Deliverable D400.6

Functionalities of Baseline Applications

WP 400

Project Acronym & Number:	FIspace – 604 123
Project Title:	FIspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics
Funding Scheme:	Collaborative Project - Large-scale Integrated Project (IP)
Date of latest version of Annex 1:	03.10.2013
Start date of the project:	01.04.2013
Duration:	24
Status:	Submitted Version
Authors:	Clarissa Cassales Marquezan (UDE), Nadeem Bari (UDE), Sokratis Barmpounakis (NKUA), Konstantina Dimtsa (NKUA), Vasileios Sarris (NKUA), Lorea Gómez (Atos), Ramón Alcarria (UPM), Harald Sundmaeker (ATB), Norman Gülcü (ATB), Serdar Arslan (KOC), Seyhun Mehmet Futaci (KOC), Marianne Hagaseth (MRTK), Asmund Tjora (MRTK), Kay Fjortoft (MRTK)
Contributors:	Fabiana Fournier (IBM), Monika Solanki (ASTON)
Document Identifier:	D400.6
Date:	30.06.2013
Revision:	007
Project website address:	http://www.FIspace.eu

The FIspace Project

Leveraging on outcomes of two complementary Phase 1 use case projects (FInest & SmartAgriFood), aim of FIspace is to pioneer towards fundamental changes on how collaborative business networks will work in future. FIspace will develop a multi-domain Business Collaboration Space (short: FIspace) that employs FI technologies for enabling seamless collaboration in open, cross-organizational business networks, establish eight working Experimentation Sites in Europe where Pilot Applications are tested in Early Trials for Agri-Food, Transport & Logistics and prepare for industrial uptake by engaging with players & associations from relevant industry sectors and IT industry.

Project Summary

As a use case project in Phase 2 of the FI PPP, FIspace aims at developing and validating novel Future-Internet-enabled solutions to address the pressing challenges arising in collaborative business networks, focussing on use cases from the Agri-Food, Transport and Logistics industries. FIspace will focus on exploiting, incorporating and validating the Generic Enablers provided by the FI PPP Core Platform with the aim of realising an extensible collaboration service for business networks together with a set of innovative test applications that allow for radical improvements in how networked businesses can work in the future. Those solutions will be demonstrated and tested through early trials on experimentation sites across Europe. The project results will be open to the FI PPP program and the general public, and the pro-active engagement of larger user communities and external solution providers will foster innovation and industrial uptake planned for Phase 3 of the FI PPP.

Project Consortium

- DLO; Netherlands
- ATB Bremen; Germany
- IBM; Israel
- KocSistem; Turkey
- Aston University; United Kingdom
- ENoLL; Belgium
- KTBL; Germany
- NKUA; Greece
- Wageningen University; Netherlands
- PlusFresc; Spain
- FloriCode; Netherlands
- Kverneland; Netherlands
- North Sea Container Line; Norway
- LimeTri; Netherlands
- Kühne + Nagel; Switzerland
- University Duisburg Essen; Germany
- ATOS; Spain
- The Open Group; United Kingdom
- CentMa; Germany
- iMinds; Belgium
- Marintek; Norway
- University Politecnica Madrid; Spain
- Arcelik; Turkey
- EuroPoolSystem; Germany
- GS1 Germany; Germany
- Mieloo & Alexander; Netherlands
- OPEKEPE; Greece
- Innovators; Greece

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.cSpace.eu // www.FIspace.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)	

Change History

Version	**Notes**	**Date**
001	Creation of the document	13.05.2013
002	Updating structure	28.05.2013
003	Updating structure	11.06.2013
004	Updating content of sections and consolidation of partners input	18.06.2013
005	Document ready for review	20.06.2013
006	Consolidation of reviewers' feedback	27.06.2013
007	Document submitted	30.06.2013

Document Summary

This document describes the work conducted within the last three months on the analysis of the baseline apps functionalities. The definition and development of baseline apps is part of the sub-task 451 (ST451 - Development of the cross-domain baseline applications) in the scope of task 450 (T450 – Generic and Domain-specific Application Development) under the Work Package 400 (WP400 – Use Case Trials). This document reports the achievements of the following objectives associated with deliverable D400 .6:

1. Consolidation of the set of baseline apps to be developed
2. Definition of main functionalities to be developed by the baseline apps
3. Identification of main relationships with the core components of WP200 (*FIspace Development Work Package*)
4. Identification of potential Generic Enablers (GEs) to be used by the baseline apps
5. Definition of preliminary test requirements and relationships with WP300 (*FIspace Hosting and Experimentation Work Package*)

The consolidated list of baseline apps includes: the Product Information Service App, Logistics Planning Service App, Business Profile App, Marketplace Operations App, Real-time Business SLA Management App. For each one of the baseline apps we describe the topics 2 – 5 above listed. In addition, we show a first list of Use Case Trials that consider using the functionalities of the baseline apps. This initial list will be revised and updated with the progress of the project.

Abbreviations

API	Application Programming Interface
App	Software Application
B2B	Business to Business
BizSLAM	Real-time Business SLA Management App
CEP	Complex Event Processing
CRUD	Create, Read, Update, Delete
D	Deliverable
DoW	Description of Work
e.g.	Exempli gratia = for example
EC	European Commission
EP	Event Processing
EPC IS	Electronic Product Code Information Services
ESB	Enterprise Service Bus
EU	European Union
FI PPP	Future Internet Public Private Partnership
FIA	Future Internet Assembly
FP7	Framework Programme 7
GA	Grant Agreement
GE	Generic Enabler
i.e.	id est = that is to say
ICT	Information and Communication Technology
IP	Intellectual Property
IPR	Intellectual Property Rights
KPI	Key Performance Indicator
Linked-USDL	Linked Unified Service Description Language
LPA	Logistics Planning App
LSC	Logistics Service Client
LSP	Logistics Service Provider
M	Month
PInfS	Product Information Service App
RDF	Resource Description Framework
RTD	Research and Technological Development
SLA	Service Level Agreement
SLO	Service Level Objective
SME	Small and Medium Sized Enterprise
ST	Sub-Task
T	Task
UI	User Interface
WP	Work Package

Table of Contents

List of Figures

List of Tables

1 Introduction

FIspace project major goal is to create a cloud-based environment to support the development of multi-domain business collaboration space (or simply FIspace Platform) that employ FI technologies for enabling seamless collaboration in open, cross-organizational business networks. The FIspace platform is organized in different main building blocks within the Work Package 200 (WP200 – FIspace Development). Based on the building blocks offered by the platform, developers will be able to create new applications (from now on called apps) or to mash up different existing apps to provide a new app with enhanced functionalities.

Considering the app development process, we envision the existence of apps that are very specific to a domain and other apps that can actually be used by different domains. For example, an app that calculates the spraying advices for famers is very specific. On the other hand, apps that notify events about violated SLAs of business partners or that provide information about products are, indeed, applicable for different domains. In FIspace project, we call ***baseline apps*** the second above described type of apps, *i.e.*, the ones that can be applicable to distinct domains.

1.1 Objectives of this Document

In this document, we present the work conducted until M3 within the scope of WP400 (*Use Case Trials*), Task 450 (T450 - *Generic and Domain-specific Application Development*), sub-task 451 (ST451 - *Development of the cross-domain baseline applications*). The work here reported documents the achievement the following objectives associated with deliverable D400 .6:

- Consolidation of the set of baseline apps to be developed
- Definition of main functionalities to be developed by the baseline apps
- Identification of main relationships with the core components of WP200 (*FIspace Development Work Package*)
- Identification of potential Generic Enablers (GEs) to be used by the baseline apps
- Definition of preliminary test requirements and relationships with WP300 (*FIspace Hosting and Experimentation Work Package*)

1.2 Consolidated List of Baseline Apps

It is defined in the DoW (Description of Work) a set of initial baseline apps to be considered for development. The baseline apps belonging to this initial list were in their majority attempts to leverage the work done in Phase I of the FI PPP Programme. Some of these apps were actually an exact projection of previous work. However, once the overall work in the project started, more details became available on how the FIspace platform should operate and which kind of functionalities from the core platform developed in WP200 could be exploited by the baseline apps. As a consequence, the team working on the baseline apps started to develop a better awareness and vision of how baseline apps should be designed and specified. The result of this awareness is a refined list of baseline apps as described in Table 1. In summary, we still leverage the work done in Phase I and we refine and extended it, so that instead of having four baseline apps as described in the DoW, we have now five baseline apps originated from two changes in relation to the initial list of baseline apps.

The first change introduced three new apps (rows 2 - 5 in Table 1) that better cover the functionalities originally planned to be developed in the “Business Services & Contract Management App”. In addition, this change reinforces the idea that baseline apps can be mashed up to other apps to provide enhanced functionalities. For example, as it will be describe later in Section 7.3, the Logistics Planning Service App uses functionalities from both Marketplace Operations App and Real-time Business SLA Management App to provide enhanced planning capabilities for logistics planning.

The second change is associated with the “Real-time Exception Detection and handling” App initially defined in the DoW. After an analysis of this baseline app we identified that its functionalities are extremely related to event processing which is a functionality provided by the ST240 (FIspace Real-time B2B Collaboration) from the core of the FIspace platform. Thus, our proposal is to provide the envisioned functionalities of this app as a generalization of the functionalities covered by the ST240 component.

Event Processing (EP) applications are mostly designed for real-time exception detection and handling. The part that we want to transform as a core functionality of ST240 is described in the DoW with respect to this base app. The idea is to provide means for configuration of new apps to ease the onboarding of new collaborations to the platform such as templates, so new apps can be built on top of some predefined business entities or rules (these two terms are defined by ST240). Furthermore, the exception handlers are also part of the pattern used in the EP engine designed in the ST240 module. These engines are commonly associated to the 4Ds: Detect, Derive, Decide, Do. Currently the EP engine in ST240 is designed to perform the first two D, i.e, detect and derive. However, we believe that the engine in ST240 could be enhanced to also “Decide” and “Do”, and this is again given as part of the FIspace platform not as an app that needs to be bought.

Taking into considerations the arguments above, we currently understand that instead of restricting the functionalities inside a baseline app, we would prefer to provide the implementation of these functionalities as part of the FIspace platform. The major advantage of this change is that every app developed on top of FIspace could benefit from these functionalities. Indeed, due to the very early stage of work in the components on WP200 (only 3 months) it is very difficult right now to take the final decision on where this functionality should sit. Nevertheless, because FIspace project uses an agile approach, our decisions can be revised at any time; and if we deem appropriated this app can be eventually included in the first, second or third release of the baseline apps, respectively in deliverables D400.7 (M9), D400.8 (M15), and D400.9 (M21).

Table 1: List of consolidated baseline apps

Baseline App	Description
Product Information Service App	This app was defined in the DoW and based on our analysis the envisioned functionalities (defined during the proposal submission) are still aligned with the vision developed in the first months of the work in FIspace project. The main purpose of this app is providing product information among the systems of stakeholders of a supply chain. The details of this baseline app will be described in Section 2.
Logistics Planning Service App	Likewise the above mentioned app, this baseline app was revised and its functionalities remained aligned with the vision of baseline apps. The main functionality of this app is to provide the support for automated planning of logistics services. Section 3 provides the details on this app.
Business Profile App	This app is associated with identification of the profile of business entities inside FIspace platform. Originally, the functionalities of this app were enclosed in the “Business Services & Contract Management App” belonging to the initial set of baseline apps described in the DoW. However, after rounds of discussions with members from WP200 and in the team developing this baseline app, we identified that the functionalities of describing the profile of business entities was self-contained and could be re-used by many other apps. Thus, following the vision of baseline apps, we defined baseline app (i.e., the Business Profile App) that is responsible for this functionality. Section 4 depicts this baseline app.

Baseline App	Description
Marketplace Operations App	Likewise the above describe situation, this baseline app contains a subset of the functionalities originally defined in the "Business Services & Contract Management App". For the same reasons as before mentioned, we believe that it better to enclose the operations associated with marketplace, such as publishing service offers and demands, in a self-contained app. Section 5 provides the details about this app.
Real-time Business SLA Management App	This baseline app encloses also a subset of functionalities previously defined in "Business Services & Contract Management App". We believe that the refinement of this subset into this specific app provides a better and more precise description to this app. The main goal of this app is to manage the SLA (Service Level Agreement) between business entities in real-time during the execution of agreed services. The details of this baseline app are presented in Section 6.

As previously mentioned, the baseline apps listed above are indeed a refinement of the work done during Phase I of the FI PPP Programme. Technological choices adopt for the development of these baseline apps were extensively analyzed during the work in Phase I and document in several deliverables from FInest and Smart Agri-food project. For instance, the three baseline apps - Business Profile, Marketplace Operations, and Real-time Business SLA Management apps – leverage the work done in the E-Contracting Module from FInest project and the Business Relationship Service from Smart Agri-Food Project. These apps use RDF as data model and the motivation for this is to be able to use the functionalities offered by some generic enablers like Marketplace GE and Repository GE from the Core Platform in the FI PPP Programme. The more details about the rationale for this specific choice can be found in the deliverables from the previous projects in Phase I, e.g., D8.1 "Requirements analysis and selection of technology baseline for logistics contract manager"[1].

In this document, we describe the alignment of the work done in Phase II related to the baseline apps to the context of FIspace project in Phase II, rather than revising the previous technological choices. Those choices are still valid and will be followed as long as they complain with the requirements for developing app based on FIspace platform. In summary, in this document we do not revise and technological choices for adapting the baseline apps to the context of FIspace project, but we revise and consolidate the functionalities of these apps. We report here the first steps towards the technical alignment with FIspace platform.

1.3 Relationship to other Tasks and Work Packages

In FIspace project there are many relationships among tasks and Work Packages. The sub-task ST451, addressed in this document, is related to all the main parts of FIspace project as illustrated in Figure 1. The dark solid and the dark dotted lines in Figure 1 indicate the relationships that are mainly related to the baseline apps. The gray lines in Figure 1 indicates relationships among other parts of FIspace, which are illustrate only to keep the consistency of the information but are not the focus of this report.

[1] http://www.finest-ppp.eu/files/deliverables/d08/finest_d8_1_final.pdf

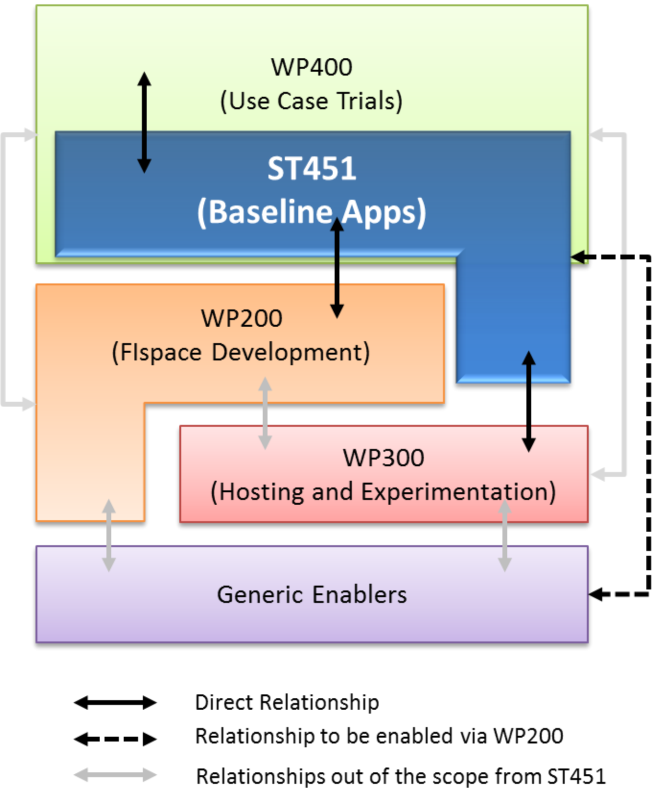


Figure 1: Baseline Apps relationships with other tasks and WPs from FIspace project

The baseline apps are tightly related to the components of WP200. These components provide the main functionalities of FIspace platform and enable the development and execution of apps. Thus, this is a very strong relationship but this does not mean that all the baseline apps will use all the functionalities of WP200. This, in fact, depends on the functionalities of the app and a mapping process has to be performed between such functionalities and the WP200 components. In this document, for each one of the baseline apps we provide the first insights about this mapping. In addition, the Table 2 provides a list of the components in WP200, their names and summary of functionalities to facilitate the understanding of the mappings done so far. These components will be later mentioned during the description of the baseline apps.

Table 2: List of components in WP200 most relevant for the baseline apps

WP200 Components	Summary of Component Scope
ST220 – Front-End Development	Serve as the main access point for end users. This module will provide ‘standard features’ for business collaboration, access to the User Interfaces of the FIspace Apps (e.g., baseline apps), support the configuration of the customized end users view, allow ubiquitous access via various devices and channels, and ensure a ‘corporate design’ of the FIspace.
ST240 – Real-time B2B Collaboration	Concerned with implementing the FIspace Real-time B2B collaboration concept. Its purpose is to manage, execute and monitor collaborative processes at the heart of FIspace. The aim is to implement the collaboration component based on artefact-centric concepts with high agility and dynamicity in defining a collaboration object lifecycle. In addition an event processing component is also implemented based on com-

WP200 Components	Summary of Component Scope
	plex event processing and enhanced to enable proactive event processing (such as proactively handling expected delays or deviations in a transport plan execution).
ST250 – System & Data Integration	Concerned with creating a communication data exchange channel between the FIspace and other external systems/sensors in an easy-to-use and bidirectional way, therewith enabling the connection of existing systems with the FIspace in order to allow for the import and export of data and relevant information.
ST260 – Operating Environment	Provide the technical infrastructure that allows all of the FIspace components to work together in harmony. Is associated with the communication channels among the FIspace modules, automation of the application lifecycle, eventual consistency, elasticity and load balancing mechanisms.
ST270 – Security, Privacy and Trust	Provide adequate security, privacy and trust management techniques in order to ensure the secure and reliable exchange of confidential business information and transactions in FIspace.
ST280 – Development Environment	Concerned with the provisioning of a Development Environment for the FIspace that consists of 2 main parts: (a) a development of a SDK (Software Development Kit) for developers to easily create new FIspace Apps, and (b) a toolkit for Business IT Engineers to customize FIspace. The goal and motivation for the former is to enable FIspace developers to re-use existing functionalities (esp. existing Apps) and extended it with new business logic (i.e. UI, rules, data structures), therewith allowing for re-use and cost-efficient application development.

As any other software components of FIspace platform, each one of the baseline apps needs to be hosted and tested. For this reason we also indicate the relationship among ST451, test requirements and WP300 as depicted in Figure 1.

In addition, some of the baseline apps will be used by some of the trials to be conducted in WP400. At this stage of the project, we are in the initial phase of identifying which trials could actually use which functionalities of the baseline apps. Thus, as illustrated in Figure 1, we are aware that we will have to exchange experience and information also with other tasks of WP400. Section 7.4 in this document shows an initial list and analysis establishing the relationship between trials and baseline apps. These trials are actually defined and developed under the scope of WP400.

Finally, we already identified that some of the baseline apps will need to rely on Generic Enablers functionalities. As illustrated by the dark dotted line in Figure 1, we still have to identify which kind of relationship the baseline apps will have with GEs, i.e., whether they will have direct or indirect access. The next section provides insights about this discussion.

1.4 Discussion on Generic Enablers, WP200 and Baseline Apps

We started the process to identify which GEs can be used by the baseline apps. According to discussions among the baseline app leads and the architects of FIspace project, we reach a consensus that it would be appropriated to avoid a direct connection from baseline apps to GEs. This means that all GE functionalities required by the baseline apps should be accessed via FIspace platform.

However, not all GEs functionalities required by the baselines apps are currently covered by the components of WP200, *i.e.*, the core FIspace platform. Most of the GEs are used, but there is not yet a clear definition of interfaces that will be exposed by the components in WP200 that enable the connection of baseline apps to the GEs.

Therefore, in this document, we list the GEs that will be required by the baseline apps and which GEs functionalities are necessary. Based on this identification we can conduct the next step of passing this information to WP200 and helping the teams in this WP to design and provide the connections that the baseline apps needs.

1.5 Organization of this Document

This document is organized as follows.

All the sections associated with the description of the baseline apps follow the same structure and provide the same type of information. We provide:

(i) an overview on the baseline;
(ii) the details in terms of functionalities to be supported by each app;
(iii) the relationship with WP200 components;
(iv) the relationship with GEs;
(v) the preliminary test requirements and relationship with WP300;
(vi) and, summary associated with the baseline app concepts.

The sections that follow the same structure are:

- Section 2 introduces the Product Information Service App.
- Section 3 describes the Logistics Planning Service App.
- Section 4 presents the Business Profile App.
- Section 5 depicts the Marketplace Operations App.
- Section 6 discusses the Real-time Business SLA Management App.

In Section 7, we provide a short overview on all baseline apps. We summary the components from WP200 associated with the baseline apps, the overall list of GEs, and importantly we show a first list of Use Case Trials that are considering using the functionalities of the baseline apps. This initial list will be revised and updated along the progress of the project.

Finally, Section 8 discusses the final remarks and next steps of the work to be conducted in ST451.

2 Product Information Service App

2.1 Overview

The main goal of the PInfS is to enable product information exchange between the stakeholders of a supply chain and towards the final consumer, based on the B2B collaborative and system integration capacities offered by the FIspace platform. The PInfS enables the exchange of product related data (e.g. quality certificates, sensor data and data requests), and finally allow an access to information that flows via several nodes in complex supply networks. The PInfS Baseline App can provide this data in a raw way, or transform it into knowledge by means of rule-setting. The interaction with the FIspace platform will allow diffusing quality alerts related information to end consumers.

Main Features

- Easy and secure exchange of product related information between supply chain partners, both from a technical and business perspective, by avoiding centralised storage of information;
- Facilities for fine-grained access control over own product data by maintaining own data sources with adjusted access management;
- Provisioning of product information from trusted sources (certified by existing business relations from FIspace);
- Federation of decentralised product data sources to increase data availability.
- On-demand and real-time data access and update functionalities reducing duration and effort of data exchange;
- Allow the access to information that is generated/offered by the supplier(s) of the own supplier, enabling to access information spread in the supply network.
- Enabling bidirectional communication through the supply chain;
- By introducing attributes related rules, this App will provide relevant information for a stakeholder based on raw data. The aim is that data stored and gathered from different stakeholders is not directly provided, but transformed by means of rules (an example of rules would be “if saturates are greater than 5,0 gr the level of saturates is high”).

The corresponding infrastructure needs to be provided by both local and FIspace based ICT resources. This Baseline App will support diverse identification schemes and the easy and secure exchange of product related information between supply chain partners towards the consumer, both from a technical and business perspective, enabling bidirectional communication through the supply chain.

Therefore, **the core objectives of the PInfS** Baseline App are:

1. Provide to the single business actor (i.e. the person that interacts with the app)
    - Capacity to add, modify, correct product information that shall be exchanged within the supply chain/network,
    - Access product data that can be introduced by a stakeholder or generated by the PInfS in raw data (data directly gathered from the stakeholders, i.e. data sets or key-value pairs) or knowledge form (e.g. processed raw data by applying rules and patterns).
    -
2. Be compatible to diverse identification schemes,
3. Enable the decentralised storage of product related information.

2.2 Baseline Apps Functionalities

This section describes the main functionalities of the Product Information Service App. These functionalities are expressed in two formats. First in user stories which can be easily understood by the domain partners and a list of functional requirements based on such user stories.

2.2.1 User Stories

The user stories are structured in three levels of hierarch moving from a high abstract or generic functionality to more specific functionalities. These levels are called: epic, feature, and story. For a story we define more concrete tasks that need to be performed to accomplish such story.

Epic 1

Description: As an actor in FIspace I can create, update or consume information related to one product so that other partners can consume it.

Feature 1: As an **actor in FIspace** I can store product data to be provided later to requesting business partners. This feature is not aimed for product or event data included in the business entities defined in ST240, but for specific information that will be stored in each of the stakeholders.

- *Story 1a*: The livestock farmer has information regarding breeding, growth conditions of his/her cows and he/she wants to provide this information to the FISpace.
  - *Task 1: A new identifier for each product entity is requested. The FIspace platform supports aggregation and division of entities in products.*
  - *Task 2*: *The stakeholder defines the information to provide and upload the information through the* ***Product Information Service (PInfS).***
  - *Task 3*: *The stakeholders define data handling permissions for the provided information.*
- *Story 1b*: The farmer or trader is compiling information with respect to its produce (e.g. fruits & vegetables as well as collaborating with laboratories that are taking samples and are providing related analysis results before as well as after the produce is delivered to the next actor (customer like a distribution centre, manufacturer or super market) in the chain. The information shall be made available to their customers.
  - *Task 1: A new identifier is requested that represents a physical objects or an aggregation of objects in a batch. This is combined with the unique identification of the supplier and the related customer.*
  - *Task 2*: *The stakeholder defines and uploads the information that shall be provided through the* ***Product Information Service (PInfS)****, while either the data itself or links to the data are uploaded (PInfS could access diverse data e.g. data stored in EPC IS, i.e. Electronic Product Code Information Services).*
  - *Task 3*: *The stakeholder defines data handling permissions for the provided information.*

Feature 2: As an **actor in FIspace** I can correct or modify information regarding a product.

- *Story 1*: The logistic stakeholder wants to automatically update the information of the carbon footprint information of a box of tomatoes, as the transportation route was longer than expected for this package.
  - *Task 1: The stakeholder reads the product identification from a bar code or QR code.*
  - *Task 2: The PInfS app checks if the stakeholder has permission to access and update this product data field.*
  - *Task 3: The stakeholders updates the data field*
  - *Task 4: The stakeholders check if this modification violates any threshold through the FIspace platform. .*

Feature 3: As an **actor in FIspace** I want to access information regarding a product.

- *Story 1*: A supermarket wants to receive different information regarding box of peppers (growing location, time of harvesting, carbon footprint of transportation, nutritional information, pesticides used, quality certifications, etc).
  - *Task 1: The stakeholder reads the product identification from a bar code or QR code.*
  - *Task 2: The stakeholder checks if he/she has permission to access to the product*
  - *Task 3: The stakeholder accesses to the allowed information fields from the different data sources.*
- *Story 2:* A customer allergic to wheat wants to get information about products that are gluten free:
  - *Task 1: The customer selects the product characteristic „gluten free“.*
  - *Task 2: The stakeholder gets the required product information.*

Epic 2

Description: As an actor in FIspace, I want to configure my instance of the Product Information Service app to provide knowledge created by application of rules and patterns from raw data.

Feature 1: The information regarding products can be filtered and evaluated by rules in order to infer other information more interesting for a stakeholder. For example the data that a product has 5,2 gr of saturates can have no meaning for a final consumer, but if we indicate the information that "*this is a high level of saturates*", this would be more useful. In this sense, it may be required that the information provided by the Product Information service is modified (by the definition of rules and new states in the information based on these rules) and delivered to the appropriate party.

- *Story 1:* The Product Information service may support traffic light labelling. The colour criteria is defined by the Food Standards Agency as in the following graph:

per 100 grams	low (green)	medium (amber)	high (red)
fat	≤ 3,0 gr.	> 3,0 gr. and ≤ 20,0 gr.	> 20,0 gr.
saturates	≤ 1,5 gr.	>1,5 gr. and ≤ 5,0 gr.	> 5,0 gr.
sugar	≤ 5,0 gr.	> 5,0 gr. and ≤ 12,5 gr.	> 12,5 gr.
salt	≤ 0,3 gr.	> 0,3 gr. and ≤ 1,5 gr.	> 1,5 gr.

The product information service can learn the terms "low", "medium" and "high" based on rules according to the previous graph, provides the number of grams of the substance along with the label "low", "medium" and "high".

  - *Task 1: Convert human knowledge (as in the previous table) to a set of rules understandable by the **PInfS**.*
  - *Task 2: The **PInfS** express this rules into a semantic language such as RDF or OWL*
  - *Task 3: The **PInfS** engine assign the execution of these rules only to the **PInfS** instances authorized by the creator of the rules.*
  - *Task 4: When an authorized instance requests data, information and knowledge is provided.*

2.2.2 Requirements based on User Stories

Based on the functionalities described in the user stories, we can define the following functional requirements for the PInfS Baseline App:

Table 3: List of requirements based on user stories of Product Information Service App

ReqID	Title	Description	Epic/ Feature/ Story
PInfS_REQ01	**Product identification**	Each product must be identified by a code that allows to manage their information in the FIspace ecosystem	E1.F1.S1
PInfS_REQ02	**Stakeholder identification**	Stakeholders must be identified so that they can be reached to gather/provide product information in the FIspace ecosystem	E1.F1.S1
PInfS_REQ03	**Product information metamodel**	The product information to be exchanged must be defined, ie. product fat, saturates, sugar, carbon footprint, breeding, growth conditions…	E1.F1.S1
PInfS_REQ04	**Data handling permissions management**	Information will be stored along with data usage policies that describe information access permissions (by which a stakeholder can authorize another to see his/her information). The PInfS provides mechanisms by which various	All

ReqID	Title	Description	Epic/ Feature/ Story
		types of access to objects and containers are authorized and permitted or denied. Access grants are bound to users or groups.	
PInfS_REQ05	**System Integration**	Implementa- tion of communication channels between the FI space and external business and legacy systems (e.g. in-house logistics solutions, ERP systems) The integration tools will be provided by the 250 team, via an SDK provided by 280 which will actually bundle all 220,240, 250 and 270 tools provided by the respective teams.	All
PInfS_REQ06	**Define product info**	Define information to provide	E1.F1.S1
PInfS_REQ07	**Store product info**	*Add/modify* the product related information	E1.F1.S1
PInfS_REQ08	**Info security**	Define which stakeholders can access which information	E1.F1
PInfS_REQ09	**Threshold validation**	Check if the info update violates any threshold.	E1.F2.S1
PInfS_REQ11	**Consumer profile definition**	So as to match the consumer's interests, it will be necessary to create a consumer profile in order to know what information the consumer is interested in	
PInfS_REQ12	**Consumer profile security**	The consumer is the owner of his/her profile and the retailer can consult the profile only if the consumer gives access. So, profile information must be protected, authorizing the access to specific parts of the profile (depending on the user se-lection) and setting the information as anonymous if necessary, etc.	E1.F3/E2
PInfS_REQ13	**Tailored information provision**	The information provided to the consumer must be adapted to the consumer’s profile	E1.F3
PInfS_REQ14	**Rule definition**	Define rules to transform product data into more comprehensive information for consumers	E2.F1.S1
PInfS_REQ15	**Rules execution**	Execute rules engine to provide required information	E2.F1.S1
PInfS_REQ16	**Security execution**	The engine assigns the execution of these rules only to the PInfS instances authorized by the creator of the rules.	All

ReqID	Title	Description	Epic/ Feature/ Story
PInfS_REQ17 (Optional)	**QR & Bar code identification**	QR & Bar code must be converted to a unique product code that will be sued by tPInf_REQ01 Access to the mobile camera must be supported.	E1.F3.S1

2.3 Relationship and Dependencies with WP200 Components

The following figure summarizes the initial approach of the relations of the Product Information Service with WP200 components. The relationships stated here about WP200 components to be used by each requirement are good estimations at this point that will have to be validated as the components definition evolves. The Real-time collaboration system from T240 will be used by the PInfS mainly for discovering the stakeholder that is the owner of the product data in order to request product information or product modification queries.

The Systems & Data integration system from T250 will be used to create communication channels between the FIspace and external business and legacy systems, so that the product information service can access to the stored product data.

The Operating Environment from task T260 provides a middleware solution for communication with the rest of the FIspace platform. Thus, in the figure, the T260 module represents the way for the PInfS to access the rest of FIspace functionalities.

Finally, the Security, Privacy and Trust system (T270) affects each one of the modules of FIspace and the communication to these modules. Therefore it is painted as a transversal system in Figure 2. In relation to stakeholder identification, it must be further analyzed in WP200, but as an initial approach it can be understood that stakeholders would be identified in T270 and PInfS would locate them through T240.

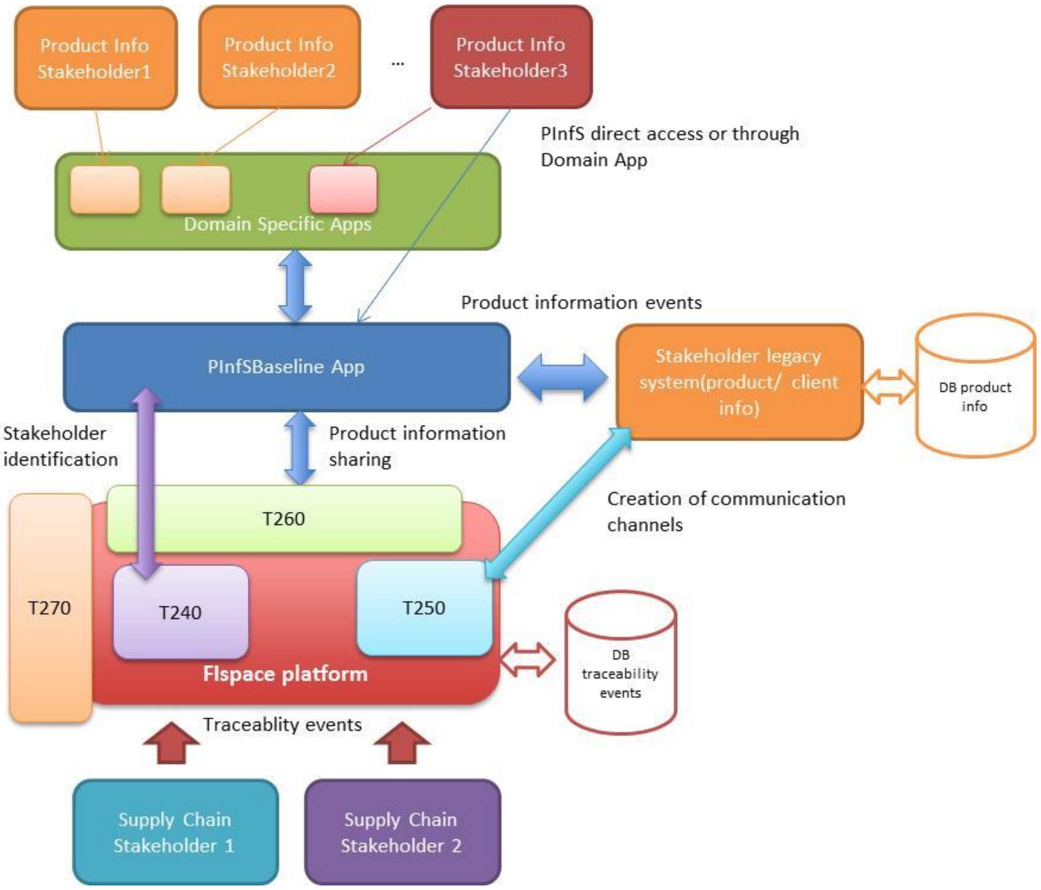

Product Info Stakeholder1
Product Info Stakeholder2
...
Product Info Stakeholder3
PInfS direct access or through Domain App
Domain Specific Apps
Product information events
PInfSBaseline App
Stakeholder legacy system(product/ client info)
DB product info
Stakeholder identification
Product information sharing
T260
Creation of communication channels
T270
T240
T250
FIspace platform
DB traceability events
Traceablity events
Supply Chain Stakeholder 1
Supply Chain Stakeholder 2

Figure 2: Relations of PinfS with WP200 components

Table 4 summarizes the mapping requirements of PInfS into modules of WP200.

Table 4: Mapping Requirements of the Product Information Baseline App into modules of WP200

Req. ID	Req. Name	ST240	ST250	ST260	ST270
01	Product Identificattion		Unique identification of products	Unique identification of products	x
02	Stakeholder identification	Stakeholder location			Stakeholder Repository
04	Data handling permissions management	Access rights over specific instances and or states of the product		Access to data access policies	Data authorization
05	System integration		Communication channels	ESB-support for channel stablishment	x
07	Store Product Info		Connection to legacy systems or new product storage solution		x
09	Threshold validation	Provide data for threshold violations. Detection of threshold violations.			x
12	Consumer profile security				Data authorization, anonymization
15	Rules execution	Rules definition and application		Rule interchange among participants	
16	Security execution				Execution permissions

Furthermore, below we present a detailed analysis of the dependencies and relationships between the PInfS App and the components of WP200 most relevant for this app.

- **Relation with Real-Time B2B Collaboration (T240)**

  This task is concerned with implementing the FIspace Real-time B2B collaboration concept. Its purpose is to manage, execute and monitor collaborative processes at the heart of FIspace. This task will define a collaboration object lifecycle and implement the event processing component to enable proactive event processing (such as proactively handling expected delays or deviations in a transport plan execution). The task includes the development of interoperability components between the collaboration and event processing components to include event-based rule generation, impact of events on collaboration objects, generating appropriate events by the collaboration management component for tracking, and more. Furthermore, the task includes work on supporting specific types of collaboration needs such as transport plan execution. This includes the ability to interpret such a collaboration in its own standard form, instantiate collaboration objects accordingly (using the results of the other subtasks) and putting in place appropriate tracking and monitoring rules and patterns to drive and manage the collaboration including deviations and exceptions.

The Product information service benefits from the B2B Collaboration system, as it will use the information provided by this component **to locate the stakeholder that is the owner of product information, in order to request the information directly to it**. This will permit to request information access to the owner of data, and also implement data access and modification policies.

PInfS products must be connected to the set of existing collaborative process that manages these objects. The rational of this connection is that once introducing a specific data field in a product definition, this could invoke an exception report to any actor in the chain concerning a specific delivery. This is based on the requirement REQ09 that states that **PInfS must check if the info update violates any threshold**.

An exception could identify a problem that would limit/restrict the usage/ consumption of the produce (e.g. contamination with bacteria or pesticides), identify a delay or describing a non-expected quality as well as a recently unexpected development of the decay/quality ratio. As some exceptions affects the service level agreement established with the supplier or customer, **the product information service must provide the address and identification of the exception reporting** module that has access to the SLA management functionality in service ecosystem.

- **Relation with Systems & Data Integration (T250)**

This task is concerned with creating a communication data exchange channel between the FIspace and other external systems/sensors in an easy-to-use and bidirectional way, therewith enabling the connection of existing systems with the FIspace in order to allow for the import and export of data and relevant information. The goal in this task is to provide a technical infrastructure to support the integration of the business and legacy systems, of the external services (in particular of IoT-enabled sensor networks and systems), including facilities for data integration and mediation to handle heterogeneous information.

The Product Information Service will use this technical infrastructure to implement **the communication channels between the FIspace and external business and legacy systems** (e.g. in-house logistics solutions, ERP systems), as stated in REQ05.

Storage of information shall be generally realized in the virtual realm of the data owner, which will decide on the cases and rules on how to share the information. This might result in different access rights with respect to different data items of a group of data as well as different access rights for different direct/ indirect business partners. This takes into account the organization internal access and the access by authenticated external parties.

As identified before, there is a **need for a unique identification of (aggregated) objects**. As some times the business entity in the chain uses its specific identification system, by the relation with Systems & Data Integration, the PInf assures unique identification when receiving the product data.

- **Relations with Operating Environment (T260)**

This task is concerned with developing the technical infrastructure that shall allow all of the FIspace components to work together in harmony. The main goal is to support SOA principles on the Cloud that is FI-WARE based, while the solution is distributed and allows “local” decision making with incomplete information.

The FIspace Operating Environment service from this task will be needed to connect all FIspace modules, components and services through a cloud-based enterprise service bus (ESB). This service bus acts as a middleware solution that incorporates different communication paradigms. The P2P communication paradigm is useful in our environment as it integrates coordination and communication mechanisms to facilitate the interchange of data in a decentralized approach.

The publish/subscribe paradigm provides time (subscribers can receive messages in a different time when publishes send the data) and space (publishes do not need to know the communication address of subscribers in order to send relevant information) decoupling. The implementation of a pub/sub communication model allows the implementation of notification systems, exception reporting systems, and also sensor networks in an efficient way. **The Product Information Service baseline application will use this middleware solution for the communication with the rest of the FIspace platform through the different communication options that will be implemented**.

- **Relations with Security, Privacy and Trust (T270)**

  This task concerned with providing employment of adequate security, privacy and trust management techniques in order to ensure the secure and reliable exchange of confidential business information and transactions in FIspace. Security, privacy and trust (SPT) technologies and mechanisms from FI-WARE and service providers shall be integrated, while deploying and integrating the security mechanisms and assurance methods. Test scenarios of the Authentication, Authorization and Accounting (AAA mechanisms) shall provide secure access along with a developer framework and a secure architecture to enable the use of security privacy & trust mechanism and integration to the relevant security mechanisms implemented. Finally, a user management & access control for the relevant components of FIspace will be realized.

  To allow actors to easily provide and access product related data, the described network should be open available in the internet. This requires the **need of identification of users and an authentication and authorization mechanism to control the access to data and decide about the trustworthiness of an information source from an ICT point of view**. Furthermore it shall specifically provide services that allow the lookup mechanism needed for the realization of the PInfS that are as follows:

  - Authorized querying for information
  - Authenticated communication (querying/propagation)
  - Respect the privacy constraints of the actors.

  Relations with Security, Privacy and Trust will support the **flow of information among the different systems in the agri-food supply chain using standarised means for interoperability, security and authorization schemes** (i.e., different stakeholders will have different access permission privileges when accessing data) This dynamic formation of links in this business environment also requires enabling “trust” among the involved entities.

2.4 Relationship and Dependencies with Generic Enablers

In order to provide the relationship of PInfS Baseline App with the FI-WARE Generic Enablers, below is presented a mapping between the functional requirements presented in the table above and the FI-WARE GEs. These GEs are suggested as potential components, some of those that will be finally used may be accessed through the FIspace platform,

Table 5: PInfS requirements on data handling permissions management

ReqID	Title	Description	Epic/ Feature/ Story
PInfS_REQ04	**Data handling permissions management**	Information will be stored along with data usage policies that describe information access permissions (by which a stakeholder can authorize another to see his/her information). The PInfS provides mechanisms by which various types of access to objects and containers are authorized and permitted or denied. Access grants are bound to users or groups.	All

PInfS_REQ04 - Data Handling GE

This GE provides a GUI where the stakeholder can attach his privacy preferences to the data he wants to share. If a third party server wants to access this data, his query will be filtered according to the privacy policy attached to the data.

The following steps demonstrate the functionality of the Data Handling GE, in relation to the PinfS_REQ04: “Data handling permissions management”. Data Handling GE works as some kind of middleware between the UI and the Cloud infrastructure where data will be stored.

- The stakeholder wants to upload data on FIspace Platform.
- The stakeholder, through his web browser, can see a user interface where he can choose which files wants to upload as well as define the sticky policy. For example, he can define a list of stakeholders who can access his data, the deletion period or life time of his data in the store, and some kind of contact (i.e. email address) so as the Data Controller of Data Handling GE can notify him on the activities and events happened to his data.
- Data is stored in the FIspace platform, along with the sticky policy. Data Controller component of Data Handling GE is responsible for the storage.

The developer of PInfS Application will have to use a RESTful API for managing Data Handling GE available functionalities, such as StorePii(), UpdatePii() etc.

PInfS_REQ04 – Gateway Data Handling GE

Gateway Data Handling GE is basically a Complex Event Processor, based on Esper library for Java. It provides a RESTful management API for the configuration and management of its components. It is flexible in the sense that you define your own xml-based event types, as well as the rules that apply on these events, and the actions that are performed when these rules trigger. The rules are EPL based, which is a form of SQL like language.

According to rules that the stakeholder chooses, data is processed and exchanged wherever it is allowed. This secure procedure starts from the things and sensors at a low level and can proceed to a higher level.

Table 6: PInfS requirements on Product info storage

ReqID	Title	Description	Epic/ Feature/ Story
PInfS_REQ07	**Store product info**	*Add/modify* the product related information	E1.F1.S1

PInfS_REQ07 – Object Storage GE

If the information is stored in file formats (e.g. RDF), Object Storage GE can be used to store the files in the cloud.

PInfS_REQ07 – Publish/Subscribe Context Broker GE – Context Awareness Platform

Possibly the information to be stored can be firstly published to a pub/sub context and then each component that is registered to this context can receive the information and store it.

Table 7: PInfS requirements on security/profile management

ReqID	Title	Description	Epic/ Feature/ Story
PInfS_REQ08	**Info security (T270)**	Define which stakeholders can access which information	E1.F1

ReqID	Title	Description	Epic/ Feature/ Story
PInfS_REQ12	**Consumer profile security (T270)**	The consumer is the owner of his/her profile and the retailer can consult the profile only if the consumer gives access. So, profile information must be protected, authorizing the access to specific parts of the profile (depending on the user se-lection) and setting the information as anonymous if necessary, etc.	

PInfS_REQ08, PinfS_REQ12 – Data Handling GE

Data Handling GE enables the specific product information owner to define who is going to access which of his/her data. *See PinfS_REQ04 for more information.*

Table 8: PInfS requirements on rules definition & execution

ReqID	Title	Description	Epic/ Feature/ Story
PInfS_REQ14	**Rule definition**	Define rules to transform product data into knowledge	E2.F1.S1
PInfS_REQ15	**Rules execution**	Execute rules engine to provide required knowledge	E2.F1.S1

PInfS_REQ14, PinfS_REQ15 – Complex Event Processing (CEP) – IBM Proactive Technology Online

In order to extract some useful knowledge for specific stakeholders about product related information not stored in business entity inside the FIspace platform, rules could be defined to filter this information. For the traffic light labeling Domain Specific App support of PinfS realisation, the Complex Event Processing (CEP) GE could be used.

REST calls can be used to place input values into CEP, while some rules have already been defined.

Rules follow the idea of the following format: "IF *Temperature* > 5 && *User* == Franz Farmer THEN *an_action*", while the REST call shall input something like "*Temperature* = 10; *User* = Franz Farmer".

In brief, the following procedure should be followed:

- Via a web interface a configuration file is created in which we define
  - The REST interface from which the input is expected
  - The rules to be monitored
  - The REST interface to which the output is sent, after an event has occurred
- Afterwards, the configuration file is sent via SCP to the VM where CEP is running and a script is being executed for the whole process to begin.

PInfS_REQ14, PinfS_REQ15 – Gateway Data Handling GE

Similarly to the above elaboration on CEP, the Gateway Data Handling GE can be used for the feature of defining and executing rules.

2.5 Preliminary Test Requirements and Relationship and with WP300

The test of the PIfnS Baseline App will base on the experimental support provided by WP300. WP300 will provide a cloud infrastructure in which the GEs and the FIspace platform will be deployed, and which the PInfS App will make use of. WP300 will also provide support on experiment execution and analysis, and the required experimentation environment to test FIspace Baseline Apps, allowing the use of real or simulated data in real or non-real time.

The test of the Baseline Apps will be done prior to the trial experimentation. In the PInfS, from a high-level perspective, the procedure that we plan to follow is:

1. Use simple test data sets (no real data, not in real time) to test the functionalities, without real connection to external info.
    a. The data sets will include different product attributes that may include some of the following:
        - Provider
        - Location
        - Local product
        - Ecological crop
        - Recycled packaging
        - CO2 footprint
        - Protected designation of origin (DOP)
        - Protected geographical indication (IGP)
        - Pesticides
        - Additives
        - Preservative
        - Quality certificate
        - Best before date
        - Contains Eggs
        - Contains Dry fruits
        - Contains Lactose
        - Contains Gluten
        - Transgenic
        - Production date
        - Traceability info
        - Season
        - Raising
        - Animal welfare certificate
        - Type of meat (kosher, halal…)
        - Especially in case of produces
            1. Origin (country, region)
            2. Place and date of harvest
        - Product identification (and delivery reference)
        - Certification scheme(s)
        - Analytical extermination(s)
        - …

2. Use real datasets from the trials in a small scale experiment.
    a. Real data from the FFV, Greenhouse and TIC trial will be used, but not in real time.
    b. It will be required the simulation from the platform traceability events and product information generation.

For a test and experimentation of the Product Information Service (PInfS) some parts of the workflow and the data need to be simulated to not disturb any processes in legacy systems as well as change any existing data. Therefore, a test setting is considered that simulates product data without a connection to any running legacy systems of stakeholders. Other functionality will be provided by the FIspace platform and its components (WP200) see Figure 3. The first tests will focus on the communication and data exchange between platform users as well as between platform services/applications and (simulated) existing/legacy systems. Processes inside of enterprises (e.g. ERP systems) will be treated as black boxes. Furthermore, the integration and processing of information will be focussed on establishing a representation of real world activities on the platform.

Applications and test cases will build upon functionalities provided by the FIspace platform, e.g. managing business relationships, creating business collaboration objects, hosting of data and managing of data access rights. Figure 3 shows the relationship of the PInfS to other platform components and external systems.

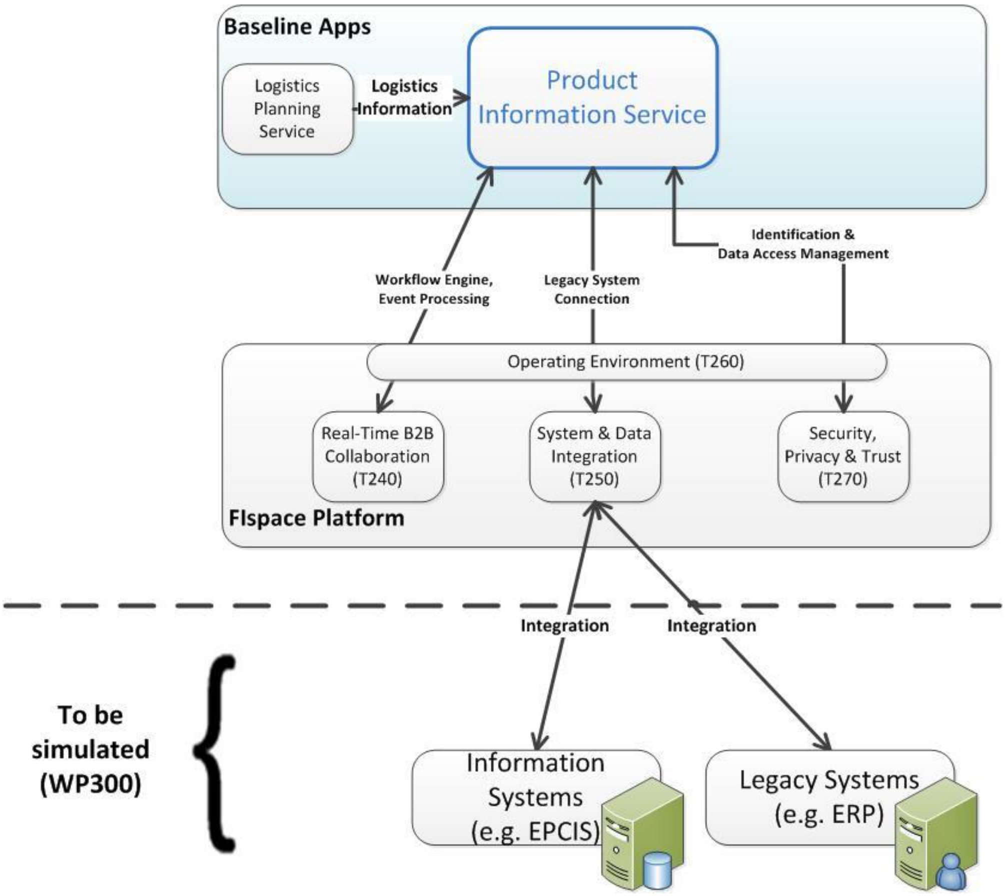

Baseline Apps
Logistics Planning Service
Logistics Information
Product Information Service
Identification & Data Access Management
Workflow Engine, Event Processing
Legacy System Connection
Operating Environment (T260)
Real-Time B2B Collaboration (T240)
System & Data Integration (T250)
Security, Privacy & Trust (T270)
FIspace Platform
Integration
Integration
To be simulated (WP300)
Information Systems (e.g. EPCIS)
Legacy Systems (e.g. ERP)

Figure 3: FIspace Components and external Systems for Test and Experimentation of the PInfS

For the simulated working environment a set of real data is preferred because this will improve the representation and understanding of real world requirements and is easy to manage if one stakeholder is willing to provide such data for testing purposes. The product test data need to be accessible via the internet (e.g. by a kind of data access layer to simulated systems).

Particularly the communication between FIspace and external systems needs to be validated and tested. For simulating data exchange with legacy systems of stakeholders (e.g. existing ERP systems) a reference system needs to be provided with similar (simulated) data access interfaces as the real legacy systems. As an example an EPCIS (Electronic Product Code Information Services) system could serve for emulating stakeholders’ back-end systems. The EPCIS provides required features (e.g. web service interfaces) and is very common while focussing on storing particular event types according to the EPCIS standard.

Other very essential components to be tested will be the workflow and event processing engines provided by task 240 because these are essential information sources about product identification and events. Therefore it is preferable to have an at least partially implemented version of these two FIspace platform components available. Other platform services and components are not of primary interest for the first tests of the PInfS, e.g. Security/ Privacy & Trust and Front-End, but of course they will be part of the test and experimentation.

Regarding deviations of product characteristics and transports the PInfS will try to use the Logistics Planning Service as additional information source for starting re-planning.

2.6 Summary

As we have stated, the main goal of the PInfS is to enable product information exchange between the stakeholders of a supply chain and towards the final consumer, based on the B2B collaborative and system integration capacities offered by the FIspace platform. The PInfS Baseline App can provide this data in a raw way, or transform it into knowledge by means of rule-setting. Moreover, the interaction with the FIspace platform event handling capabilities will allow diffusing quality alerts related information to end consumers.

So as to fulfill its purpose, the PInfS App must cover a set of requirements identified in previous sections, some of which will be provided by the FIspace platform (product and stakeholder identification, system integration, security…). The Systems & Data integration system from T250 will be used to create communication channels between the FIspace and external business and legacy systems, so that the product information service can access to the stored product data. The Operating Environment from task T260 provides a middleware solution for communication with the rest of the FIspace platform. Finally, the Security, Privacy and Trust system affects each one of the modules of FIspace and the communication to these modules.

In relation to GEs usage, a first analysis on the potential GEs to be used to fill these requirements has been carried out and as a first approach we identified the following: Data Handling GE, Gateway Data Handling GE, Object Storage GE, Publish/Subscribe Context Broker GE – Context Awareness Platform,

Complex Event Processing (CEP), which will be accesses through the FIspace platform.

Finally, for test and experimentation of the Product Information Service (PInfS) some parts of the workflow and the data need to be simulated to not disturb any processes in legacy systems as well as change any existing data, this will be done with the experimental support provided by WP300.

3 Logistics Planning Service App

3.1 Overview

The Logistics Planning App supports the generation and maintenance of transport chain plans by using the description of the demand, real time information of logistics services, real time contract information, execution statuses, and details of relevant events, as shown in Figure 4.

For the logistics service client (LSC), the baseline app supports the building of a transport chain plan based on the client's demand and online available services, using the latest available information for service descriptions.

For the logistics service provider (LSP), the baseline app enables the description of transport services that can be used by the LSCs, enables the use of marketplaces to find demands that match the provided services during planning time, as well as for planning the use of subcontractors.

For both clients and providers, the baseline app will have facilities for detailing out the execution plans and for negotiation and booking of the services. The app will also provide functionality for replanning, in case the execution of the original plans fails. In such case, user receives a notification indicating there is a problem. If needed; user can manually do the replanning by using LPA application.

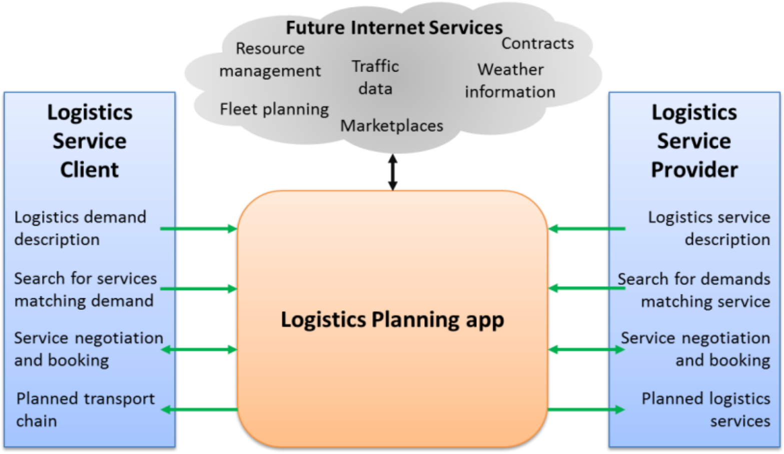

Future Internet Services
Resource management
Traffic data
Contracts
Weather information
Fleet planning
Marketplaces
Logistics Service Client
Logistics demand description
Search for services matching demand
Service negotiation and booking
Planned transport chain
Logistics Planning app
Logistics Service Provider
Logistics service description
Search for demands matching service
Service negotiation and booking
Planned logistics services

Figure 4: Logistics Planning App Overview

3.2 Baseline Apps Functionalities

This section described the main functionalities of the Logistics Planning App (LPA). These functionalities are expressed in two formats. First in user stories which can be easily understood by the domain partners and a list of functional requirements based on such user stories.

3.2.1 User Stories

The user stories are structured in three levels of hierarch moving from a high abstract or generic functionality to more specific functionalities. These levels are called: epic, feature, and story. For a story we define more concrete tasks that need to be performed to accomplish such story.

Epic 1: LSC Functionality

Description: As a logistics service client using FIspace, I can set up a transport plan so that service and contract searches and bookings can be done automatically.

Feature 1: As a logistics service client I can enter a Logistics Demand Description so that a set of possible services/legs/transport plans fulfilling the demand can be fetched from LSPs and marketplaces based on real time information on availability of services and real time information on relevant contracts.

- *Story 1*: Inbound logistics from Korea to Turkey: A LSC enters a transport demand for the transportation of washing machine motors to be sent from a factory in South Korea to a factory in Turkey.
  - *Task 1: The transport demand is set up based on a given purchase order. The transport demand covers at least the pickup and delivery places and times, and a description of the actual goods.*
  - *Task 2: The LSC can choose to publish the demand on a marketplace for matching by some LSP's services (done independent of task 3).*
  - *Task 3: The LSC continues with the planning of the demand by identifying the priorities that he wants the app to list the alternatives in (best price, shortest time, etc.).*
- *Story 2:* Fish transport from Norway to Brazil: A fish exporter enters a transport demand for a certain amount of containerized dried fish based on an export order.
  - *Task 1: The transport demand is set up by specifying the pickup and delivery locations and times, and a description of the actual goods.*
  - *Task 2: The LSC can choose to publish the demand on a marketplace for matching by some LSP's services.*
  - *Task 3: The LSC continues with the planning of the demand by identifying the priorities that he wants the app to list the alternatives in (best price, shortest time, etc.).*

Feature 2: As a logistics service client I can select a transport plan generated by the app to complete service negotiation and booking.

- *Story 1*: Select a transport plan containing a service based on a long-term contract
  - *Task 1*: *Since this service is based on a long-term contract, it is probable that the LSC can just confirm the booking.*
  - *Task 2: Confirm or reject the complete, selected transport plan.*
- *Story 2*: Select a transport plan containing a service based on a spot-market offer:
  - *Task 1*: *The LSC must examine the service offer closely before confirming the booking, or reply to the offer by setting it non-confirmed.*
  - *Task 2: Negotiation must be done.*
  - *Task 3*: *Eventually confirm or reject the service offer.*

Epic 2: LSP Functionality

Description: As a logistics service provider using FIspace, I can enter logistics services so that real time service information is made available to the LSCs during logistics planning.

Feature 1: As a logistics service provider I can enter a Logistics Service Description.

- *Story 1: As a forwarder, I want to publish a service covering container transport of dried fish from Ålesund to Rotterdam, including a storage service, loading service, and feedering service.*
  - *Task 1: Set up a complex service containing all three different services, for loading, storing, and feedering transport.*

Feature 2: As a logistics service provider I can search for demands matching a service.

- *Story 1: A fish exporter has to change its original export and transport of dried fish from Ålesund, Norway to Rotterdam, and enters this updated demand on a Marketplace. The LSP enters their voyage schedule to the marketplace and gets back a list of possible demands that fits the actual service with a given probability.*
  - *Task 1: The LSP adds service details to a marketplace to try to find matches.*

Feature 3: As a logistics service provider I can do service negotiation to achieve a confirmed booking.

- *Story 1: A container line carrier has found a matching demand that he wants to get confirmed.*
  - *Task 1: Send offer to the LSC*
  - *Task 2: Receive reply from the LSC, and generate response*
  - *Task3: Continue until confirmed or rejected.*

Epic 3: Replanning

Description: As a LSC using FIspace, I can replan a logistics transport plan so that events and changes reported on the execution are taken into account.

Feature 1: As a logistics service client I receive a transport plan together with the current execution status and a list of events saying what the problem is.

- *Story 1*: A forwarder[2] gets notified that a cargo containing dried fish from Ålesund to Rotterdam will be delayed due to a missing import license.
  - *Task 1: The existing booking is checked, and possibly cancelled if the service cannot be completed within the new conditions.*
  - *Task 2: Changes related to the cancelled booking must be done (changes in loading list, changes in discharge list etc.)*

Feature 2: As a logistics service client I set up a new plan based on given details, and commits the new plan which is forwarded to FIspace for further handling and execution.

- *Story 1*: The forwarder sends a new request to the carrier, and starts negotiation.
  - *Task1: Create a new offer and send to carrier.*
  - *Task 2: Complete negotiation of the booking.*
  - *Task 3: Complete booking*

3.2.2 Requirements based on User Stories

Based on the functionalities described in the user stories, we can define the following functional requirements for the LPA:

Table 9: List of requirements based on user stories of Logistics Planning App

Requirement-tID	Title	Description	Epic/Feature/ Story
LPA_R001	**Check user credentials**	Check user credentials if user is signed on, has an access to LPA, and right to manage certain aspects	All
LPA_R002	**Create offer or demand**	Create or re-plan offer (service desc.) or demand	E1-F1-S1 E1-F1-S2 E2-F1-S1
LPA_R003	**Search and match offers and demands**	Search/match algorithm for offers and demands	E1-F1-S1 E1-F1-S2 E2-F2-S1
LPA_R004	**Set priorities**	User set priorities (customization of best price, shortest time etc)	E1-F1-S1 E1-F1-S2
LPA_R005	**Send offer to LSC**	Send offer to LSC	E2-F3-S1
LPA_R006	**Send demand to LSP**	Send demand to LSP	E1-F2-S1 E1-F2-S2 E3-F2-S1

[2] A Forwarder can have the role of both a LSC and a LSP.

Requiremen-tID	Title	Description	Epic/Feature/ Story
LPA_R007	**Negotiation**	Negotiation interactions between client and provider	E1-F2-S2 E2-F3-S1 E3-F2-S1
LPA_R008	**Confirmation or rejection of offer**	Confirmation or rejection of an offer	E1-F2-S1 E1-F2-S2 E2-F3-S1 E3-F2-S1
LPA_R009	**Send plan to execution**	Send plan to execution (B2B Collobration Module)	All
LPA_R010	**Multiple plan owners**	A transport chain plan can have one or more owners	E1-F1
LPA_R011	**Change of plan owner**	The owner of the transport chain plan can change, for instance, from the shipper to the for-warder.	E1-F1
LPA_R012	**System integration**	System integration to legacy booking systems	
LPA_R013	**Data integration**	Mapping between LPA internal formats and EDIFACT messages	
LPA_R014	**Receive transport plan status for replanning**	Receive transport plan status from "B2B Collaboration Module" if replanning is necessary	E3-F1-S1
LPA_R015	**Receive transport plan status when chain plan is ready**	Receive transport plan status notification from "B2B Collaboration Mod-ule" if transport chain plan is ready	
LPA_R016	**Publish demand**	Publish demand on e.g. a marketplace	E1-F1-S1 E1-F1-S2
LPA_R017	**Publish service description**	Publish service description on e.g. a marketplace	E2-F1

3.3 Relationship and Dependencies with WP200 Components

Even though LPA baseline application will handle most of the requests internally, there are still some functionalities require LPA to communicate with other components of FIspace WP200 tasks. The table created below contains possible tasks that LPA supposed to handle certain requirements. 'X' refers that associated subtask of WP200 can be used to achieve the objective. 'E' is used to point out that ESB happens to manage all the interactions and each task should use it. 'S' refers to Security, Privacy and Trust and many tasks supposed to use SPT attributes to control user credentials, authentication etc.

Table 10: Mapping requirements of Logistics Planning App into WP200 components

RequirementID	Req. Name	ST220	ST230	ST240	ST250	ST260	ST270	ST280
LPA_R001	Check user credentials if user is signed-on, has an access to LPA, and right to manage certain aspects					E	S	
LPA_R002	Create or re-plan offer (service desc.) or demand					E		
LPA_R003	Search/match algorithm for offers and demands					E		
LPA_R004	Identifying priorities (customization of best price, shortest time etc)					E		
LPA_R005	Send offer to LSC					E		
LPA_R006	Send demand to LSP					E		
LPA_R007	Negotiation interactions between client and provider					E		
LPA_R008	Confirmation / rejection of an offer					E		
LPA_R009	Send plan to execution (B2B Collobration Module)			X		E		
LPA_R010	A transport chain plan can have one or more owners					E	S	
LPA_R011	The owner of the transport chain plan can change, for instance, from the shipper to the forwarder.					E	S	
LPA_R012	System and data integration: System integration to legacy booking systems				X	E		
LPA_R013	Data integration to legacy booking systems: Map-				X	E		

RequirementID	Req. Name	ST220	ST230	ST240	ST250	ST260	ST270	ST280
	ping between LPA internal formats and EDIFACT messages							
LPA_R014	Receive transport plan status from "B2B Collaboration Module" if re-planning is necessary			X		E		
LPA_R015	Receive transport plan status notification from "B2B Collaboration Module" if transport chain plan is ready			X		E		

3.4 Relationship and Dependencies with Generic Enablers

Some GEs will be used indirectly by the logistics planning baseline App via the WP200 Modules and via the other baseline Apps.

LPA will use functionalities provided by Marketplace Operations baseline App which will provide the necessary functionality to the business actors to announce as well ask for offers and demands according to their specific criteria and preferences. Automatic mechanisms will also be supported for matchmaking between announced and requested services. LPA will use Marketplace GE and Repository GE indirectly via Marketplace Operations baseline App.

Logistic service provider can enter description of the logistics services offerings in LSP and Logistic service client can enter logistics demand descriptions in the LSC baseline application. In order to provide these functionalities, uploading offer or demand information functionality of Marketplace Operations baseline App is used.

Logistics service provider using LSP App can search for demands matching a service and Logistic Client using LSC App can search for service provider offerings according to specific consumer requirements.

LPA baseline App user can re-plan a logistics transport plan so that events and changes reported on the execution are taken into account. Generation of events of expected delays or deviations in a transport plan execution is handled B2B collaboration subtask of WP200. Purpose of B2B collaboration subtask is to manage, execute and monitor collaborative processes. LPA needs to make necessary interactions with B2B collaboration module of FIspace for submitting the transport plan and receive events on the execution.

Table 11: Mapping requirements of Logistics Planning App into GEs

Requirement ID	Req. Name	GEs considered to be indirectly used via Marketplace Operations App	GE considered to be indirectly used via WP200 components
LPA_R001	Check user credentials if user is signed-on, has an access to LPA, and right to manage certain aspects		OneIDM GE

LPA_R002	Create or re-plan offer (service desc.) or demand	Marketplace GE Repository GE	
LPA_R003	Search/match algorithm for offers and demands	Marketplace GE	

3.5 Preliminary Test Requirements and Relationship and with WP300

Functional and System Testing

Functional testing involves ensuring that the functionality specified in the requirement specification works. System testing involves putting the new program in many different environments to ensure the program works in typical customer environments with various versions and types of operating systems, devices and applications. System testing is testing conducted on a complete, integrated system to evaluate the system compliance with its specified requirements. Both of these tests should be done by unbiased testers, therefore programmers shouldn't take part in any of the testing process.

After functional testing is completed, there are three stages to be completed for the system testing; these stages can confirm if the software is ready for beta version or not.

Stress testing

Testing conducted to evaluate a system or component at or beyond the limits of its specification or requirement. For example; multiple testers should create multiple demands for the same provided service constantly to see, if LPA can handle the stress and still manages to provide acceptable solution to clients.

Performance testing

Testing conducted to evaluate the compliance of a system or component with specified performance requirements. Since there will be constant requests from the system, performance should be tested. Each individual case takes n amount of time to be executed; if the execution takes longer than expected it should be reported as a bug.

Usability testing

Testing conducted to evaluate the extent to which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component. If LPA user is having hard time understanding the user interface, or spending too much time to achieve targeted process then tester should discuss the problem with developer or designer.

Unit Testing

Unit testing is the testing technique; testers (usually the developers creating the code implementation) verify that the code or queries do what it is intended to do and if correctly optimized at a very low structural level. For example, the tester wants to control if "LPA's demand screen" is working properly. When available, the tester will examine the low-level design of the code; otherwise, the tester will examine the structure of the code by looking at the code itself. Unit testing is generally done within a class or a component.

Preliminary Test Requirements

Initial plan for testing the LPA baseline application prior to the actual testing should include the following steps;

Preconditions/requirements

- WP200 modules used by the app should be already tested
- Marketplace & Repository GEs already set-up & configured

- Potential Legacy/External Systems connected (if finally external marketplaces will be used for example)

Happy Paths

- Defining step by step actions per feature and task
- Clarifying expected outcome per defined step.

Test Data

LPA testing requires data to be available to test all the defined cases.

Test Input Data

Testing of LPA already requires LSC and LSP to provide certain data to begin with. Therefore, there is no need for preset data to be ready for testing creation of offer and demand testing. It would even be better not to have any preset data, so we can test our cases by easily entering new data for offers and demands. Once there are enough data, we can continue with the rest of the cases. However, regardless the case, having real data (not necessarily real time) with real life example would make a positive difference during testing.

In addition, test data preferably should be real data from the real life scenarios and companies. If FIspace project cannot access such data, LPA should still request similar data to achieve successful test results.

Not all the requirements/functionalities of LPA baseline app requires specific datasets to be tested.

3.6 Summary

Using the features and tasks explained in the overview of this document, functional requirements of LPA were defined. LPA baseline application creates environment for clients and providers to find the best suitable solution for their logistics needs without extra man effort. Also, LPA uses matchmaking services of Marketplace Operations App to allow users to find most suitable match so that the LAP can plan the logistics services among the partners of those matches.

Potential connection and dependencies to the modules, GEs and infrastructure were defined and initial test plan and requirements were explained.

Development of the LPA would be possible by starting the technical alignment with Marketplace Operations App, modules and GEs which are planned to be used. Preliminary testing place were created and needs were defined.

4 Business Profile App

4.1 Overview

This baseline app is designed to enable of business partners joining FIspace Platform to create their profile. In this profile, business partners can describe their interests and business scope. The ultimate goal of this app is to enable an automatic discovery of potential business partners with aligned business interests and scope.

4.2 Baseline Apps Functionalities

This section described the main functionalities of the Business Profile App. These functionalities are expressed in two formats. First in user stories which can be easily understood by the domain partners and a list of functional requirements based on such user stories.

4.2.1 User Stories

The user stories are structured in three levels of hierarch moving from a high abstract or generic functionality to more specific functionalities. These levels are called: epic, feature, and story. For a story we define more concrete tasks that need to be performed to accomplish such story.

Epic_1

Description: As an actor in FIspace I can announce myself within FIspace Platform so that I become visible to other stakeholders inside the platform.

Feature_1.1: As an actor in FIspace I can define my business profile in FIspace platform and be aware of other actors' profiles also belong to FIspace Platform (The stakeholder connects to FIspace Platform. Through Business Service App, there is a GUI where stakeholder can create his/her basic profile that identifies such a stakeholder and can get information about other stakeholders in the FIspace Platform for future collaboration.)

- ***Story_1.1.1***: As an actor in FIspace I can define my business profile so that other potential partners could know that I am a stakeholder of the business sector/domain**.**
  - Task 1: Define which initial set of actors in specific domains that will be served for this app
  - Task 2: Identify the minimum set of information describing an actor profile or find already defined data models to describe actors based on Linked-data
  - Task 3: Model the set of information following the Linked USDL[3] approach
- ***Story_1.1.2***: As an actor in FIspace I want to associate different users to a single business profile so that I create only one profile for my institution and different accounts for my employees in FIspace platform but keep them associated**.**
  - Task 1: Define mechanisms to associate user accounts of FIspace to the business profile
  - Task 2: Create mechanisms to manage the association

[3] Linked USDL is the data model used in the Application chapter of FI-WARE (Core Platform). This data model is used by the Marketplace GE. Thus, in order to use the Marketplace GE we also use the Linked-USDL data model in this baseline app.
http://www.linked-usdl.org/
http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Architecture_of_Applications_and_Services_Ecosystem_and_Delivery_Framework

- ***Story_1.1.3***: As an actor in FIspace I want to find other stakeholders connected to FIspace so that we can collaborate**.**
    - Task 1: Define search criteria to find stakeholders
    - Task 2: Define the software support for showing the information to the users
    - Task 3: Define mechanisms that can continuously do match making of actor profiles so to provide possible recommendations of potential partners for collaborations.

4.2.2 Requirements based on User Stories

Based on the functionalities described in the user stories, we can define the following functional requirements for the Business Profile App:

Table 12: Functional Requirements of Business Profile App

Req. ID	Title	Description	Epic/ Feature/ Story
BP_REQ01	**Business profile management**	There must be the ability to define and manage individual business profiles in FIspace platform.	E1-F1.1-S1.1.1
BP_REQ02	**User account and business profile association**	There must be a mechanism to guarantee that users belonging to the same business profile have a single association.	E1-F1.1-S1.1.2
BP_REQ03	**Stakeholder/potential partner matchmaking support**	There must exist a recommendation mechanisms that continuously do match making of business profiles and indicate the results to the end users	E1-F1.1-S1.1.3

4.3 Relationship and Dependencies with WP200 Components

The Table 13 shows how each of the above defined requirements is related to the components of WP200. The relationships identified so far need to be further analyzed once we get more information about the details of these components interfaces and features.

Table 13: Mapping requirements of Business Profile App into components of WP200

Requirement ID	Req. Name	ST220	ST240	ST250	ST260	ST270	ST280
BP_REQ01	Business profile management	(1)			(2)	(3)	
BP_REQ02	User account and business profile association		(4)		(2)	(3)	
BP_REQ03	Stakeholder/potential partner matchmaking support	(1)	(4)		(2)	(3)	

(1) UI support for showing notifications to users
(2) Use of the ESB to communicate with components outside this app

(3) Use of security and trust support to guarantee only authorized users have access to the information

(4) Use of B2B Collaboration functionalities in order to keep the life-cycle of functionalities that keep their computation running in background even when the users not logged in the platform directly accessing these functionalities.

4.4 Relationship and Dependencies with Generic Enablers

The Table 14 maps the requirements of the Business profile App to Generic Enablers (GE's) that can be potentially used by this app. The GEs below reported are the ones that we believe to have a direct relationship to them. Nevertheless, there might be the case in which other GE's will be used indirectly by this baseline App via WP200 modules. It is out of the scope of this document to list the GEs used by each component in WP200.

Table 14: Mapping requirements of Business Profile App into potential GEs to build upon.

Requirement ID	Req. Name	GEs considered to be directly used by App
BP_REQ01	Business profile management	Marketplace GE (To be further analyzed) Repository GE
BP_REQ02	User account and business profile association	Repository GE
BP_REQ03	Stakeholder/potential partner matchmaking support	Repository GE

Marketplace GE: We indicate in Table 14 the possibility of also using the Marketplace GE for the business profile App. The idea is to use the registration API and offering API to support the matchmaking of business profiles. However, we need to better analyze and determine whether this GE can support this type of use. The reason for this is the fact that originally the Marketplace GE is designed to provide search and matchmaking of services and not participants as we intent to do.

Repository GE: The business profile information will be stored in the Repository GE. The information will be stored as resources (according to the definition of this term in the Repository GE description) and this baseline app will be able to retrieve and manage such information. The requirements listed in this table will need to interact with this GE to retrieve and store the information.

4.5 Preliminary Test Requirements and Relationship and with WP300

The preliminary test plan and requirements for the Business Profile app include the following points:

- WP200 components used by the app should be already tested
- Repository GE and Marketplace GE (if really necessary) must be configured and functional
- The input test data will have to be provided by the domain partners of FIspace project. The idea is to request those partners to provide examples of their own business profiles, which would constitute real data, or examples of business profiles according to their experience in the domain.
- The user acceptance test for this app will be based on verifying whether the proposed functionalities work.

Table 15: List of dataset required for testing requirements of the Business Profile app

Requirement ID	Req. Name	Test Dataset
BP_REQ01 & BP_REQ03	Business profile management & Stakeholder/potential partner matchmaking support	Examples of business profile in Linked-USDL format
BP_REQ02	User account and business profile association	Examples of users belonging to the same business profile

In addition, we also verified that for conducting the tests with the Business Profile app we will need support from the WP300. One important point is to guarantee the connectivity with GEs. Thus, we will reach out to the Task 330 – Core Platform GE integration and deployment – and make sure that this will be properly configured. We also need use support for the experimentation set-up and execution (Task 340) of the WP300, as well as, experimentation facilities (Task 350) for managing the tests with this app.

4.6 Summary

In this section we described the main functionalities and relationships associated with the Business Profile app. We used user stories and requirements to describe such functionalities. The main goal of this app is to enable the definition and management of individual business profiles and provide automatic mechanisms to recommend actors, i.e., business profiles, with common interests or scope. Furthermore, we investigated the relationship of the Business Profile app with WP200 components, GEs and experimental support from WP300.

The main next step is to start technical alignment with the WP200 components that we are going to use in the development of this app. After this we can then design of the basic components and data models for the app.

Concerning the data models, we will use Linked-USDL as the language for specifying the business profile information. The data models defined in Phase I for the transport and logistics domain will be used as start point. One important aspect we need to observe during the development of this app is to guarantee the uniqueness description of each business profile. We need to create mechanisms to avoid data replication and conflicts in updating and saving data to the repository GE.

5 Marketplace Operations App

5.1 Overview

Marketplace Operations Baseline App will provide the necessary functionality to the business partners to announce as well ask for offers and demands according to their specific criteria and preferences. Automatic mechanisms will also be supported for matchmaking between announced and requested services.

The following section describes the functionality of Marketplace Operations Baseline App of FIspace platform. As a Baseline App, Marketplace Operations is a cross-domain, general application, with close relationship to the FIspace trials. In order to define the functional requirements, we defined first User Stories which are then transformed into the requirements. In addition to the requirements, the connection between the App and other FIspace core components is presented, i.e. the WP200 core modules, the GEs that will be provided via the platform, as well as Work Package 300.

5.2 Baseline Apps Functionalities

In this section the User Stories are presented, based on which the requirements of the Baseline App are afterwards defined. The user stories are structured in three levels of hierarch moving from a high abstract or generic functionality to more specific functionalities. These levels are called: epic, feature, and story. For a story we define more concrete tasks that need to be performed to accomplish such story.

5.2.1 User Stories

Epic_1

Description: As an actor in FIspace I can announce my services and/or demands so that I become visible to other stakeholders and to the FIspace marketplace and optionally to external marketplaces.

Feature_*1.1*: As an **actor in FIspace** I can define offers and demands descriptions of my services so that other partners can automatically find my services.

- ***Story_1.1.1***: As an **actor in FIspace** I can define offer information model so that SLOs (Service Level Objectives) of services can be standardized, and facilitate the exchange of information**.**
  - Task 1: Define which kind of domain will be served for this app
  - Task 2: Identify the minimum set of information describing a service offer for each domain to be supported by this app. The actor has to define valid period of the offering alongside with it price plan.
  - Task 3: Associate a service level profile with this offering
  - Task 4: Model the set of information following the Linked-USDL approach
  - Task 5: Get feedback from domain partners whether the information model matches their expectations
  - Task 6: Create RDF data models with the final model accepted by the domain partners, upload it to the repository and upload the description to the marketplace.

- ***Story_1.1.2***: As an **actor in FIspace** I can define demand information model so that SLOs of services can be standardized and facilitate the exchange of information**.**
  - Task 1: Define the domain which will be served by this app.
  - Task 2: Identify which set of information should be included in a demand following a unified way amongst actors.
  - Task 3: Model the set of information in an independent vocabulary created for client description for demand (as Linked-USDL vocabulary is suitable only for provider side)
  - Task 4: Create an RDF data model representation of the information.
  - Task 5: Create mechanism to enable the constant automatic match making between demands and offers.

Feature_1.2: As an **actor in FIspace** I can search for offers and demands so that I can **automatically find needed services. (**The stakeholder connects to Marketplace of FIspace Platform. Through Business Service App, there is a GUI where stakeholder can see list of demand and offer services, as well as bids. Also, there are search filters for faster search.**)**

- ***Story_1.2.1***: As an **actor in FIspace** I can search for service offerings fitting my demands.
  - Task 1: Define appropriated mechanisms for match making between one specific demand and available offers.
  - Task 2: Store and manipulate the returned service information offerings
  - Task 3: Create mechanism to enable the constant automatic match making between offers and demands, and notifies the owners of the demands.

- ***Story_1.2.2***: As an **actor in FIspace** I can search for services demands to see if I can serve it with my offerings.
  - Task 1: Create mechanisms to select demands that match specific offer requirements
  - Task 3: Present the identified set of demands to the user so that he/she can choose the demand.
  - Task 4: Create mechanism to enable the constant automatic match making between demands and offers, and notifies the owners of the demands.

- ***Story_1.2.3***: As an **actor in FIspace** I would like to have access to information of offers and demands belonging to external marketplaces.
  - Task 1: Identify which kind of external marketplaces the domain partners of FIspace currently use
  - Task 2: Create a list of external marketplaces considered to be integrated into FIspace Marketplace and prioritize which will be effectively tested/integrated in this App
  - Task 3: Create mechanisms to keep the consistency between information from external marketplaces and the FIspace marketplace.

5.2.2 Requirements based on User Stories

Table 16 lists the functional requirements of the Marketplace Operations App extracted from the above presented User Stories. For each one of the requirements there is a short description is provided and their mapping to features and stories is established.

Table 16: Functional Requirements of Marketplace Operations App

Req. ID	Title	Description	Epic/ Feature/ Story
MO_RE001	**Define offer/demand information**	Define which kind of business domains will be served for this app, define valid period of the offering/demand, define price plan, associate service level profile	E1-F1.1-S1.1.1 & E1-F1.1-S1.1.2
MO_RE002	**RDF data model creation for final information model of offers /demands**	After the domain partners accept the offer/demand information using Linked-USDL approach, there must be the ability to create RDF data model and handle them accordingly	E1-F1.1-S1.1.1 & E1-F1.1-S1.1.2
MO_RE003	**Upload information model of offer/ demand**	There must be the ability to upload the RDF data model to the repository, as well as the description to the marketplace.	E1-F1.1-S1.1.1 & E1-F1.1-S1.1.2

Req. ID	Title	Description	Epic/ Feature/ Story
MO_RE004	**Offer/demand matchmaking support**	There must be automated mechanisms to find service descriptions (offers and demands) that match according to established criteria	E1-F1.2-S1.2.1 & E1-F1.2-S1.2.2
MO_RE005	**Feature to choose among suggested demands to serve with own offerings**	After the offer-demand matchmaking support returns a specific list of matches, the user (either client or provider) must be able to choose which match will be considered.	E1-F1.2-S1.2.2
MO_RE006	**Storing/manipulation of matchmaking support's returned services**	Database supporting the storage as well as manipulation of services that matchmaking support will return	E1-F1.2-S1.2.1
MO_RE007	**Access to external marketplaces' offers & demands (optional)**	After identifying which external marketplaces the domain partners use, there should be a mechanism for integration as well mechanisms for keeping consistency between external and FIspace marketplace information.	E1-F1.2-S1.2.3

5.3 Relationship and Dependencies with WP200 Components

Baseline apps like the Marketplace Operation are designed and developed based on the support offered by the FIspace platform. Therefore, we need to identify the connection and dependency of between the functional requirements of this app with the WP200 Core Components, which are the building blocks of the FIspace platform.

The aforementioned mapping is introduced in Table 17. In addition to the simple indication of existing relationships denoted with "x", the numbers in the table are used to explain a particular dependency. The symbol "?" indicates that the relationship between the requirement and the component should be better understood and further analyzed.

Table 17: Mapping requirements of Marketplace Operations App into modules of WP200

Requirement ID	Req. Name	ST220	ST230	ST240	ST250	ST260	ST270	ST280
MO_RE001	**Define offer/demand information**	x				(3)		
MO_RE002	**RDF data model creation for final information model of offers /demands**					?		
MO_RE003	**Upload information model of offer/ demand**	x				(3)		
MO_RE004	**Offer/demand matchmaking support**			(1)		(3)		
MO_RE005	**Feature to choose among suggested demands to serve with own offerings**	x				(3)	(4)	

Requirement ID	Req. Name	ST220	ST230	ST240	ST250	ST260	ST270	ST280
MO_RE006	**Storing/manipulation of matchmaking support's returned services**	x				(3)	(4)	
MO_RE007	**Access to external marketplaces' offers & demands (optional)**				(2)	(3)		(2)

Description of the tags in Table 17:

(1) Possibly when a new offer/demand is submitted, there will be a background process, during which the matchmaking will take place among the existing demands/offers respectively, and suggestions/notifications will be sent to the stakeholders
(2) If T250/T280 provide tools for integrating external/legacy marketplaces
(3) If finally ESB manages all interactions, almost all functionalities of the App should use T260
(4) There should be some kind of privacy after certain collaborations (offer-demand matching) are created.

5.4 Relationship and Dependencies with Generic Enablers

Similarly with the relationship with the WP200 components, the table below is linking specific requirements of the Marketplace Operations App to FIWARE Generic Enablers (GEs) for implementing a particular functionality. It has to be noted though, that apart from the GEs mentioned here, some GEs might be used indirectly by the this app via the WP200 Modules. The report on which GEs are used in the core of the FIspace platform is out of the scope of this document.

Table 18: Mapping requirements of Marketplace Operations App into potential GEs used to build upon

Requirement ID	Req. Name	GEs considered to be directly used by App
MO_RE003	**Upload information model of offer/ demand**	Marketplace GE Repository GE
MO_RE004	**Offer/demand matchmaking support**	Marketplace GE

MO_REQ003

- Marketplace GE has a core component named "Offering and Demand", which is responsible for retrieving offerings from stores or other sources and publish them to Marketplace. This component provides functionalities such as create offerings, so as to push new service offering from a registered store into marketplace, update offering, end offering etc. The Marketplace itself relies on functionality provided by the Repository GE, which is described right below. For example, the functionality of creating offerings needs Repository, so as to link this offering with its Linked-USDL description in Repository.

- Repository GE is a place to store service models, especially in Linked-USDL format. The Repository is composed of two core objects, *resources* and *collections*. The resources are mainly the USDL service descriptions, while collections are containers for collecting resources. Thus, the core functionality of the Repository is to store resources and retrieve them when necessary, as well as update and delete resources. Also, it offers functionalities for creating collections, retrieving collections as well as deleting them. All the above functionalities use RESTful APIs.

MO_REQ004

- Marketplace GE has a core component named "Discovery & Matching", which is responsible for discovering and matching offerings and demands according to stakeholder's criteria and needs. In particular, this component offers operations such as free text search operation, which can be used to search the marketplace for offerings using a search string with wildcards and filters, a variety of filter options so as to get the possible filter options for a free text search, comparison of offerings as well as search for stores.

5.5 Preliminary Test Requirements and Relationship and with WP300

The preliminary list of test requirements and plan of the Marketplace Operations Baseline App shall involve:

- Preconditions/requirements:
  - WP200 modules used by the app should be already tested
  - Marketplace & Repository GEs already set-up & configured
  - Potential Legacy/External Systems connected (if finally external marketplaces will be used for example)

- User acceptance test:
  - Actions/steps to be performed
  - Expected outcome per action/step

Test Input Data: For the Marketplace Operations App, in order to test the defined requirements, test data should be available.

Test Input Data

We believe that ideally the datasets to test this baseline app should be based on real data eventually collected from the systems from the domain partners of FIspace project. In case we cannot have access to real data associated with the business operation of the FIspace domain partners, we will ask them to provide realistic examples based on their experience. We can accomplish this by requesting the domain to use our app and insert examples of offers and demands.

Another issue that needs to be considered during test in general in FIspace platform is the use of real-time or simulated input data stream. In the case of the Marketplace Operation App, we will need real-time access to the information stored inside the GEs, specially the Repository and Marketplace GEs.

Not all the requirements/functionalities of the this app will require specific datasets to be tested. Among the various requirements defined earlier for the Marketplace Operations App, the ones requiring either real or near-real (due to the use of experts to create examples) test datasets for testing are presented in the table below. In all cases, the required test dataset type is common, i.e. Offers and Demands data in Linked-USDL format.

Table 19: List of dataset required for testing requirements of the Marketplace Operations app

Requirement ID	Req. Name	Test Dataset
MO_RE002	**RDF data model creation for final information model of offers /demands**	**Offers/demands datasets in Linked-USDL format**
MO_RE003	**Upload information model of offer/**	**Offers/demands datasets in Linked-**

	demand	**USDL format**
MO_RE004	**Offer/demand matchmaking support**	**Offers/demands datasets in Linked-USDL format**
MO_RE005	**Feature to choose among suggested demands to serve with own offerings**	**Offers/demands datasets in Linked-USDL format**
MO_RE006	**Storing/manipulation of matchmaking support's returned services**	**Offers/demands datasets in Linked-USDL format**

The Offers/Demands data shall have different fields according to the domain to which they refer. Some examples of these fields would be:

- Domain of service
- Service provider / service inquirer details
    - Name
    - Location
    - Contact details
- Type of service – product
- Cost of service
- Locations served
- Date availability
- Quantity (if the service is a product)
    - Product details (particular type, other specifications)

In WP300, a set of experimentation facilities will be provided in addition to hosting the software developed in FIspace. We will certainly use the hosting facilities to deploy the Marketplace Operations (associated with Task 320 – Cloud Hosting) and also collaborate with the team from WP300 in order to integrate and connect our solution to GEs that will be used (associated with Task 330 – Core Platform GE integration and Deployment). We also believe that we will need to use the experimentation set-up and execution (Task 340) support from WP300. This will be decided once we have a better understanding of the actual test plan for this app. Finally, there is also the possibility of using the experimentation facilities from Task 350 for example for managing the execution of the tests associated with the Marketplace Operation App.

5.6 Summary

In this section we defined the Marketplace Operations App that is one of the baseline apps to be developed and made available in FIspace platform. We used user stories and functional requirements to describe such functionalities.

This app will give the opportunity to the user to announce services, in the context of offers and demands as well as search for services. We will provide automatic matchmaking between offers and demands that can be initiated either by need of matches for covering a demand, or need of matches for covering an offer. According to the needs of the trials in FIspace projectt, we could also consider to integrate external marketplaces information into the marketplace inside FIspace.

For each one of the requirements defined, we investigated the potential connection and dependecy to the WP200 modules, the GEs of the core platform. In addition we also discussed the preliminary test requirements and the relationship of this app with WP300.

In order to start with the development of the Marketplace Operations App, it would be necessary to start a technical alignment with the WP200 modules that this app relies upon. This way, we can design the basic components of the Marketplace Operations App, *i.e.*, the architecture of the App, the connection with the other components etc. Finally, example datasets of offers and demands should be collected from the domain partners in order to be able to further elaborate on the App's functionality as well as test it.

6 Real-time Business SLA Management App

6.1 Overview

In this section we describe the Real-time Business SLA Management App (BizSLAM). The main purpose of this app is to bring into the operation of business activities the SLO (Service Level Objectives) established in the SLA (Service Level Agreement) between 2 partners. The difference of this SLA management solution for many others is the fact that services here are indeed real-life or cyber-physical services (e.g., transportation of goods by air carrier) and not computational services (e.g., buying a slice of storage from a cloud provider). Management of SLAs of cyber-physical services requires domain specific data models and different mechanisms to detect violations and to perform compliance check. Thus, this baseline app provide the solution for managing the information inside the SLAs of cyber-physical services (from now on called business services) and enabling notifications about events associated with the SLA information itself as well as violations on the SLOs established in this SLA.

The core capabilities of the Real-time Business SLA Management app are:

- Support for online representation of SLOs of business services.
- Management of CRUD operations (Create, Read, Update, Delete) without compromising the privacy and security of the stored SLAs.
- Provide notifications about events and deadlines of each SLA a partner is part of.
- Detection and signaling in real time of deviations on the SLOs of SLAs.
- Notify partners about opportunities to establish SLAs.

In the next sections, we present the user stories, requirements mapping to user stories, relationship of this app with WP200, GEs and WP300.

6.2 Baseline Apps Functionalities

This section described the main functionalities of the BizSLAM app. These functionalities are expressed in two formats. First in user stories which can be easily understood by the domain partners and a list of functional requirements based on such user stories.

6.2.1 User Stories

The user stories are structured in three levels of hierarch moving from a high abstract or generic functionality to more specific functionalities. These levels are called: epic, feature, and story. For a story we define more concrete tasks that need to be performed to accomplish such story.

Epic_1

Description: As a business partner in a SLA I want to have real-time and online access to SLOs of agreed SLA with my partners so that I use this information during the daily operation of my services.

Feature_1.1: As a business partner in a SLA I want to have a standardize way of describing and using the SLOs of my domain so that my partners and I can use this information automatically and in real-time.

- ***Story_1.1.1***: As a business partner in a SLA I want to model the SLOs of my domain so that I can interact with other partners in an easier way.
  - Task 1: Define which kind of business domain will be served for this app
  - Task 2: Identify the minimum set of information describing the most important SLOs for the services in this domain
  - Task 3: Model the set of information following the Link-USDL approach
  - Task 5: Get feedback from domain partners whether the information model matches their expectations
  - Task 6: Create RDF data model with the final model accepted by the domain partners.

Feature_1.2**:** As a business partner in a SLA I want to manage the information of my SLOs so that I can have this information available for the downstream of my business operation.

- ***Story_1.2.1:*** As a business partner in a SLA I want to have certain level of CRUD (Create, Read, Update and Delete) operations on my SLOs so that I can have this information available online for the downstream of my business operation.
    - Task 1: Define mechanisms to insert the information of agreed SLA outside FIspace platform into the BizSLAM app control.
    - Task 2: Define mechanisms to guarantee that both partners of the same SLA can visualize the stored information avoiding duplication of the same contract information inside the FIspace platform and/or inconsistencies
    - Task 3: Create mechanisms that can guarantee the stored SLOs of the SLA cannot be unilaterally changed by any of the parties of such SLA without agreement from both parts
    - Task 4: Create a UI so that business partner can perform the management operations for the types of domain SLAs supported by this app

- ***Story_1.2.2:*** As a business partner in a SLA I want to search and visualize the information of my SLOs so that I can have this information available for the downstream of my business operation.
    - Task1 : Define mechanisms to search among the SLAs stored for a business partner
    - Task 2 : Define search criteria for each type of domain SLAs supported by this app
    - Task 3: Create mechanisms to guarantee that only allowed entities can access the information of the SLAs
    - Task 4: Create mechanisms to present to the user the information of the SLOs in the SLA

- ***Story_1.2.3:*** As a business partner in a SLA I want to downstream the SLOs so that I can integrate this information with any other activity from my business operation.
    - Task1 : Define interfaces to provide information about SLOs in the SLAs to other software entities (e.g., Logistics Planning App)

Epic_2

Description: As a business partner in a SLA I want to receive analytics about my SLOs so that I can use this information to improve my business operation.

Feature_2.1**:** As a business partner in a SLA I want to automatically detect deviations from agreed SLOs in my SLAs so that I can react in real-time (or near real-time) to these deviations.

- ***Story_2.1.1:*** As a business partner in a SLA I want to detect deviations before even the execution of my offered or demanded services so that I can avoid the high costs of exception handling.
    - Task 1: Identify for each type of SLA if there is the possibility of conducting conformity check before a business process between the contracted partners starts to execute
    - Task 2: Determine which kind of checks between pre execution of business services and the agreed SLOs should be done for each case (e.g., a pre-execution violation could be if the amount of containers to be booked in the Carrier system is higher than the agreed capacity between the shipper and the Carrier)
    - Task 3: Create the mechanisms to enforce the check for each situation
    - Task 4: Create mechanisms to notify users of the app about SLO deviations from the agreed SLA

- ***Story_2.1.2:*** As a business partner in a SLA I want to detect deviations during the execution of my offered or demanded services so that I can be aware as early as possible of eventual penalties.
    - Task 1: Identify for each type of SLA if there is the possibility of conducting conformity check during the execution of the business process between the contracted partners
    - Task 2: Determine which kind of checks during the execution should be conducted (e.g., check if the amount of goods to be loaded in the truck are the same one booked/expected)
    - Task 3: Create the mechanisms to enforce the check for each situation
    - Task 4: Create mechanisms to notify users of the app about SLO deviations from the agreed SLA during execution of the business process

Feature_2.2**:** As a business partner in a SLA I want to be notified about ordinary events associated with stored SLAs so that I can take actions if necessary.

- ***Story_2.2.1:*** As a business partner in a SLA I want to receive alerts about the lifecycle of my contract so that I can be aware of the status of my contract and not specifically about my SLAs.
    - Task 1: Identify events associated with the SLA lifecycle and not to the SLOs (e.g.: about to expire, expired, under review, request from one partner to change, unilateral change, request to cancel, etc..)
    - Task 2: Create the mechanisms to monitor and report these events
    - Task 3: Create the mechanisms for reporting these events to the users

- ***Story_2.2.2:*** As a business partner in a SLA I want to receive alerts of recurrent spot market SLAs I agree with the same partner so that I can negotiate a long term SLA with this partner and get better SLOs for the service.
    - Task 1: Create the mechanisms to constantly monitor the recurrence of spot market SLAs among two business partners
    - Task 2: Create the mechanisms for reporting these recurrences to the users

Epic_3

Description: As a business partner in a SLA I want to be able to find new partners so that I can propose and initiate new SLAs with them.

Feature_3.1**:** As a business partner in a SLA I want to be aware of other business partners in FIspace platform so that I can initiate new SLAs with them.

- ***Story_3.1.1:*** As a business partner in a SLA I want to search for other business partners inside FIspace Platform so that I can propose new SLAs to these business partners.
    - Task 1: Identify which functionalities of the Business Profile Apps can be used to search for business partners
    - Task 2: Define the search criteria based on the API offered by the Business Profile Apps
    - Task 3: Create the mechanisms for showing in the Real-time Business SLA Management the information of business partners retrieved from the Business Profile App

- ***Story_3.1.2:*** As a business partner in a SLA I want to contact business partners so that I can start the process to establish a SLA.
    - Task 1: Create the mechanisms to notify business partners about possible engagement in a SLA negotiation

6.2.2 Requirements based on User Stories

Based on the functionalities described in the user stories, we can define the following functional requirements for BizSLAM App:

Table 20: List of requirements based on user stories of the BizSLAM App

Req. ID	Title	Description	Epic/ Feature/ Story
BIZ_SLA_REQ01	Define SLA Information	Define what kind of business domains will be served for this app and the SLOs of the SLAs.	E1-F1.1-S1.1.1
BIZ_SLA_REQ02	RDF data model creation with information model of SLA	After the domain partners accept the SLA information using Linked-USDL approach, there must be the ability to create RDF data model and handle them accordingly.	E1-F1.1-S1.1.1
BIZ_SLA_REQ03	Store information model of SLA	There must be the ability to store SLA file in the Repository GE.	E1-F1.1-S1.1.1
BIZ_SLA_REQ04	Define mechanisms to insert SLA Information and management operations	There must be a UI in which information of SLAs between partners can be inserted into FIspace platform and mechanism to manage this stored information.	E1-F1.2-S1.2.1
BIZ_SLA_REQ05	Define mechanisms to avoiding duplication of the same SLA.	There must be a support for avoiding and detection of duplication of SLAs, once each partner of each SLA could unilaterally insert this information into FIspace platform.	E1-F1.2-S1.2.1
BIZ_SLA_REQ06	Define mechanisms to guarantee SLO information integrity.	There must be a support for avoiding and detecting unilateral changes in the SLOs attributes of a SLA stored in FIspace platform.	E1-F1.2-S1.2.1
BIZ_SLA_REQ07	Mechanisms to search SLA information	There must be mechanisms to support searching information in SLAs.	E1-F1.2-S1.2.2
BIZ_SLA_REQ08	Define Mechanisms to guarantee access control to SLA information visualization.	There must be security mechanisms to guarantee that only authorized partners can visualize the information of the stored SLAs.	E1-F1.2-S1.2.2
BIZ_SLA_REQ09	Define search criteria for each type of SLA	Define what kind of fields could be searched for each type of supported SLA information model.	E1-F1.2-S1.2.2

Req. ID	Title	Description	Epic/ Feature/ Story
BIZ_SLA_REQ10	Integrate SLA information with business operation.	Create mechanisms to allow information from the stored SLA to be pushed into the core of FIspace platform, other Apps, and/or external systems during the execution of services associated with the SLA.	E1-F1.2-S1.2.3
BIZ_SLA_REQ11	Qualify SLOs that need to be checked	There must be mechanisms to indicate which SLOs should be checked either before execution or during execution of business services.	E2.F2.1.S2.1.1 & E2.F2.1.S2.1.2
BIZ_SLA_REQ12	Detect deviations before the execution of offered/demanded services.	There must be a continuous and automated mechanism to check for violations of SLOs before the execution of business services associated with stored SLAs.	E2.F2.1.S2.1.1
BIZ_SLA_REQ13	Detect deviations during the execution of offered/demanded services.	There must be an automated and continuous mechanism to check for violations of agreed SLOs during the execution of a business service associated with such SLOs.	E2.F2.1.S2.1.2
BIZ_SLA_REQ14	Notify the app users about deviations in SLA.	There must be a support to automatically inform partners about deviations in SLA either before or during execution of a business service.	E2.F2.1.S2.1.1 and E2.F2.1.S2.1.2
BIZ_SLA_REQ15	Detect ordinary events associated with SLAs.	There must be an automated and continuous mechanism to identify the events associated with stored SLA.	E2-F2.2-S2.2.1
BIZ_SLA_REQ16	Notify app users about ordinary events associated with the SLA itself.	The users must be automatically notified about ordinary events related to their stored SLAs.	E2-F2.2-S2.2.1
BIZ_ LA_REQ17	Alert about potential partner to establish long term SLA.	There must be automated and continuous analysis to identify and notify recurrent short term SLAs between the same two partners.	E2-F2.2-S2.2.2
BIZ_SLA_REQ18	Search for other business partners inside FIspace platform.	There must be a mechanism to support business partners in FIspace to search for other potential partners also inside FIspace platform. This mechanism should be based on the APIs from Business Profile App.	E3-F3.1-S3.1.1

Req. ID	Title	Description	Epic/ Feature/ Story
BIZ_SLA_REQ19	Allow two unknown possible business partners to start SLA negotiation.	There must be a mechanism to allow business partners to become aware of potential business opportunities among them and enable the future establishment of SLAs among them.	E3-F3.1-S3.1.2

6.3 Relationship and Dependencies with WP200 Components

The Table 21 shows how each of the above defined requirements is related to the components of WP200. The relationships identified so far need to be further analyzed once we get more information about the details of these components interfaces and features.

Table 21: Mapping requirements of BizSLAM App into components of WP200.

Requirement ID	ST220	ST240	ST250	ST260	ST270	ST280
BIZ_SLA_REQ01		(1)				(6)
BIZ_SLA_REQ02						(6)
BIZ_SLA_REQ03				(3)	(4)	
BIZ_SLA_REQ04				(3)	(4)	
BIZ_SLA_REQ05		(5)		(3)	(4)	
BIZ_SLA_REQ06		(5)		(3)	(4)	
BIZ_SLA_REQ07				(3)	(4)	
BIZ_SLA_REQ08				(3)	(4)	
BIZ_SLA_REQ09						
BIZ_SLA_REQ10				(3)	(4)	
BIZ_SLA_REQ11		(5)		(3)	(4)	(7)
BIZ_SLA_REQ12		(5)		(3)	(4)	
BIZ_ SLA_REQ13		(5)		(3)	(4)	
BIZ_SLA_REQ14	(2)	(5)		(3)	(4)	
BIZ_SLA_REQ15		(5)		(3)	(4)	

BIZ_SLA_REQ16	(2)	(5)		(3)	(4)	
BIZ_SLA_REQ17	(2)	(5)		(3)	(4)	
BIZ_SLA_REQ18		(5)		(3)	(4)	
BIZ_SLA_REQ19	(2)	(5)		(3)	(4)	

Description of the tags:

(1) Information that is important for alignment with models in ST240
(2) UI support for showing notifications to users
(3) Use of the ESB to communicate with components outside this app
(4) Use of security and trust support to guarantee only authorized users have access to the information
(5) Use of B2B Collaboration functionalities in order to keep the life-cycle of functionalities that have computation running in background even when the users are not logged in the platform directly accessing these functionalities.
(6) Tools for helping future extension of SLA data model
(7) Tools for helping future developers to define check constraints and combine this with the tools from ST240

6.4 Relationship and Dependencies with Generic Enablers

The BizSLAM app directly uses the **Repository GE** as the storage for the SLA information. Thus all the requirements related to access information from the stored SLAs are associated with this GF. Below you can see list of requirements:

- BIZ_SLA_REQ03 - Store information model of SLA
- BIZ_SLA_REQ04 - Define mechanisms to insert SLA Information and management operations
- BIZ_SLA_REQ07 - Mechanisms to search SLA information
- BIZ_SLA_REQ17 - Alert about potential partner to establish long term SLA.

In addition to the direct use of the Repository GE, we believe that other GEs such as CEP engine and the GE associated with the Security will be also used, but indirectly due to the fact that our requirements will be built on top of WP200 components that use the aforementioned GEs. For example, the list below shows some of the requirements of the BizSLAM app that will build upon ST240 component which uses the CEP GE:

- BIZ_SLA_REQ05 - Define mechanisms to avoiding duplication of the same SLA.
- BIZ_SLA_REQ06 - Define mechanisms to guarantee SLO information integrity.
- BIZ_SLA_REQ12 - Detect deviations before the execution of offered/demanded services.
- BIZ_SLA_REQ13 - Detect deviations during the execution of offered/demanded services.

6.5 Preliminary Test Requirements and Relationship and with WP300

The preliminary test plan and requirements for the BizSLAM app include the following points:

- WP200 modules used by the app should be already tested
- Repository GEs must be set-up & configured
- The input test data will have to be provided by the domain partners of FIspace project. The idea is to request those partners to provide examples of their SLAs, which would constitute real data. However, we know that this is very sensitive information for the partners business and it will be

almost impossible to get this real data. Therefore, we will request the domain partners to provide or examples of SLAs according to their experience in the domain.

Currently we already have the data model for SLA in the transport and logistics domain which is a result from Phase I. We will leverage this work done in Phase I and use it as the startup point for the activities in the BizSLAM app. During the progress of these activities we will analyze and evaluate the possibility of also defining an SLA data model for the agri-food domain. However, this will be a secondary thread of work, because most important is to have the implementation itself of the BizSLAM app until the end of FIsapce project. Nevertheless, extensions of SLA models for the BizSLAM app could be also provided by partners coming from the Open Call of FIspace project.

In addition, in this section we also provide a brief discussion on the types of datasets and examples that will be needed to test the requirements defined for this app, as depicted in Table 22. We believe that the we can join forces with the team on ST240 for identifying examples and datasets when it comes to testing the requirements of the BizSALM app that are dependent on ST240 components (as we previously indicated in Table 21).

Table 22: List of data required for testing requirements of the BizSLAM app

Requirement ID	Req. Name	Data
Biz_SLA_REQ03 & Biz_SLA_REQ04 & BIZ_SLA_REQ06	Store information model of SLA & Define mechanisms to insert SLA Information and management operations & Define mechanisms to guarantee SLO information integrity	SLA examples in Linked-USDL data format
BIZ_SLA_REQ11	Qualify SLOs that need to be checked	List of SLOs for a type of SLA that can be associated with exceptions and thus need to be checked
BIZ_SLA_REQ12	Detect deviations before the execution of offered/demanded services	Examples of deviations values and types for each identified SLO that needs to be checked
BIZ_SLA_REQ13	Detect deviations during the execution of offered/demanded services	List of types of checks that need to be conducted in some SLOs before the execution of the service agreed in the SLA containing such SLOs.
BIZ_SLA_REQ14	Notify the app users about deviations in SLA	Examples of values for situations in which exceptions happen and affect the SLOs of a SLA
BIZ_SLA_REQ15 & BIZ_SLA_REQ16	Detect ordinary events associated with SLAs & Notify app users about ordinary events associated with the SLA itself	Identification of situations associated only to the SLA document itself that need to be checked and reported. Examples of values for these situations.

Requirement ID	Req. Name	Data
BIZ_SLA_REQ17	Alert about potential partner to establish long term SLA	Short term SLA examples in Linked-USDL data format. Minimal set of examples is 3, this means with same partners are involved in 3 short term SLAs.

Finally, we also verified that for conducting the tests with the BizSLAM app we will need support from the WP300. One important point is to guarantee the connectivity with the Repository GEs. Thus, we will reach out to the Task 330 – Core Platform GE integration and deployment – and make sure that this will be properly configured. We also need use support for the experimentation set-up and execution (Task 340) of the WP300, as well as, experimentation facilities (Task 350) for managing the tests with this app.

6.6 Summary

In this section we described the main functionalities of the Real-time Business SLA Management (BizSLAM) app and its relationships with the WP200 components, GEs and WP300. User stories and requirements are to describe such functionalities. The core capabilities of the BizSLAM app are: (i) support for online representation of SLOs of business services; (ii) management of CRUD operations (Create, Read, Update, Delete) without compromising the privacy and security of the stored SLAs; (iii) provide notifications about events and deadlines of each SLA a partner is part of; (iv) detection and signaling in real time of deviations on the SLOs of SLAs; and (v) notify partners about opportunities to establish SLAs.

The next steps are: technical alignment with the WP200 components that we are going to be used in the development of this app and design of the basic components and data models for the app. We will use Linked-USDL as the specification language for the SLA information representation. The data models defined in Phase I for the transport and logistics domain will be used as start point.

7 Overview of Baseline Apps

This section presents an overview about the main issues associated with the baseline apps detailed in the previous section. We first describe the main components from WP200 and WP300 that are important for the baseline apps. Then, we list the all GEs that need to be accessed by the baseline apps. As we discussed in Section 1.4, how exactly the GEs will be accessed by the baseline apps is a topic that need to be better discussed with WP200 team. We also describe in this section two other important sections. One describing the so far identified relationships among the baseline apps, and another summarizing which baseline apps are intended to be used by the trials from WP400.

7.1 Relationship with WP200 and WP300

The baseline apps have to be designed based on the FIspace platform. This platform is based on a set of components. As described in the previous sections, we mapped the requirements of the baseline apps to the WP200 components. Table 23 summarizes the components of WP200 that are used by each baseline app. As it can be observed in this table, not all the components of WP200 must necessarily be used by the baseline apps. The next step for all baseline apps is to refine and reach the technical level of their relationship with the WP200 components.

Table 23: Summary of WP200 components to be used by the baseline apps

Baseline Apps	ST220	ST240	ST250	ST260	ST270	ST280
Product Information Service App		X	X	X	X	
Logistics Planning App		X	X	X	X	
Business Profile App	X	X		X	X	
Marketplace Operations App	X	X	X	X	X	X
Real-time Business SLA Management App	X	X	X	X	X	X

Regarding the relationship with WP300, we observed that there are three main areas in this work package that will be needed by the baseline apps:

- ST320 (Cloud Hosting) - We need to be sure that the developed baseline apps are properly deployed inside the FIspace platform in the cloud environment of the FIspace.
- ST330 (Core Platform GE integration and Deployment) – We need to collaborate with the team from WP300 in order to make sure that our apps are correctly connected to the required GEs
- ST340 (Experimentation Set-up and Execution) - We need to check and analyze the support from WP300 for conducting the actual test plans of each baseline app.
- ST350 (Experimentation Facilities) - We could, for instance, use the experimentation facilities to manage the execution of the baseline app tests.

7.2 Use of Generic Enablers

The Table 24 shows the summary of the GEs consider to be used by the baseline apps. The form how the baseline app will access the GEs interfaces is not yet defined. The current assumption is that WP200 components should provide APIs to the baseline apps. These APIs will actually expose in a transparent form the functionalities of the GEs.

Nevertheless, we needed to identify which of the GEs and the functionalities are important to the baseline apps. Table 24 summarizes the GEs discussed by the baseline apps in the previous chapters. The next step is to check if all the listed GEs are covered by the WP200 components and what are the APIs offered to the baseline apps.

Table 24: Summary of GEs considered to be used by the baseline apps

GE related to the baseline app	Product Information Service App	Logistics Planning App	Business Profile App	Marketplace Operations App	Real-time Business SLA Management App
Data Handling	X				
Gateway Data Handling	X				
Object Storage	X				
Publish/Subscribe Context Broker	X				
Complex Event Processing (CEP)	X				X
OneIDM		X			
Marketplace			X	X	
Repository			X	X	X

7.3 Relationship among Baseline Apps

The analysis of the initial list of user stories and functional requirements of the baseline apps showed the existence of functionalities that were duplicated in some of the baseline apps. Based on this analysis, we then identified which baseline app should actually provide the functionality and which ones could use such functionalities. Table 25 depicts the main relationship so far identified among the aforementioned baseline apps. The columns represent the baseline apps that are provided functionalities to other baseline apps, while rows indicate the baseline apps that require functionalities from other baseline apps. Inside each cell we indicate which functionality is provided and/or use. The symbol “NA” denotes that the relationship is not applicable, and “-” indicates that the relationship between the apps has not yet been identified.

Table 25: Summary of relationship among baseline apps

Column: Provide Functionality Row: Require Functionality	Product Information Service App	Logistics Planning App	Business Profile App	Marketplace Operations App	Real-time Business SLA Management App
Product Information Service App	NA	-	-	-	-
Logistics Planning App	-	NA	-	Uses marketplace operation functionalities	Uses functionalities to access SLA information
Business Profile App	-	-	NA	-	-
Marketplace Operations App	-	-	-	NA	-
Real-time Business SLA Management App	-	-	Uses functionalities to find business profiles	-	NA

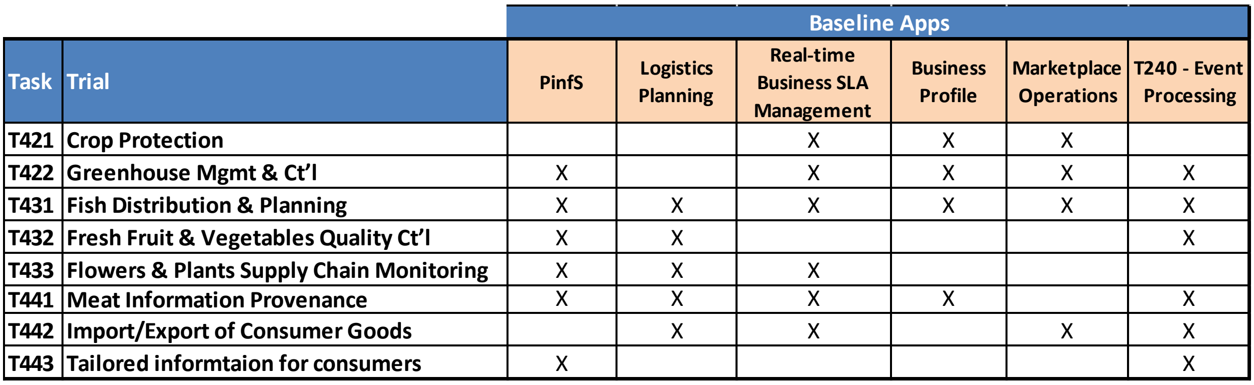
We could so far identify three relationships involving four baseline apps as described below:

- The **Logistics Planning App** is related to two baseline apps: the **Marketplace Operations App** and the **Real-time Business SLA Management App**. Using the functionalities offered by these apps the logistics partners (client or provider) can have access to offers and demand information through the Marketplace Operations. In addition, during the planning of a logistics service, the users of the LPA app can access their SLA information, and, thus, they become aware of which partnerships they can use for generating the plan.
- The **Real-time Business SLA Management** app uses functionalities of the **Business Profile App**. The former app uses the business profile information stored and managed by the latter in order to recommend new SLAs between business partners that are not aware of each other yet.

The content of Table 25 can change over time according to the evolution of the work in the baseline apps teams. There is the possibility of identification of new relationships when we start the work on the technical specification. Based on the agile approach for the development of the baseline apps will be able to identify these new relationships and adapt the work accordingly.

7.4 Trials from WP400 Using Baseline Apps

The trial experimentations to be developed and tested in FIspace project will use some of the functionalities provided by the Baseline Apps. In FIspace project there are eight trials. Each one of these trials analysed the functionalities of the defined in previous chapters. The output of this analysis is an initial list mapping trials to baseline apps. Figure 5 summaries this list indicating the the relationship among trials and baseline apps.



		Baseline Apps					
Task	Trial	PinfS	Logistics Planning	Real-time Business SLA Management	Business Profile	Marketplace Operations	T240 - Event Processing
T421	Crop Protection			X	X	X	
T422	Greenhouse Mgmt & Ct'l	X		X	X	X	X
T431	Fish Distribution & Planning	X	X	X	X	X	X
T432	Fresh Fruit & Vegetables Quality Ct'l	X	X				X
T433	Flowers & Plants Supply Chain Monitoring	X	X	X			
T441	Meat Information Provenance	X	X	X	X		X
T442	Import/Export of Consumer Goods		X	X		X	X
T443	Tailored informtaion for consumers	X					X

Figure 5: Relationship among Trials and Baseline Apps

Nevertheless, the trials also performed a more details analysis and some of them were able to identify which functional requirements of the baselines apps could be mapped to the functionalities they require. The complete and detailed mapping between baseline functionalities and trial needs is reported in D400.1. In this document, instead, we provide two examples of the identified relationships between the PInfS app and two trials: the Fresh Fruits and Vegetables Quality Assurance and the Greenhouse Management trials.

Relationship with Fresh Fruits and Vegetables Quality Assurance Trial

The Fresh Fruits and Vegetables (FFV) trial focusses on communication within a supply chain including several sources of information about products' characteristics and status. To enable chain actors to retrieve and distribute information from and to business partners a service is needed to connect the different information sources. The Product Information Service (PInfS) shall be used to facilitate the provision and consumption of product information according to business relationships. The FFV trial addresses the following use cases:

- Provision of product-related characteristics between business partners along a supply chain
- Provision of dynamic transport-related data to involved/interested partners
- Report of deviations and exceptions related to transport or food safety
- Facilitation of business processes (e.g. RTI (Returnable Trade Item) pool management)

To realize the use cases of the FFV trial required functionality needs to be provided to app developers by the PInfS. Table 26 maps requirements for the FFV trial defined in D400.1 on requirements of the PInfS defined in Section 2.2.2 of this document:

Table 26: Mapping between required functionalities for FFV trial use cases and defined PInfS requirements

Required functionalities (D400.1)	Related requirements (Section 2.2.2 of D400.6)
Aggregating and assigning of IDs and data	Product identification (PInfS_REQ01)
Identification of products	Product identification (PInfS_REQ01)
Mapping between products and deliveries	Product identification (PInfS_REQ01), System integration (PInfS_REQ05)
Definition of access rights for product, transport and business data	Data handling permissions management (PInfS_REQ04), Define/Store product info (PInfS_REQ06/07), Info security (PInfS_REQ08)
Discovery of available information	Product identification (PInfS_REQ01), Stakeholder identification (PInfS_REQ02), Product information meta-model (PInfS_REQ03), Data handling permissions management (PInfS_REQ04) System integration (PInfS_REQ05)

Required functionalities (D400.1)	Related requirements (Section 2.2.2 of D400.6)
Tracking of dynamic information	Product identification (PInfS_REQ01), Data handling permissions management (PInfS_REQ04), Define/Store product info (PInfS_REQ06/07), Info security (PInfS_REQ08), System integration (PInfS_REQ05)
Handling of deviations and exceptions	Product identification (PInfS_REQ01), Stakeholder identification (PInfS_REQ02), System integration (PInfS_REQ05), Rule definition/execution (PInfS_REQ14/15), Security execution (PInfS_REQ16)

PInfS relationship with Greenhouse Management & Control Trial

In the context of Greenhouse Management & Control Trial (T422) several different scenarios will take place, involving various business actors, legacy systems and baseline applications. In all T422 trial scenarios, the Product Information Service Baseline App is functional, and various services shall request information from it in order to complete successfully the business chain in which they participate.

The Greenhouse Management scenarios, which are directly related and dependent to the PInfS Baseline App, are:

- Advice Request
- Contracts
- Managing Complaints
- Product Recall
- Task Planning

The functionality of PinfS in all scenarios above is similar. Thus, below, one example of these scenarios is presented. The first version of the scenarios has already been modelled using the Archimate modelling language.

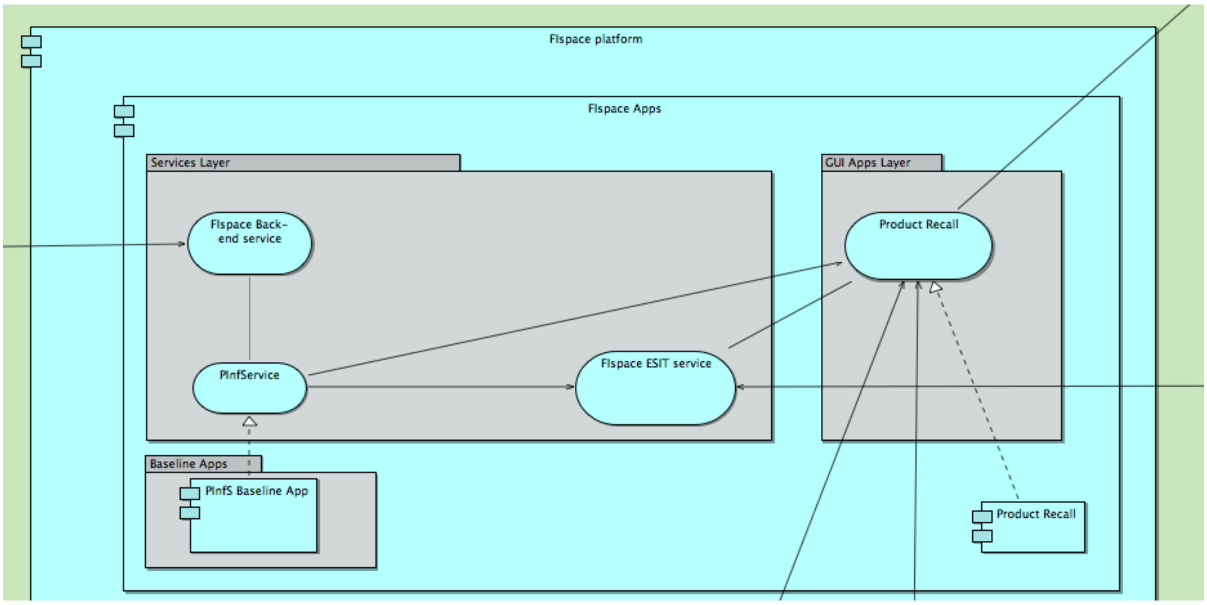


Figure 6:. Technology layer of Product Recall scenario in Archimate

The above figure shows the Technology Layer of the scenario description in Archimate. At the bottom-left part of the figure, the PinfS Baseline App is shown. The technology (service) layer of the procedure is as follows:

- A State Agency for Agricultural Policy is using the Product Recall Fispace App, downloaded from FIspace marketplace.
- At some point, a health hazard event breaks out. The State Agency receives a notification via the Product Recall FIspace App (***PinfS_REQ01 – Product Identification***, ***PInfS_REQ09 – Threshold Validation)***
- FIspace platform receives the notification and via the B2B Collaboration Module notifes the involved business actors about the particular product.
- FIspace ESIT Service, one of the services involved in the particular scenario, is a Product Traceability Platform. The particular service must retrieve specific information about the particular product, in order to be able to perform the traceability. (***PinfS_REQ01 – Product Identification, PinfS_REQ04*** **- Data handling permissions management,** ***PinfS_REQ08*** **- Info Security**)
- Similarly, the FIspace Back-end System Service must retrieve specific information concerning the product (***PinfS_REQ01 – Product Identification, PinfS_REQ04*** **- Data handling permissions management,** ***PinfS_REQ08*** **- Info Security**)
- Based on this information both the FIspace Back-end System Service and the FIspace ESIT platform will forward information back to the FIspace Product Recall App, which will be handled accordingly by the State Agency, to notify the appropriate Stakeholders (Farm Manager, Trader etc.)

8 Final Remarks and Next Steps

This document encloses the report of the activities until M3 within the scope of WP400 (Use Case Trials), Task 450 (T450 - Generic and Domain-specific Application Development), sub-task 451 (ST451 - Development of the cross-domain baseline applications). We reported the achievement the following objectives associated with deliverable D400 .6:

- Consolidation of the set of baseline apps to be developed
- Definition of main functionalities to be developed by the baseline apps
- Identification of main relationships with the core components of WP200 (FIspace Development Work Package)
- Identification of potential Generic Enablers (GEs) to be used by the baseline apps
- Definition of preliminary test requirements and relationships with WP300 (FIspace Hosting and Experimentation Work Package)

The baseline apps described in this document are:

- **Product Information Service App** - The main goal of the PInfS is to enable product information exchange between the stakeholders of a supply chain and towards the final consumer, based on the B2B collaborative and system integration capacities offered by the FIspace platform.
- **Logistics Planning Service App** - Supports the generation and maintenance of transport chain plans by using the description of the demands, real time information of logistics services, real time SLA information, execution statuses, and details of relevant events.
- **Business Profile App** - Designed to enable of business partners joining FIspace Platform to create their profile with the description of their interests and business scope. The ultimate goal of this app is to enable an automatic discovery of potential business partners with aligned business interests and scope.
- **Marketplace Operations App** - Provide the necessary functionality to the business partners to announce as well ask for offers and demands according to their specific criteria and preferences. Automatic mechanisms will also be supported for matchmaking between announced and requested services.
- **Real-time Business SLA Management App** - Bring into the operation of business activities the SLOs (Service Level Objectives) established in the SLA (Service Level Agreement) between two partners. Provide the solution for defining and managing the information inside the SLAs, enabling notifications about events associated with such SLA information itself as well as violations on the SLOs established in this SLA.

We provided the following description for each one of the baseline apps listed above: an overview on the baseline; the details in terms of functionalities to be supported by each app; the relationship with WP200 components; the relationship with GEs; the preliminary test requirements and relationship with WP300; and, summary associated with the baseline app concepts.

The next steps to be conducted by each team associated with the baseline apps are:

- Technical specification of the baseline app based on the support provided by WP200 components. Major challenges associated with this step are:
  - Identification of how the functional requirements of the baseline apps could be fulfilled by the use of the engines and model provided by ST240 (Real-time B2B Collaboration).
  - Alignment and providing feedback to WP200 components about the APIs offered and required.
- Plan and prepare the backlog, internal software releases, and sprints for the development of the baseline app functionalities until M9, when we have to provide the first release of the baseline apps implementation.
- Identify sources for the data necessary for testing the functionalities of the baseline apps as detailed in the each section associated with each app.
- Refinement and more analysis and alignments to determine how the trials are willing to use the functionalities of the baseline apps.

FIspace

Business Collaboration