Deliverable D200.3

Final Report on Validation Activities, Architectural Requirements and Detailed System Specification

WP200

Project acronym & number:	SmartAgriFood – 285 326
Project title:	Smart Food and Agribusiness: Future Internet for Safe and Healthy Food from Farm to Fork
Funding scheme:	Collaborative Project - Large-scale Integrated Project (IP)
Date of latest version of Annex I:	18.08.2011
Start date of the project:	01.04.2011
Duration:	24
Status:	Final
Authors/ Contributors	Panagis Magdalinos, Alex Kaloxylos, Vassilis Sarris, Aggelos Groumas, Lampros Katsikas, Matina Lala, Ioanna Lampropoulous Zoi Politopoulou, Eleni Antoniou, Liisa Pesonen, Koistinen Markku, Frederick Teye, Esther Mietzsch, Egon Schulz, Nikola Vucic, Carlos Maestre Terol, Leena Noros
Due date of deliverable:	31.12.2012
Document identifier:	D200.3
Revision:	1.0
Date:	19.12.2012

FUTURE INTERNET PPP

SEVENTH FRAMEWORK PROGRAMME

Smart Agri-Food

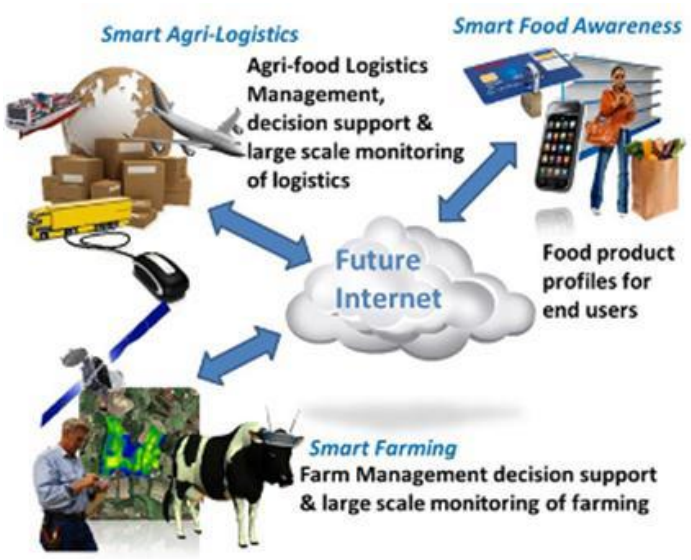
The SmartAgriFood Project

The SmartAgriFood project is funded in the scope of the Future Internet Public Private Partnership Programme (FI-PPP), as part of the 7th Framework Programme of the European Commission. The key objective is to elaborate requirements that shall be fulfilled by a "Future Internet" to drastically improve the production and delivery of safe & healthy food.

Project Summary

SmartAgriFood aims to boost application & use of Future Internet ICTs in agri-food sector by:

- Identifying and describing technical, functional and non-functional Future Internet specifications for experimentation in smart agri-food production as a whole system and in particular for smart farming, smart agri-logistics & smart food awareness,
- Identifying and developing smart agri-food-specific capabilities and conceptual prototypes, demonstrating critical technological solutions including the feasibility to further develop them in large scale experimentation and validation,
- Identifying and describing existing experimentation structures and start user community building, resulting in an implementation plan for the next phase in the framework of the FI PPP programme.



Project Consortium

- LEI Wageningen UR; Netherlands
- ATB Bremen; Germany
- TNO; Netherlands
- CentMa GmbH; Germany
- ATOS ORIGIN; Spain
- ASI S.L.; Spain
- Huawei; Germany
- MTT Agrifood Research; Finland
- KTBL e.V.; Germany
- NKUA; Greece
- UPM; Spain
- BRI Magyarország, Hungary
- Aston University; United Kingdom
- VTT; Finland
- OPEKEPE; Greece
- John Deere; Germany
- Wageningen University; Netherlands
- EHI Retail Institute GmbH; Germany
- GS1 Germany GmbH; Germany
- SGS S.A.; Spain
- Condis Supermercats S.A.; Spain
-

More Information

Dr. Sjaak Wolfert (coordinator)
LEI Wageningen UR
P.O. Box 35
6700 AA Wageningen

e-mail: sjaak.wolfert@wur.nl
phone: +31 317 485 939
mobile: +31 624 135 790
www.smartagrifood.eu

Dissemination Level

PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document Summary

This document intends to provide a report on the validation activities, architectural requirements and system specifications of the WP200 pilots. Towards this end, it builds on the outcome of the two previous deliverables (Deliverables 200.1 [2] and 200.2 [4]). The scope of this deliverable is manifold:

1. Provide the System Specifications:

   System Specifications in terms of available indicative APIs, including key input/output parameters, deployment architecture and preliminary performance results will be provided through D500.5.2 [8]. In the context of this document we provide, in Sections 3 and 4, details regarding deployment and usage of the system from a user perspective. Architectural specifications and implemented functionalities are validated against the functional requirements identified in D200.1 [2] (Section 9.1).

2. Ensure cross workpackage alignment and integration of FI-WARE generic enablers in the context of the implementation:

   Both pilots integrate a large number of FI-WARE Generic Enablers. The integration effort is overall presented in Section 2.4 and then divided per pilot in Sections 3.3 and 4.3. Additional effort was taken in order to align the implementation effort with the work of WP500, presented in D500.3 [6].

3. Describe the supported scenarios:

   Building on top of the implemented prototypes, the supported scenarios are presented in terms of high level functionality. The presentation serves a dual purpose; on one hand highlight the added value brought to the end user by the WP200 effort while on the other provide technical details regarding the exploitation and integration of the various hardware and technical components.

4. Highlight the cooperation between pilots:

   Finally, this document details the cooperation between the two pilots. The current status of this effort is captured in Section 5. Additional items and the final results will be provided in D500.5.2 [8].

Special care has been taken towards keeping the document small and concise; additional material has been placed in the annex section of this document (Section 9). The document however is not standalone per se; it should be evaluated in conjunction with D200.4 (assessment report by end-users - [5]), D500.5.2 (full version of the SmartAgriFood prototypes - [8]) and D100.3 (policy issues and UC harmonisation - [2])

Abbreviations

API	Application Programming Interface
ESB	Enterprise Service Bus
FI	Future Internet
FM(I)S	Farm Management (Information) System
GCP	Global Customer Platform
GE	Generic Enabler
GLN	Global Location Number
GUI	Graphical User Interface
ICT	Information Communication Technologies
IdM	Identity Management
IP	Internet Protocol
LLRP	Low Level Reader Protocol
MoSCoW	Must - Should - Could - Won't
OSGi	Open Services Gateway initiative
PPP	Public Private Partnership
RCP	Reader Core Proxy
REST	Representational State Transfer
RFID	Radio Frequency Identification
SensorML	Sensor Model Language
SLA	Service Level Agreement
SME	Small-Medium Enterprise
SOA	Service Oriented Architecture
SQL	Structured Query Language
SSCC	Serial Shipping Container Code
STD	Standard Deviation
UML	Unified Modeling Language
URL	Uniform Resource Locator
USDL	Unified Service Description Language

Table of Contents

List of Figures

List of Tables

1 Introduction

The purpose of this document is to provide a holistic view of the work carried out in the context of the WP200. Specifically, D200.3 aims to present the System Specification and Validation results of the Greenhouse and Smart Spraying pilots that have been designed in the context of WP200. In parallel it links the work of WP200 with WP500 thus providing a homogenized view of the technical accomplishments. The content of this document in conjunction with the content of D500.5.2 ([8] - to be released on M24) enable –in principle- any third party to implement the two pilots and exploit their merits.

The document commences by describing the final version of the architecture and the updates that took place since D200.2 [4]. The architecture is enhanced by the outcome of the development process as well as the work carried out in the context of WP500 and specifically in D500.3 [6] and D500.4 [7]. FI-WARE's Generic Enablers employed in the context of the architecture are also presented and mapped on the functional entities of the FMS controller. In parallel, SAF Domain Specific enablers are outlined. Finally, the architectural specifications are validated against the functional requirements identified in D200.1 [2]. This exercise ensures backward traceability in the context of WP200 (i.e. the functional requirements identified by the use cases in the beginning of the project are adequately addressed by the functionalities offered by the architecture) and work package alignment between WP200 and WP500 (i.e. the generic services identified in the context of WP500 are incorporated in the refined version of the architecture). This work is presented in Section 2.

Sections 3 and 4 provide the technical specifications of the two pilots implemented in the context of WP200, namely the Greenhouse and the Smart Spraying pilot. The presentation assumes a top down approach. In the beginning, a high level view of the implemented pilot is given in terms of functionalities and added value to the end user. Then we identify the domain specific and FI-WARE's generic enablers encapsulated in the implementation. Core technical parts are omitted and will be reported in D500.5.2 [8]. More specifically, software and hardware design accompanied by detailed UML diagrams, API descriptions and message sequence charts will all be reported through WP500. The presentation is concluded with the validation of the prototype (similar to the validation of the architecture) and the identification of potential standardization opportunities stemming from the implementation effort.

The document concludes with the integration effort of the two pilots. Essentially, in the end of the WP200 lifetime, the outcome will not be two pilots but one core-integrated pilot with application in greenhouse and spraying management. The integration effort is presented in terms of design and implementation Section 5. Additional material related to the integration of the WP200 work with other EC projects and FMISs are also provided.

2 Refined Architecture of FMS

In the context of this section, the refined system architecture will be presented. The work builds on the outcome of D200.2 [4] and extends it by taking into account the progress of the last months. Additionally, the refined architecture attempts to capture design decisions and updates that were dictated by outside factors (e.g. development of FI-WARE enablers). Figure 2-1 provides a high level view of the architecture. The architecture consists of two parts, the Cloud FMS and the Local FMS. Essentially, both entities serve the purpose of farm management; however the Cloud FMS is empowered with additional capabilities stemming from FI Services and Enablers. Therefore, it is appointed the task of farm management while the Local FMS is used for local tasks (e.g. interaction with sensors and actuators) as well as the overall management in case network errors appear.

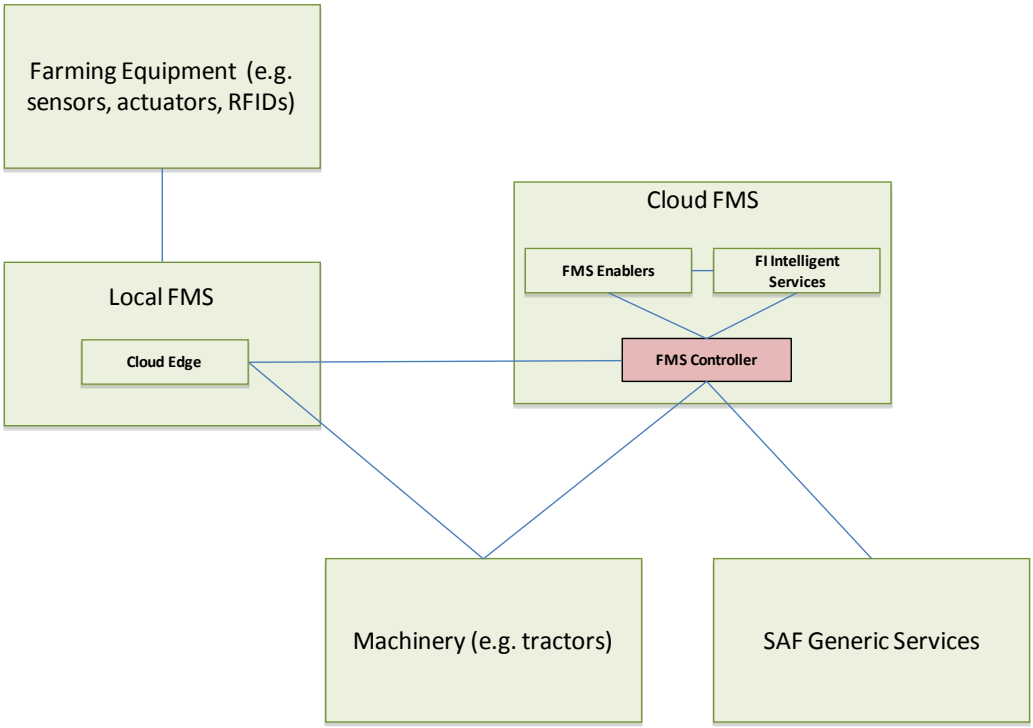


Figure 2-1: The overall architecture

2.1 Refined System Architecture

Within the FI-PPP, all the innovations that are proposed by the SmartAgriFood [1] project for improving the stakeholders' life along the food chain are in line with the breakthroughs of the Future Internet. As far as the smart farming area is concerned, a thorough research about existing systems was made that revealed their imperfections. The identification of end users and the acquisition of their requirements had led to determine the conceptual design of the "TO-BE" Farm Management System (FMS). At the same time, the evolution of the generic principles that the Future Internet should follow, enabled us to convince the few disbelievers that our proposed system is not only feasible but also has potentials to become the basic architecture for any prospective farming system.

The refined architecture of a Cloud Farm Management System (Cloud FMS) consists of four layers, each performing specific operations. This novel perception of the FMS architecture is an abstract conception of the one presented in the previous deliverable (i.e. D200.2). In general,

all the modules that have been presented so far remain the same. What is changing is their classification due to the development of FI-WARE GEs. As depicted in Figure 2-2, on the top of this stack lays the FMS User, the system's end user. The end user of the FMS can be a farmer, a worker in a farm or any stakeholder along the food chain such as a trader, an agriculturist, a service provider, a consumer, etc.

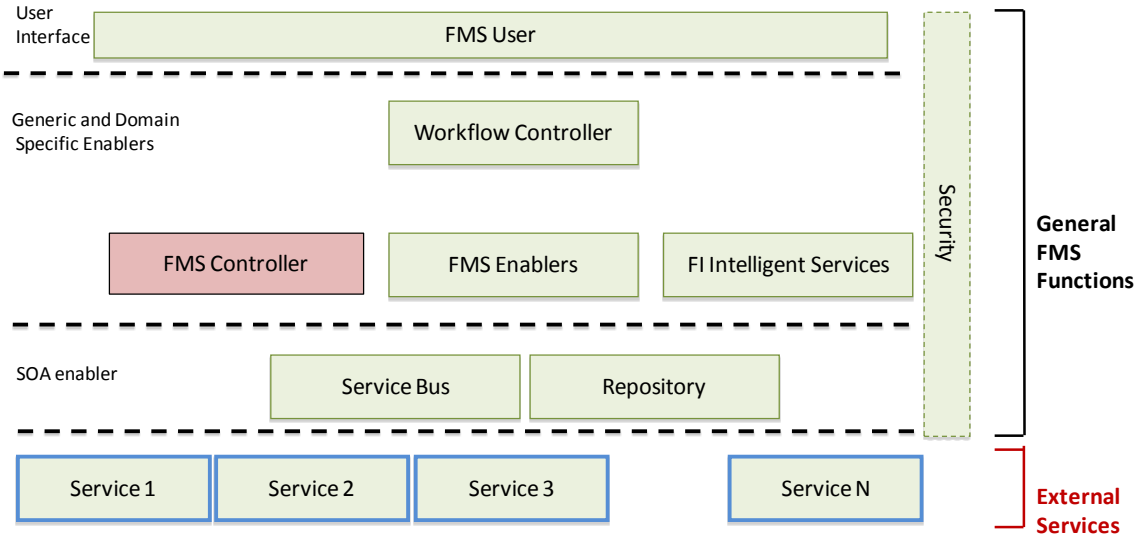


Figure 2-2: Cloud FMS Architecture

An FMS User wishes to interact with the FMS and receive high quality services and experience. For this reason, the bottom layer depicts a variety of External Services that the user would like to employ. Any developer, can create his own agricultural service e.g., weather service, e-agriculturist, etc that could be advertised by any FMS instance and subsequently consumed by an FMS user.

The FMS, in order to enable the usage of any service, has the ability to communicate with them and transfer from or to any kind of data requested. The actual transfer is achieved via the third layer from the top, named SOA enabler layer (a layer enabling the realization of Service Oriented Architecture - SOA) that sets the communication links between the core modules of an FMS and a service that is available in the internet.

The core functionality of the FMS is abstractly depicted in Figure 2-2 as the second layer from the top and identified as Generic and Domain Specific Enablers Layer. The role of Workflow Controller is to capture all the incoming messages of an FMS User and forward them to one of the following entities/modules:

- appropriate internal modules that support the realization of the FMS controller functionalities (FMS Enablers),
- the module Intelligence used for implementing intelligent services (FI Intelligent services),
- the FMS Controller, which hosts modules implementing the Domain Specific Enablers of our future FMS.

It should be pointed out that that the realization and implementation of the FMS Controller and FMS Enablers may exploit available FI-WARE GEs or instantiate them.

Generic and Domain Specific Enablers Layer

The modules building up this layer (c.f. Figure 2-3) are the same as those presented in D200.2 [4]. Nevertheless, since FI-WARE is developing its work and reconsiders new classification of the development of the GEs, we are forced to adjust our architecture in line with these alterations.

For these reasons, we have made the following considerations that led to a more refined architecture:

1. Group all sub - modules that imply functionalities of GEs that will be released within phase I or the next phases.

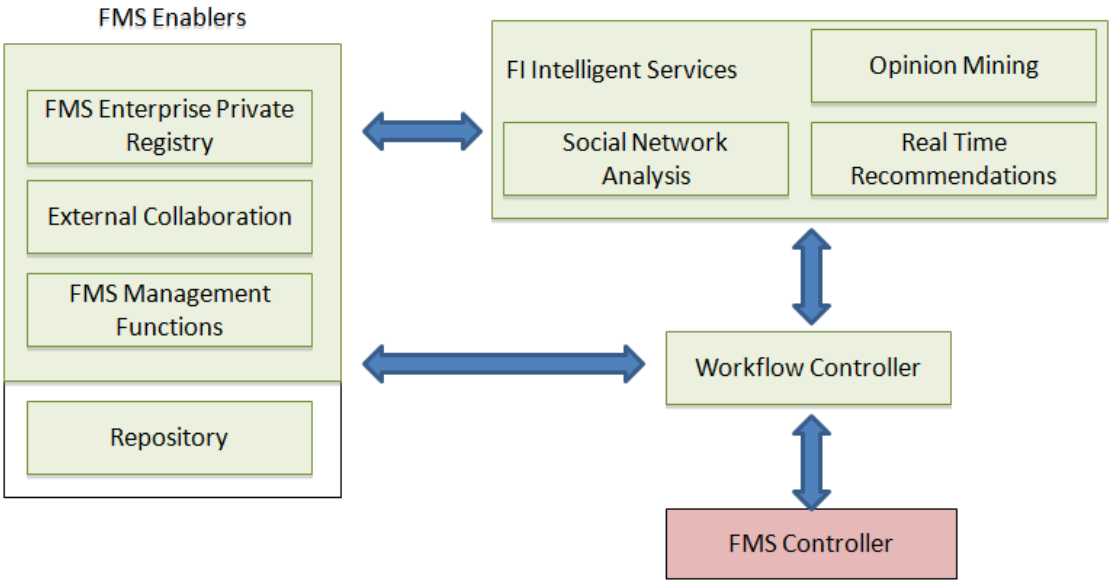


Figure 2-3: Generic and Domain Specific Enablers Layer

2. All sub-modules that will not be released by FI-WARE, will be grouped in a module called FI Intelligent Services. Its name is derived by the fact that within Product Vision these generic principles were classified as GEs for implementing Intelligent Services. Due to the fact that FI-WARE has not provided any information about their development we have grouped them in a new module call FI Intelligent Services, since their functionalities are necessary for developing high quality services. Figure 2-4 shows the module FI Intelligent Services that consists of three sub-modules. These are:
   - Opinion Mining:
     In order to increase the trustworthiness of the proposed FMS, moving towards the FI principles, we will provide mechanisms for evaluating and rating the overall system, the services provided through the Marketplace and the stakeholders involved.

Figure 2-4: FI Intelligent Services Module

- Real Time Recommendation

  The profiles that will be created from the Web analysis profiling will be used in order to provide personalized recommendations / suggestions to the end users. For example, a farmer who browses for advisory service for tomatoes will be suggested to subscribe to a monitoring service for the prices of the tomatoes in the markets.

- Social Network Analysis

  The last years a tremendous evolution of social networking has been made. An FMS user cannot be isolated; the proposed architecture enables him to become an actual node in the web agricultural sector and gives him the opportunity to communicate via the Internet with linked users who are involved to the food sector. Moreover, the link between different users enables services to process data of neighboring users in order to provide more qualitative results.

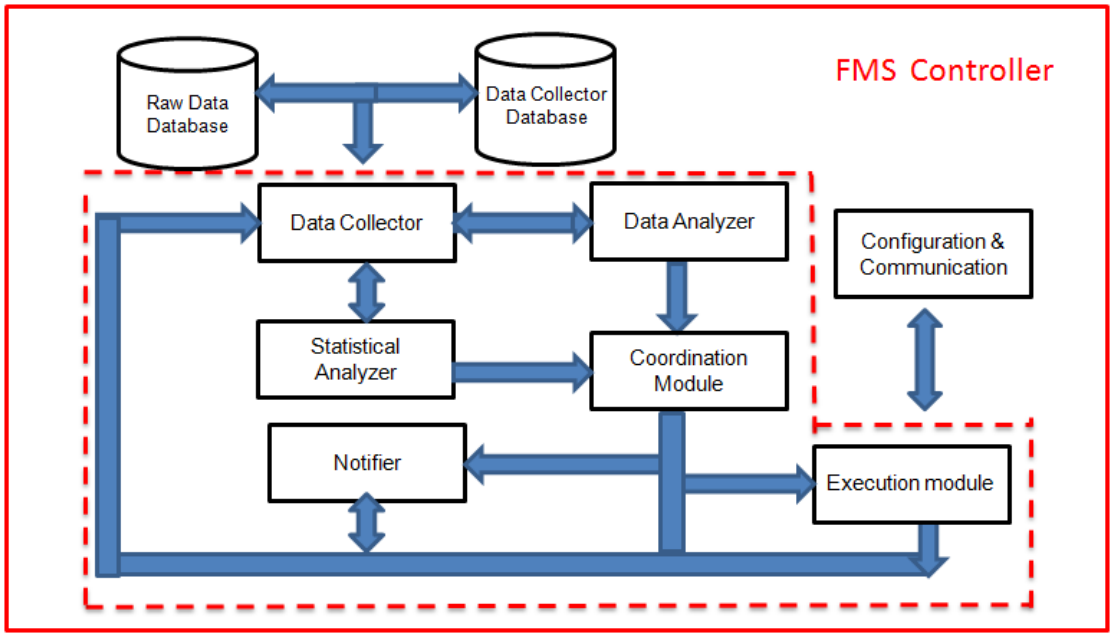


Figure 2-5: FMS Controller

3. The Workflow Controller operates as a message dispatcher for the second layer. It is used to process all the incoming messages from the top layer and forward them to the most suitable sub-module that is located in the GEs or in the Intelligence or to the message dispatcher of the FMS Controller .
4. As far as the FMS Controller is concerned, the modules depicted in Figure 2-5, act as the edge between the cloud and the local system as well as the point between the core FMS with the end user. Their role is to process incoming data and perform numerous actions (e.g. aggregate, classify, produce statistical analysis, etc) enable the communication with external services and notify the FMS User about significant events of his interest.

The name of the layer (i.e. Generic and Domain Specific Enablers) denotes that:

- Each sub-module is designed to act according to the generic principles of the Future Internet that the FI-WARE project has set.

- Each sub-module will be implemented with the potential to interact with the relevant GEs of the Core Platform.
- The FMS Controller consists of modules that perform tasks completely relevant with the smart farming sector. Thus, its internal modules are considered Domain Specific Enablers. The innovation introduced is that since a domain specific enabler uses the generic principles of the Generic Enablers, all the modules will be able to directly interact with the GEs of the Core Platform when needed.
- In the context of FMS, Intelligence is a Domain Specific Enabler. Its sub-modules will be used to provide high quality of experience to an end user interacting with external services. Assuming that an FMS User wants to consume the best service at the best possible price, the provision of recommendations for using a specific service that possible concerns him (Real time Recommendations), the fact that he can interact with other end users and enable the exchange of data between their subscribed services (Social Network Analysis) as well as the opportunity to be informed about the rating of a service or even rate a service (Opinion Mining), would definitely increase the trust to his FMS provider and the services provided.
- The module FMS Enablers consists of sub-modules that implement generic principles of FI-WARE and interact with the GEs of the Core Platform when needed. For example External Collaboration facilitates the interaction with the Identity Management GE and the Marketplace GE. Also the FMS Management Functions cater for charging, billing and accounting of services. Additionally, these modules support interactions with the four SAF Generic Services (Section 2.3)

SOA enabler

This layer comprises a channel that provides interconnection between heterogeneous services. The Enterprise Service Bus (ESB) architecture will therefore help the Configuration and Communication Module to enable the appropriate transactions between services. The ESB is currently the most innovative middleware layer that assists the integration of heterogeneous systems, supports web services, XML, message routing and event-based interactions.

External Services Layer

The External Services Layer encapsulates the variety of services/applications that will be offered through the architecture. Some of them may be:

- E-agriculturist services: Software that can assist farmers in their daily tasks by providing them with suggestions for complex situations. For example, if the environmental conditions (e.g., humidity, temperature) are such that a disease may occur, the farmer is notified to take specific actions.
- Spraying services: This service can be used in order to enable farmers or tractor drivers to schedule their spraying tasks inside a farm.
- Meteorological services: The farmer needs to have access to meteorological data (current condition and forecasts). These data can be used by him in order to plan his forthcoming tasks about his crops. They can also be give as inputs to multiple services for providing complex outputs e.g. the combination of an e-agriculturist service and a meteorological service that predicts the weather could give to the farmer suggestions on how to handle possible future alarms.

- State's and policies services: These types of services would link efficiently the governmental authorities with the farmers. In this way, farmers can be easily informed about possible subsidies, their policies about the use of chemicals or the rules for certifying a crop as "bio", potential hazards, etc.

Numerous services that can be provided to the stakeholders of the farming area have been described in the previous WP200 deliverables, namely D200.1 [2] and D200.2 [4]. Technical details regarding the actual implementation of these services are provided in subsequent sections.

2.2 Validation of System Architecture

The identified system architecture is validated against the functional requirements identified in D200.1 [2]. The purpose of this exercise is to ensure backward traceability and validate that the provided functionalities indeed address the functional requirements posed by the identified set of UCs. The results signify that the refined version of the architecture adequately addresses the initial set of requirements. The full set of requirements as well as the work conducted in the context of this document are provided in tabular form in Section 9.1.

The next two chapters present the Domain Specific Enablers and the Generic Enablers that are directly linked with the FI-WARE. Last, we need to state that since FI-WARE develops more and more GEs, we are in duty bound to track any modification or addition in the FI-WARE architecture and Open Specifications and adjust our conception with it since our design is planned to give the basic functionalities of any future FMS.

2.3 Domain Specific Enablers

One of the SmartAgriFood domain specific enablers is the FMS Controller which encapsulates functionalities specific to the project. The detailed specifications of the FMS Controller have been provided in D200.2 [4]. Here we provide a short overview for reasons of completeness (e.g. provide a standalone document). Figure 2-6 provides an overview of the modules building up this functional block.

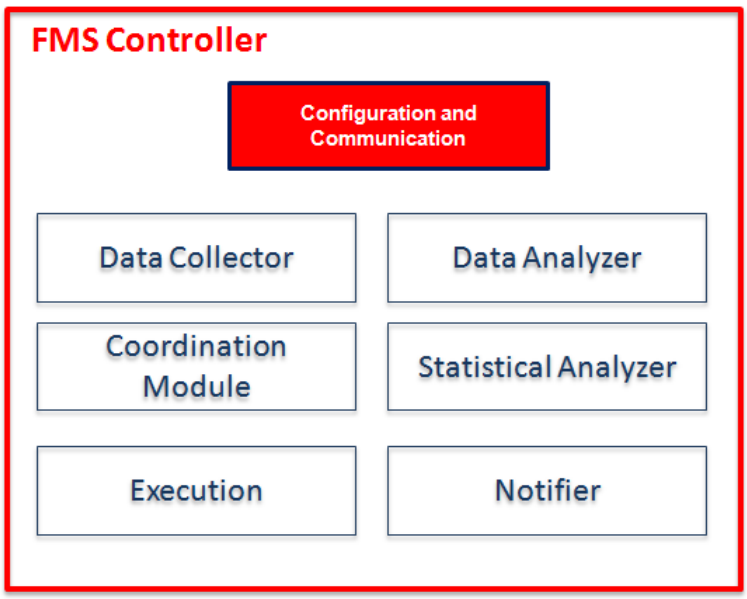


Figure 2-6: FMS Controller

The functionalities of the various modules appear in the following list:

- **Configuration and Communication:** It contains three sub-modules. The *Message Dispatcher* , the *Configurator* and the *Authentication.*
- **Data collector**: Its main task is to transfer data to and from the databases of the system.
- **Data Analyzer:** Mainly involved with the processing and analysis of different types of data.
- **Coordination module:** This module receives input from the Data Analyzer and the Statistical Analyzer and has the "intelligence" to handle a situation.
- **Notifier module:** It is used to inform stakeholders about events of the existing system.
- **Statistical Analyzer:** It processes an amount of data using mathematical and statistical functions.
- **Execution module:** It is used for actions that can be executed automatically inside a farm.

Moreover, based on the *w*ork conducted in the context of D500.3 [6] and D500.4 [7] a set of four SAF generic services has been identified. These services are integrated in the overall architecture as presented in Section 2 of the D500.4. In the context of this document we briefly describe their functionality and integration in order to ensure inter-WP alignment and consistency of the document.

Certification Service: The service consists of two sub-services, the Certification Validation Service and the Logo Validation Service. The functional requirement implemented by this service is related to the reliability and trustworthiness of the collected and transmitted information. In fact, that information is what will be communicated at the end to the consumers/end-users and one of the aspects to improve for the future internet is the trustworthiness of the supply chain information management for all the stakeholders, including end-users.

Product Information Service: The Product Information comprises a generic service that can be developed, implemented and provided by ICT services providers. In SmartAgriFood, its main focus is to enable the exchange of product related information and facilitate the control in complex supply networks while in parallel drastically reduce reaction times with respect to quality issues. Moreover, the service aims to offer basic technical environment that provides the generic functionalities while in parallel allow a participation of food chain actors with diverse ICT maturity levels.

Business Relations Service: The Business Relations Service provides an interoperability infrastructure to maintain interactions of business partners, enabling connectivity and information exchange and facilitating the addressing and search of information in a Future Internet. This service pursues two main objectives: i. Creating long term and quality relationships between partners playing different roles, supporting business-to-business, consumer-to-business and consumer-to-consumer relationships and ii. Investigating how a customer responds to provided services, managing their feedback and distributing the information to the appropriate business entity.

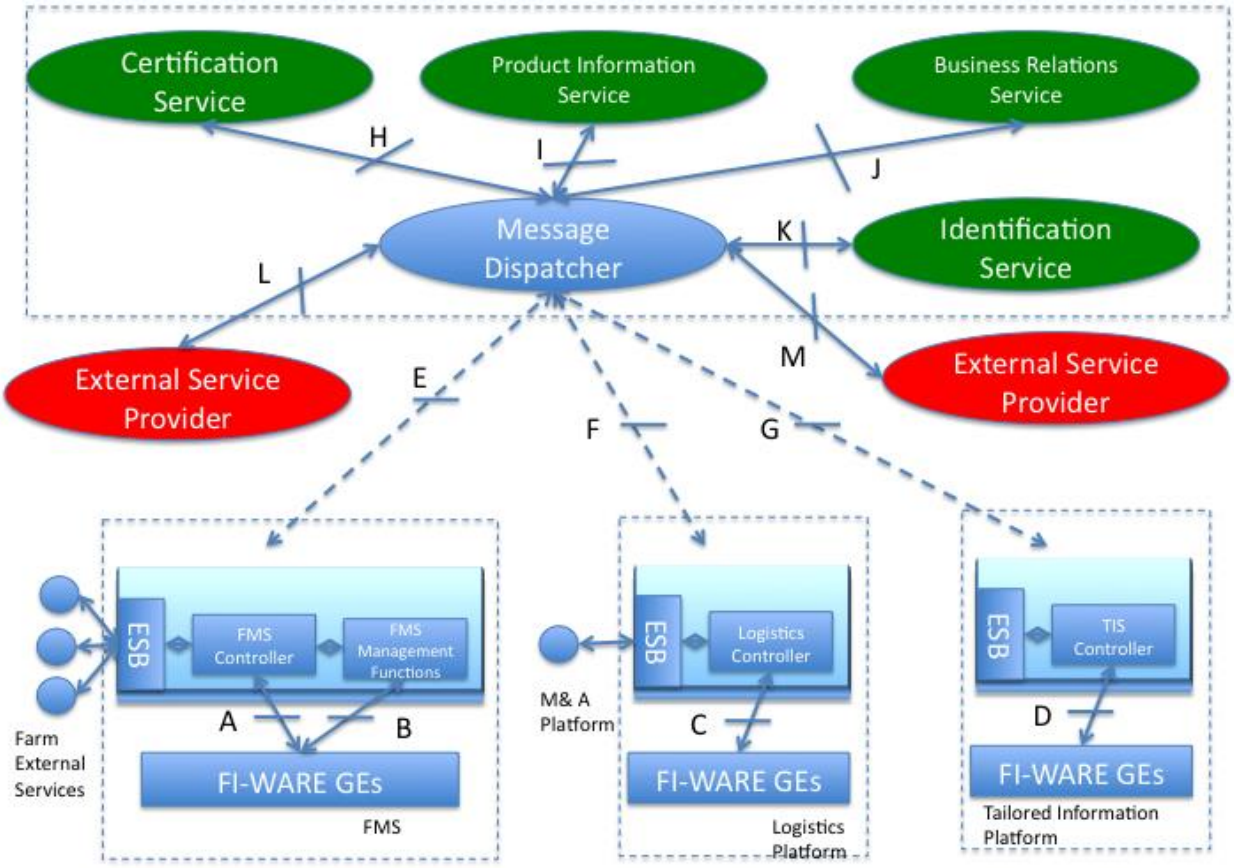


Figure 2-7: Cross-Workpackage SAF architecture incorporating the four generic services.

Identification Service: The Identification Service provides functions for: i. Registration of user, systems, and service provider accounts, ii. Login for registered users, systems, and providers by means e.g. of user name and password, iii. Service licensing and pricing agreements management and iv. Self-administration for users (e.g., password management, changes to user data, security management, access to recourses and trustees' management).

Based on the latter, the overall system architecture of the SAF project (cross-workpackage) is depicted in Figure 2-7, which is also described in the section 2 of the D500.4.

2.4 Link with FI-WARE's generic enablers

Several modules of the architecture, depending on its role, are able to act as a "client" to the Core Platform modules. Thus, it is important to clarify which sub – module shall interact with which GE located in the Core Platform, as described in Table 2-1. It should be pointed out that this is a theoretic exercise which attempts to link all GEs to the FMS modules presented in the previous paragraphs. Specific usage of the available/released GEs by the two WP200 pilots is provided in the respective sections.

Table 2-1: Correlation between GEs and FMS modules

Generic Enabler	**FMS Sub-Module**
Application Ecosystem and Delivery Framework	
Service Mashup	It works as a standalone. No mapping to an FMS module.
Application Mashup	It works as a standalone. No mapping to an FMS module.
Service Composition	It works as a standalone. No mapping to an FMS module.
Light Semantic Composition	It works as a standalone. No mapping to an FMS module.
Marketplace	External Collaboration (public USDL), FMS registry.
Mediator	Configuration Communication.
Registry	External Collaboration (public USDL), FMS registry.
Repository	External Collaboration (public USDL), FMS registry.
Cloud Hosting	
Object Storage	Data Collector, Notifier.
IaaS DCRM	FMS Management Functions.
IaaS CERM	FMS Management Functions.
IaaS Service Management	FMS Management Functions.
Internet of Things Service Enablement	
(Backend) Things Management	Configuration Communication.
(Backend) Device Management	Configuration Communication.
(Gateway) Data Handling	Local Configuration Communication
(Gateway) Protocol Adapter	Local Configuration Communication.
(Gateway) Device Management	Local Configuration Communication.
Data / Context Management Services	
Complex Event Processing GE	Coordination module
Publish / Subscribe Broker GE	Configuration Communication, Notifier.
Semantic Annotation GE	Data Collector, Notifier.
Big Data Analysis GE	Data Analyzer.
Multimedia Analysis Generation GE	Data Analyzer.
Query Broker GE	Data Collector, Notifier.

Generic Enabler	**FMS Sub-Module**
Location GE	Data Collector.
Semantic Application Support GE	Data Collector, Notifier.
Unstructured Data Analysis GE	Data Analyzer.
Meta – data pre-processing GE	Data Collector, Notifier.
Interfaces to Network and Devices	
Cloud Edge GE	Local FMS
Connected Devices Interface (CDI) GE	Local Configuration Communication.
Network Information and Control GE	Configuration Communication.
Service, Capability, Connectivity, and Control GE	Configuration Communication.
Security	
Identity Management GE	Configuration Communication.
Security Monitoring GE	It works as a standalone. No mapping to an FMS module.
Context Based Security & Compliance GE	Configuration Communication.
Data Handling GE	Configuration Communication.
Privacy GE	Configuration Communication.

3 Greenhouse Pilot Specification

In the context of this paragraph the Greenhouse pilot is going to be presented in terms of functionalities and specifications. Section 3.1 provides a high level view of the pilot while Sections 3.2 and 3.3 detail the domain specific and FI-WARE generic enablers employed in the prototype. The section is concluded with the pilot validation process (Section 3.4) and the roadmap towards potential standardization (Section 3.5).

3.1 High level view of the Greenhouse Pilot

The main objective of the Greenhouse Management prototype is a Future Internet compliant framework which will be able to take into account real data (e.g. weather data) from sensors and provide it to a Farm Management System (FMS) in order to take smart decisions regarding actions that need to be done and will eventually lead to the increase of the farm's productivity. External services have access to the real data collected and produce results related to smart planning of farming actions. Notifications and alerts about the current situation and actions are forwarded to the farmer. In this way, a farmer achieves having a complete surveillance of his farm.

The Greenhouse Management prototype has been implemented in order to fulfill a number of innovative concepts. In particular:

- Lower investment cost since the intelligence of the system is located in the cloud.
- Automatic communication of the system with any equipment/machinery using a service oriented architecture.
- Storage of raw data and guaranteeing user-independence from any FMS.
- Service adaptation according to user preferences and end-device capabilities.
- One-stop market place facilitating the end-user in his everyday needs.
- Integration of domain specific services (e.g., advisory services, meteo services, task planning services, policy state services etc).
- Learning schemes focusing on improving operations through proper exploitation of accumulated data.
- Using IoT technologies it is possible to tailor scalable and flexible farm specific control hardware and customized farm systems where components origin from different manufacturers thus enabling integration of existing infrastructure.

The Greenhouse pilot testbed consists of the "Greenhouse part" and the "Cloud Part". Inside the greenhouse, the deployed wireless nodes send their measurements periodically to the gateway which is deployed on a commodity PC located at the farmer's office, not far from the greenhouse itself. From there, the information is propagated to NKUA premises and specifically to the FMS controller which is deployed in a server with public internet IP. The processed information and the extracted knowledge are subsequently presented to the farmer via a web based portal, deployed on another server. The testbed topology is depicted in Figure 3-1.

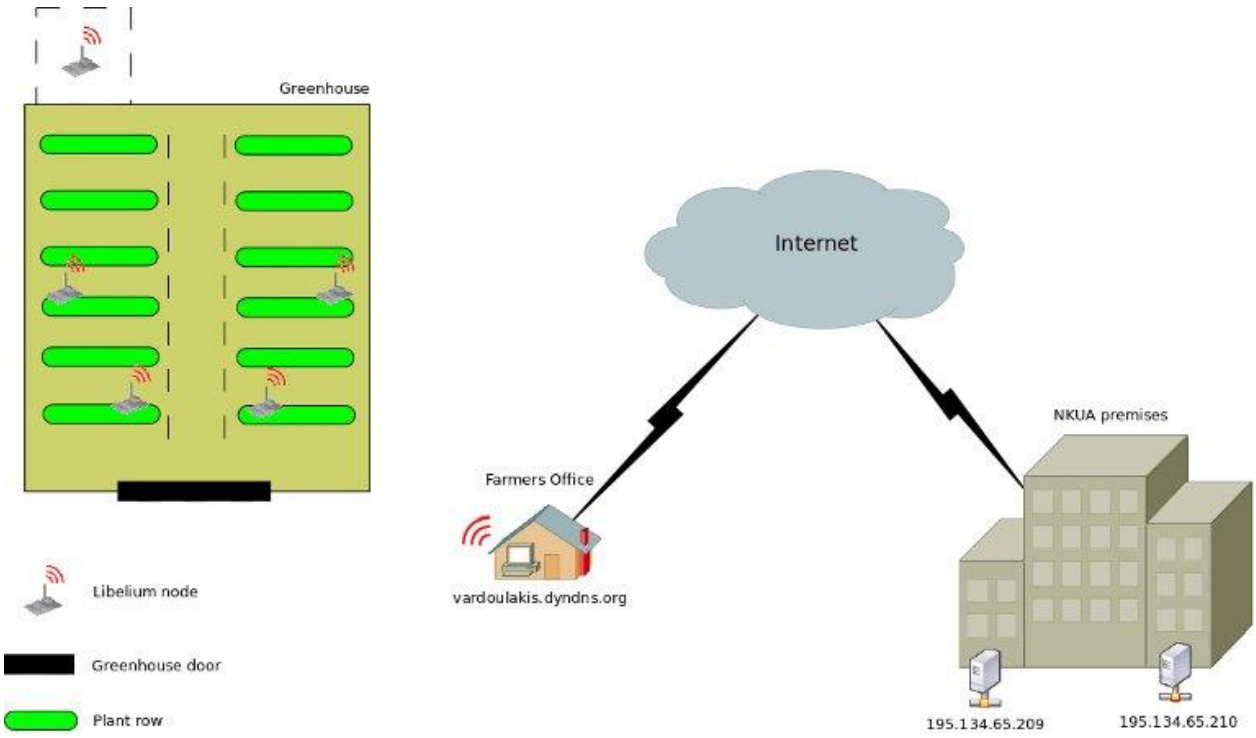


Figure 3-1: Greenhouse Pilot actual deployment

Figure 3-2: The sensors deployed in the greenhouse

Figure 3-3: The Greenhouse

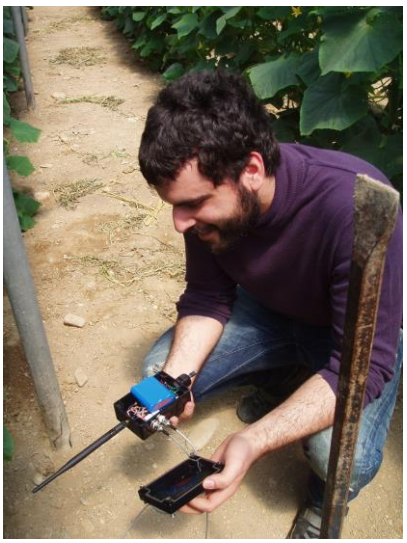


Figure 3-4: Sensors ready to be deployed

Figure 3-5: Sensor deployed in the Greenhouse

Figure 3-6: Sensors with solar panels

Figure 3-7: Sensors getting updated

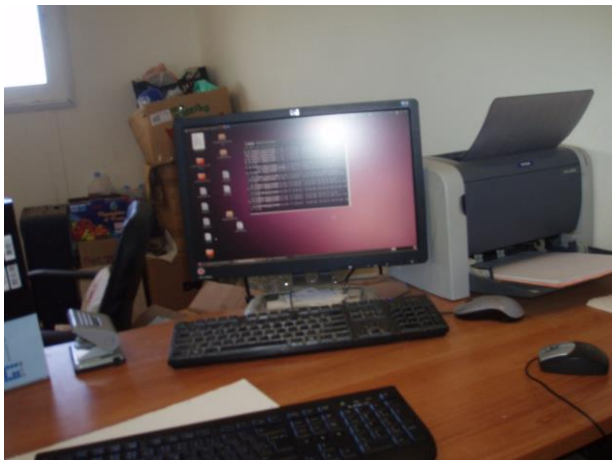


Figure 3-8: A low end commodity PC running Ubuntu Linux serves as the Greenhouse's cloud proxy

The greenhouse is approximately 10,000 m^2 big, having an almost rectangular shape. The deployed nodes are equipped with 3 soil moisture, 3 temperature, 3 relative humidity, 1 CO2 and 1 PH sensors. There is also a node outside the greenhouse equipped with a temperature sensor. Inside the farmer's office there is a low-end commodity PC while in NKUAs premises two high-end servers are deployed. Figure 3-2 to Figure 3-8 provide snapshots of the greenhouse, the sensors, the deployment process as well as the actual control environment in the Greenhouse in Crete.

3.2 Domain Specific Enablers of the Greenhouse Pilot

In the context of the Greenhouse Pilot the following domain specific enablers have been implemented:

- Configuration Communication of FMS controller.
- Data Collector of FMS controller.
- Data Analyzer of FMS controller.
- Statistical Analyzer of FMS controller.
- Coordination module of FMS controller.
- Execution module of FMS controller.
- Notifier module of FMS controller.

Detailed information regarding their implementation, APIs and deployment is provided in the subsequent paragraphs of this section.

3.3 Related FI-WARE's GEs to the Greenhouse Pilot

As far as the real implementation of Greenhouse pilot is concerned, within phase I, we have identified the degree of importance for integrating each GE in the proof – of – concept. The prioritisation was done according to the MoSCoW [25] prioritization technique and reported in previous deliverables in the context of WP200 and WP500. As discussed previously, in principle, all modules may interact with the FI-WARE's GEs. The consortium was granted access to the GEs implementations on the 1st of October 2012 (beginning of project month 19), and as a consequence, the integration of the prototype with the GEs commenced on M19. The current status is briefly presented in Table 3-1. The reader should take into account that all Generic Enablers that appear in Table 2-1 but are not present in Table 3-1 were not implemented on M21, but may be integrated afterwards, and documented in D500.5.2 [8].

Table 3-1: Usage of FI-WARE GEs from Greenhouse pilot modules

Generic Enabler	**Provided by FI-WARE on M21**	**Exploited by module (c.f. Section 3.4 – further results will be reported in WP500)**
Application Ecosystem and Delivery Framework		
Application Mashup	Yes	It works as a standalone. Integrated in Cloud frontend GUI.
Service Composition	Yes	Exploited by Application Mashup in the background
Light Semantic Composition	Yes	Not used
Marketplace	Yes	FMS Registry
Mediator	Yes	Configuration and Communication
Repository	Yes	FMS Registry.
Cloud Hosting		

Generic Enabler	Provided by FI-WARE on M21	Exploited by module (c.f. Section 3.4 – further results will be reported in WP500)
Object Storage	Yes	Data Collector
IaaS DCRM	Yes	Currently no APIs are exposed. Used DCRM's portal.
IaaS Service Management	Yes	Not used
Internet of Things Service Enablement		
(Backend) Things Management	Yes	Data Collector
(Gateway) Data Handling	Yes	Not used
(Gateway) Protocol Adapter	Yes	Not used
(Gateway) Device Management	Yes	Not used
Data / Context Management Services		
Complex Event Processing GE	Yes	Not used
Publish / Subscribe Broker GE	Yes	Data Collector or Notifier (critical information regarding APIs is missing on M21)
Semantic Annotation GE	Yes	Not used
Big Data Analysis GE	Yes	Not used
Multimedia Analysis Generation GE	Yes	Not used
Query Broker GE	Yes	Not used
Location GE	Yes	Not used
Semantic Application Support GE	Yes	Not used
Interfaces to Network and Devices		
Cloud Edge GE	Yes	Local Configuration Communication.
Security		
Identity Management GE	Yes	Configuration and Communication
Security Monitoring GE	Yes	Not used – Standalone tool

The Service Description Repository enabler is integrated in the smart farming pilot for storage of service descriptions. More specifically, service descriptions, captured in linked-USDL RDF files are uploaded on the repository provided by this GE implementation. Afterwards, information can be extracted from the RDF files in order for the service to be deployed. Thus, there is going to be a connection between the Data Collector module of the FMS controller and the Service Description Repository GE so as to enable Services Information exchange between the Smart Greenhouse Pilot and the GE implementation.

3.4 Validation of the Greenhouse Pilot

Taking into consideration the architecture description provided in the above sections alongside with the software description of Table 3-2 we can show a mapping of classes to specific architecture functional blocks. The latter appears in Table 3-3.

Table 3-2: Main classes of the FMS Controller

Class Name	Description
ObjectHandler	This class is responsible for mapping objects into database table data. In addition it also executes specific queries for these objects.
RestObjectHandler	This class is responsible for providing the REST interfaces that enable access to the ObjectHandler class' functionalities.
ConfCom	This class is encapsulating the functionalities provided by the FMS Configuration Communication module.
Coordination	This class is implementing the FMS coordination module.
DataAnalyzer	This class is implementing the FMS DataAnalyzer's functionalities.
DataCollector	This class is responsible for the FMS Data collector's actions.
Notifier	This class is the implementation of the FMS Notifier module.
StatisticalAnalyzer	This class is implementing the statistical functions of the FMS Statistical analyzer module.
FmsMngtFunctions	This class is including the functionality for the FMS Management.
FmsRegistry	This class is implementing the functional procedures of the FMS Enterprise Registry.
Workflow	This class is implementing the Workflow Controller which enables access of the user to the FMS.
Execution	This is the implementation of the Execution module.
ShiroManager	Class in charge of security issues utilized by Apache Shiro framework.

In principle, Table 3-2 presents the classes implementing the FMS controller and their associated functionalities while Table 3-3 maps these classes to the architecture specifications of Section 2. Additional information regarding the classes and the internal software design will be reported in D500.5.2.

Table 3-3: Validation mapping

Class	Architecture functional module
ConfCom	Configuration Communication of FMS controller.
DataCollector	Data Collector of FMS controller.
DataAnalyzer	Data Analyzer of FMS controller.
Statistical	Statistical Analyzer of FMS controller.
Coordination	Coordination module of FMS controller.
Execution	Execution module of FMS controller.
Notifier	Notifier module of FMS controller.
Workflow	Workflow Controller.
FMSRegistry	FMS enterprise registry.

Information regarding the validity of the implementation with respect to the scenarios identified in previous deliverables appears in Section 9.1.

3.5 Greenhouse Pilot and Standardization

The Greenhouse Pilot employs an excerpt of the agroXML [9], SensorML [10] and Observations and Measurements [11] standards. However, the implementation effort in conjunction with the envisaged novelties has indicated the need for extensions. These extensions could possibly be accommodated as future additions to the standard.

For example, in the case of FarmManager, Contact, Person and other employee relevant elements in the agroXML, there is a reference to employeeId but there is no information regarding his speciality; this could be farmer, agriculturist, spray contractor etc. Such kind of information would be useful, and can be accommodated with an additional element describing the type of the employee. Similarly in the case of machine specifications there could be some information regarding interconnections with other machines or devices like protocols capabilities (e.g ISOBUS).

Another type of information that agroXML is lacking and became apparent during the development of the pilot is related to multimedia content (e.g. photos from the greenhouse, videos etc). Towards this end, elements describing multimedia type, encoder, resolution, container, location - both geographic and internet (e.g. where was the photo taken and where it is stored) - should be accommodated. These elements while facilitate the provision of visual information regarding the farm.

Finally, information concerning food-chain and product traceability is also missing. Following the project's vision of supporting traceability from farm to fork, information like the Serial Shipping Container Code (SSCC) of the container used for the products, packaging material that keeps the product protected, the water source used for the plants or the Global Location Number (GLN) should be incorporated. This data will facilitate end-to-end information availability, from the grower to the consumer.

A novel aspect of the developed system that could be standardized is related to the API exposed by the Service Bus (ESB) and enables communication with the FMS controller. An indicative API for standardization stemming from the design and implementation of the Greenhouse Pilot appears in the appendix (Section 9.2, Table 9-1).

A need for standardization also arises from the communication between the FMS and the GEs. The gaps that have been identified are listed in Table 3-4. This list is restricted to domain specific data. Any format might be used for calendar dates, IDs, etc.

Table 3-4: Gaps in standardisation

	Gap	Possible solution	Comment
Communication with the Certification Service			
	Type of certificate		
	Certification authority		
Product Information Service			
	Field of plot of harvest	IACS field identifier	
	Commodity, variety	CPVO Identifier	lacks any in-depth description
	Link to additional crop information (pointers to FMS)	agroXML	
	Cause of exception		
Business Relations Service			
	Participant_type		
	Offering		
	Category		
	Type of business service		Core component

This table shows that several gaps exist where no standard seems to be available, or where existing standards need extension. An important issue is to identify the types of available business services. These are domain specific e.g. spraying contracts. The definition of the service semantics is a task for the next phase of the project. However, the standardisation of IT related services as needed for the Identification Service is a generic task which must also be performed later.

4 Spraying Pilot Specification

The Smart Spraying pilot demonstrates advanced functionalities of a future farm management infrastructure that support the planning, execution and documentation of plant protection spraying operations in potato production. The functionalities enable disease alerting for the setup of a sprayer, spraying, documenting operations, remote monitoring of spraying operations and decision support for handling sprayer machine breakdown. The functional logic is shown in the figure below.

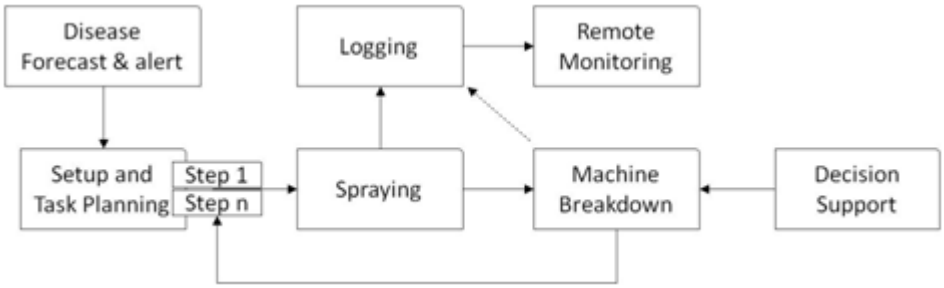


Figure 4-1: Functional logic of the Smart Spraying Pilot

The main aim of the spraying pilot was to develop a service framework that enables the implementation of an enhanced Farm Management System (FMS) that provides innovative services not only to the farmers but to actors in the food chain for potato production. As shown in Figure 4-1, the pilot demonstrates the optimized use of plant protection agents through decision support services regarding the weather, soil, and machine performance monitoring to predict disease onset, optimize spraying, assist during machine breakdown and document operations for use by stakeholders.

Value proposition of the spraying pilot includes:

- Networking - access to services:
  - Seamless spraying experience
- Provide more high quality food products:
  - avoiding possible crop damages caused by diseases and chemicals
  - avoiding chemical residues in the product
- Documentation of all processes:
  - increase consumer and processing industry trust
  - advertising products information for government and research
- Environmentally friendly production:
  - reduce risk of wasting spraying chemicals and causing environmental emission
  - knowing the demand/need for spraying to justify the spraying
- Improved the resource management:
  - better spatio-temporal work allocation and chemical allocation

Value proposition leads to a set of functional requirements that includes:

- Access to services in the market
  - A service framework enables review and purchasing of available services
- Tailoring of services

  - It enables building meaningful farm service ecology
- Awareness of service events
  - Update and delivery of relevant information generated by services
- Offering services to the market
  - Creation and offering of services
- Cumulative farming experience
  - Enables building meaningful and established practice based on collecting and synthesizing of business and production related data
- Food chain communication
  - Enables food chain transparency through meaningful communication and interaction
- Service synergy
  - Services can exchange and make available meaningful data within service framework

The value proposition as well as the functional requirements will be described in more detail in the final WP200 deliverable, D200.4.

4.1 High level view of the Spraying Pilot

The Smart Spraying Pilot implements a service framework (hereinafter called spraying pilot service framework, service framework or framework) that provides innovative services not only to the farmers but to actors in the whole food chain. The service framework of the Smart Spraying Pilot is hosted on the CropInfra platform [17].

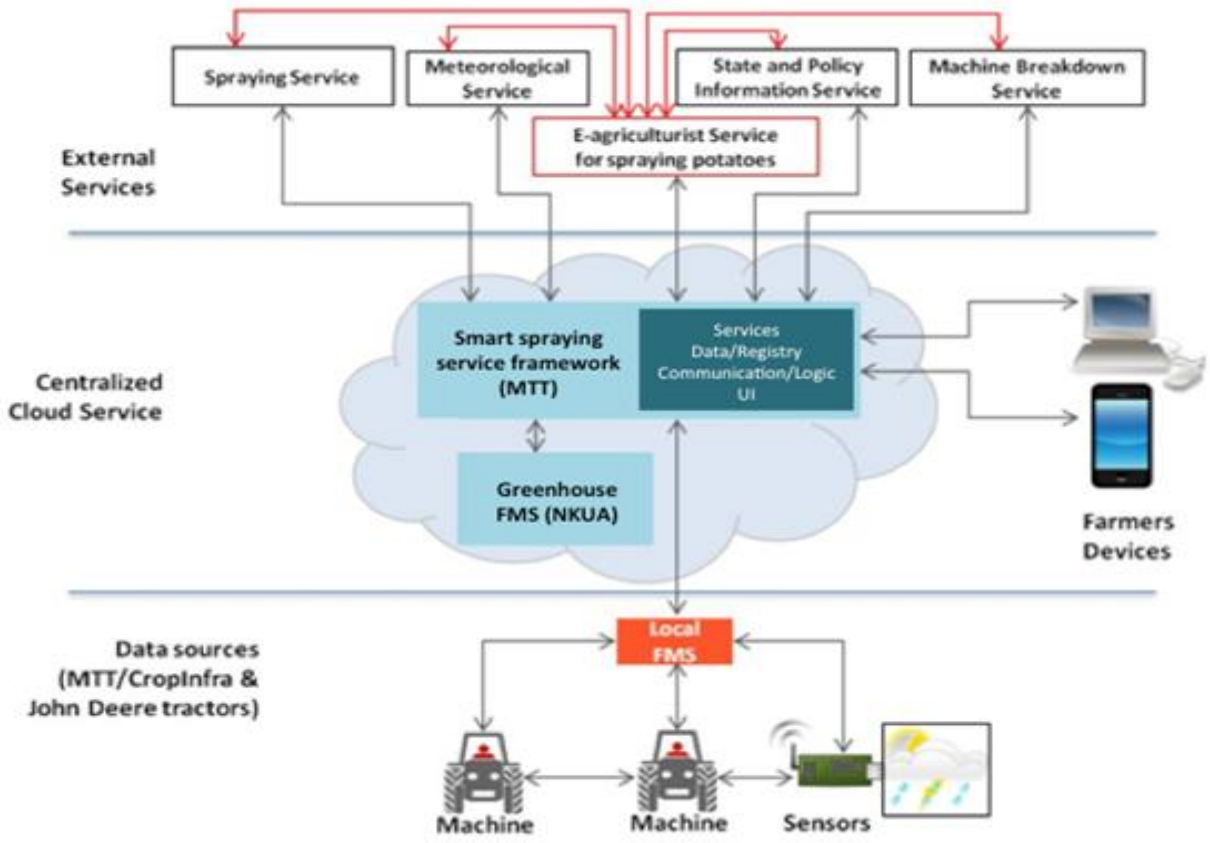


Figure 4-2: Implementation of the Smart Spraying Pilot at MTT, Finland with JD partnership

The Smart Spraying pilot has been also integrated with the greenhouse pilot described in the previous section. More details are provided in Section 5. Figure 4-2 illustrates the implementation and integration.

The functionalities implemented in the Smart Spraying Pilot are divided into two parts; namely, the general service framework functions and the E-agriculturist functions. The architecture and infrastructure of the general service framework functions provides user registration, service registration, log-in, searching of services, subscribing to services, and rating of services and enables information exchange and user interface embedding between registered services. The E-agriculturist functions enable Spraying Setup Functions and Machine Breakdown support.

The spraying setup function supports visualization of weather information from different sensor networks around the farmers' field parcels. This weather information is used to forecast disease warnings (Disease Pressure Service). On reception of the warnings, the farmer uses the FMS tools (Task Planning Service) provided by the cloud to prepare a task plan and schedule for spraying. The setup functions provide possibilities to the farmer to download planned tasks for the sprayer.

The Setup System has also logging (Data Service) functionalities in order to maintain and store the process of the spraying operation. In case there are several work units operating at the same time, real-time remote and fleet monitoring is implemented in the Smart Spraying Pilot. The Machine Breakdown support functions provide information about status of farmers' spraying machines. They provide alerts and solutions for repairing in addition to task rescheduling in case of machine malfunction or breakdown. The main functionality is to accumulate information from machines, labour, crop as well as weather, disease and market information in order to provide recommendations for fixing machines and task replanning so as to enable the completion of an ongoing spraying task. The classification of the functions is given in Figure 4-3.

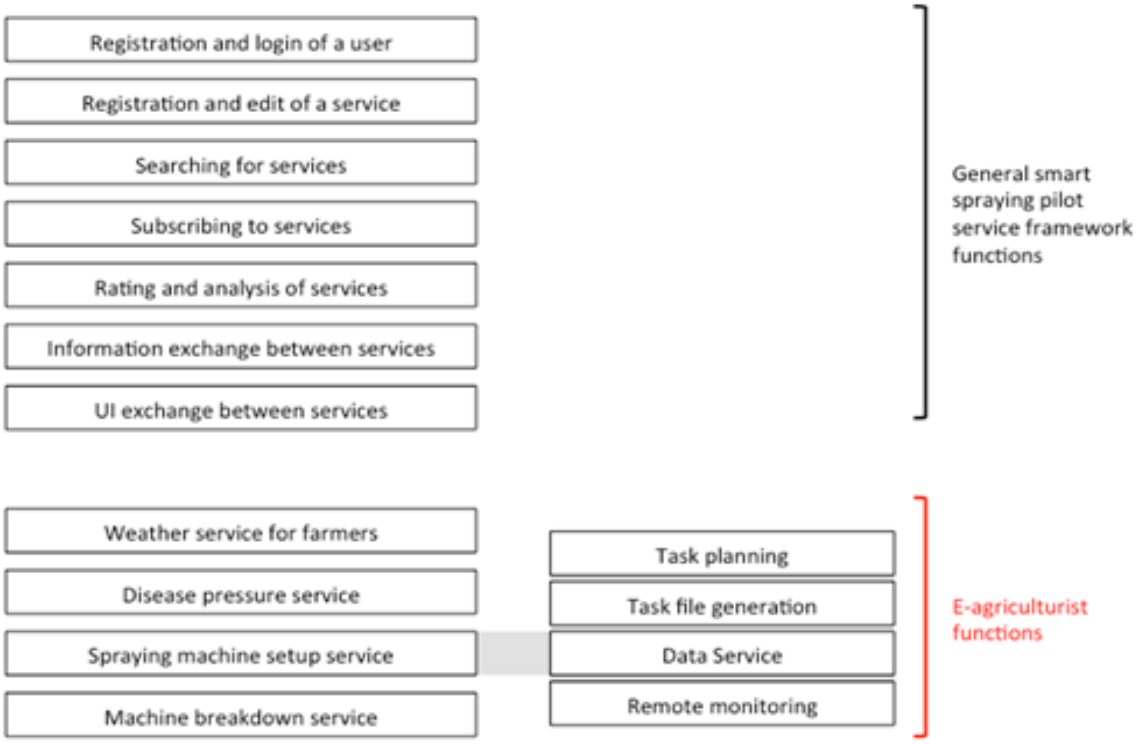


Figure 4-3: Functional components of the Smart Spraying pilots.

Based on the initial analysis of the design, the functional architectural composition is described in detail in the next section, the web based graphical interface for standard web browsers and handheld devices was composed. The current status of development prior to first prototype release is presented here. The Graphical User Interface is implemented on the "CorpInfra platform" of MTT Agrifood Research Finland and is accessible at http://www.cropinfra.com/SAF. The mobile user interface version of the pilot is accessible at http://www.cropinfra.com:8080/framework/framework.jsp?username=demo&password=demopassword&x=0&y=0&pageID=m_verifySignin.

4.2 Domain Specific Enablers of the Spraying Pilot

The service framework developed for the Spraying Pilot supports functionalities such as User Registration, Sign In, Service Registration, Searching and Subscribing to Services and Rating Services. It also supports information exchange and user interface embedding between the registered third party services. The functions per se can be seen as generic and suitable for different domains (the functions essentially comprise an instantiation of the Identification Service from the family of SAF Generic Services – Section 2.3). Additionally the User Registration and Sign In procedures comprise an implementation of the Identification Generic Service as the latter is described in Section 2.3.

4.3 Related FI-WARE's GEs to the Spraying Pilot

Based on the requirements for the conceptual smart spraying service framework two essential GEs were identified: Global Customer Platform IdM (GCP) and Marketplace with SLA management.

GCP handles such IdM functionalities as user registration and login, user service registration and registry related to that, session management and overall administration of registered services including customer care / self care management. When the GCP takes care of the service registration the Marketplace makes it possible for services to be found. The Marketplace includes USDL registry and repository but also SLA management, revenue settlement and sharing system. The spraying pilot service framework aims to implement tight integration with these two GEs.

The GCP IdM provides the core functionality of the identity management of the service framework. On the pilot software implementation phase an IdM based on the ideas addressed on the FI-WARE security chapter on Global Customer Platform was implemented in some extent. The aim of the Spraying Pilot is to replace the local implementation with the one provided by the FI-WARE. One goal in the service framework design has been to easily replace third party services. This goal applies also to integrated services such as the identity management. The functionality is realized by abstracting the underlying application logic implementing REST interfaces also for the internal calls of the implement. Since the first steps of GCP integration has just been taken the information on API is restricted to the specification provided by the FI-WARE mediawiki. The OpenID[23] protocol is used in the FI-WARE provided GCP integration.

The Marketplace integration is based on the two functional requirements derived from the value proposition of the spraying pilot: the Service Framework enables review and purchasing of available services (access to services in the market) and enables building meaningful farm service ecology (tailoring of services). On the consumer side the marketplace makes it possible to search and discover services and offerings according to specific consumer requirements as

well as functions like reviews, rating and recommendation. The integrated marketplace view may be pre-filtered to offer only the relevant services and offerings based on the context of the marketplace implement for registration and licensing.

The spraying use case contextual service framework implements the Marketplace Search RESTful Web API for finding the offerings from the marketplace. The requests for search for offerings are listed in Table 4-1.

Table 4-1: API Operations as defined in FI-WARE mediawiki

Verb	URI	Additional Path Parameters	Description
GET	/search/offerings/fulltext/{searchString}	filter, index, limit, sortBy, order, minScore	search for offerings where the services description matches the specified search string
GET	/search/offerings/filteroptions		returns a list of possible filter options for offering search
Example: **http://[SERVER_URL]?filter=name:Simth*&index=20&limit=10&order=asc&sortBy=name,storeName&minScore=0.75**			

As part of spraying pilot relating e-agriculturist service research the use of some of the GEs introduced in the FI-WARE chapter Internet of Things (IoT) Services enablement was investigated. The target of the study was to enable automatic discovery and utilization of location aware local weather stations and weather station networks. In the spraying pilot prototype a farmer has two individual weather stations registered into the GCP IdM. When the service framework is accessed with the same credentials or the ones within the trusted credential pool the weather stations become visible in the globally registered services enabling the use of the own local weather data with any weather data dependent framework registered third party service. In addition, when an IoT compatible weather station or weather station network is registered as an offering into the marketplace the owner of the weather station can easily sell the weather data to other actors like weather service providers or neighboring farmers.

When we think about the spraying event itself the wind speed data of the IoT compatible weather stations can be used to adjust the sprayer nozzle for the best spraying outcome. With the location awareness the sprayer can always connect to the nearest weather station or the possible third party assisting service can make suggestions to the nozzle controlling system based on the location of the sprayer and the nearest weather station.

4.4 Validation of the Spraying Pilot

As much as possible, the functionalities described in section 2.4.4 of D200.2 are implemented. The general service framework functionalities provides for the following:

- Registration of user: Account creation and Login of a user.
- Registration of service: Classification of Service, Upload Service Files, Validate of service.
- Searching for services: General Search, Categories.
- Subscribing to services: Viewing Rating and details, Subscribing and unsubscribing.
- Rating of services: Star rating, Feedback provision.
- Information exchange: Automatic information exchange between framework registered third party services

- UI Exchange and Embedding: User interface exchange and embedding mechanism for enhanced scalability

The idea behind the spraying pilot service framework is to enable a developer or a user to freely select and replace the services to be used within the framework. This applies also for the underlying functionality of the FMS Controller. As described in earlier chapters, the service framework has build-in capability to utilize the underlying FMS Controller functionalities by using the RESTful Web APIs. The spraying pilot service framework treats the FMS Controller APIs as third party services and uses the same mechanism for discovery and registration of them as it uses for any other service. In this case, a user can decide to use the whole offering of services provided by the FMS Controller or use a different service provider for example for statistical analysis or for notifications. The current status of the spraying pilot's usage of the FMS Controller sub modules is presented in Table 4-2.

Table 4-2: Validation mapping

Class	Architecture functional module	Coupled
restRequestHandler implementing frameworkApplicationLogic	Configuration Communication of FMS controller.	No
	Data Collector of FMS controller.	No
	Data Analyzer of FMS controller.	No
	Statistical Analyzer of FMS controller.	Yes
	Coordination module of FMS controller.	No
	Execution module of FMS controller.	No
	Notifier module of FMS controller.	No
	FMS enterprise registry.	No

Additional information regarding the validity of the implementation with respect to the scenarios identified in previous deliverables appears in Section 9.1. The Smart Spraying service framework prototype was introduced to end-users in KoneAgria 2012 trade fair ([18]). The end-user response and comments will be reported in D200.4.

4.5 Spraying Pilot and Standardization

In the spraying pilot the agroXML and the ISOBUS XML standards are deployed to some extent. The agroXML is used in farm data related data exchange and the schema is mirrored to the underlying database structure when applicable. The ISOBUS XML is used conceptually in task file exchange between the sprayer setup service and the working unit.

The spraying pilot views on the agroXML are mostly convergent with the ideas introduced in Section 3.5. However in addition the weather station XML description needs to be addressed. The description has the elements for common weather data exchange attributes but seems to be lacking two relevant ones: the information on wind direction and air pressure.

The wind direction has relevance for example in a spraying event when the change of potentially dangerous fallout to the neighbouring parcel should be taken into account. It may also assist the driver in the spraying line decisions. In the disease pressure models the information on wind direction can be used to estimate the possible infecting spore spreading direction when applicable.

The air pressure can be used for example to predict the changes in weather type. This information becomes relevant in the scenario where the information provided by a weather station or weather station network is sold to and used by a third party weather forecast service.

5 Integration of Greenhouse Pilot and Smart Spraying Pilot

The Greenhouse pilot in its current form implements the functionalities identified by the requirements described in D500.3 [6]. In particular, the system supports constant monitoring of the conditions that vegetables are grown (e.g. humidity, luminosity, soil ph ...) while in parallel is able to store information, transmit it to the cloud and exploit the added value of a basic set of services. Future work in the Greenhouse pilot will focus on harmonizing the interfaces and the interactions of the system with the generic services of D500.3.

A preliminary version of integration is reported in the following. Any further results will be reported in WP500. Also, the integration with other EC projects has been investigated, specifically with Aspire and ENVIROFI. Further details are omitted in this section and are provided in the appendix (Sections 6.2 and 6.3). In Section 6.4 the integration with the AgroSense FMIS is also provided.

5.1 High Level Scenario

The spraying and the greenhouse pilots will be eventually integrated. Two integration options have been identified; integrate with Greenhouse Statistical Analyzer, Notifier and Data Analyzer or core integration in the GUI. The first option will be pursued since it is easier in terms of integration, demonstration and ease of implementation. The second option will be investigated as soon as concrete results have been extracted from the first one. The supported storyline follows:

- A service is deployed
- The service provides some weather/soil information to the agriculturist
- The service starts reporting abnormal values
- The Statistical Analyzer and the Data Analyzer identify the problem.
- The Notifier issues a notification to the relevant farmer

A key requirement of this scenario is the availability of a large set of values. This set will ensure that we can properly identify what operational status is normal and what is abnormal. Abnormal values can be identified by means of standard deviation. In principle we can assume that the normal set of values X follows a normal distributions, thus all values appearing outside m(X) +/- c*std(X) (e.g. c=1). A more sophisticated approach could entail the approximation of the distribution based on the available samples and the dynamic definition of the c factor.

5.2 Message Exchanges

The natural integration point for the spraying pilot and the greenhouse pilot is the FMS Controller. The spraying pilot can utilize underlying FMS Controller functionalities like the Statistical Analyzer, the Notifier and the Data Analyzer by using RESTful Web APIs.

In this exercise, the weather station backend, one of the third party services in the spraying pilot, communicates with the statistical analyzer to determine if the measurements coming from the temperature sensors are reliable or not (Figure 5-1). The weather station backend implements a data handling functionality with a local repository for storing historical information on the measurements of the sensors in the weather station. The 24 hour sample of temperature measurements is used for analysis. The weather station sends the temperature measurements every 15 minutes to the backend which leads us to total 95 values and one reference value per analysis cycle. A 48 hour sample (191 measurement values) was tested but the test result analysis indicated that the deviation on temperature measurements became too even for making reliable conclusion on the health of the sensor.

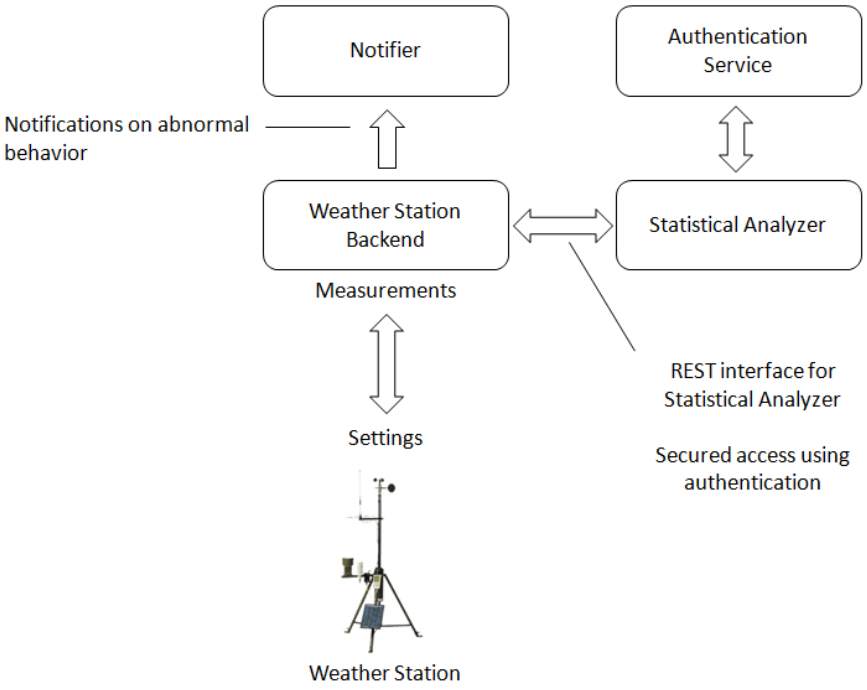


Figure 5-1: Actors in the abnormal temperature sensor behavior detection scenario

The communication between the weather backend and the statistical analyzer is secured by using an authentication service for session management (Figure 5-2). After successful session opening the statistical analyzer API becomes accessible. The Statistical Analyzer API is presented in more details in D500.5.2.

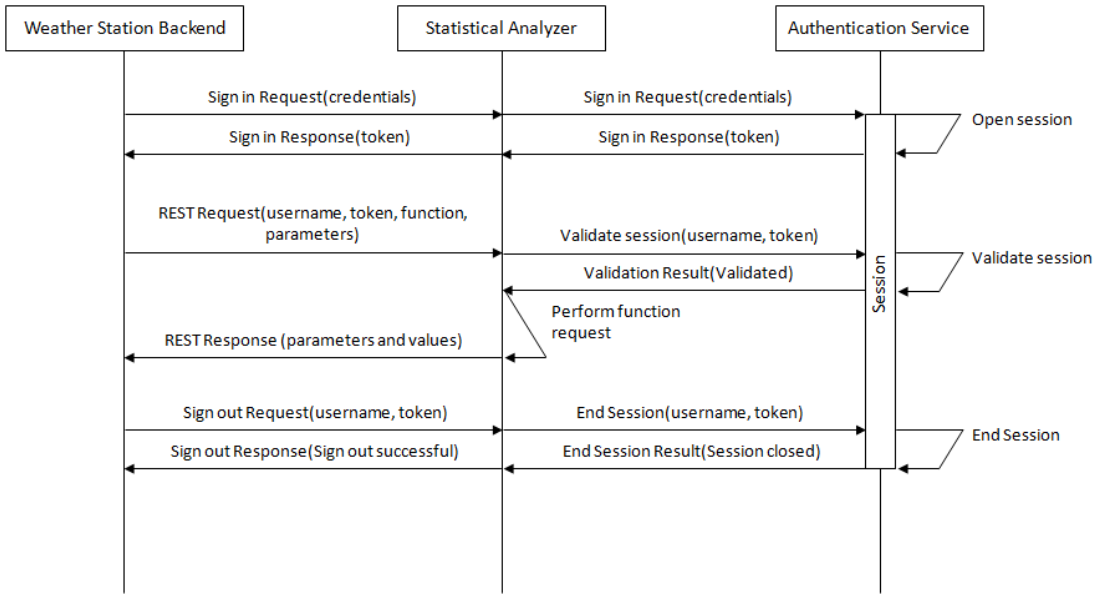


Figure 5-2: Statistical Analyzer access message sequence chart

The weather station backend requests for deviation and average for the 24 hour measurements. The decision on the validity of the latest measurement is done internally by the backend and is based on three-step score of expected range of the latest measurement. The weather station statistical report (Figure 5-3) can be accessed at http://www.cropinfra.com/saf/integration.

Statistical Analysis of the Hovi 6 SW36 (Vihti, Finland) Weather Station Temperature Sensor

Location	WGS84LAT 60.4246, WGS84LON 24.3729
Statistical Analyzer	Successfully connected to NKUA Statistical Analyzer RESTfull Web API
Latest measurement time	Today at 16:00
Latest measurement value	-12.9
Previous measurement time	Latest - 15 minutes
Previous measurement value	-13.2
Direction	Up
24h average	-17.78
24h deviation	3.4

Got -12.9
Expexted range based on average and deviation from -21.18 to -14.38 -> Latest measurement is higher
Expected range based on previous measurement and deviation -16.6 to -9.79 -> Latest measurement is in range
Expected range based on previous measurement +/- 5 -18.2 to -8.19 -> Latest measurement is in range

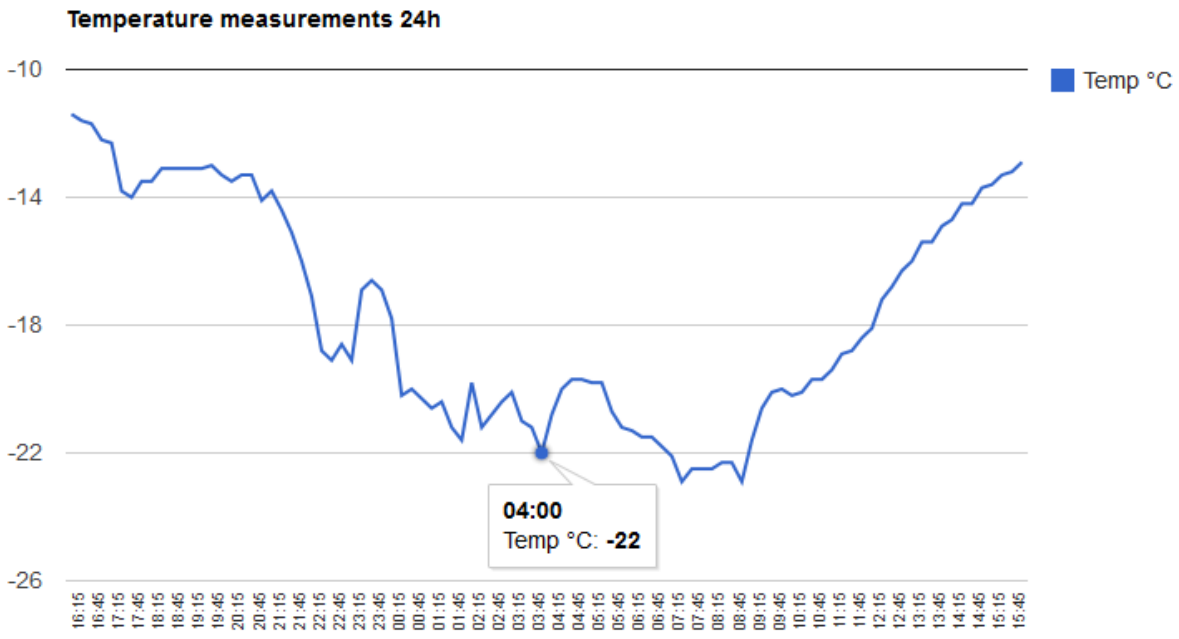


Figure 5-3: Weather station temperature sensor statistical report

6 Integration with other projects and external systems

In the final section of the document we present additional material, EC projects and external systems that have been, or will, be integrated in the overall environment. We start with the Expert System which is used in the pilots at the time of this writing and continue with the integration possibilities with the ASPIRE and the ENVIROFI project. The section is concluded with the description of the integration opportunity with AgroSense.

6.1 Expert System

In order to facilitate the identification of problematic situations in the context of the pilots described in the main part of the document, an expert system has been designed and implemented. The functional requirements of such a system were presented in D200.1 [2] while its design and implementation in D200.2 [4].

Here we briefly present the system in order to highlight the extensibility of our work. We initially provide the mathematical formulation and subsequently the input used to populate the knowledge base and identify the solutions to the problems.

6.1.1 Mathematic Formulation

The Expert System enables the transition from the continuous domain (i.e. values in R) to nominal (e.g. values categorized as High/Normal/Low). The latter is achieved with the use of a sigmoid mapping of the real-valued input data:

0	:=	low
0.5	:=	normal
1	:=	high

The sigmoid maps any value in-between the interval (0…1), thus a value of e.g. 0.6 indicates a state between “normal” and “high”, with trend to be “normal”. For each input value, an “expected range” is specified with a lower threshold and a higher threshold, e.g. for the temperature the normal range could be [22°C … 29°C] (e.g. Figure 6-1.

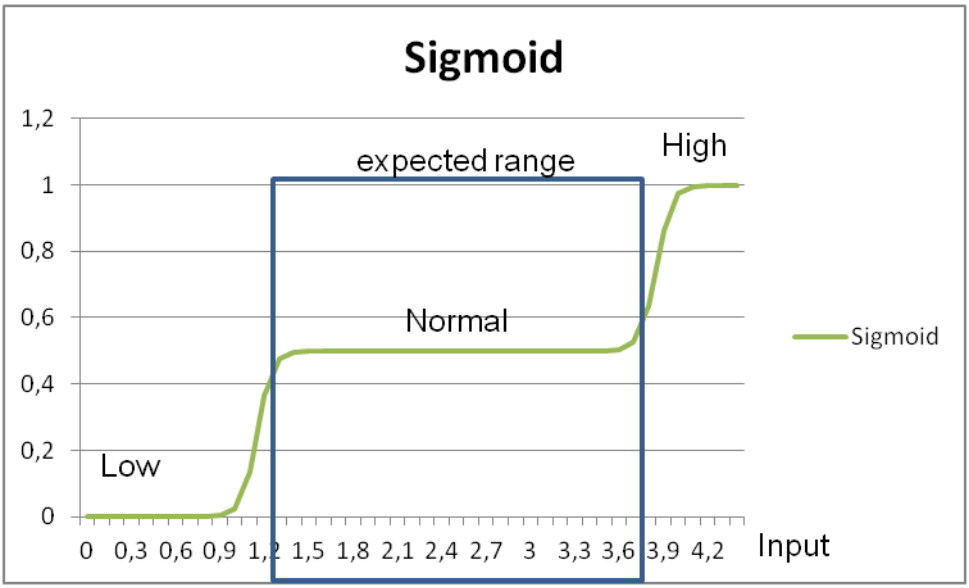


Figure 6-1: Ranges

Thus, a ranking of best-matching rules can be achieved. These rules are provided in the next section.

6.1.2 Rules and Actions

In the context of this section we will present the information which was provided by a professional agriculturist and was used as input to the expert system. The presented rules and actions are about tomatoes cultivation. Similar data is available for cucumbers as well.

At first, a categorization of the various parameters was done according to Table 6-1.

Table 6-1: Categorization of values according to expert's advice

	High	Normal	Low
Temperature	*33°C*	*between*	*10°C*
Luminosity	*40 klux*	*between*	*5 klux*
Air Humidity	*0.8*	*between*	*0.5*
PH	*7*	*between*	*5*
EC	*3,5 ds/m*	*between*	*1,5 ds/m*
Soil Moisture	*90% of water capacity*	*between*	*60% of water capacity*
CO_2	*1000 ppm*	*between*	*200 ppm*

The knowledge base was then constructed according to a large number of combinations. In principle, the attempt is to identify all possible value combinations (scenarios) and link them to the available set of solutions. As one can see in Table 6-2 a problem can be fixed with a multitude of solutions.

Table 6-2: Scenarios and link to the solutions

Luminosity	Air Humidity	Soil Moisture	**Solution**
H	H	H	*7,2,3*
H	H	L	*7,2,3,15*
H	L	H	*7,9,14,4,17*
H	L	L	*7,15,9,4,17*
H	H	H	*Complicated scenario*
H	H	L	*Complicated scenario*
L	L	H	*6,9,4,18,17,14*
L	L	L	*15,9,4,6,8,18,17*
L	H	H	*14,2,3,18,6*
L	H	L	*14,2,3,18,6,15*
L	L	H	*Complicated scenario*
L	L	L	*Complicated scenario*

The solutions (identified by numbers in the previous table) are coming from a finite set of possible remedy actions which appears in Table 6-3.

Table 6-3: Solutions and alerts related to the tomatoes cultivation expert system

ID	Actions
1	Control sensors
2	Open the windows
3	Start the ventilation system
4	Close the windows
5	Start the heating system until the interior temperature becomes 21^{o}C
6	Close the shade curtains (we increase the luminosity)
7	Open the shade curtains (we decrease the luminosity)
8	Enhance the atmosphere with CO2 until the value 1000ppm
9	Start the water sparying until the air humidity be 70%
10	Irrigation with solution-fertilizer 80cc H3PO4 / m3 of water
11	Irrigation with solution-fertilizer CaO 12% 150 ml/m3 of water
12	Flusing with rainwater 8 m3 / arce two times
13	Irrigation with nutrient Solution depending the growth stage e.g N 200mg/lit, P 30mg/lit, K 200mg/l Mg ,30mg/lit Ca ,200mg/lit, μέχρι EC 2,5dc/m
14	Start the heating system until temperature 32^{o}C
15	Irrigation with water until the maximum level that the development phase accept eg 70% of the water capacity
16	Irrigation with nutrient solution depending on the development phase of the tomato eg N 200mg/lit, P 30mg/lit, K 200mg / l Mg, 30mg/lit Ca, 200mg/lit, by EC 2,5 dc / m, = 0.015 NH4/TotalN
17	Close the Ventilation system
18	Start the light - supplementary system
19	Irrigate with rain water, 4 m^3 / acre three times
20	Alert: dangerous for the human health. Do Not enter to the greenhouse

6.2 Integration with ASPIRE Project

Within the Aspire project [12], a sophisticated, royalty-free, RFID middleware was developed, with the intention of boosting RFID usage by European SMEs. After investigating possible integration scenarios, we concluded that Aspire functionalities could be used within our architecture, specifically within the Cloud Proxy, located inside the greenhouse, facilitating the deployment of a low-cost RFID solution.

The modules that are going to be used are the Reader Core Proxy and the Filtering & Collection Server. Their location in the Aspire architecture is depicted in Figure 6-2.

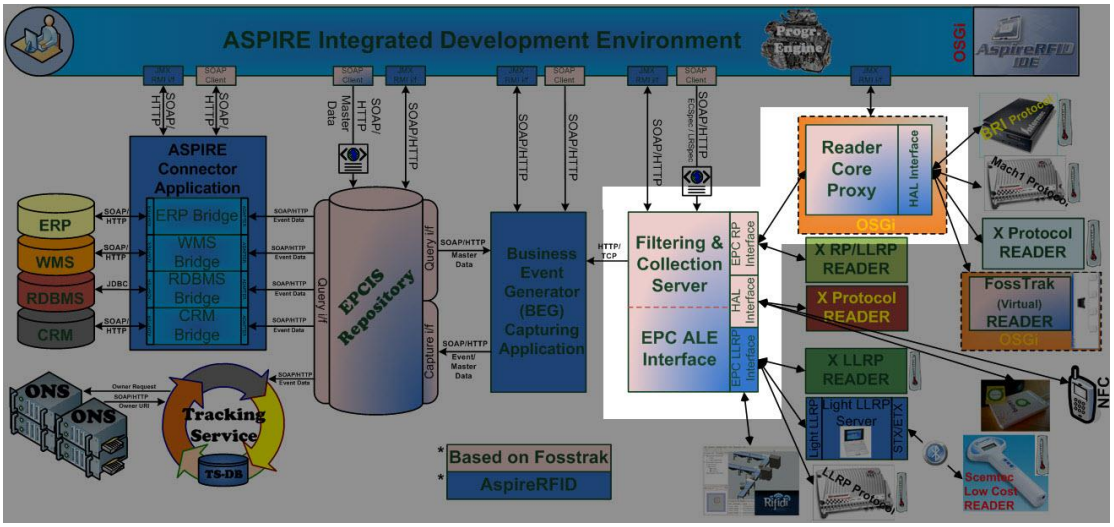


Figure 6-2: The ASPIRE architecture

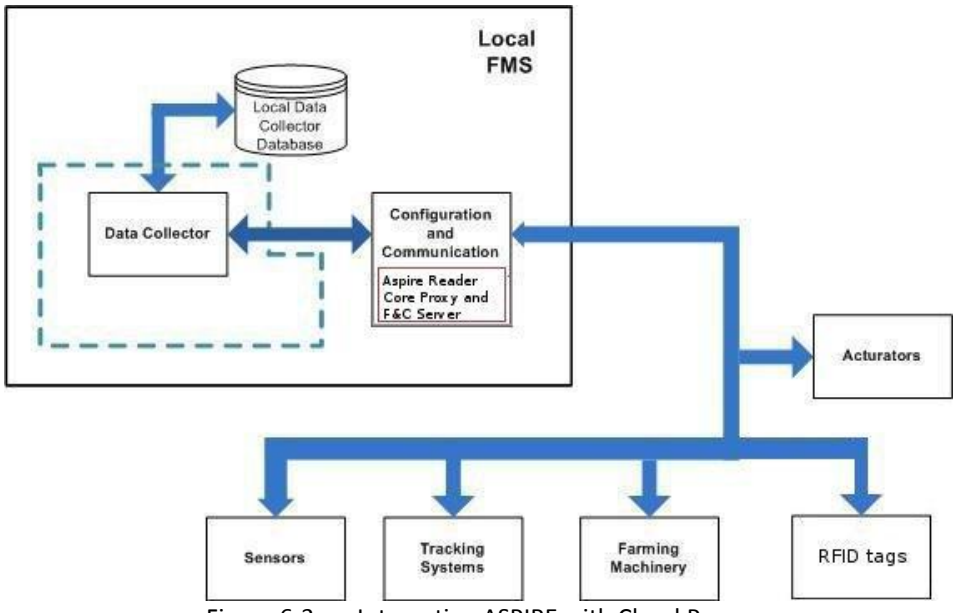


Figure 6-3: Integrating ASPIRE with Cloud Proxy

Aspire supports readers that implement Reader Protocol (RP) [13] and Low Level Reader Protocol (LLRP [14]) directly, while readers using vendor-dependent protocols are supported with the introduction of a Hardware Abstraction Layer (HAL) and the Reader Core Proxy (RCP) application which acts as a mediator between non- RP-LLRP compliant readers and the Filtering & Collection Server. The F&C Server is responsible for filtering out non-actionable reads from the environment and providing operations regarding the aggregation and counting of reads.

The F&C Server along with the Reader Core Proxy can be integrated inside our Cloud Proxy, and specifically, in the Configuration & Communication Module as depicted in Figure 6-3.

As RFID tag read events are produced, they are captured by RCP and forwarded to the F&C Server. From there, they are sent to the Cloud FMS Controller in order to be stored and furthered processed, using F&C Server's HTTP/TCP interface. From a technical point of view, necessary steps must be taken in order for the integration to be realized, the most important one being the development of a HAL adaptor so our RFID reader can communicate with the RCP. Furthermore, there is a need for deploying the OSGi platform [15] and Apache Tomcat, as well as the configuration of the aforementioned modules.

6.3 Integration with ENVIROFI project

Following the reviewers' suggestions, the potential integration with Envirofi was investigated. The outcome of these discussions was that Envirofi environment can be integrated with the Greenhouse and the Smart Spraying pilots of WP200. In fact, various scenarios were proposed in order to facilitate the task, while the potential of an actual integration will also be investigated. The scenarios (starting from the simplest one) are presented in the list below:

- Provide the geolocation of a farm and then download a map and related information from Envirofi platform.
- Inter-platform information exchange (e.g. send pollution data of an area to a farmer and vice versa meaning what fertilizer and in what quantity a farmer has used)
- Integration of Envirofi as a service with UI embedding

The envisaged integration is depicted in the following figure (commonly agreed between SAF and Envirofi):

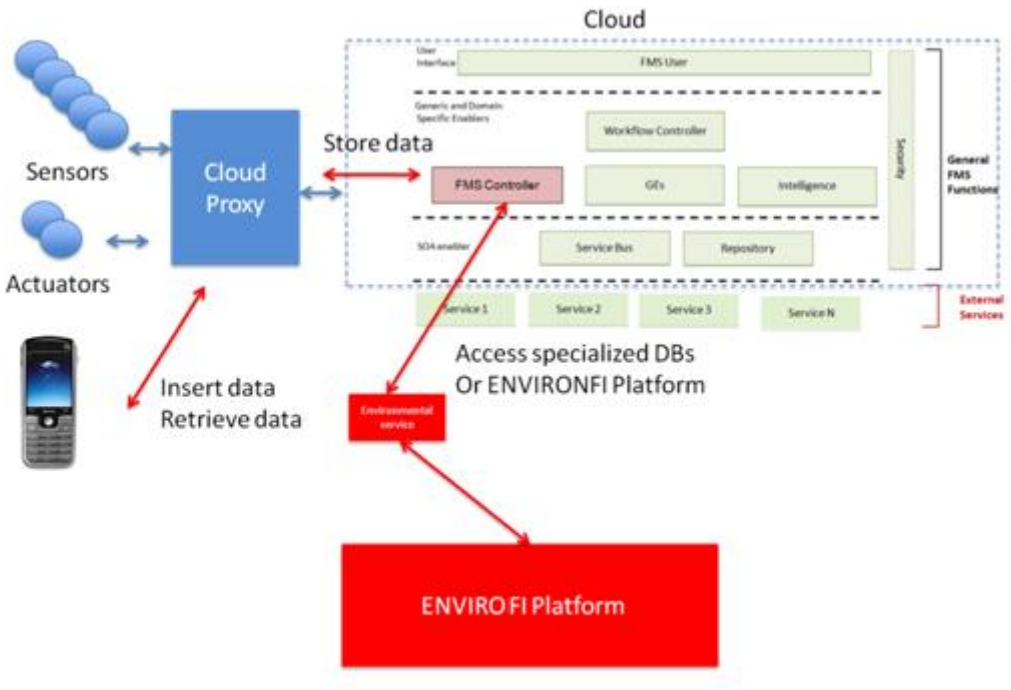


Figure 6-4: Conceptual integration of Envirofi and SAF

6.4 Integrating FMS Controller with other tools

During the International Smart Agrimatics 2012 conference [20] where WP200 work was presented, the SAF team had the chance to work with people from the AgroSense Project [19]. The focus of the work was to examine if and how WP200 architecture can interoperate other tools or integrate with them. This is an important aspect since a fundamental prerequisite of the architecture is to enable interworking with legacy systems as well as currently developing tools. Agrosense is an open source FMIS tool that is currently being sponsored by Ordina [21] and is being rapidly developed.

AgroSense consists of two parts, the client and the server. The AgroSense desktop client is developed as open source software and a free functional version is available. It already contains modules supporting three mobile sensors CropCircle, Greenseeker and Fritzmeier and is capable of processing rough sensor data and producing meaningful information by linking it to a CropField. The SAF team worked after the conference together with the AgroSense team to produce some meaningful scenarios on how these tools can interoperate. The result of this work has been used in the submitted proposal called cSpace that is planned to be the second phase of the SAF project. More details of this integrated work follow in the subsequent paragraphs.

Agrosense's currently supported functionality appears in the following list:

- importing mobile sensor data (CropCircle, Greenseeker and Fritzmeier)
- import ESRI shape files containing either treatment zones or fields
- convert treatment zones to different tasks (spray, fertilize etc)
- export tasks to working ISOBUS task files
- create production units and crop fields
- draw new fields on the map
- split fields or crop fields in two or more parts

Active development, that will be available before April 2014, appears in the following:

- Basic crop planning templates (plant, spray, harvest etc)
- Default flow for obtaining data from a data service
- Default flow for sending data to and receiving data from an expert system
- AgroSense Server component.

The AgroSense server is capable of synchronizing data between the server and multiple clients. The data connection is secured by encryption with unique keys for the server and all client instances. If needed, multiple server instances can be synchronized as well. Furthermore the AgroSense server exposes a public API; when an application (with its own unique key) is authorized by the AgroSense user for a certain subset of data, it may use the public API to retrieve it. Authorization is done by the standard Oauth [22] protocol, authentication by the openID [23] protocol.

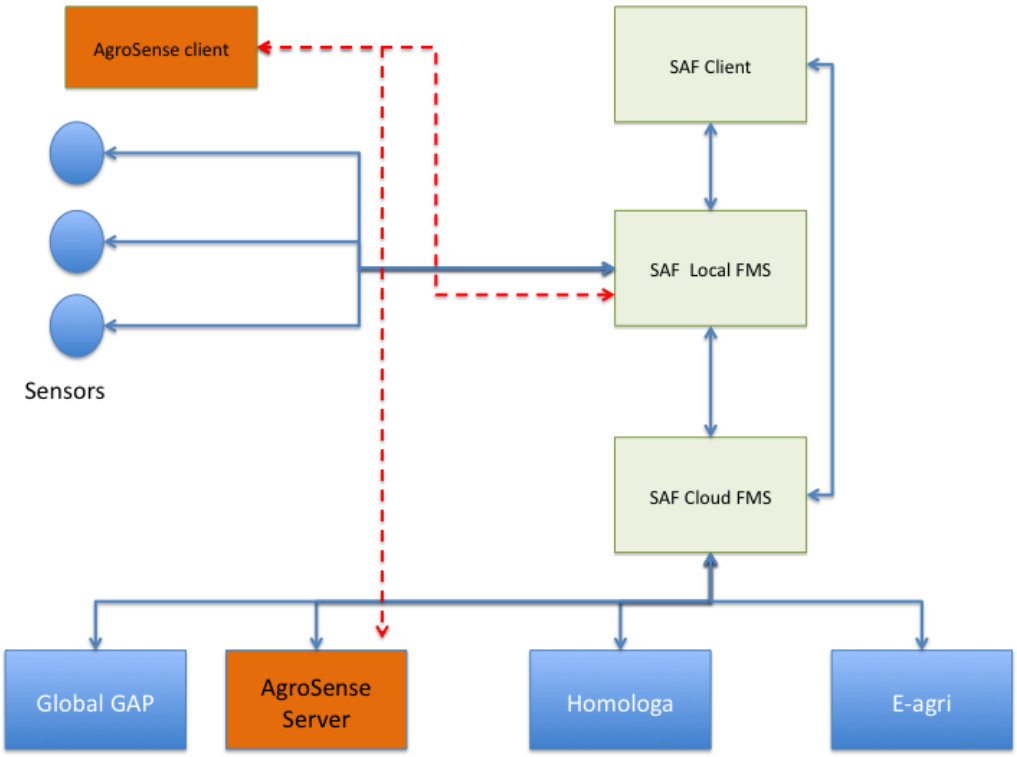


Figure 6-5: Interworking between SAF's FMS and Agrosense

A possible Interworking scenario:

1. In the greenhouse a local sensor net is present, measuring environment variables like temperature, humidity, luminosity etc

2. This real time collected sensor data needs to be stored on a local hub. The role of the local hub is implemented by the local FMS. The local AgroSense client has a configurable module that is able to retrieve the data from the hub. Additionally, the AgroSense client is equipped with mobile sensors like the CropCircle and can produce sensor data for the open fields. Rough sensor data has no meaning. This data can become information, by providing the context. AgroSense is capable of providing context to the sensor data because it can link the GPS coordinate of the sensor to the crop/greenhouse etc. AgroSense publishes the sensor information (with context) through an API. When properly authorized (OAuth key exchange) another application can retrieve this data in a standardized message (EDITeelt 4.0).

3. The AgroSense server is fully synchronized with AgroSense client. The AgroSense server exposes data through an API with standardized message (EDITeelt 4.0), authorized with OAuth key exchange.

4. The AgroSense API could be considered as the interface for the SAF's cloud FMS since it could implement the same standard authorization mechanism (which is already used by twitter, facebook and many other applications) and the same standardized messages (EDITeelt 4.0). Thus, the cloud FMS can communicate with AgroSense to retrieve information if this is required and forward it to other services (e.g., an expert system)

5. Expert systems enrich standard business process implementations; e.g. the business process for crop protection on potatoes. In the standard implementation, the farmer can choose a time for his spray task, choose an agent assign a worker etc. When he chooses, a farmer can use an expert system to determine the best moment and/or agent to protect his crop. Expert systems like the phytophthora model can use distributed calculation generic enablers, or data storage GE's for example to run/store their models. Additional services can be used to acquire data

needed to feed expert systems (like weather data)

6. The Cloud FMS can communicate with all services (e.g., AgroSense, expert system the Homologa DB [24], weather stations etc) and forward this information to the farmer. It can also forward this information to the AgroSense Server. This Server can synchronize this information with its client.

7. The AgroSense client can be used to produce ISOBUS task data xml files to have detailed control over what amount of agent is applied where. Also a module in the AgroSense client can be used to send new schedules to the greenhouse climate control system. Such schedules can be created by expert systems based on crop etc.

The Agrosense client can communicate with the Local FMS to store (like position, sensor data, task planning information etc) and retrieve collected data (sensor's data etc). The Cloud FMS and the AgroSense Server can communicate directly in order to execute complex processes by collecting information and knowledge through different external services. A high level view of this scenario is presented in Figure 6-5.

Overall this exercise has proven that the cooperation of our work with other tools is not only feasible but it can also provide value added services to the farmers. For example the following figure presents in Archimate such an example (Figure 6-6). In this example a disease is found and a plan for spraying is needed. This planning is executed by services orchestrated by the Greenhouse management tools (i.e., the Cloud and Local FMSs) while the execution is performed by the Agrosense client (i.e., running in a Smartphone). The checking of the spraying operation is performed by the Cloud FMS.

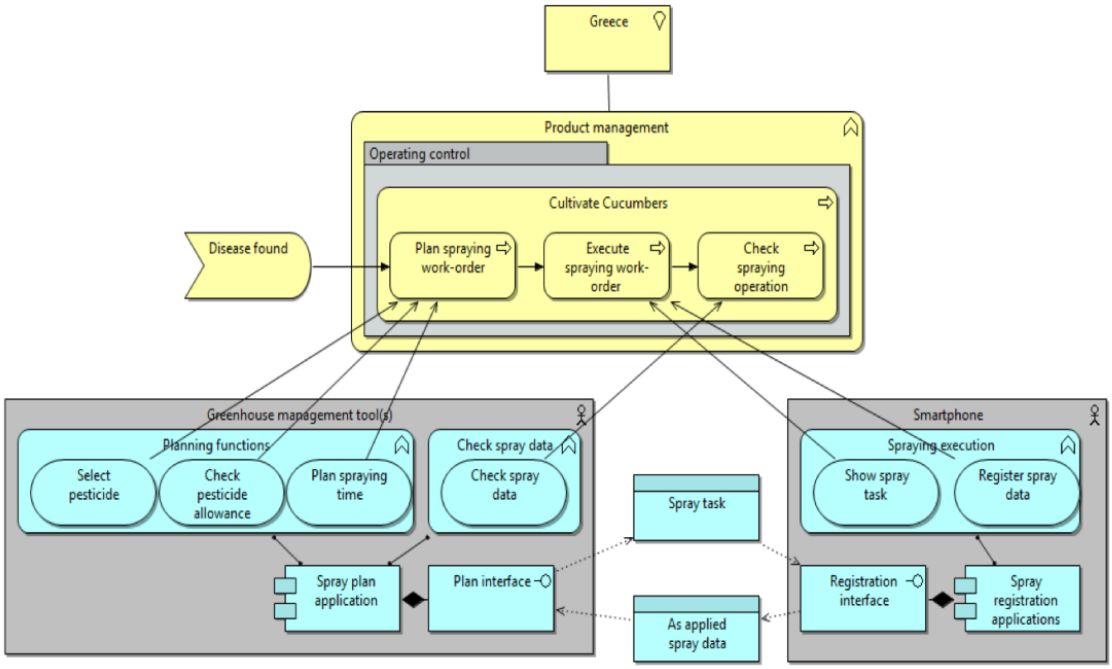


Figure 6-6: Linking SAF's FMS with Agrosense – The business case

7 Conclusions

The objectives of WP200 were captured in the project Description of Work and achieved through the four deliverables. We conclude this document by providing a brief summary of these objectives and how they were achieved:

Obj.1: The overall aim of this WP was to define the key technical aspects related to smart farming, spanning from the architectural requirements and the specification of the required mechanisms and domain sub-systems to the definition of the pilot system for experimentation.

→This work has been captured in D200.1 [3] and D200.2 [4]. In the first document, numerous use cases were identified and used in order to identify functional and non-functional requirements; these requirements were further processed in D200.2 and aggregated into functional blocks which formed the FMS architecture.

Obj.2: Develop a small scale pilot system to demonstrate the key features of the smart farming use case

→ Based on the design and the specifications of the two former deliverables, in the context of this document (Sections 3 and 4) we provided the specifications of the two pilots implemented in the context of WP200, namely the Greenhouse and the Smart Spraying pilot. The document concluded with the integration effort of the two pilots. Further, pure technical information, will be reported in D500.5.2 [8].

Obj.3: Evaluate and assess the architectural aspects and defined mechanisms and assess the penetration of smart farming services and their impact to the end users

→ Evaluation and assessment of the mechanisms took place through various activities which will be reported in D200.4 [5] and WP600. In the context of this document we evaluated the prototypes against the requirements and use-cases of D200.1 [3].

Obj.4: Define the architectural requirements of the smart farming area and their links with the Generic Enablers implementing the key objectives of the core platform for the future internet.

→ This document described the final version of the architecture and the updates that took place since D200.2 [4]. In Section 2, we provided updated information regarding the architecture, identified the potential usage of Generic Enablers and highlighted the integration of the four SAF generic services. Additionally, in Sections 3 and 4 we identified a subset of Generic Enablers that has been evaluated and integrated in the prototypes.

Obj.5: Monitor and coordinate the standardization activities related to smart farming focusing on sensor data harmonization and interoperability.

→ WP200 has been involved in standardization activities; the major part of this work is reported in D200.4 [5]. In this document, based on the feedback from the actual implementation process we identified potential standardization issues. In the appendix we outline an indicative API for standardization which stems from the actual integration. Additional information regarding standardization requirements for large scale experimentation will be reported in D600.2.

8 References

[1] Smart Agri-Food, "Smart Food and Agribusiness: Future Internet for Safe and Healthy Food from Farm to Fork", http://www.smartagrifood.eu/

[2] D100.3, "Use Case Harmonization"

[3] D200.1, "First Report on Smart Farming Architectural Requirements and Subsystem"

[4] D200.2, "Detailed Specification for Smart Farming Experimentation: Generic Enabler, Subsystem and Architecture"

[5] D200.4, "Smart Farming: Final Assessment Report"

[6] D500.3, "Specification on network elements and functions of Core Platform"

[7] D500.4, "Specification on protocols between domain networks of stakeholders and Core Platform"

[8] D500.5.2, "Second Release of SmartAgriFood conceptual prototypes"

[9] AgroXML, http://www.agroxml.de/

[10] Sensor Model Language, http://www.opengeospatial.org/standards/sensorml

[11] Observations and Measurements, http://www.opengeospatial.org/standards/om

[12] ASPIRE project: http://www.fp7-aspire.eu/

[13] Reader Protocol: http://www.epcglobalus.org/Standards/EPCglobalStandards/ReaderProtocolStandard/tabid/361/Default.aspx

[14] Low Level Reader Protocol: http://www.gs1.org/gsmp/kc/epcglobal/llrp

[15] *OSGi,* www.osgi.org

[16] ENVIROFI: *The **Enviro**nmental Observation Web and its Service Applications within the **F**uture **I**nternet, http://www.envirofi.eu/*

[17] CropInfa: http://www.cropinfra.com

[18] KoneAgria: http://www.koneagria.fi/

[19] The Agrosense project, http://agrosense.org/

[20] Smart AgriMatics 2012, http://www.smartagrimatics.eu/

[21] Ordina, http://www.ordina.com/

[22] Open Authentication Protocol, http://oauth.net/

[23] OpenID, http://openid.net/

[24] Homologa, http://www.homologa.com/

[25] Requirements Prioritization Technique – MoSCoW Analysis: http://project-management.learningtree.com/2011/01/18/requirements-prioritization-technique-moscow-analysis/

9 Annex

9.1 Validation Matrixes

A crucial part in system validation is the verification backward tracebality i.e. trace back the initial user requirements and identify whether they have been met or not. Towards this end, we extract the requirements tables from D200.1 [2] and try validate whether the requirements have been encapsulated in the architecture design and finally implemented (and to which extent) in the pilots.

Indicative text from scenario	Functional requirement	Implemented	Addressed by design
Information about the expected yield	Predictions about estimated yield should be possible	Yes	Yes
Information about the development of crops, fruits and vegetables	Recognize if products are developed properly should be performed	Yes	Yes
To use information for planning of the production, for sales forecast, for scheduling harvesting, for irrigation network and for nutrients management	Planning the daily tasks inside the farm should be available	Yes	Yes
When agricultural machines (tractors, harvesting machines, etc.) are working on the field camera systems connected to foreign body identification systems can identify the foreign bodies and their location and provide a map to guide their removal.	Identification of extraneous and foreign bodies should be performed	No	Yes
Identification can be very exact and the information can be collected in a large database of pictures, images, data and characteristics about identified foreign bodies.	Gathering multimedia information for further analysis should be possible	Partial. Only video capturing is supported. No post-processing takes place.	Yes
Provide a map to guide the removal	Removal of extraneous and foreign bodies should be performed	No	Yes
To distribute the national quota and to monitoring and recording the production volumes compared with the allocated quota, so every member need to operate production volumes registration.	Receive notifications from the authorities about the national quota	N/A	Yes
To distribute the national quota and to monitoring and recording the production volumes compared with the allocated quota, so every member need to operate production volumes registration.	Farmers notify the authorities about the produced milk quota	N/A	Yes
After he has worked the half of the area the sprayer system informs him that the tank will be empty in 15 minutes, and 1/3 of the area will remain unsprayed.	The spraying system should inform the machine operator for existing alarms	Yes	Yes
The sprayer shares the information with the	Proper notifications should be	Partial.Adapted	Yes

Indicative text from scenario	Functional requirement	Implemented	Addressed by design
other sprayers nearby, alarming them about the situation.	sent to neighbouring stakeholders for any emergencies	on conceptual level. Can be implemented in notifier service using access rights management.	
Several self propelled forage harvesters (SPFH) and transporting vehicles (tractor with trailer) are coordinated for harvesting for a biogas plant.	Plan for cooperative harvest should be produced	N/A	Yes
Defect on SPFH • Trailers are reassigned to another harvesting chain Defect on a trailer • Reorganisation of trailer assignments Silo is full. Trailers are reassigned to another silo	Reorganization of cooperative harvest when a problem occurs should be performed	N/A	Yes
If a new firmware version is available the tractor asks the tractor owner/driver for permission to download and install	Automatic firmware updates should be supported	Yes	Yes
This data is transferred to a data mining application in the cloud that processes this data in order to find optimal settings and conditions for a given machine and task.	Data mining techniques should be enabled	Partial. Specific implementation related to the Expert System.	Yes
In this scenario a tractor with a trailer is steered remotely by a SPFH during loading.	Remote control of agricultural machines should be possible	Yes	Yes
Via remote machine diagnostic and wireless connection the dealer, after getting permission from the operator, connects to the machine for diagnostics and trouble shooting.	Fault operation of agricultural machines should be detected	Yes	Yes
Based on a disease forecast model, criteria for disease onset can be predicted	Disease forecast as well as recommendations should be available	Yes	Yes
When the criteria for disease risk prediction from the PDFS are reached, a warning is sent to the farmer's mobile phone	Notifications should be sent when it is predicted that plants will be infected by a disease	Yes	Yes
After spraying is executed, the documentation of the spraying operation is updated to the PDFS for performing future predictions	All transactions between different services and modules will be used for making the system learn.	Partial. The spraying event is recorded and stored in the farm data storage service that can communicate with other relevant services.	Yes
To provide information about spraying process and other spraying fleet for informed	Notifications about spraying processes should be sent to	Partial. Adapted on	Yes

Indicative text from scenario	Functional requirement	Implemented	Addressed by design
decision making	interested parties	conceptual level. Can be implemented in notifier service using access rights management.	
Prioritization of weather, data and information transfer between machine and internet	Notifications with high priority should be sent using appropriate protocols	Yes	Yes
Modification of task on-line is done when spraying; this task is based on to the present/changing weather parameters and forecasts, spatial rules	Decision for recommending cancellation of scheduled tasks inside the farm when weather is not proper should be given	Yes	Yes
A farmer has installed a number of sensors in his greenhouse collecting information for a number of parameters such as temperature, humidity etc	Collection of sensed data should be performed	Yes	Yes
All data that comes from different sensor networks are gathered and aggregated by a sensor network aggregator.	Data aggregation should be dynamically performed	Yes	Yes
The farmer is informed about possible actions that he can do in order to handle the alarms.	Recommendations should be given for handling alarms	Yes	Yes
Modules have to control all FMISs as well as services that are located in and over the cloud for producing the best possible results	Mechanisms should be developed for managing and controlling all up – coming services and applications	Yes	Yes
The farmer has defined during the installation of the system or at a later time, the thresholds for receiving notifications and alarms for their cultivations	Configuration of parameters for existing methods should be available at any time	Yes	Yes
If the problem is simple and can be handled autonomously by the locally installed systems of sensors & actuators system, which is installed is the farm, it is resolved (e.g., open the windows, start the ventilation system).	Automatic reaction of the system should be present	Yes	Yes
The respective notification entity undertakes the task to notify the farmer to take a certain action	Notifications should be sent to every stakeholder when an abnormal state takes place in a farm	Yes	Yes
A module informs the farmer only about the disordered values, in one or more types of communications (e.g. SMS, email etc) in one or more types of terminals (desktop, mobile phone etc)	Adaptability of content on different devices should be possible	Yes	Yes
The farmer can use his FMIS or other FMISs in order to discover and call another player (e.g., his spray contractor or an agriculturist) who could help him.	Defined level of control and management of the network should be available	Partial. A simple mechanism is implemented.	Yes
A software module (either located locally, or in the cloud) checks periodically the integrity of the incoming information to detect a faulty sensor	Faulty sensors should be detected	Yes	Yes

Indicative text from scenario	Functional requirement	Implemented	Addressed by design
If a sensor is detected to send data with values not in accordance with the other sensors in the area, its values are no longer taken into account	Actions should be performed in order to isolate malfunctioning sensors	Yes	Yes
Different types of sensors have been connected to a sensor network and automatically establish working configurations	Self-configuration mechanisms should take place	Yes	Yes
News that comes from the outside world and may affect the farm is gathered.	Collection of news related to a farm's operation should be performed	Yes	Yes
The farmers, who have claimed their interests for specific kind of data, are informed from the respective service	A farmer should be informed about news he is interested of	Yes	Yes
The system (i.e., the smart farming system - FMIS) is informed by other services (registered or not) about relevant news.	Cooperation between different FMISs should be possible	Partial. Implemented through services notifications	Yes
Different services i.e. policies' and information service, meteorological service, etc give permission to access their data or want to provide the system with data	Only authenticated and authorized users would have access to information	Yes	Yes
Decisions may be modified according to specific rules (e.g., based on a new governmental policy)	The system must be re-configurable based on new data received over the Internet	Yes	Yes
The system (i.e., the smart farming system or his FMIS) is informed by external services (registered or not) about relevant news or according to farmers behaviour and personal interests a new service may interest him	Real – time recommendations should be sent according to stakeholder's behaviour	Yes	Yes
The farmer could also advertize himself through his profile by uploading data, such as photos or videos in order to make his profile friendlier	Different stakeholders' profiles should be loaded for different services	Yes	Yes
When an event takes place and is recorded in the data collector the appropriate information profiles are updated.	Periodically updates of profile should be possible depending on triggers	Yes	Yes
Farmer's related data are securely accessed over the Internet	Security issues are critical for any service and FMIS	Yes	Yes
A farmer subscribes to an electronic advisory service (e-agriculturist) to receive sophisticated advices	Subscriptions should be provided for stakeholders to all the available services	Yes	Yes
In this task the e-agriculturist can also suggest specific contractors based on price, their location and availability, reviews from other farmers etc	Finding other players and link to them should be available	Yes	Yes
A farmer will provide feedback to the whole system, possibly fills questionnaires, rates	Stakeholders should be given with the opportunity to give	Yes	Yes

Indicative text from scenario	Functional requirement	Implemented	Addressed by design
specific methods of farming and evaluates the recommendations given from information/advisory service	feedback for every action of the system		
All the farmers are registered to the system	Registrations could be performed for every service	Yes	Yes
Each farmer could share his opinion about the predictions, the suggestions that have been given as well as the specialized personnel that have been proposed by different FMISs.	Stakeholders should be given with the opportunity to give feedback for every action of the system	Yes	Yes
Geo – spatial issues have to be considered in order to help end – users find other interested parties	Geo – located users activity data and mobility profiles should be available	No	Yes
A farmer wants to have access to statistical data that concerns him.	Statistical process of data should be done	Yes	Yes
After the statistical processing, a module can analyze these data in order to produce a proper recommendation	Suggestions should be available according to statistical data	Yes	Yes
These multimedia data are processed by an appropriate analyzer to identify possible issues (e.g., a disease)	Multimedia analysis should be performed	No	Yes
The notifier checks the available identities of the farmer, their priorities and possibly the fixed time period of each device in a certain location	Automatic end – terminal selections should be available	Yes	Yes
After his entrance in to his desktop, all the received data by his smart phone, as well as newly arrived data, are handed over to his desktop	Seamless transition between different devices should be performed	Yes	Yes
All the local functionalities have to be set into a local mode and adjust their results based on local configurations and capabilities (devices inside the farm and outside, CPUs, cache memory, RAM, ROM, etc)	Devices linked to the system announce their capabilities	Yes	Yes
Although, the respective cloud proxy that will be located in the home network will provide with acceptable storage and computing capabilities.	Local system has to take control when internet connection fails	Yes	Yes
The farmer has the need to print a basic barcode label for his final product before their storage or shipment.	Tracing/tracking capabilities should be enabled	Yes	Yes
All data can be formatted automatically in an appropriate format for presentation that will be useful by other involved entities such as certification authorities, government authorities, payment authorities etc	External services should be able to access farmer's information	Yes	Yes
Consequently, the system combines data collected by the already installed sensors and the data from the electronic advisory system (described in UC17) and a cultivation plan is derived.	Cultivation plans should be produced	Yes	Yes
The farmer provides data about the location of operation, infrastructure available, the	A stakeholder should be able to customize the information	Yes	Yes

Indicative text from scenario	Functional requirement	Implemented	Addressed by design
properties of the parcel, etc	loaded into his profile		
The farmer or any visitor can search the material uploaded in the system using multiple criteria	Efficient indexing techniques should be used to have access to stored data	Yes	Yes
He enters the system (if he enters for the first time he should create a user profile) and performs a search for producers that are active nearby.	Data coming from neighbouring infrastructures should be gathered for further use.	Partial. Information aggregation is implemented. Search is not implemented to full extent	Yes
Full information for the milk's components such as fat, protein, lactose, number of Somatic Cells as well as indications that alarms the values of the above factors	Collection of information that refer to dairy products should be possible	N/A	Yes
The consumer is searching for qualitative products coming from dairy farms	Aggregation of data should be done in order to specify the quality of a product	Yes	Yes
He has access to the data collected by sensors and various measurements concerning animals' health.	Data management of data that refers to animals should be possible	N/A	Yes
- Farmer interesting in selling his animals subscribes to the system and states his interest - Farmer interesting in buying animals subscribes to the system, searches with multiple criteria to find other farmers	Linking sellers and buyers should be possible	Partial. Indirect link through services.	Yes

9.2 Indicative API for Standardization

Table 9-1: Indicative API for standardization stemming from the implementation of the Greenhouse Pilot

Method name	Description	Associated REST-URI (ServerRootUrl/)	METHOD-PARAMETERS
find_stakeholder_by_id	Return information about the stakeholder identified by the specified ID.	stakeholder_by_id/	GET, stakeholder_id
find_service_by_id	Return information about the service identified by the specified ID.	service_by_id/	GET, service_id
get_notifications	Return the list of notifications related to the user identified by the specified ID.	notifications/	GET, user_id
get_alerts	Return the list of alerts related to the user identified by the specified ID.	alerts/	GET, user_id
show_farmplan	Return the list of measurements related to the user identified by the specified ID grouped by farm and sector	farmplan/	GET, user_id
show_sensors	Return the information (type, last measurement, etc…) for all sensors of all sectors and farms for the user identified by the specified ID.	sensors/	GET, user_id
get_all_services	Return the list of all services of an FMS.	services/	GET
get_all_stakeholders	Retrieves the list of all stakeholders of an FMS.	stakeholders/	GET
get_farms	Returns a list with information of all the farms that belong to the user identified by the specified id.	get_farms/	GET, user_id
get_sectors	This method requires as an input a valid id of a farm that belongs to a farmer. It returns the information of all sectors that are part of a specific farm.	get_sectors/	GET, farm_id
get_equipment	This method takes as a parameter the id of a farm's sector. It retrieves the information of all equipment that are installed on a specific sector of a farm.	get_equipment/	GET, sector_id
get_mote	This method takes as a parameter the id of an equipment and returns the information (if there exists) associated with it	get_mote/	GET, equip_id
get_sensor	This method requires as input an id of a mote and the type of sensor. It returns the description of the sensor.	get_sensor/	POST, mote_id, type_of_sensor

Method name	Description	Associated REST-URI (ServerRootUrl/)	METHOD-PARAMETERS
get_measurements	This method takes as parameters the number of measurements the user wants to find and the sensor id that these measurements are taken from.	get_measurements/	POST, count, sensor_id
find_average	This method is taking as input a list of double values, finds their average and returns it.	ReceiveStatistical/ function/average	POST, list of Double values
find_deviation	This method takes as parameter a list of double values and returns the standard deviation that stems from these numbers.	ReceiveStatistical/ function/deviation	POST, list of Double values
find_min	This method receives as input a list of double values and finds the minimum.	ReceiveStatistical/ function/min	POST, list of Double values
find_max	This method receives as input a list of double values and finds the maximum.	ReceiveStatistical/ function/max	POST, list of Double values