls (any businesses with a physical location interface with customers) could benefit from the traffic to their locations generated through a location-based game such as the Scavenger Hunt game experiment. When a player visits a point of interest to complete a task, they simultaneously become a potential customer. For instance, a cafe can design a hunt whose last task is at their shop, or a museum can design a hunt inside their exhibition, potentially making the tour more entertaining.

6.4 Replication guidelines

The Scavenger Hunt game prototype and its Blockmoji companion app were open-sourced under the Apache License 2.0 for development and research purposes. The source files are included in the SOFIE project Github repository [<https://github.com/SOFIE-project>].

6.4.1 Replication guidelines

Prerequisites

- Unity 2019.3 or higher (2019.1 or higher for Blockmoji) with Android / iOS build module.
- BLE beacons that can use the Eddystone UID protocol.
- Android or iOS phone.
- Purchase your own iBeacon plugin from the Unity asset store. More detailed instructions below.

Setting up the Scavenger Hunt client

The main game's Unity project (the client) is the ScavengerHuntTemplate-UnityClient folder in the Open source github. After you open the project in Unity for the first time, you will encounter compilation errors. This is expected. Many scripts depend on an "iBeacon" named plugin that is not included in this repository. In order to make the game work, you must purchase the iBeacon plugin from the Unity Asset Store and place the iBeacon folder under the Assets folder. After that, the game should compile.

Select the IBeacon object in the "MainScene" scene. If it does not have the scripts "IBeaconReceiver" and "BluetoothState" attached to it or if Unity throws compilation errors about missing scripts, attach both of these scripts to that game object.

In addition, before the game is playable, the backend host address must be set. After setting up the backend as described in the next section, update the "_host_address" variable in the ServerHandler.cs file. Now the game can be built for a mobile phone and played if there are hunts on the backend (instructions for the backend below).

For testing purposes, getting nearby hunts also works in the Unity Editor on a machine without GPS capabilities. In that case, GPS latitude and longitude are hardcoded in function StartRefreshingHunts in the NearbyManager.cs script, which can be changed to your current coordinates.



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

When playing the game on mobile and completing hunts, your device ID will be used as the Player ID. If exiting the game and returning, your progress will be retained, and you will not be able to complete old hunts again. While ideal for a real-life scenario, you might want to be able to replay hunts for testing purposes. For this, `_newPlayerEveryTime` in the `Player.cs` script can be set to true and a new account will be created every time you enter the game. Optionally, you can manually delete player data entries from the backend database.

This template version of the game does not have any sound effects. Many buttons and UI elements have empty sound clip references that can be populated with custom sounds, and the game prototype itself can be extended to your liking.

Setting up the Blockmoji client

Blockmoji is a companion prototype application where the user can browse item rewards that they have acquired from Scavenger Hunt. If you create another game that uses the Blockmoji standard and backend, items received from there will also be visible in this application. In the Blockmoji client, the user can see the items' attributes and the items can be equipped and unequipped from their avatar. A game using the Blockmoji framework could, if it wanted to, interpret these attributes to provide in-game benefits.

Once the Blockmoji backend has been set up, populate the `_host_address` field in the `APIHandler.cs` script.

Setting up the backend

The Backend consists of the following components:

- RESTful API, running a Python flask application on AWS LAMBDA and database on AWS DynamoDB, for the core game related tasks.
- Fabric RESTful API, running as a Node.js application on AWS EC2 node, using the Hyperledger Fabric Client SDK to query and invoke chaincode on AWS Managed Blockchain.

This client does not communicate with the blockchain directly. It communicates to the AWS Lambda game server, which forwards the request to the blockchain when necessary.

The Blockmoji backend must also be set up, as Scavenger Hunt depends on it (it awards Blockmoji items to the player).

Setting up hunts and beacons

In Unity, in the Scavenger Hunt project, select the `IBeacon` object in the scene. In the inspector for the `IBeaconReceiver` script, create a new region with an arbitrary name (such as `"com.test.ibeacon"`), and specify that beacons are of type `"Eddystone UID"`. Next, define a namespace for the beacons in 20 hexadecimal digits, which can be, for example `"00000000000000000000"`.

We found the following detection parameters to work well: `timespan = 6`, `scan period = 3` and `between scan period = 0`.

Next, let's set up a hunt! POST the content provided in `exampleHunt.json` to `LAMBDA-API-URL/hunts`, with a header `"Content-Type"` set to `"application/json"`. In the hunt example, you can see that a clue points to a beacon with ID `17592186044416` as an integer, which translates to `100000000000` in hexadecimal. You can add new clues as you wish to the json that you post, as long as you increase `"task_num"` by 1 with each subsequent task and accompany each new task with a new entry in `"clues"` and `"hints"`.

Hunts may also have virtual item rewards. Before using this functionality, you should create the items on the Blockmoji backend. To do this, POST a json to `FABRIC-API-URL/item` in the scheme defined by `exampleItem.json` (again, with the header `"Content-Type"` set to `"application/json"`). After the item exists on the backend, you can POST hunts that offer the item



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

as a reward. Simply include the item's "UniqueID" string in the "assetRewards" list of the hunt. NOTE: For the game to work, at least one reward item (and a maximum of four) must be included in every hunt. If you accidentally POST a hunt without item rewards, you will have to delete that entry from the database.

To set up beacons, you can use a mobile app, such as "iBKS", to configure them. When configuring a beacon, make sure that they use the Eddystone UID protocol. In this protocol, each beacon has an UID which consists of 32 hexadecimal digits. The first 20 digits should be the namespace as defined in the Unity IBeaconReceiver script (as instructed above). The remaining 2 digits should be a unique identifier of the beacon. When creating hunts, you can, for simplicity, use small numbers as IDs, in order for them to be similar in appearance both as decimal integers and hexadecimal. For instance, if you configure a beacon's ID to be "000000000001", the integer "instanceID" in the hunt json should conveniently be 1.

6.4.2 Limitations

One limitation of using IoT beacons for positioning is that one has to trust that they will not be physically moved in the future. For instance, if a cafe uses a small detachable BLE beacon for hosting a POI location, the beacon could be stolen, and cheaters could earn rewards unfairly. Theoretically, combining BLE positioning with another positioning method, such as GPS, could improve the validity of positioning.

On the blockchain side, race conditions must be handled properly. If two players complete the same hunt at the same time, rewards should be moved from escrow to the players' account so that the global sum of coins remains the same as before.

Blockchain only works well where latency is not a factor, limiting the amount of available gameplay features. Almost all of the current Blockchain games do not have real gameplay and suffer from having a simple play mechanism and a short life cycle. Our proof of concept tries to solve this by adding a centralized server for most of the game operations and blockchain for added functionalities, but the theoretical maximum number of concurrent players still remains a significant limitation. Despite their potential, blockchains are having trouble effectively supporting a large number of users on the network. The technical debate to improve scalability has been hindered by the trade-off between the performance and security goals of the blockchain system.

For a game to be fun, technology must not stand in the way of player experience. Transaction latency, wallet creation and other possible quirks of DLTs are possible pain points for the players. Moreover, a network transaction fee in a Blockchain can become problematic and some games may require necessary transactions for which fees are simply unacceptable to the players. If making a game for the masses, the benefits of DLTs should outweigh their shortcomings, and the DLT aspect itself could even be invisible to the player. Popular game markets, such as Google Play and Apple App Store currently do not accept cryptocurrencies as a payment method. Games or applications that accept cryptocurrencies need to perform payments using third-party exchanges, increasing security risks and costs. For the player, if it is harder to participate in the game economy than in a traditional system (banks, simple virtual currencies), then the game would not necessarily appeal to the masses beyond DLT enthusiasts.



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

7. SMAUG

7.1 Overview

SMAUG has been developed as a reference implementation with the goal to prove the functional integration of all the SOFIE components into a single use case. The non-functional testing, such as performance and system characteristics, is part of pilot validation testing scope in accordance with the original project plan.

The source code of SMAUG is available on the SOFIE project GitHub page¹⁹. The possibility of deploying the system in an environment that would outlive the scope of the SOFIE project and that would be accessible to the general public is currently being discussed internally at LMF.

A detailed description of the system architecture and its relationship with SOFIE is provided in D3.4 [D3.4] and integrated with additional documentation in the software repositories.

7.2 Validation

SMAUG does not mandate the validation of any requirements specifically for the reference implementation. Nevertheless, one of the goals of SMAUG is to support requirement validation for the SOFIE framework components, specifically the functional requirement RF23, relative to the SOFIE Marketplace component, and the architectural requirement RA05, relative to the SOFIE framework architecture as a whole. The process of validating those requirements is described in D2.7 [D2.7]. SMAUG reached CI/CD level 5 (as described in D3.3 [D3.3]) using employing integration tests.

7.3 Evaluation

7.3.1 TRL

SOFIE's reference application has achieved TRL-3 which is summarized below:

TRL	Justification
3	SMAUG is a reference implementation. It has not been deployed on production environments, nor has it been extensively tested in lab environments.

7.4 Replication Guidelines

All replication instructions are provided in the relative GitHub repositories that is released as an open-source contribution under the Apache 2.0 licensing model.

¹⁹ <https://github.com/orgs/SOFIE-project>



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

8. Cross pilot scenarios and testing plan

Two types of cross-pilot use-cases are described (also mentioned in deliverable D5.3 [D5.3]): one that focuses on how data can technically be exchanged between pilot platforms and one that focuses on the business value from combining pilot platforms. The implementation focus is on the Energy Data exchange cross pilot case in order to showcase that SOFIE can enable and also support cross pilot interoperability, while the Reward exchange case focuses on the business value aspects that emerge from combining different pilot use cases.

8.1 Cross pilot data exchange

8.1.1 Overview

In this scenario, we federate the Decentralized Energy Data Exchange (DEDE) pilot with the Decentralized Energy Flexibility Marketplace (DEFM) pilot and enable secure data exchange between them. We use the Federation Adapter (FA) developed for the DEDE pilot to achieve this. Although the main goal of the DEDE pilot is to liberate energy data, the technical solution is not limited to this single domain. The exchanged data can be anything, and the solution is thus suitable for a cross-pilot scenario. The architecture of the DEDE pilot and its Federation Adapter is described in D5.3.

8.1.2 Technical Description

8.1.2.1 Architecture and implementation

This cross-pilot scenario required no changes to the existing platforms of the federated pilots. The only requirement for each pilot is to be able to describe the services that it offers in the OpenAPI 3.0 format. That is the only format currently supported by the FA. If the pilot does not already offer services that can be described in the OpenAPI 3.0 format, it is possible to develop a converter on top of the existing platform services. In the case of the DEFM pilot, it was not necessary. Although all the federated pilots could easily consume services offered by other pilots, we deploy a separate client dedicated to the purpose of testing and evaluating this federation approach. This way, the cross-pilot testing does not force the pilots to implement functionality that does not align with their business goals. The deployment diagram is shown in Figure 20.

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

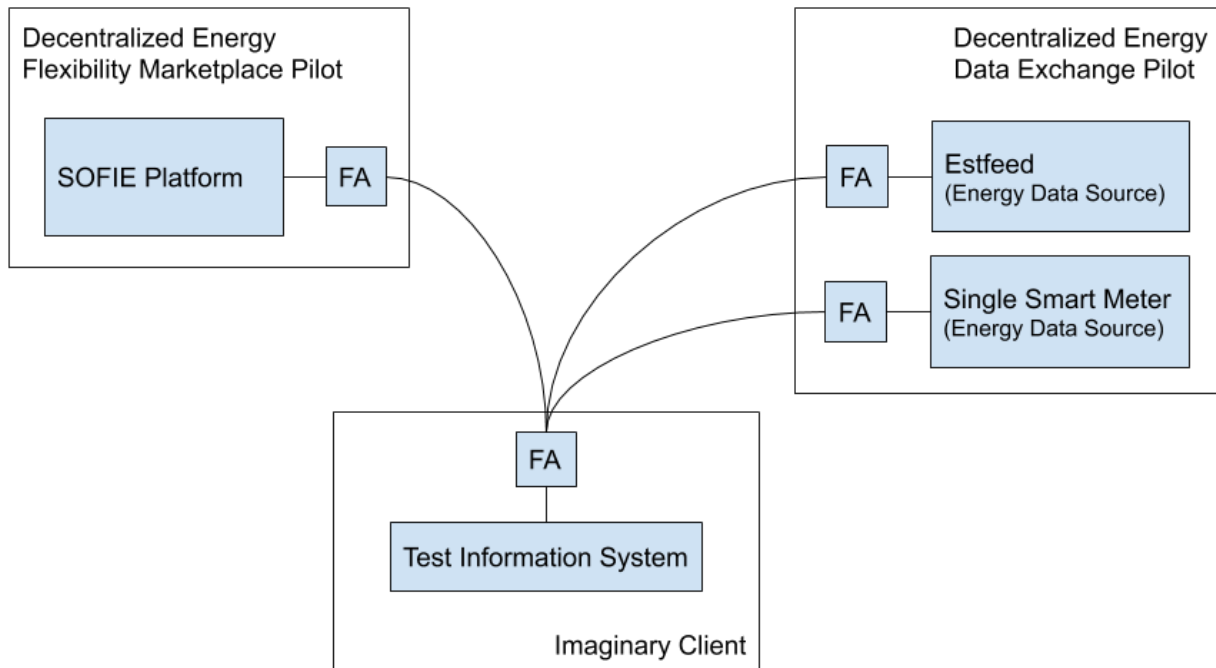


Figure 20: Deployment diagram of the cross-pilot scenario using the FA from the DEDE pilot.

The following services are offered by the DEFM pilot:

- *getLoadData* - returns the average load as measured by a metering point given the metering point ID

The following services are offered by the two data sources in the DEDE pilot:

- *getMeteringPoints* - returns a list of metering points this data source has data for
- *getConsumptionData* - returns electricity consumption data given a metering point ID

Although there is no service to list the available metering points in the DEFM pilot, we were provided with two IDs that the test information system can request data for. The main difference between the *getConsumptionData* in the DEDE pilot and the *getLoadData* in the DEFM pilot is that the former returns exact numbers about consumed electricity and can be used for billing purposes, while the latter returns the estimated numbers about the load and can be used to manage the network.

Both pilots give access to all of their services for the Test Information System. DEDE services are shown in Figure 21 while DEFM services are shown in Figure 22. In each figure the respective services are shown as they become available through the FAs.



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Federation Adapter *Estfeed*

Identifiers Credentials Services Permissions Discover Log

Services

DID	Name	Server URL	Proved Attributes
VcGwzAh3Dp1vceyaXEeJJJC	getConsumptionData	http://172.27.27.42:8888/	
VcGwzAh3Dp1vceyaXEeJJJC	getMeteringPoints	http://172.27.27.42:8888/	
BRJn9eMPrSmVQT4WGsNwYj	getMeteringPoints	http://172.27.27.42:7777	
BRJn9eMPrSmVQT4WGsNwYj	getConsumption	http://172.27.27.42:7777	

VcGwzAh3Dp1vceyaXEeJJJC

OpenAPI service description URL

Import services

Figure 21: List of the DEDE pilot services as seen in the configuration UI of the FA. Provided by the single smart meter (VcG...) and Estfeed (BRJ...).

Federation Adapter

Identifiers Credentials Services Permissions Discover Log

Discover

Discover services of target DID Discover

```
{
  "openapi": "3.0.1",
  "info": {
    "description": "Provides access to metering points and their energy consumption data.",
    "title": "SOFIE Energy Metering Data Source",
    "version": "1.0.0"
  },
  "servers": [],
  "paths": {
    "/STH/v1/contextEntities/type/SMX/id/urn:ngsi-Id:SMX:{id}/attributes/P": {
      "get": {
        "tags": [
          "Get data of meter"
        ],
        "description": "It returns an array of data from initial date to final date.",
        "operationId": "getDataFromSTH",

```

Figure 22: Test client discovering the services of the DEFM pilot (ABf...).

8.1.3 Validation

Latency overhead and throughput of the FA

We measured the latency overhead added by both the service consumer FA and the service provider FA of the request-response cycle. This metric is constant as the network grows. There are several potentially time-consuming steps that the FA on both sides needs to go through before it can pass a request or response forward. For example, looking up the public key for the service provider DID, looking up the endpoint of the service provider FA, looking up the hash value of the currently valid TLS certificate, sending out proof requests to get the values for proved attributes, or signing and verifying the signatures of all the messages. However, most of



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

these steps can be optimized with a cache. Only signing and signature verification remain part of each request-response cycle. Service consumer does one signing operation for the request and one signature verification for the response. Likewise, the service provider does one signature verification for the request and one signing operation for the response. Neither of these require interaction with the ledger. The latency measurements reflect this scenario, where everything that can be cached is cached, and only the steps that must be performed every time contribute to the measured latency.

Table 24: Measured latency overhead of the FA. Mean value over 400 requests.

Action	Latency overhead
Constructing a signed SOFIE request in the service consumer FA	33ms
Verifying the SOFIE request in the service provider FA	11ms
Constructing a signed SOFIE response in the service provider FA	34ms
Verifying the SOFIE response in the service consumer FA	9ms
Total	87ms

As expected, the latency overhead (shown in Table 24) is similar on both sides (service provider and service consumer). The total value of 87ms is rather an upper limit and can probably be optimized further. The main result is that the messages are exchanged without interacting with the ledger and the costs it would incur.

Signing operation is also the main limiting factor for the throughput of the FA. On an 8-core Intel i7-8550U the FA on either side was able to process about 300 req/s.

Integration Effort and Comparison to Current Situation

The following steps were taken to federate the DEDE pilot with the DEFM pilot:

1. Downloading the FA Docker image and running it as a Docker container
2. Checking that the services of the DEDE pilot are reachable given their DIDs
3. Choosing the services to expose and constructing the service description in the OpenAPI 3.0 format
4. Making sure that the FA can reach the service implementation. Some network configuration was necessary as the FA was started in an isolated network by default.
5. Publishing the external IP address of the FA and making necessary firewall rules so that other FAs on the internet can reach it. This step took a couple of days due to internal processes of the organization.
6. Switching the service implementation. It is important to expose the right layer of the IoT platform as the service. A back-end service designed to serve a front end is not a good fit, as it usually has its own authentication and authorization solution which complicates the use of the FA. The FA is designed to offer a common authentication and authorization solution for all federated platforms. Having a custom solution beneath it can considerably hamper the federation process.
7. Adding support for HTTP header parameters. Proxying header parameters was not initially implemented in the FA but was added once it was needed.

It took about 1-2 weeks of intermittent effort from both sides to get from the initial instructions to a new service provided by the DEFM pilot.

From the client perspective, consuming the services of either pilot was identical and required the following steps:

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

1. Downloading the FA Docker image and running it as a Docker container
2. Discovering the services of both pilots given their DIDs
3. Requesting access rights from both pilots to use their services
4. Constructing requests based on the service descriptions

With the uniform access to any of the federated IoT platforms, the benefits become clear. The initial burden lies on the IoT platform to expose its services, but it will immediately add value to all of the existing participants on the network. For the client, it is the matter of acquiring access rights and understanding the service description to start using a new service. Both steps are necessary in any system integration and cannot be avoided. Still, we have secured the communication between the service providers and the client without relying on centralized components or the client having to do custom work to set up secure communication channels to each of them.

8.2 Cross pilot reward exchange

8.2.1 Overview

The cross-pilot reward exchange focuses on the added value for the end users derived from the ability to spend their token in different contexts.

Figure 23, below, represents the actors involved and the main technologies considered. The three pilots combined provide a unique environment so that the same actor participates to different pilots at once but, in the scenario represented, is still possible to identify the different roles from the DEFM pilot, the FSC pilot, and the CAMG pilot.

In more detail, we can observe the DSO and Fleet Managers operating in the decentralized flexibility marketplace, the store owner and transporters considered in the food supply chain pilot, and the end users, which can be either users of the mobile game, store customers, or EV users.

The roles overlap even more considering that, for instance, a DSO operator in turn can be one of the mobile game users or one of the transporters can participate to the flexibility marketplace if the transport fleet includes some electric vehicles.

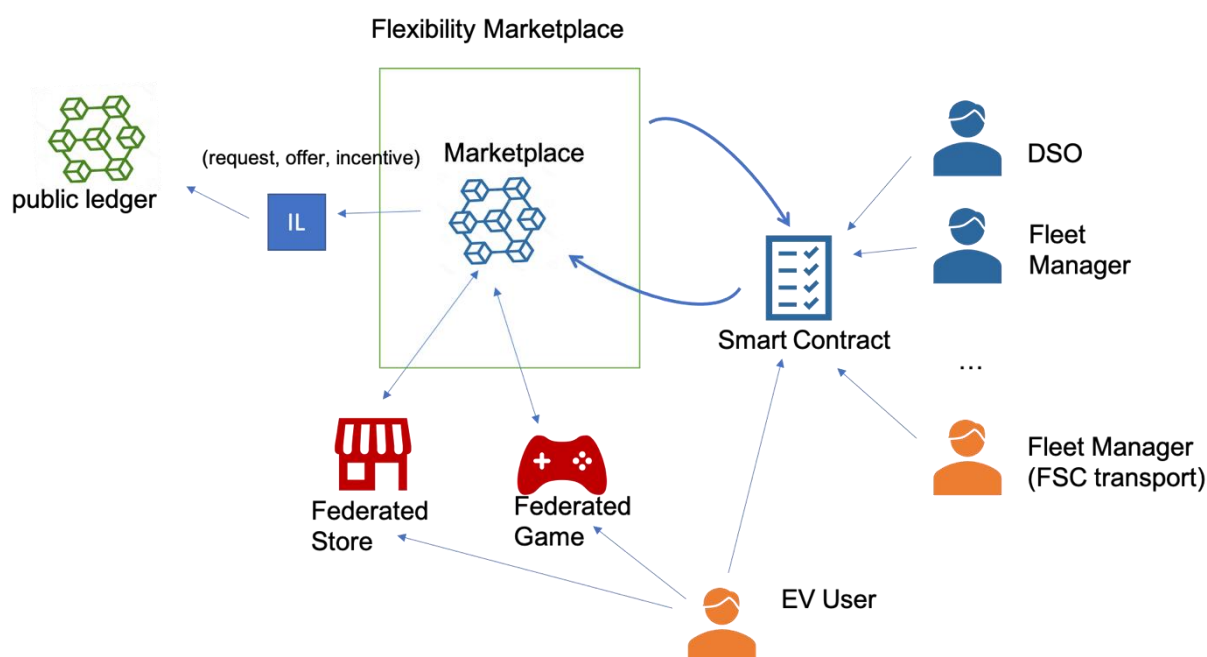


Figure 23: Cross pilot reward exchange scenario.



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

The principle behind this cross-pilot scenario is a direct network effect, in which an increase in usage leads to an increase in value for other users in a similar way to what happens for social networks, online games, or telephone systems.

8.2.2 Technical Description

To make it possible to share the same kind of reward obtained in one of the three different pilots by one of the actors, it is necessary to agree on the definition of this reward and how to handle it across the different platforms. Since each of the platform considered is linked to the Ethereum blockchain, it is possible to natively support *tokens* on each of the pilot platforms.

Unlike Ethereum's native cryptocurrency (ETH), tokens are not held directly by accounts. The tokens exist within a contract, structured as an independent database. The contract specifies the rules for tokens and maintains a list of users' balances. To move tokens, users send a transaction to the contract asking to allocate part of their balance somewhere else.

Ethereum offers a relatively simple format to define tokens, the ERC-20 standard. Following the ERC-20 guidelines developers can define tokens that are automatically interoperable with existing services and software like software and hardware wallet and exchanges.

Ethereum support the definition of interfaces that can be implemented by the different smart contracts. The ERC-20 software interface is shown below in Table 25.

Table 25: ERC-20 software interface.

```
pragma solidity ^0.4.23;
pragma experimental ABIEncoderV2;

interface ERC20Interface {
    function totalSupply() external view returns (uint);
    function balanceOf(address tokenOwner) external view returns (uint
balance);
    function allowance(address tokenOwner, address spender) external view
returns (uint remaining);
    function transfer(address to, uint tokens) external returns (bool
success);
    function approve(address spender, uint tokens) external returns (bool
success);
    function transferFrom(address from, address to, uint tokens) external
returns (bool success);

    event Transfer(address indexed from, address indexed to, uint tokens);
    event Approval(address indexed tokenOwner, address indexed spender,
uint tokens);
}
```

In this way, to enable a cross-pilot reward exchange, it is sufficient to implement to define a unique token smart contract based on ERC-20, that will be used in the different separate pilots.

8.2.2.1 Architecture

To make it possible to circulate the same kind of token across the different pilots, it is necessary that the smart contracts used inside each pilot extend the same generic “token” object. The generic token, as already mentioned, will extend in turn the ERC-20 interface. Figure 24 represents this relation.

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

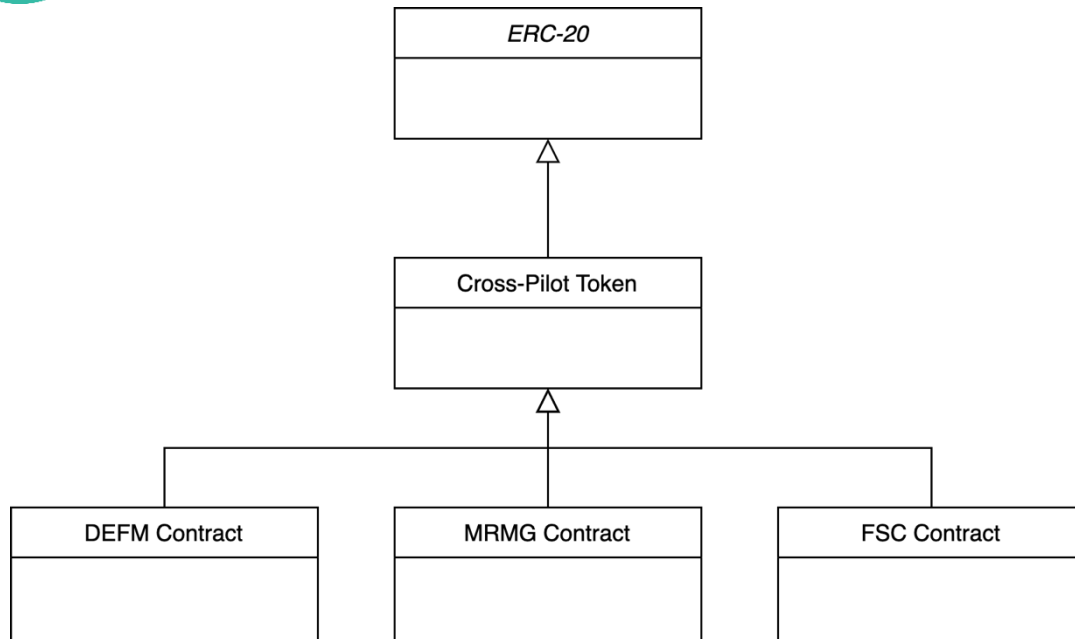


Figure 24: Class Diagram.

Each time one of the actors wants to transfer some of the tokens in exchange for goods or services provided by the other actors involved, the smart contract will manage the request invoking the transfer method exposed by the ERC-20 interface. Once received the user's updated balance and after verifying that the user has enough tokens, the smart contract will finalise the operation. The list of the operations is described in the sequence diagram in Figure 25.

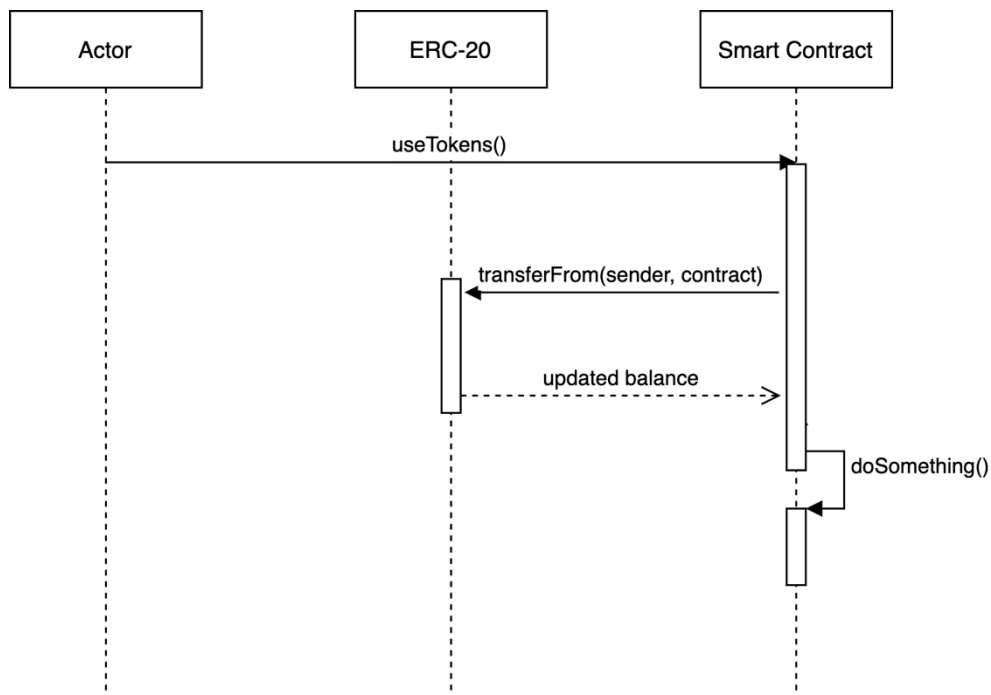


Figure 25: Sequence Diagram.



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

The architecture described above considers a single ledger. In such a case, an actor's balance is consistent across different pilots by having the smart contracts on each pilot update the same cross-pilot token balance (based on ERC-20 tokens), thus avoiding any double spending.

If the pilots use a different ledger, the token balance consistency across different pilots, hence ledgers, requires additional mechanisms. A key contribution of the SOFIE project is the implementation and evaluation of the interledger functionality to support transaction atomicity across two or more ledgers. Transaction atomicity refers to the property that either all the transactions (belonging to the set of transactions that should be executed in an atomic manner) are executed on all ledgers or none of the transactions are executed. This functionality is achieved through the use of *Hashed Time-Lock Contracts* (HTLCs) that cryptographically link transactions and events on two or more ledgers.

The Interledger component's sequence of actions to support the aforementioned transaction atomicity property is presented in the asset transfer scenario considered in deliverable D4.5, "Final Architecture, System, and Pilots Evaluation Report", where assets are transferred between a game asset ledger and a marketplace ledger. In the context of the cross-pilot reward exchange scenario, the two ledgers can be assumed to belong to different pilots, namely the DEFM and CAMG pilots. We emphasize the HTLC mechanism implemented in the interledger component is applicable to more than two ledgers. The only requirement is that all the ledger must support the same hash function.

8.2.2.2 Implementation

The cross-pilot reward exchange scenario was not intended for implementation within the SOFIE project, but more as an example on how to connect different business platforms to a SOFIE powered architecture from a business point of view.

Since each different pilot utilizes its own smart contract, we assumed that each operation on the platform may be associated with a certain value in tokens. Some of the pilots, like the DEFM pilot, already need tokens to operate while the remaining ones already operate totally or in part over Ethereum smart contracts, meaning that tokens are easily integrated.

Although tokens and cryptocurrencies are not strictly equivalent, we can consider them, for the sake of simplicity, comparable in this case. The value for a cryptocurrency network is related to the network of people who use it. The value of a network is determined by a law, accredited to Bob Metcalfe, which states that a network's value is proportional to the square of the number of its users ($V=N^2$). The generalized Metcalfe's law was tweaked to fit the analysis of cryptocurrencies²⁰ by using an exponent of 1.69 considering that each user is linked, on average, to $N^{2/3}$ other users and resulted effective in predicting the value of Bitcoin.

The assumption behind the cross-pilot reward exchange scenario is that a similar law can be applied to the network of users and providers linked to the scenario. Based on this assumption, the scenario aims to pool together the different actors involved in the pilot platforms instead of keeping several different non-connected clusters of users, benefiting from a higher network value.

The implementation of asset transfer between two ledgers, which can be assumed to belong to different pilots, is documented in deliverable D4.5, "Final Architecture, System, and Pilots Evaluation Report". The results in D4.5 also assess the additional cost and time overhead incurred from using the interledger components compared to manually conducting the transactions, showing that it is less than 6% and 1%, respectively. Furthermore, the validation of the atomicity of interledger components is demonstrated in deliverable D4.3, "First Architecture and System Evaluation Report", which also assesses the gains from using the

²⁰ Spencer Wheatley, et al. "Are Bitcoin Bubbles Predictable? Combining a Generalized Metcalfe's Law and the LPPLS Model." (2018).



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

interledger component to interconnect public and private ledgers in terms of the significant reduction of the transaction cost and the transaction delay. Finally, the end-to-end delay of transactions on a public ledger (Ethereum testnet) and a permissioned ledger (Hyperledger Fabric) are assessed in deliverable D4.4, “Second Architecture and System Evaluation Report”. These results show that the added value of cross-pilot reward exchange scenario can be achieved with different tradeoffs between transparency and trust, which is supported in a wide-scale and decentralized manner using public ledgers, and performance gains in terms of transaction cost and transaction delay, which can be significantly reduced with private and permissioned ledgers.



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

9. Conclusions

WP5 aimed at setting up the four pilots of the SOFIE project and validating its federation architecture in real operating conditions. This deliverable presented the final versions of the pilot platforms which include the latest versions of the SOFIE framework components used in the pilots. All pilots are using components from the last SOFIE Framework release. In particular, the Interledger component is used in all pilot platforms, confirming its cross-domain applicability and value as an integral part in pilots' operation: federating different ledgers. DLT-federation is in the core of the pilots and enabled via the IL component of the SOFIE framework.

The validation results of the final pilot platform versions which were deployed on-site are also reported with accompanying pictures from the results generated during the validation steps are included.

Following the validation, the evaluation of the four platforms was presented, based on the KPI tables defined in a previous deliverable. All KPIs specified were achieved, the ones related to the underlying technology but also those that are more business related. Also, three pilots reached TRL-7 as they were demonstrated in an operational environment and one pilot reached TRL-6.

In order to support external parties that would be willing to join the pilot platforms, this document included replication guidelines for others to make their own SOFIE-compliant modules for federating their platforms. In all cases that was applicable, links to open-source repositories were provided and specific examples of how-to replicate were described.

An update on the status of SOFIE's reference application, SMAUG, was provided, as well as updates on the cross-pilot cases that illustrate the capabilities of SOFIE to provide interoperability across different domains which can, in turn, offer additional business value. In the cross-pilot context, two cases were presented, one that focused on the implementation technical aspects, and a second one which was more focused on how, through SOFIE components, and the Interledger in particular, value could be generated in a cross-domain manner.

Results of the validation of SOFIE pilots have also been included in Appendix II of this deliverable.



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

10. References

- [D4.5] V. Siris et al. “SOFIE Deliverable 4.5 – Final Architecture, System, and Pilots Evaluation Report”, December 2020.
- [D2.6] Y. Kortensniemi et al. “SOFIE Deliverable 2.6 - Federation Architecture, final version”, October 2020.
- [D2.7] Y. Kortensniemi et al. “SOFIE Deliverable 2.7 - Federation Framework, final version”, December 2020.
- [D3.3] Mikael Jaatinen et al. “Business Platform, Pilot Release”, September 2019.
- [D3.4] Mikael Jaatinen et al. “Business Platform, Final Release”, December 2020.
- [D5.1] I. Oikonomidis et al. “SOFIE Deliverable 5.1 – Baseline Systems and Measurements”, June 2018. Resubmitted December 2019.
- [D5.2] I. Oikonomidis et al. “SOFIE Deliverable 5.2 – Initial Platform Validation”, July 2019.
- [D5.3] I. Oikonomidis et al. “SOFIE Deliverable 5.3 - End-to-end Platform Validation”, July 2020.
- [FAs] SOFIE Federation Adapters, open-source implementations of the FAs used in the SOFIE pilots, available at: <https://github.com/SOFIE-project/Federation-Adapters>



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

11. Appendix I: Food Supply Chain user questionnaire

		1 to 5: Definitely NOT – Definitely YES					
#	Topic	N/A	1	2	3	4	5
Q1	Do you find the web application useful?						
Q2	Does the web application suit your needs?						
Q3	Do you think that the web application may be used by others?						
Q4	Do you think the product history information included in the QR code is useful for you?						
Q5	Would you say that the product history information helped towards not-optimal product recall reduction by half?						
Q6	Does the product history information included in the QR code help you toward deciding on purchasing the product?						
Q7	Do you think more product history information would be useful to be included in the QR code?						
Q8	Do you think the product history information available in the QR code will attract more customers to the product?						
Q9	Do you find the auditing capability useful?						
Q10	Would you use the auditing capability in case of a not-optimal product?						
Q11	In case you found a not-optimal product, did you find the explanation provided via the audit service sufficient for identifying the reason behind it?						
Q12	Do you find the web application easy to use?						
Q13	Do you find the QR code information easy to access?						
Q14	Do you find the information presented (either in the web application pages or in the QR code) easy to understand?						
Q15	Do you think the information presented (either in the web application pages or in the QR code) is well presented?						
Q16	Do you think the web application is fast enough?						



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

Q17	Did you encounter any delays while using the web application?						
Q18	Did you encounter any other issues when using the web application (e.g., pages not loading, disconnections, etc)?						
Q19	Do you find the concept (product history information and auditing capabilities) innovative?						
Q20	Would you like to see the same concept applied to more products of the Food sector?						
Q21	Would you have more trust in a product if it includes history information?						



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

12. Appendix II: Pilot Validation Matrix

12.1 Food Supply Chain

<i>Food Supply Chain</i>			
<i>ID</i>	<i>Validation Process</i>		<i>Result</i>
REQ_FSC0.1	<i>Requirement Description</i>	The services must be provided (to the actors) through the same web application.	OK
	<i>Test approach</i>	Field test	
	<i>Test Description</i>	Each registered actor of any type (e.g. producer, transporter, warehouse, supermarket employee) can access and perform all the services provided by the FSC web application based on its role.	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC02, as defined in D5.1. The results for FSC_TC02 are found on D5.4 chapter 3.2.1	
REQ_FSC0.2	<i>Requirement Description</i>	The services must be accessible (by the actors) under a Role-based Access Control (RBAC) policy.	OK
	<i>Test approach</i>	Field test	
	<i>Test Description</i>	Each registered actor of any type (e.g. producer, transporter, warehouse, supermarket employee) can access and perform all the services provided by the FSC web application based on its role. The actors have already registered on the pilot platform. Roles for the actors are granted by the Keycloak server (which is a component of the platform) during their registration.	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC02, as defined in D5.1. The results for FSC_TC02 are found on D5.4 chapter 3.2.1	
REQ_FSC0.3	<i>Requirement Description</i>	Each actor must be identified in a unique way	Private test
	<i>Test approach</i>	Unit test	
	<i>Test Description</i>	Authorization server is configured so each registered actor is bound to a unique ID. One of the attributes in an actor's profile (e.g. name) is used as a key to avoid duplicate IDs for the same actor.	
	<i>Test location</i>	The unit test is embedded in the code.	
REQ_FSC0.4	<i>Requirement Description</i>	Each federated IoT environment must have a unique identifier in the system architecture.	Private test
	<i>Test approach</i>	Unit test	
	<i>Test Description</i>	Each federated IoT platform is bound to a unique ID. Federated platforms register themselves by using an Ethereum client such as Geth to create accounts by using built-in encryption policies. The smart contract executed in the private ledger (consortium ledger) implements a number of checks to verify the identities of the IoT platforms requesting transactions. The test will compare datasets which are sent by each federated platform for a specific period to the records in the consortium ledger to decide whether each IoT platform is bound to a unique ID or not.	
	<i>Test location</i>	The unit test is embedded in the code.	



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

REQ_FSC0.5	<i>Requirement Description</i>	Authentication and access control logic must be applied to common storage resources.	OK
	<i>Test approach</i>	Documentation	
	<i>Test Description</i>	An authorization and access management server had been integrated to enable actors' registration in the supervisor data management layer and establish role-based accessibility to the provided services (D5.2, Chapter 3.3.2, page 38). The test will verify that each registered actor can make transactions to the private ledger based on its role (and the defined use cases).	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC06, as defined in D5.1 The results for FSC_TC06 are found on D5.4 chapter 3.2.1	
REQ_FSC1.1	<i>Requirement Description</i>	Registration of a crop must be timestamped.	OK
	<i>Test approach</i>	Integration test	
	<i>Test Description</i>	Data and metadata provided by the actors through the FSC web application are recorded in DLTs. The payload of any transaction is verified.	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC06, as defined in D5.1. The results for FSC_TC06 are found on D5.4 chapter 3.2.1	
REQ_FSC2.1	<i>Requirement Description</i>	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)	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	Readability of all included information in QR codes is confirmed.	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC08, as defined in D5.1. The results for FSC_TC08 are found on D5.4 chapter 3.2.1	
REQ_FSC3.1	<i>Requirement Description</i>	Handovers must be recorded in an immutable way where all federated IoT environments must have access.	OK
	<i>Test approach</i>	Integration test	
	<i>Test Description</i>	Data and metadata provided by the actors through the FSC web application are recorded in DLTs. The payload of any transaction is verified.	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC06, as defined in D5.1. The results for FSC_TC06 are found on D5.4 chapter 3.2.1	
REQ_FSC3.2	<i>Requirement Description</i>	The boxes could be sealed upon the delivery to the transportation company (from the producers).	N/A
	<i>Test approach</i>	N/A	
	<i>Test Description</i>	N/A, Open-top boxes were used during on-site deployment	
	<i>Test location</i>	N/A	

Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

REQ_FSC4.1	<i>Requirement Description</i>	Upon delivery to the WH employee, boxes could be unsealed by the TR employee.	N/A
	<i>Test approach</i>	N/A	
	<i>Test Description</i>	N/A, Open-top boxes were used during on-site deployment	
	<i>Test location</i>	N/A	
REQ_FSC5.1	<i>Requirement Description</i>	Each box must have a unique RFID tag identifier.	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	Test that box reuse is possible (after its release) and that registration of a box with an ID that is already used by another box is impossible (box unique identifier).	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC05, as defined in D5.1. The results for FSC_TC05 are found on D5.4 chapter 3.2.1	
REQ_FSC5.2	<i>Requirement Description</i>	Boxes must be considered as things of the transportation IoT platform.	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	An RFID tag is attached to each box. The test will verify that the RFID reader detects all tags which are placed within its range at any moment.	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC06, as defined in D5.1. The results for FSC_TC06 are found on D5.4 chapter 3.2.1	
REQ_FSC5.3	<i>Requirement Description</i>	Box registration in the supply chain must also define the producer from whom it will be used.	OK
	<i>Test approach</i>	Integration test	
	<i>Test Description</i>	The payload of the transaction which corresponds to the specific use case (FSC_UC5 “register session”) is verified to also include the ID of the farmer who will use the boxes.	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC06, as defined in D5.1. The results for FSC_TC06 are found on D5.4 chapter 3.2.1	
REQ_FSC5.4	<i>Requirement Description</i>	Registration of a box must be timestamped.	OK
	<i>Test approach</i>	Integration test	
	<i>Test Description</i>	The payload of the transaction which corresponds to the specific use case (FSC_UC5 “register session”) is verified to also include a timestamp.	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC06, as defined in D5.1. The results for FSC_TC06 are found on D5.4 chapter 3.2.1	
REQ_FSC6.1	<i>Requirement Description</i>	Transportation trucks must have internet connection to communicate and exchange data with the transportation IoT platform.	OK
	<i>Test approach</i>	Field test	



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

	<i>Test Description</i>	The SOFIE platform receives data from the transportation GW deployed in the truck i) as the vehicle moves, and ii) as the vehicle engine is turned off.	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC05, as defined in D5.1. The results for FSC_TC05 are found on D5.4 chapter 3.2.1	
REQ_FSC6.2	<i>Requirement Description</i>	A TR employee (driver) must be able to use different transportation trucks on different occasions.	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	By using the FSC web application, the TR employees can select any of the available trucks to transport boxes between two sites.	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC05, as defined in D5.1. The results for FSC_TC05 are found on D5.4 chapter 3.2.1	
REQ_FSC7.1	<i>Requirement Description</i>	Measurements from IoT devices are stored locally in the corresponding IoT platform.	OK
	<i>Test approach</i>	Integration test	
	<i>Test Description</i>	Measurements from each deployed sensing device are collected by the corresponding IoT platform and they are properly stored in its database system.	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC01, as defined in D5.1. The results for FSC_TC01 are found on D5.4 chapter 3.2.1	
REQ_FSC8.1	<i>Requirement Description</i>	Upon delivery to the SM employee, boxes could be unsealed by the TR employee.	N/A
	<i>Test approach</i>	N/A	
	<i>Test Description</i>	N/A, Open-top boxes were used during on-site deployment	
	<i>Test location</i>	N/A	
REQ_FSC9.1	<i>Requirement Description</i>	The temperature within each storage room of the WH must be continually monitored.	OK
	<i>Test approach</i>	Integration test	
	<i>Test Description</i>	The test will verify that temperature measurements from each deployed sensing device are collected by the corresponding IoT platform	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC01, as defined in D5.1. The results for FSC_TC01 are found on D5.4 chapter 3.2.1	
REQ_FSC9.2	<i>Requirement Description</i>	In the WH, a notification appears in the monitoring service of the Aberon IoT platform each time a predefined temperature range is violated.	OK
	<i>Test approach</i>	Integration test	
	<i>Test Description</i>	The test will create datasets that include some values out of the predefined temperature domain and verify that corresponding notifications are created.	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC01, as defined in D5.1.	



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

		The results for FSC_TC01 are found on D5.4 chapter 3.2.1	
REQ_FSC10.1	<i>Requirement Description</i>	The (unreleased) boxes in the WH must contain either raw or packetized products.	N/A
	<i>Test approach</i>	Documentation	
	<i>Test Description</i>	The req. has been merged into the workflow (action in the physical space) that accompanies the use of services. Not a technical requirement, no test is applied	
	<i>Test location</i>	N/A	
REQ_FSC11.1	<i>Requirement Description</i>	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	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	Creation of QR codes by using the FSC web application.	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC08, as defined in D5.1. The results for FSC_TC08 are found on D5.4 chapter 3.2.1	
REQ_FSC11.2	<i>Requirement Description</i>	The same QR label must be attached to every packet containing grapes which were transferred into the same box.	N/A
	<i>Test approach</i>	Documentation	
	<i>Test Description</i>	The req. has been merged into the workflow (action in the physical space) that accompanies the use of services. Not a technical requirement, no test is applied	
	<i>Test location</i>	N/A	
REQ_FSC11.3	<i>Requirement Description</i>	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.	N/A
	<i>Test approach</i>	Documentation	
	<i>Test Description</i>	Contents of QR codes are verified that they are well accepted by customers.	
	<i>Test location</i>	The req. has been merged into the workflow (action in the physical space) that accompanies the use of services. Not a technical requirement, user questionnaires are used as described in D5.4.	
REQ_FSC11.4	<i>Requirement Description</i>	The QR codes must be self-contained, so internet connection is not needed to read their content.	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	A number of QR codes are scanned and it is verified that they include product information from all segments of the supply chain	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC08, as defined in D5.1. The results for FSC_TC08 are found on D5.4 chapter 3.2.1	
REQ_FSC11.5	<i>Requirement Description</i>	The QR codes must contain product information that relate to all the segments of the chain.	OK
	<i>Test approach</i>	Functional test	



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

	<i>Test Description</i>	A number of QR codes are scanned and it is verified that they include product information from all segments of the supply chain	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC08, as defined in D5.1. The results for FSC_TC08 are found on D5.4 chapter 3.2.1	
REQ_FSC12.1	<i>Requirement Description</i>	Boxes must be able to be re-used in the future (to carry other products) after they have been released of the current transfer.	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	Test that box reuse is possible (after its release) and that registration of a box with an ID that is already used by another box is impossible (box unique identifier).	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC03, as defined in D5.1. The results for FSC_TC03 are found on D5.4 chapter 3.2.1	
REQ_FSC13.1	<i>Requirement Description</i>	QR labels must be accessible by everyone by using a smartphone device.	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	A QR code which is created by the supermarket employee using the FSC web application can be read offline by using different smartphone devices. Readability of all included information is confirmed.	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC08, as defined in D5.1. The results for FSC_TC08 are found on D5.4 chapter 3.2.1	
REQ_FSC14.1	<i>Requirement Description</i>	In the case of an audit, requested organizations must be able to provide proof of their claims about the historic data of assets which are stored locally.	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	The tests verifies that all measurements of interest can be retrieved by the API of the corresponding IoT platform	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC09, as defined in D5.1. The results for FSC_TC09 are found on D5.4 chapter 3.2.1	
REQ_FSC14.2	<i>Requirement Description</i>	Transfer of responsibility over boxes (assets) must be timestamped.	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	The payload of transactions is verified to include correct timestamps.	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC06, as defined in D5.1. The results for FSC_TC06 are found on D5.4 chapter 3.2.1	
REQ_FSC14.3	<i>Requirement Description</i>	A transaction must be confirmed by both transacting parties.	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	The test will verify that both transacting parties have to confirm the transaction (i.e., box handover) by using the FSC web application.	



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

	<i>Test location</i>	This requirement is tested as part of the FSC_TC06, as defined in D5.1. The results for FSC_TC06 are found on D5.4 chapter 3.2.1	
REQ_FSC14.4	<i>Requirement Description</i>	Both parties of a transaction must be able to access the details of the transaction at any time.	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	The test will verify that metadata related to an actor's transaction is accessible by that actor at any time and is invisible to any other actor.	
	<i>Test location</i>	This requirement is tested as part of the FSC_TC07, as defined in D5.1. The results for FSC_TC07 are found on D5.4 chapter 3.2.1	

12.2 Decentralized Energy Data Exchange Pilot

Decentralized Energy Data Exchange Pilot			
ID	Validation Process		Result
REQ_DEDE1.1	Requirement Description	Data owner can access info about his data, full visibility of data use.	Test not found
	Test approach	Functional test	
	Test Description	The tests verify that existing data owners can access the system (through supported authentication methods) and can see its metering points and audit log for data use by other services.	
	Test location	-	
REQ_DEDE1.2	Requirement Description	Each actor must be identified.	OK
	Test approach	Functional test	
	Test Description	The test verifies that participants in data exchange are uniquely identified by their DID-s.	
	Test location	This requirement is tested as part of the EDE_TC02, as defined in D5.1 Results are found in D5.4 chapter 4.2.1	
REQ_DEDE2.1	Requirement Description	Owner must be able to decide who gets access to his/her data.	OK
	Test approach	Functional test	
	Test Description	The test verifies that the data owner can add and revoke credentials to its data.	
	Test location	This requirement is tested as part of the EDE_TC02, as defined in D5.1 Results are found in D5.4 chapter 4.2.1	
REQ_DEDE2.2	Requirement Description	All user info must be GDPR compliant.	OK
	Test approach	Documentation	



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

	Test Description	Data handling in the system will be GDPR compliant	
	Test location	D5.2, Chapter 4.2.2.1, pag. 55	
REQ_DEDE2.3	Requirement Description	Data handover must be registered and proved at every transaction.	OK
	Test approach	Functional test	
	Test Description	Every interaction in the system will have an audit log that can be verified later stages.	
	Test location	This requirement is tested as part of the EDE_TC03, as defined in D5.1 Results are found in D5.4 chapter 4.2.1	
REQ_DEDE2.4	Requirement Description	Service providers must be able to define the energy consumption data parameters.	OK
	Test approach	Integration test	
	Test Description	The test verifies that it's possible to define service parameters that affect the dataset retrieved	
	Test location	This requirement is tested as part of the EDE_TC01, as defined in D5.1 Results are found in D5.4 chapter 4.2.1	
REQ_DEDE2.5	Requirement Description	Service providers must be able to download the energy consumption data.	OK
	Test approach	Integration test	
	Test Description	When secure connection is established and credentials exchanged, consumption data can be fetched and validated.	
	Test location	This requirement is tested as part of the EDE_TC01, as defined in D5.1 Results are found in D5.4 chapter 4.2.1	
REQ_DEDE2.6	Requirement Description	Authentication toolkit for all actors (eIDAS compliant).	OK
	Test approach	Documentation	
	Test Description	Authentication with existing approaches (e.g. eIDAS) will be supported	
	Test location	D5.2, Chapter 4.2.2.1, pag. 55	
REQ_DEDE2.7	Requirement Description	Processes monitoring the system must be logged, stored (in local environment)	OK
	Test approach	Functional test	
	Test Description	The test verifies that an audit log is created for action happening in the system.	
	Test location	This requirement is tested as part of the EDE_TC03, as defined in D5.1 Results are found in D5.4 chapter 4.2.1	



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

REQ_DEDE5.1	Requirement Description	Service provider must be able to get proof of receiving the energy consumption data	OK
	Test approach	Functional test	
	Test Description	Proofs can be downloaded and verified	
	Test location	This requirement is tested as part of the EDE_TC03, as defined in D5.1 Results are found in D5.4 chapter 4.2.1	
REQ_DEDE5.2	Requirement Description	System logs integrity must be 3rd party verifiable (auditor)	OK
	Test approach	Functional test	
	Test Description	3rd parties can verify the interactions	
	Test location	This requirement is tested as part of the EDE_TC03, as defined in D5.1 Results are found in D5.4 chapter 4.2.1	

12.3 Decentralized Energy Flexibility Marketplace

Decentralized Energy Flexibility Marketplace			
ID	Validation Process		Result
REQ_DEFM1.1	Requirement Description	DSO shall be able to forecast of electricity production/consumption	OK
	Test approach	Integration test	
	Test Description	Described in D5.1 v2.0 as DEFM_TC01 and DEFM_TC03	
	Test location	Test definition: pilots_deployments/italian-energy-pilot/tests/test_dso.tavern.yaml Sample result: https://ci.sofie-iot.eu/jenkins/job/italy-energy-pilot/job/italian-energy-pilot-cd/29/testReport/tests.test_dso.tavern.yaml/	
REQ_DEFM1.2	Requirement Description	DSO shall be able to check the load and production forecasting of the whole distribution grid	OK
	Test approach	Integration test	
	Test Description	Described in D5.1 v2.0 as DEFM_TC01 and DEFM_TC03	
	Test location	Test definition: pilots_deployments/italian-energy-pilot/tests/test_dso.tavern.yaml Sample result: https://ci.sofie-iot.eu/jenkins/job/italy-energy-pilot/job/italian-energy-pilot-cd/29/testReport/tests.test_dso.tavern.yaml/	
REQ_DEFM1.3	Requirement Description	DSO shall be able to forecast of electricity production / consumption at the grid level	OK
	Test approach	Integration test	



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

	<i>Test Description</i>	Described in D5.1 v2.0 as DEFM_TC01 and DEFM_TC03	
	<i>Test location</i>	Test definition: pilots_deployments/italian-energy-pilot/tests/test_dso.tavern.yaml Sample result: https://ci.sofie-iot.eu/jenkins/job/italy-energy-pilot/job/italian-energy-pilot-cd/29/testReport/tests.test_dso.tavern.yaml/	
REQ_ DEFM1.4	<i>Requirement Description</i>	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	OK
	<i>Test approach</i>	Integration test	
	<i>Test Description</i>	Described in D5.1 v2.0 as DEFM_TC01 and DEFM_TC03	
	<i>Test location</i>	Test definition: pilots_deployments/italian-energy-pilot/tests/test_dso.tavern.yaml Sample result: https://ci.sofie-iot.eu/jenkins/job/italy-energy-pilot/job/italian-energy-pilot-cd/29/testReport/tests.test_dso.tavern.yaml/	
REQ_ DEFM1.5	<i>Requirement Description</i>	DSO shall be able to estimate the energy flexibility availability; Assess flexibility availability by using available historical data.	OK
	<i>Test approach</i>	Integration test	
	<i>Test Description</i>	Described in D5.1 v2.0 as DEFM_TC01 and DEFM_TC03	
	<i>Test location</i>	Test definition: pilots_deployments/italian-energy-pilot/tests/test_dso.tavern.yaml Sample result: https://ci.sofie-iot.eu/jenkins/job/italy-energy-pilot/job/italian-energy-pilot-cd/29/testReport/tests.test_dso.tavern.yaml/	
REQ_ DEFM1.6	<i>Requirement Description</i>	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.	OK
	<i>Test approach</i>	Integration test	
	<i>Test Description</i>	Described in D5.1 v2.0 as DEFM_TC01 and DEFM_TC03	
	<i>Test location</i>	Test definition: pilots_deployments/italian-energy-pilot/tests/test_dso.tavern.yaml Sample result: https://ci.sofie-iot.eu/jenkins/job/italy-energy-pilot/job/italian-energy-pilot-cd/29/testReport/tests.test_dso.tavern.yaml/	
REQ_ DEFM2.1	<i>Requirement Description</i>	When the Fleet Manager obtains the responsibility to provide the flexibility required by the DSO, a micro contract between the Fleet Manager and the DSO is executed.	OK
	<i>Test approach</i>	Integration test	
	<i>Test Description</i>	Described in D5.1 v2.0 as DEFM_TC02	



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

	<i>Test location</i>	Test definition: pilots_deployments/italian-energy-pilot/tests/test_dso.tavern.yaml Sample result: https://ci.sofie-iot.eu/jenkins/job/italy-energy-pilot/job/italian-energy-pilot-cd/29/testReport/tests.test_dso.tavern.yaml/	
REQ_ DEFM2.2	<i>Requirement Description</i>	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.	OK
	<i>Test approach</i>	Integration test	
	<i>Test Description</i>	Described in D5.1 v2.0 as DEFM_TC01	
	<i>Test location</i>	Test definition: pilots_deployments/italian-energy-pilot/tests/test_dso.tavern.yaml Sample result: https://ci.sofie-iot.eu/jenkins/job/italy-energy-pilot/job/italian-energy-pilot-cd/29/testReport/tests.test_dso.tavern.yaml/	
REQ_ DEFM4.1	<i>Requirement Description</i>	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.	Private tests
	<i>Test approach</i>	Unit test	
	<i>Test Description</i>	Charging stations deployed in Italian pilot site are compatible with all models of electric vehicles	
	<i>Test location</i>	Unit tests are part of Emotion internal codebase and are not released to the public. They are executed before each new release to ensure the proper implementation of functionalities. From a functional point of view, it is possible to assess the operability of the Fleet Manager platform used in the pilot.	
REQ_ DEFM4.2	<i>Requirement Description</i>	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 should be consistent, reliable, transparent and accessible to the partners. Furthermore, to perform optimized DR campaigns 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.	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	When secure connection is established, data from electric vehicles and charging stations are collected	
	<i>Test location</i>	The functionality can be assessed by operating the Fleet Manager platform used in the pilot	
REQ_ DEFM4.3	<i>Requirement Description</i>	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.	OK
	<i>Test approach</i>	Integration test	
	<i>Test Description</i>	Data collected from electric vehicles and charging stations are stored in the fleet manager server	



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

	<i>Test location</i>	Test definition: pilots_deployments/italian-energy-pilot/tests/test_dso.tavern.yaml Sample result: https://ci.sofie-iot.eu/jenkins/job/italy-energy-pilot/job/italian-energy-pilot-cd/29/testReport/tests.test_dso.tavern.yaml/	
REQ_ DEFM4.4	<i>Requirement Description</i>	As there will be more than one charging station on the pilot site, each individual charging station must have its own unique identifier.	Private Tests
	<i>Test approach</i>	Unit test	
	<i>Test Description</i>	All charging stations in the pilot system will have unique identifiers (IDs)	
	<i>Test location</i>	Unit tests are part of Emotion internal codebase and are not released to the public. They are executed before each new release to ensure the proper implementation of functionalities. From a functional point of view, it is possible to assess the operability of the Fleet Manager platform used in the pilot.	
REQ_ DEFM4.5	<i>Requirement Description</i>	As there will be more than one electric vehicle on the pilot site, each individual electric vehicle must have its own unique identifier.	Private Tests
	<i>Test approach</i>	Unit test	
	<i>Test Description</i>	All electric vehicles in the pilot system will have unique identifiers (IDs)	
	<i>Test location</i>	Unit tests are part of Emotion internal codebase and are not released to the public. They are executed before each new release to ensure the proper implementation of functionalities. From a functional point of view, it is possible to assess the operability of the Fleet Manager platform used in the pilot.	
REQ_ DEFM4.6	<i>Requirement Description</i>	To allow the EV user to realize the available charging stations and the fees associated with them, a web platform is required.	OK
	<i>Test approach</i>	Integration test	
	<i>Test Description</i>	Fleet manager and EV users can authenticate on the web platform and check the electric vehicles and charging stations real time status and historical data	
	<i>Test location</i>	Test definition: pilots_deployments/italian-energy-pilot/tests/test_dso.tavern.yaml Sample result: https://ci.sofie-iot.eu/jenkins/job/italy-energy-pilot/job/italian-energy-pilot-cd/29/testReport/tests.test_dso.tavern.yaml/	
REQ_ DEFM4.7	<i>Requirement Description</i>	Both charging stations and electric vehicles must be connected to the internet in order to send data.	OK
	<i>Test approach</i>	Integration test	
	<i>Test Description</i>	Charging stations and electric vehicles must be connected to internet to communicate with the fleet manager server	
	<i>Test location</i>	Test definition: pilots_deployments/italian-energy-pilot/tests/test_dso.tavern.yaml Sample result: https://ci.sofie-iot.eu/jenkins/job/italy-energy-pilot/job/italian-energy-pilot-cd/29/testReport/tests.test_dso.tavern.yaml/	
REQ_ DEFM5.1	<i>Requirement Description</i>	The charging station must be remotely controlled to start/stop charging sessions and to modulate the power output.	Private Tests



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

	<i>Test approach</i>	Unit test	
	<i>Test Description</i>	The charging station must be remotely controlled to start/stop charging sessions and to modulate the power output	
	<i>Test location</i>	Unit tests are part of Emotion internal codebase and are not released to the public. They are executed before each new release to ensure the proper implementation of functionalities. From a functional point of view, it is possible to assess the operability of the Fleet Manager platform used in the pilot.	
REQ_ DEFM7.1	<i>Requirement Description</i>	DSO shall be able 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	OK
	<i>Test approach</i>	Integration test	
	<i>Test Description</i>	From the DSO local network, load forecast for the two network zones can be fetched	
	<i>Test location</i>	Test definition: pilots_deployments/italian-energy-pilot/tests/test_dso.tavern.yaml Sample result: https://ci.sofie-iot.eu/jenkins/job/italy-energy-pilot/job/italian-energy-pilot-cd/29/testReport/tests.test_dso.tavern.yaml/	
REQ_ DEFM8.1	<i>Requirement Description</i>	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 charge electric vehicles	OK
	<i>Test approach</i>	Integration test	
	<i>Test Description</i>	The marketplace backend exposes the APIs needed by the actors for interacting with the system, participating with requests and offers.	
	<i>Test location</i>	Test definition: pilots_deployments/italian-energy-pilot/tests/test_dso.tavern.yaml Sample result: https://ci.sofie-iot.eu/jenkins/job/italy-energy-pilot/job/italian-energy-pilot-cd/29/testReport/tests.test_dso.tavern.yaml/	

12.4 Context-Aware Mobile Gaming Pilot

Context-Aware Mobile Gaming Pilot			
ID	Validation Process		Result
REQ_ MRMG0.1	<i>Requirement Description</i>	Each person interacting with the game should have a unique identifier.	OK
	<i>Test approach</i>	Unit test	
	<i>Test Description</i>	The test passes if all player IDs are different.	
	<i>Test location</i>	D5.4, Chapter 6.2	
REQ_ MRMG1.1	<i>Requirement Description</i>	Game challenges are accessible using the Android application	OK



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	In the test, the user opens the Scavenger Hunt game application and enters the Nearby Challenges tab. The user should see a list of (uncompleted) challenges that start in GPS coordinates that are within a set radius from the user. The requirement is met if the nearby challenges that exist on the backend are indeed visible in the Nearby Challenges tab.	
	<i>Test location</i>	D5.3, Chapter 6.3, Page 69	
REQ_MRMG1.2	<i>Requirement Description</i>	Players can join any nearby challenge from the game app.	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	The requirement is met if a challenge is added to the list of the player's current challenges, after the player presses the Start button in the client.	
	<i>Test location</i>	D5.3, Chapter 6.3, Page 70	
REQ_MRMG1.3	<i>Requirement Description</i>	Each challenge should have a unique identifier	OK
	<i>Test approach</i>	Unit test	
	<i>Test Description</i>	The test passes if IDs of all Scavenger Hunt challenges are different.	
	<i>Test location</i>	D5.4, Chapter 6.2	
REQ_MRMG1.4	<i>Requirement Description</i>	Time should be recorded for each player, starting after joining the challenge till the player completes it.	OK
	<i>Test approach</i>	Field test	
	<i>Test Description</i>	The requirement is met if, after a user plays the challenge, the completed challenge's start and end time fields are populated.	
	<i>Test location</i>	D5.4, Chapter 6.2	
REQ_MRMG1.5	<i>Requirement Description</i>	Players should receive unique tasks when near the IoT beacons based on their challenge.	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	The requirement is met if a user standing next to a BLE beacon receives a task in the mobile application.	
	<i>Test location</i>	D5.3, Chapter 6.3, Page 71	
REQ_MRMG1.6	<i>Requirement Description</i>	Players should be able to skip any task and receive the location of next IoT beacon using the In-App tokens.	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	If a player has Star items in-game, they can use one start to skip a task. The requirement is met if, when presented with a task and using	



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

		a star, the current task auto-completes and the user receives the clue to the next beacon.	
	<i>Test location</i>	D5.3, Chapter 6.3, Page 72	
REQ_MRMG1.7	<i>Requirement Description</i>	Players can buy In-App tokens using in-game currency	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	The requirement is met if the player can spend in-game coins in the application to buy Gem and Star tokens - increasing Gem and Star amounts in possession and decreasing Coins in possession.	
	<i>Test location</i>	D5.4, Chapter 6.2	
REQ_MRMG1.8	<i>Requirement Description</i>	System should automatically calculate rewards after player has completed a challenge	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	After a player completes a challenge, the requirement is met if the player sees rewards in the client application.	
	<i>Test location</i>	Others	
REQ_MRMG2.2	<i>Requirement Description</i>	System should automatically add the rewards to the player's account after the challenge ends.	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	The test passes if, after the reward transaction, the amount of coins in the escrow has decreased and the amount of coins in the player's account has increased by the reward amount.	
	<i>Test location</i>	Others	
REQ_MRMG3.1	<i>Requirement Description</i>	Player should be given the option to view advertisements while playing a challenge.	N/A
	<i>Test approach</i>	N/A	
	<i>Test Description</i>	This requirement has been replaced with REQ_MRMG 9	
	<i>Test location</i>	N/A	
REQ_MRMG3.2	<i>Requirement Description</i>	Player should receive tokens for viewing the advertisement.	N/A
	<i>Test approach</i>	N/A	
	<i>Test Description</i>	This requirement has been replaced with REQ_MRMG 9	
	<i>Test location</i>	N/A	
REQ_MRMG3.3	<i>Requirement Description</i>	Every ad viewability data should be recorded as a transaction on the blockchain.	N/A



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

	<i>Test approach</i>	N/A	
	<i>Test Description</i>	This requirement has been replaced with REQ_MRMG 9	
	<i>Test location</i>	N/A	
REQ_MRMG4.1	<i>Requirement Description</i>	Players can buy and sell Blockmoji assets on the blockchain	OK
	<i>Test approach</i>	Functional tests	
	<i>Test Description</i>	The requirement is met if players are able to buy and sell Blockmoji items on the blockchain.	
	<i>Test location</i>	D5.4, Chapter 6.2	
REQ_MRMG4.2	<i>Requirement Description</i>	Every asset traded on the platform should be recorded as a transaction on the blockchain.	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	The test passes if, after a Blockmoji trading transaction has occurred, that transaction can be read from the blockchain.	
	<i>Test location</i>	D5.4, Chapter 6.2	
REQ_MRMG5.1	<i>Requirement Description</i>	Web application for designing new challenges and uploading advertisements.	Partial
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	The test passes if the developer can add a new challenge to the game.	
	<i>Test location</i>	Others	
REQ_MRMG5.2	<i>Requirement Description</i>	Access control to the web services based on the role of the user.	Partial
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	Only developers can add the challenge to the game.	
	<i>Test location</i>	Others	
REQ_MRMG7.1	<i>Requirement Description</i>	Blockmoji item rewards be can offered to players through challenges	OK
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	If a challenge offers a Blockmoji item reward, the player should see it in their mobile application reward screen after completing the challenge.	
	<i>Test location</i>	D5.3, Chapter 6.3, Page 74	



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

REQ_ MRMG7.2	<i>Requirement Description</i>	Blockmoji rewards should be added and recorded on the blockchain.	Test location not available
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	The requirement is met if, after a player completes a challenge that awards a Blockmoji item, the receiving of the item can be read as a transaction on the blockchain.	
	<i>Test location</i>	Others	
REQ_ MRMG8.1	<i>Requirement Description</i>	Ads manager should publish any ad video using the web application	N/A
	<i>Test approach</i>	N/A	
	<i>Test Description</i>	This requirement has been replaced with REQ_ MRMG 9	
	<i>Test location</i>	N/A	
REQ_ New REQ_ MRMG9.1	<i>Requirement Description</i>	Every user that signs in with Decent ID should have a unique decentralized ID with the connection.	Test location not available
	<i>Test approach</i>	Unit test	
	<i>Test Description</i>	The test passes if all connection DIDs of a user are different.	
	<i>Test location</i>	-	
New REQ_ MRMG9.2	<i>Requirement Description</i>	Companies send pieces of the user's ad profile data to the user as credentials	Test location not available
	<i>Test approach</i>	Functional test	
	<i>Test Description</i>	The test validates that ad profile data is received by the connection if the user has allowed the request.	
	<i>Test location</i>	-	
New REQ_ MRMG9.3	<i>Requirement Description</i>	Companies request access to the user's ad profile credentials, and user can accept them	Test location not available
	<i>Test approach</i>	Field test	
	<i>Test Description</i>	The requirement is met if, after a service requests for credentials, the user sees a prompt in the mobile application to accept the request.	
	<i>Test location</i>	-	
New REQ_ MRMG9.4	<i>Requirement Description</i>	User can revoke connections' access to credentials by resetting the decentralized ID for the connection	Test location not available
	<i>Test approach</i>	Field test	
	<i>Test Description</i>	The requirement is met if the player can unlink connections from the mobile application.	



Document:	H2020-IOT-2017-3-779984-SOFIE/ D5.4 – Final Validation & Replication Guidelines					
Security:	Public	Date:	7.5.2021	Status:	Completed	Version: 1.10

	<i>Test location</i>	-	
--	----------------------	---	--