Deliverable D200.7

System and Data Integration (SDI) component

WP 200

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)
Latest version of Annex 1:	2013-10-03
Start date of the project:	01.04.2013
Duration:	24
Status:	Draft
Editor:	Said Rahma (ATOS)
Contributors (to the "R" part of the deliverable[1]; ordered by project partner)	**ATOS**: Said Rahma Rodriguez, Javier Hitado
Document Identifier:	FIspace-D200.7-FIspace_Integrated_Release_V3-SDI-v0.4.docx
Date:	27.02.2015
Revision:	004
Project website address:	http://www.FIspace.eu

[1] Contributors to FIspace code ("P") include ATB, UDE, IBM, ATOS, KOC, TOG, AST, NKUA, UPM and LimeTri; contributing persons are listed at https://bitbucket.org/fispace/profile/members

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
- BO-MO; Slovenia
- MOBICS; Greece
- Fraunhofer IML; Germany
- Q-ray; Netherlands
- FINCONS; Italy
- 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
- CIT; Spain
- SDZ; Germany
- Snoopmedia; Germany
- EECC; Germany
- CBT; Spain

More Information

Harald Sundmaeker (coordinator) e-mail: sundmaeker@atb-bremen.de
Bert Vermeer (deputy coordinator) e-mail: bert.vermeer@wur.nl
Project Website Web link: http://www.fispace.eu/

Dissemination Level

PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	**X**
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 SDI component	09.12.2014
002	Updated of the contents, minor changes related to rephrasing and typo error, update of the list of modules that compose the SDI	20.01.2015
003	Internal review process, checking URL links to the FIspace Web online documentation Update of the abbreviation table, update of the references table Final version ready for submission of approved document	06.02.2015
004	Update of the coordinator information in the section "*More Information*" Added FIspace development repository and documentation references, formatting improvement Final version ready for submission to EC	27.02.2015
005		
006		

Abbreviations

AAA	Authentication, Authorisation, and Accounting
ACSI	Artifact-Centric Service Interoperation
AdvB	Advisory Board
AJAX	Asynchronous JavaScript + XML
API	Application Programming Interface
App	Software Application
B2B	Business-to-business
B2C	Business-to-Consumer
BCM	Business Collaboration Module in FIspace
BCO	Business Collaboration Objects in FIspace
BE	Business Entities
BPPC	Business Process Participant Configuration
BSS	Business Support Systems
CDR	Charging Detailed Records
CEP	Complex Event Processing
CSB	Cloud Service Bus
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
D	Deliverable
DAO	Data Access Object
DB	Database
DoW	Description of Work
EC	European Commission
EDI	Electronic Data Interchange
EE	Experimentation Environment
e.g.	Exempli gratia = for example
EPA	Event Processing Agent
EPM	Event Processing Module in FIspace
ESB	Enterprise Service Bus
EU	European Union
FIA	Future Internet Assembly
FI-PPP	Future Internet Public Private Partnership
FP7	Framework Programme 7
GA	Grant Agreement
GE	Generic Enabler
GUI	Graphical User Interface
HTML	HyperText Markup Language
IaaS	Infrastructure as a Service
ICT	Information and Communication Technology
IDE	Integrated Development Environment
IDM	Identity Management
i.e.	id est = that is to say
IE	Integration Environment
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
I/O	Input / Output
IoT	Internet of Things
IP	Intellectual Property
IP (protocol)	Internet Protocol
IPR	Intellectual Property Rights
IPsec	Internet Protocol Security
IT	Information Technology
ITU	International Telecommunication Union
ISO	International Standardization Organisation
J2SE	Java 2 Platform, Standard Edition
JDK	Java Development Kit
JDT	Related to Eclipse Java Development Tools
JMX	Java Management Extensions
JRE	Java Runtime Environment
JS	JavaScript
JSON	JavaScript Object Notation
JSP	Java Server Page
JVM	Java Virtual Machine
KPI	Key Performance Indicator
LPA	Logistics Planning Application
M	Month
MTBF	Mean Time Between Failures
MVC	Model–View–Controller
OASIS	Organization for the Advancement of Structured Information Standards
OAuth	Open standard Authentication protocol
OMG	Object Management Group
OSS	Operational Support Systems
P2P	Peer-to-peer
PaaS	Platform as a Service
PDE	Related to Eclipse Java Development Tools
PE	Production Environment
PIA	Product Information App

PIE	Preliminary Integration Environment
PKI	Public Key Infrastructure
PM	Person Month
POM	Project Object Model (used by maven tools)
Proton	IBM Proactive Technology Online
QoS	Quality of Service
RBAC	Role-Based Access Control
RCP	Rich Client Platform
REST	Representational State Transfer
RFC	Request for Comments
RSS	Revenue Sharing System
RTD	Research and Technological Development
SaaS	Software as a Service
SDI	System and Data Integration layer in FIspace
SDK	Software Development Kit
SME	Small and Medium Sized Enterprise
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOA-RM	(OASIS) Reference Model for Service Oriented Architecture
SPT	Security, Privacy and Trust Framework
SSH	Secure Shell
SSL	Secure Sockets Layer
SSO	Single Sign On
ST	Sub-Task
SWT	Standard Widget Toolkit
T	Task
TCP	Transmission Control Protocol
TIC	Tailored Information for Consumers
TLS	Transport Layer Security
TPM	Transport Planning Module
UAA	User Management, Authentication and Authorisation
UI	User Interface
UML	Unified Modeling Language
URI	Universal Resource Identifier
URL	Universal Resource Locator
USDL	Unified Service Description Language
VM	Virtual Machine
VPN	Virtual Private Network
W3C	World Wide Web Consortium
WADL	Web Application Description Language
WLAN	Wireless Local Area Network
WP	Work Package
WS	Web Service
WSDL	Web Services Description Language
XLS/XLSX	Microsoft Excel file Format
XML	eXtensible Markup Language
XSD	XML Schema Definition

Table of Contents

List of Figures

List of Tables

1 Introduction

This document aims at describing the third release (V3) of the FIspace, encompassing the implementations along with usage guidance and technical documentation of each FIspace component.

It reports on the description concerning the **SDI core component**, the description of the development and implementation of the SDI core components that is part of the FIspace platform.

The System and Data Integration Layer allows for the integration and continued usage of existing legacy and business systems as well as the integration of external systems and services, including support for:

- **Connecting business and legacy systems** used by individual users by means of Tool-supported mechanisms, supporting the creation of "connectors" (using common interface standards such as EDI) to business and legacy systems.
- **Connecting external services** (e.g., IoT or 3rd party services) by means of APIs for importing / exporting data (such as REST or SOAP).
- Handling heterogeneous data by means of mechanisms for **data mediation**.

The overarching purpose of System and Data Integration is to provide a robust and scalable infrastructure that enables seamless integration of external legacy systems/IoT systems with the FIspace platform and applications deployed on it. Outputs from the task will facilitate the implementation of Web based, FIspace-driven applications by providing unifying data models, data mediation tools and system integration APIs.

Online documentation for SDI: http://dev.fispace.eu/doc/wiki/sdi

1.1 Scope

The aim of this document is mainly to describe and detail the **FIspace SDI core component** at development and implementation level, giving detailed and technical information related to the design and the implementation as well as information about the related technologies and standard taken as a .reference to build each component.

Along this development activities and tasks, there is a set of resources, online documentation, tutorial and other external resource that refer to the Generic Enablers that can provide more technical information and user guides for the community and people who want to use the FIspace platform for Business collaboration or developers who want to create and develop business application (Apps developer) for a specific domain of application.

Table 1 shows the links to other online resources related to FIspace project and FI-WARE.

Description	Link
FIspace Business collaboration web site	http://www.fispace.eu/
FIspace Developer Documentation	http://dev.fispace.eu/doc/wiki/Home

web site	
FIspace Deliverables web site	http://www.fispace.eu/deliverable.html
FIspace Tutorial web site	http://www.fispace.eu/tutorials.html
FIWARE web site	http://www.fi-ppp.eu/projects/fi-ware/
FIWARE Catalogue of the Generic Enablers (GEs)	http://catalogue.fi-ware.org/
FIWARE community web site	http://www.fi-ware.org/community/

Table 1: Other FIspace and FIWARE resources

Table 2 shows the links to the Wirecloud online documentation.

Description	Link
FIWARE - Catalogue - Application Mashup - Wirecloud	http://catalogue.fi-ware.org/enablers/application-mashup-wirecloud
FIWARE - Catalogue - Application Mashup - Wirecloud Documentation	http://catalogue.fi-ware.org/enablers/application-mashup-wirecloud/documentation
FIWARE - Application Mashup - Wirecloud - User and Programmer Guide	https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/Application_Mashup_-_Wirecloud_-_User_and_Programmer_Guide
Dashboard - Wirecloud home page	http://conwet.fi.upm.es/wirecloud/
Dashboard - The WireCloud Mashup Platform	http://conwet.fi.upm.es/docs/display/wirecloud/The+WireCloud+Mashup+Platform
Dashboard - Welcome to CoNWeT-Wirecloud Confluence	http://conwet.fi.upm.es/docs/dashboard.action
Dashboard - User Guide	http://conwet.fi.upm.es/docs/display/wirecloud/WireCloud+User%27s+Guide
Dashboard - WireCloud Installation and Administration Guide	http://conwet.fi.upm.es/docs/display/wirecloud/Wire-Cloud+Installation+and+Administration+Guide

Table 2: Wirecloud online documentation

Table 3 shows the links to the WStore online documentation.

Description	Link
FIWARE - Catalogue - Store - WStore	http://catalogue.fi-ware.org/enablers/store-wstore

FIWARE - Catalogue - Store - WStore Documentation	http://catalogue.fi-ware.org/enablers/store-wstore/documentation
FIWARE - Store - W-Store - User and Programmer Guide	https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/Store_-_W-Store_-_User_and_Programmer_Guide
FIWARE - Store - W-Store - Store - W-Store - Installation and Administration Guide	https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/Store_-_W-Store_-_Installation_and_Administration_Guide

Table 3: Store online documentation

Table 4 shows the external development tools references.

Description	Link
Java Environment, JVM, JRE, JDK (Oracle)	http://www.oracle.com/technetwork/java/javase/downloads/index.html
Eclipse IDE (Integrated Development Environment)	https://www.eclipse.org/ , https://www.eclipse.org/downloads/
Maven	http://maven.apache.org/ , http://maven.apache.org/download.cgi

Table 4: External development tools references

Table 5 shows the FIspace development repository and documentation references based on the bitbucket tools for collaborative development.

Bitbucket is a hosting site for the distributed version control systems (DVCS) Git (http://git-scm.com/) and Mercurial (http://mercurial.selenic.com/). The service offering includes an issue tracker and wiki, as well as integration with a number of popular services such as Basecamp, Flowdock, and Twitter.

Description	Link
Bitbucket FIspace repository home page	https://bitbucket.org/fispace
Bitbucket FIspace core component home page	https://bitbucket.org/fispace/core/wiki/Home
Bitbucket FIspace Roadmap page	https://bitbucket.org/fispace/core/wiki/roadmap

Table 5: Bitbucket collaborative environment for FIspace development

1.2 Intended audience

The main interest groups of this deliverable are the participating teams and the responsible partners of FIspace project involved in the development activities, setup and preparation of the development phase. This document is relevant to the software engineer, programmers and developers who are the persons directly involved in the development, participating effectively on the design and implementation of the FIspace platform and the underlying components and sub-systems who want to know more about some technical information intrinsic to the FIspace platform.

At the technical level this document is relevant to: system architects; information systems designers; system developers and application developers; software engineers; other audiences who provide design services and applications using relevant standards and the recommendations of standards bodies like IETF, ITU, ISO, W3C, etc.

Partners involved in the integration tasks include: system integrators; people to test, validate and evaluate the FIspace platform and associated systems; can be also interested.

1.3 General remark

This document follows the ISO/IEC Directives, Part 2: Rules for the structure and drafting of International Standards w.r.t. the usage of the word “shall”. The word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this specification.

This document describes the corresponding core components involved in the FIspace core platform. It presents the development currently done and the corresponding implementation, the main features developed, as well as the related technologies and environment requirements.

In most of the following sections the structure is organized as:

- **Overview**: provides an overall introduction to the component, a description, of the internal architecture and features among other.
- **Interfaces or Application programming interface (API)**: describes the API accessible for the users or entities of the component (typically applications, but a component may also be used by other components).
- **Information model**: describes or specifies the component from an information perspective describing information objects of the component domain.
- **Interaction model**: describes or specifies main usage component “scenarios” associated with the component/GEs, sequence diagrams.
- **High level composite architecture**: describes or shows the main components constituting the set of components (this perspective is optional, since some component consists of only one main component).

Notice that some components only need to describe some of the item above described.

2 System and Data Integration (SDI)

2.1 Overview

The component called System and Data Integration (SDI) is a part of the core platform of the FIspace Project, and it can be defined as the gateway or entry point of the platform. In other words, this module is in charge of the communication between the “outside” world and FIspace. This communication is done in a bidirectional manner.

To communicate with external systems, the SDI offers a RESTful API. Also, whenever the SDI has to send information to an external system, the SDI expects the external system to provide a REST API according to FIspace SDI’s approach/guidelines, i.e the so-called “capabilities model” (see below). The next figure shows an overall picture of the SDI and its interactions with other modules. While the communication with the “outside world” is done via REST, the communication with other FIspace components is done via the Cloud Service Bus (CSB).

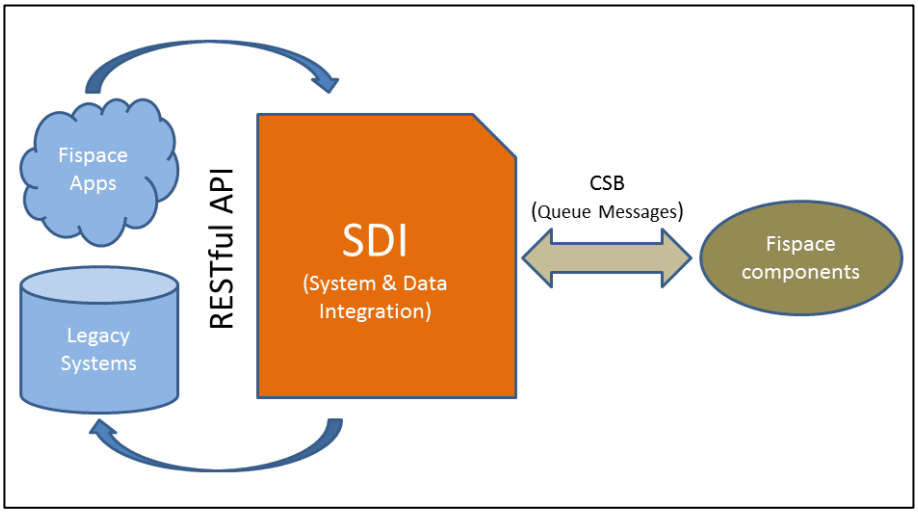

Figure 1: SDI interaction model overview

It should be noted that the so called “FIspace apps” are, from a technical point of view, seen as external systems and therefore have to use the SDI as gateway in case they want to send messages to other FIspace components (e.g. business collaboration messages to the B2B module or notification messages to the GUI). Therefore, in the following text the term “external system” also includes “FIspace apps”.

As an integration tool, the SDI uses Apache Camel[2] technology, which is a versatile open-source integration framework supporting known Enterprise Integration Patterns.

In order to develop as much as possible a scalable and generic component, recently a new concept has been introduced to expose the communication interface through the so-called “capabilities model”. This new approach is meant to facilitate the business to business collaboration, by offering capabilities that can be used in a business process

[2] Camel provides the tools to define routing and mediation rules in a variety of domain-specific languages, including a Java-based Fluent API, Spring or Blueprint XML Configuration files, and a Scala DSL. Camel provides the tools to define routing and mediation rules in a variety of domain-specific languages, including a Java-based Fluent API, Spring or Blueprint XML Configuration files, and a Scala DSL. Apache Camel uses URIs to work directly with any kind of Transport or messaging model such as HTTP, ActiveMQ, JMS, JBI, SCA, MINA or CXF, as well as pluggable Components and Data Format options permitting to work with the same API regardless which kind of Transport is used. (http://camel.apache.org/)

(an example could be the capability “provide greenhouse advice”, which is provided by a greenhouse expert and can be used in a “greenhouse advice business process” by a farmer who is searching for advice). In other words, capabilities are services that FIspace or a connected external system offers. From a technical point of view, each capability defines how a specific message exchange between FIspace and an external system works (e.g. which request message is sent, what is the expected response, REST method used, URL of the external system that provides the capability). As some typical capabilities like “provide product information” or “provide shipment information” could be used in several business processes and could be provided by different types of systems, there is also a generalization of capabilities foreseen: the so-called “capability types” define capabilities in a reusable way, so that different systems that implement a specific capability type can be used (replacing each other) in the same process.

2.2 Interfaces / API

The SDI exposes the external API defined in the FIspace core module. Although there exists a specific API, it is important to clarify how external systems communicate with SDI.

To perform the correct communication, before using the API, the business process to be managed by the FIspace platform should be clearly identified. In this manner, the next steps should be followed:

1. Clearly identify and define the businesses processes to be supported. The use case or trial is the responsible of this task.
2. Message definition. Once the business processes are identified, the messages that will be exchanged between FIspace and external systems have to be defined to be used during the business process. The business architect of the project should work together with the trial to define the correct messages.
3. Create the necessary capabilities in the SDI to be able to process in the FIspace platform the messages defined in the previous step. These capabilities have to be connected to the business process definition in the B2B.
4. Add the implementation of the capability (provision or usage of the capability) to the external system(s).
5. Then messages can be exchanged between the external systems and the SDI.

The definition of messages and capability types is regularly updated (at least once per each 4-week development cycle) and published at http://mvnrepository.com/artifact/eu.fispace/api/.

2.2.1 API operations

In the next table it can be found an explanation of the different methods and their description of the proposed external API that can, besides other tasks, also be used to add new capabilities to the FIspace registry, which can then be used in business processes. Nowadays working is aiming to have all those functionalities completely operative.

This table represents the current status of the Core API. Due to the continuous work on the task and the creation of new messages, this information might suffer minor changes due to the improvements and developments of future releases.

Method	URL	Description
GET	/api/capabilities	get all available capabilities (i.e. specific implementations of capability types by specific systems)
GET	/api/capabilities/(id)	get specified capability
GET	/api/capability-types	get all available capability types (i.e. reusable definitions of capabilities)
GET	/api/capability-types/(id)	get specified capability type
GET	/api/capabilities/for-request/(requestMessageType)	get a capability for a request message type
GET	/api/business-processes	get all available business processes (i.e. specific instances of business process templates)
GET	/api/business-processes/(id)	get business process for specified id
GET	/api/business-process-templates	get all available business process templates (i.e. reusable definitions of business processes)
GET	/api/available-resources	get all available resources (i.e. expert system resource URI)
GET	/api/available-resources/(capabilityId)/(businessProcessId)	get all available resources for specific capability and business process id
GET	/api/channels	get all channels (channel represents a data channel from an external system to FIspace)
GET	/api/channels/(id)	get channel for specified id
GET	/api/triggers	get all triggers (trigger represents and event detected by EPM)
GET	/api/triggers/(id)	get trigger for specified id

PUT	/api/capabilities	update or create a capability
PUT	/api/capability-types	update or create a capability type
PUT	/api/business-processes	update or create a business processes
PUT	/api/business-process-templates	update or create a business processes template
PUT	/api/available-resources	update or create an available resource
PUT	/api/channels	update or create a channel
PUT	/api/triggers	update or create a trigger
POST	/api/capabilities/{request_message}/(capabilityId)	i.e. request_message is “provide_weather_conditions”. This would be an example of a typical call of a client that uses a specific capability of type “provide weather conditions”. The capabilityId is used to define the specific implementation (i.e. the specific weather conditions provider) that should be used.
DELETE	/api/capabilities/(id)	remove a capability
DELETE	/api/capability-types/(id)	remove a capability type
DELETE	/api/business-processes/(id)	remove a business process
DELETE	/api/business-process-templates/(id)	remove a business process template
DELETE	/api/channels/(id)	remove a channel
DELETE	/api/triggers/(id)	remove a trigger

Table 6: SDI – Core REST API

In the next lines it is provided an example about how to retrieve a list of capabilities, retrieve a capability by Id and to add and remove a capability:

- Retrieve a list of capabilities:
  - Request: http://37.131.251.57:8080/fispace/api/capabilities (GET)
  - Response: a list of capabilities, including their id, the URI of the external system that provides the capability, the business process ID (bps_id) as well as the capability type id (cte_id)

Figure 2: SDI response to get capability list

- Retrieve a capability by id:
  - Request: http://37.131.251.57:8080/fispace/api/capabilities/1 (GET)
  - Response:

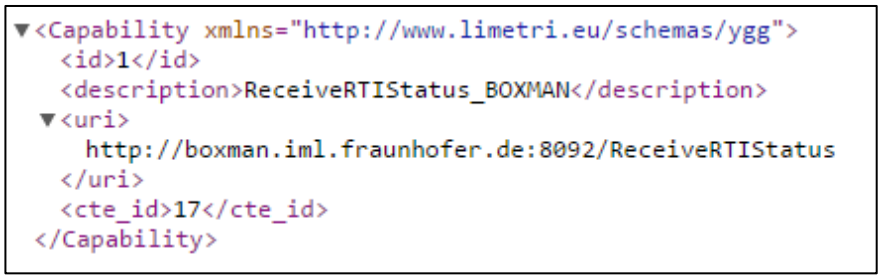

Figure 3: SDI response to get capability by id

- Add a capability:
  - Request: http://37.131.251.57:8080/fispace/api/capabilities (PUT)
  - Payload:

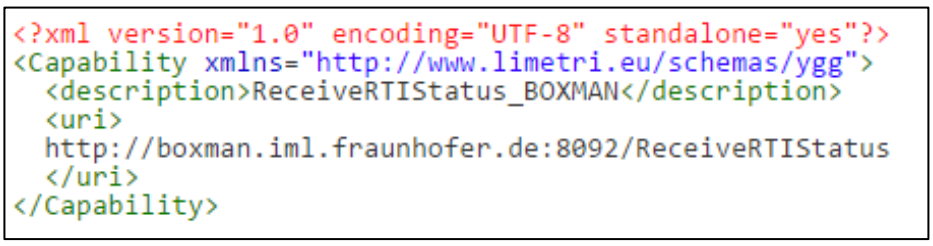

Figure 4: Payload example to add a capability

  - Response: 200 – OK

- Remove a Capability:
  - Request: http://37.131.251.57:8080/fispace/api/capabilities/1 (DELETE)
  - Response: 200 – OK

2.3 Information model

The SDI uses messages to exchange information. Those messages are defined in the core module of FIspace. In the next lines is explained how those messages should be generated.

The Business Architect is responsible for the definition of the business processes to address. Once the business processes are clearly defined, they are uploaded to the Collaboration Engine supported by the FIspace B2B module. During this process one or more capabilities and messages will have to be defined. In case of developing new capabilities, new messages will also have to be defined.

The next class diagram provides a graphical view of the process to be followed, depending on the role of the user (Business Architect or FIspace User):

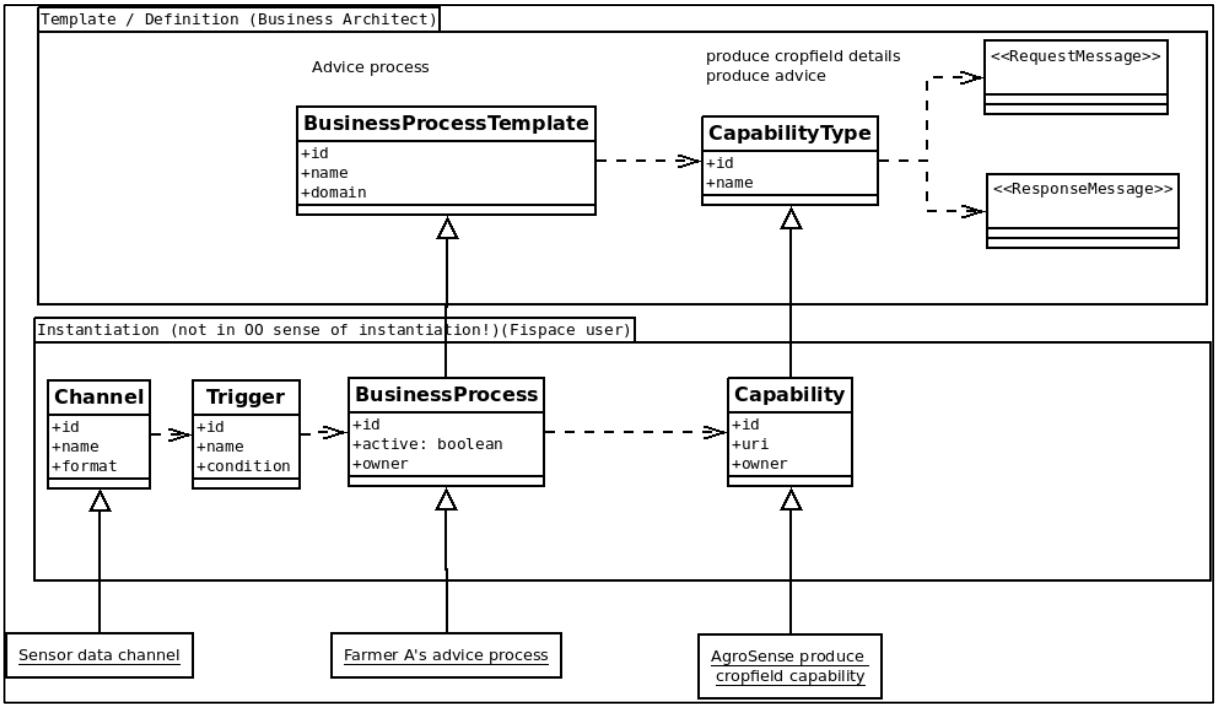

Figure 5: Class diagram representing information model process

The classes in the diagram are explained in the following:

- BusinessProcessTemplate represents a reusable definition of a business process, e.g. the generalized “greenhouse advice process” that involves a farmer entity who needs advice and an expert entity who provides the advice
- BusinessProcess represents an instance of a template, based on a contract between specific process participants, e.g. the greenhouse advice process involving Frank Farmer and Ed Expert
- CapabilityType is a reusable definition of a type of capability that can be provided by an external system and used in a business process template, e.g. the greenhouse advice process includes the capabilities “provide greenhouse sensor in-

formation" (to be implemented by the farmer entity) and "provide advice" (to be implemented by the expert entity)

- Capability is the actual implementation of a capability type by a specific system, e.g. in the greenhouse advice process this would be the capability "provide greenhouse sensor information" (implemented by Frank Farmer's farm management system) and "provide advice" (implemented by Ed Experts expert system)
- Channel represents a "lightweight" data channel from an external system to FIspace' Event Processing Module, e.g. in the greenhouse advice process this would be Frank Farmers greenhouse sensor network connected to FIspace, sending current greenhouse sensor information in regular time intervals
- Trigger represents an event detected by the FIspace Event Processing Module, e.g. in the greenhouse advice process the business process would be triggered when the Event Processing Module would detect that greenhouse sensor values from Frank Farmer's greenhouse are "out of defined boundaries"

All these information generated:

- Business Process Templates and Capability Types from the Business Architect perspective
- Channel, Trigger, Business Process and Capability from the FIspace User perspective

will be stored in the Registry module of the SDI by using the specific methods (already explained in the API operations chapter) of the core API exposed. Additionally, the system will be able to use this information during runtime execution what will allow the SDI to properly complete its main functions, i.e. message routing

2.4 Interaction model

In this section it is presented, as an example, the Weather Condition Scenario where the main actors involved are the weather condition app (external system), the weather condition provider (external system) and the FIspace platform.

In this example, the weather condition app requests weather conditions from the weather condition provider. But instead of direct communication between the two participants, communication is routed through FIspace in order to decouple the weather condition app from a specific implementation of the weather condition provider.

The process starts when the app asks for new weather condition information and ends when the app has successfully fetched the requested information from the provider.

All communication indicated in the diagram is technically realized as REST web service calls, therefore for each HTTP request there is a corresponding HTTP response indicated.

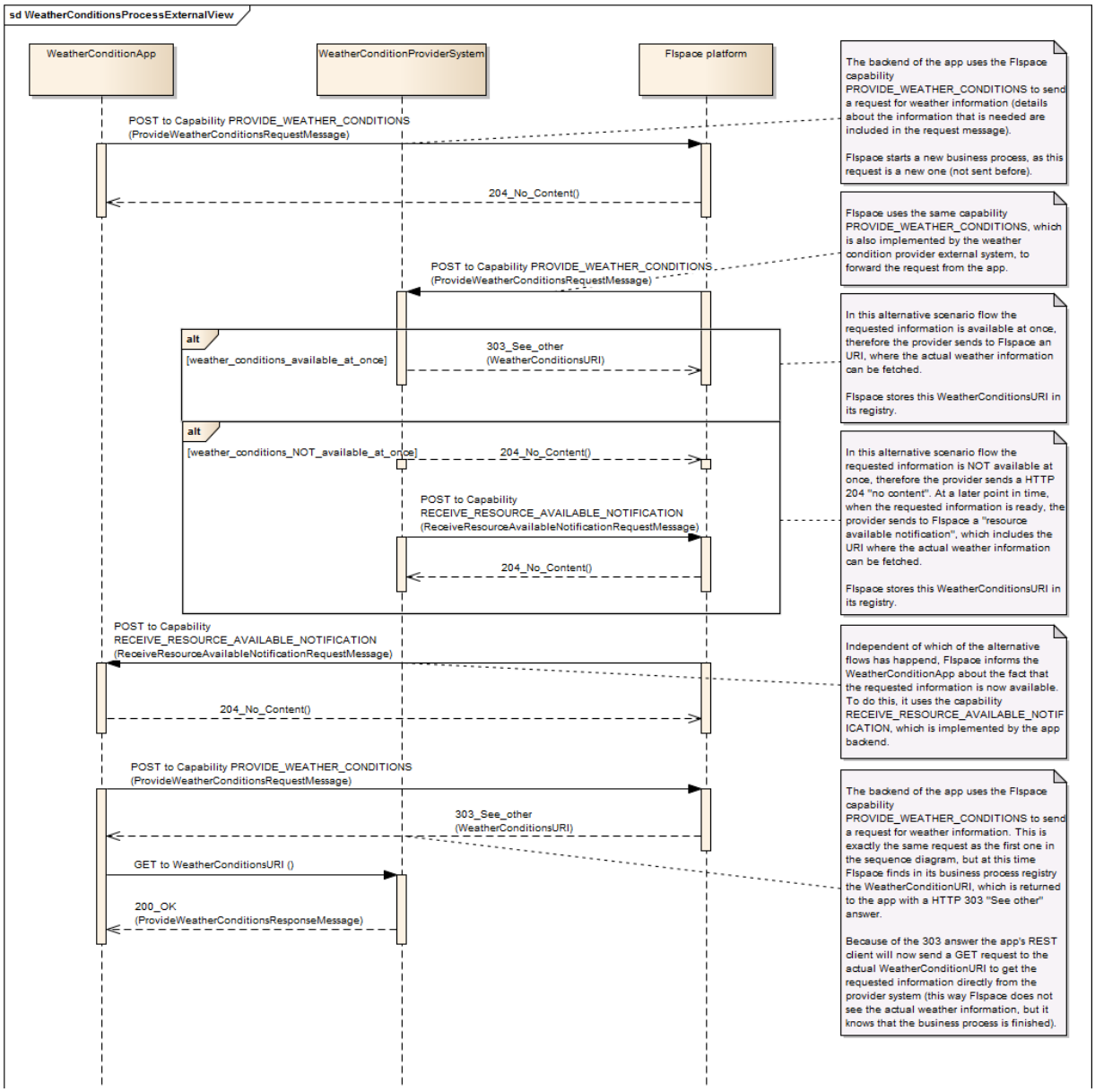

Figure 6: Weather Condition scenario diagram

2.5 High level composite architecture

At the time of writing this deliverable, the high-level architecture of the present status of the SDI is depicted in the figure below. Due to the continuous work on the task, minor changes could be found in future releases of the component.

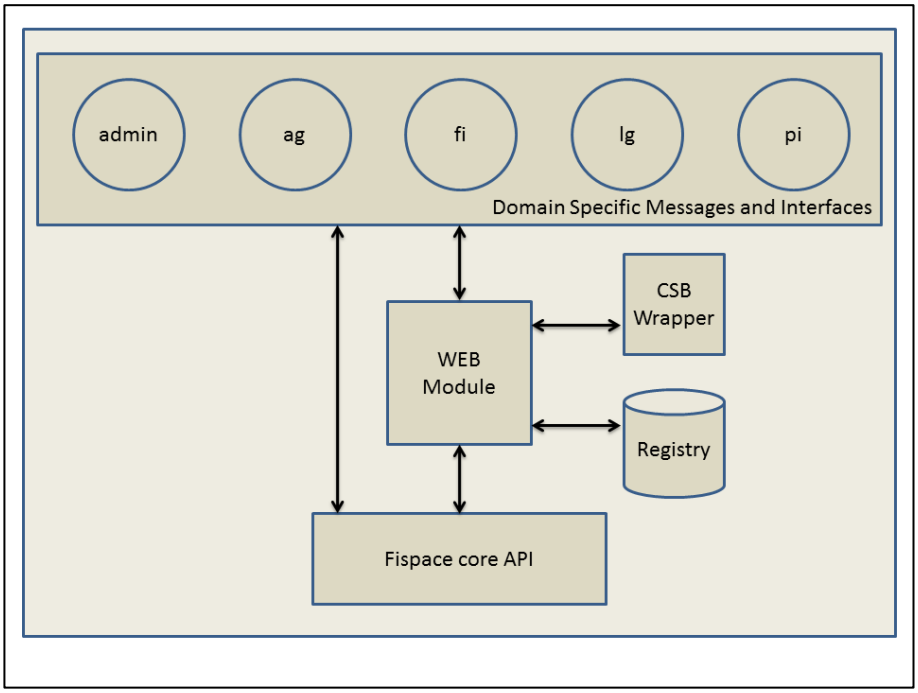

Figure 7: SDI high-level architecture.

As it can be seen in the figure, there several modules that compose the SDI are:

- **WEB**: This module is implemented using Apache Camel technology, and it is the resulting application of the junction of the different SDI components. It enables the communication between the external systems or apps and the different modules of the SDI providing a complete set of functionalities. The different required modules that compose the WEB[3] module are included using maven technology, in other words, included as maven dependencies.
- **Core**: SDI provides an implementation of the FIspace Core API. The FIspace Core is directly related with the different domains, as it the responsible for the management and maintenance of the different messages or interfaces that the FIspace platform is going to expose. This module can be considered as the "core" of the FIspace platform.
- **Domains Specific Messages and Interfaces**: FIspace offers its platform for different domains of application. Depending on the requirements of each "FIspace app", each domain module will be the responsible for its correct management among the platform. At the time of writing this document, the domains available are:
  - AG: Agriculture
  - LG: Logistics
  - PI: Product Information
  - ADMIN: To provide communication between the components that compose the platform
  - FI: Fish

  Due to the intensive work done on the integration of new Trials, the number of domains will continuously be increased.

[3] Packaged as a WAR file and it can be run in any web application server. For FIspace project it is being using the WildFly application server, which is flexible, lightweight, managed application runtime server based on JBOSS. (http://wildfly.org/)

- **CSB Wrapper**: Java code in charge of the exchange of information between the SDI and the CSB.
- **Registry**: This component represents the database used by the SDI. Depending on the requirements of the user it can be configured to use an Apache Derby[4] database or a MySQL database.
- **Security:** SDI has been successfully integrated in the security single sign-on solution provided by FIspace (Keycloak [32]). The access type that has to be used by SDI is “bearer-only” although for testing purposes it has been used “confidential” as access type. Using bearer-only access type means that the application will only allow bearer token request and so the SDI will not provide browser login while providing confidential access-type there is a need to perform a browser login, Using this new security feature, whenever a capability is created, it is assigned to the user that has created it.

[4] Lightweight embedded open source relational database implemented entirely in Java and based on SQL standards (http://db.apache.org/derby/)

3 Glossary

The glossary provides the coherent terminological framework used in this document.

3.1 Terms and definitions

This section provides definitions of any terms that may be needed in order for the reader to understand the terminology used in the document. The author should define any definition/acronym or technical term used in the document that may be unfamiliar to the reader, and it is best to err on the side of too many rather than too few definitions. This also allows the author to frame a word within a specific context, which provides the reader with a common understanding of the author's definition.

Access control

Authorisation (or denegation) for performing a certain action (based on privileges management). The access control is carried out once the Identification and Authentication procedures have been performed.

Accounting

Process of gathering information about the usage of resources by subjects.

Acceptance and trust

Acceptability indicates the degree of approval of a technology by the users. It depends on whether the technology can satisfy the needs and expectations of its users and potential stakeholders. Within the framework of introducing new technologies, acceptability relates to social and individual aspects as well.

Application

Use of capabilities, including hardware, software and data, provided by an information system specific to the satisfaction of a set of user requirements in a given application domain.

Application Domain

Integrated set of problems, terms, information and tasks of a specific thematic domain that an application (e.g. an information system or a set of information systems) has to cope with.

Application Schema [ISO/FDIS 19109:2003]

Conceptual schema for data required by one or more applications.

Architecture (of a system) [ISO/IEC 10746-2:1996]

Set of rules to define the structure of a system and the interrelationships between its parts.

Architecture (of a system) [ISO/IEC 10746-2:1996]

Set of rules to define the structure of a system and the interrelationships between its parts.

Authentication

Process of verifying the identity of a certain subject. In other words authentication indicates whether a subject is who/what it seems to be.

Generally speaking, this proof can depend on a secret that can be, e.g. what somebody has (key, smart card, …), what somebody knows (password, …), what somebody is (biometrical data, …)

Authorisation

Process of determining whether a subject is allowed to have the specified types of access to a particular resource. This is done by evaluating applicable access control information contained in a so called authorisation context. Usually, authorisation is carried out after the identification and authentication. Once a subject is identified and authenticated, it may be authorized (or not) to perform different types of access.

Availability

Availability refers to the degree to which a system, subsystem, or equipment is in a specified operable and committable state at the start of a mission, when the mission is called for at an unknown, i.e., a random time. So, availability is the proportion of time that a system is in operating condition.

Capability

Capabilities are a set of functionalities, through a combination of software and hardware, used to provide services and data. They can reside in a system or for example in a terminal itself as embedded capabilities or they can be available through the network services and infrastructure and others communication technologies as external capabilities.

Catalogue [derived from http://www.opengeospatial.org/resources/?page=glossary]

Collection of entries, each of which describes and points to a feature collection. Catalogues include indexed listings of feature collections, their contents, their coverages, and of meta-information. A catalogue registers the existence, location, and description of feature collections held by an Information Community. Catalogues provide the capability to add and delete entries. A minimum Catalogue will include the name for the feature collection and the locational handle that specifies where these data may be found. Each catalogue is unique to its Information Community.

Certificate Authority

A Trusted Third Party, responsible for ensuring the binding between the public keys and the personal data of their respective owners.

Component

Hardware component (device) or Software Component.

Conceptual model [ISO/FDIS 19109:2003(E); ISO 19101]

Model that defines concepts of a universe of discourse.

Conceptual schema [ISO/FDIS 19109:2003(E); ISO 19101]

Formal description of a conceptual model.

Coverage [ISO 19123]

Function from a spatial, temporal or spatiotemporal domain to an attribute range. A coverage associates a position within its domain to a record of values of defined data types. Thus, a coverage is a feature with multiple values for each attribute type, where each direct position within the geometric representation of the feature has a single value for each attribute type.

Data acquisition

Methods of data acquisition include methods to collect background data, digitally acquire data from sensors, and subjective data (such as data acquired from questionnaires). In addition, data in the form of manually or automatically transcribed data and reductions of collected data is also considered sensor acquired data (but with a manual sensor – the analyst).

Description Logics

Family of logic based knowledge representation languages that are a decidable subset of first order logic with well-defined semantics and inferencing (problem decision procedures). In Description Logics, a distinction is made between the terminological knowledge and the assertional knowledge. This distinction is useful for knowledge base modelling and engineering: for modelling it is just natural to distinguish between concepts and individuals; for engineering it helps by separating key inference problems.

Digital Certificate

A kind of digital document that contains structured information about the identity of its owner along with her/his public key, signed all together with a Certificate Authority's private key.

Digital Signature

The encrypted form of a message with the private key of the owner, indicating in a secure way the creator of the message, as well as the identity of a signed data.

Encryption

The act of modifying the contents of a message in an algorithmic and secure way, so that it can not be observed or altered in while in transit.

End-User

All users that are involved in an application domain and that use the applications, the services built by the system users according to the system and service Architecture.

Feature [derived from ISO 19101]

Abstraction of a real world phenomenon [ISO 19101] perceived in the context of an Application. In this general sense, a feature corresponds to an "object" in analysis and design models.

Framework [http://www.opengeospatial.org/resources/?page=glossary]

An information architecture that comprises, in terms of software design, a reusable software template, or skeleton, from which key enabling and supporting services can be selected, configured and integrated with application code.

Generic

A service is generic, if it is independent of the application domain. A service infrastructure is generic, if it is independent of the application domain and if it can adapt to different organisational structures at different sites, without programming (ideally).

Identification

The identification process allows relating a person/device with the service environment. The “electronic identity” is something like a credential or a “business card”, suitable to be verified throughout the authentication process.

Implementation [http://www.opengeospatial.org/resources/?page=glossary]

Software package that conforms to a standard or specification. A specific instance of a more generally defined system.

Info-structure Service

Service that is required to operate a system oriented service in the sense that it plays an indispensable role in the operation of an architecture or system oriented service.

Interface [ISO 19119:2005; http://www.opengis.org/docs/02-112.pdf]

Named set of operations that characterize the behaviour of an entity.

The aggregation of operations in an interface, and the definition of interface, shall be for the purpose of software reusability. The specification of an interface shall include a static portion that includes definition of the operations. The specification of an interface shall include a dynamic portion that includes any restrictions on the order of invoking the operations.

Interoperability [ISO 19119:2005 or OGC; http://www.opengeospatial.org/resources/?page=glossary]

Capability to communicate, execute programs, or transfer data among various functional units in a manner that require the user to have little or no knowledge of the unique characteristics of those units [ISO 2382-1]. (http://www.opengeospatial.org/ogc/glossary/i)

Loose coupling [W3C; http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#loosecoupling]

Coupling is the dependency between interacting systems. This dependency can be decomposed into real dependency and artificial dependency: Real dependency is the set of features or services that a system consumes from other systems. The real dependency always exists and cannot be reduced. Artificial dependency is the set of factors that a system has to comply with in order to consume the features or services provided by other systems. Typical artificial dependency factors are language dependency, platform dependency, API dependency, etc. Artificial dependency always exists, but it or its cost can be reduced. Loose coupling describes the configuration in which artificial dependency has been reduced to the minimum.

Middleware [http://www.opengeospatial.org/resources/?page=glossary]

Software in a distributed computing environment that mediates between clients and servers.

Open Architecture [based on (Powell 1991)] [33]

Architecture whose specifications are published and made freely available to interested vendors and users with a view of widespread adoption of the architecture. An open ar-

chitecture makes use of existing standards where appropriate and possible and otherwise contributes to the evolution of relevant new standards.

Operation [ISO 19119:2005; http://www.opengis.org/docs/02-112.pdf]

Specification of a transformation or query that an object may be called to execute. An operation has a name and a list of parameters.

Performance indicators definition (PI)

PIs are quantitative or qualitative measurements, agreed on beforehand, expressed as a percentage, index, rate or other value, which is monitored at regular or irregular intervals and can be compared with one or more criteria.

Platform (Service)

Set of infrastructural means and rules that describe how to specify service interfaces and related information and how to invoke services in a distributed system.

Reference Model [ISO Archiving Standards; http://ssdoo.gsfc.nasa.gov/nost/isoas/us04/defn.html]

A reference model is a framework for understanding significant relationships among the entities of some environment, and for the development of consistent standards or specifications supporting that environment. A reference model is based on a small number of unifying concepts and may be used as a basis for education and explaining standards to a non-specialist.

Reliability

Reliability is the ability of a system or component to perform its required functions in routine circumstances, as well as hostile or unexpected circumstances, under stated conditions for a specified period of time.

Resource

Functions (possibly provided through services) or data objects.

Service [ISO 19119:2005; ISO/IEC TR 14252; http://www.opengis.org/docs/02-112.pdf]

Distinct part of the functionality that is provided by an entity through interfaces.

REST

Representational state transfer (REST) is an abstraction of the architecture of the World Wide Web; more precisely, REST is an architectural style consisting of a coordinated set of architectural constraints applied to components, connectors, and data elements, within a distributed hypermedia system. REST ignores the details of component implementation and protocol syntax in order to focus on the roles of components, the constraints upon their interaction with other components, and their interpretation of significant data elements.

Service [ISO 19119:2005; ISO/IEC TR 14252; http://www.opengis.org/docs/02-112.pdf]

Distinct part of the functionality that is provided by an entity through interfaces.

Session

Temporary association between a subject and a principal as a result of an authentication process initiated by the subject. Information about a session is stored in authentication session information.

SOAP

Simple Object Access protocol is a protocol specification for exchanging structured information in the implementation of web services in computer networks. It uses XML Information Set for its message format, and relies on other application layer protocols, most notably Hypertext Transfer Protocol (HTTP) or Simple Mail Transfer Protocol (SMTP), for message negotiation and transmission.

Software Component [derived from component definition of http://www.opengeospatial.org/resources/?page=glossary]

Software program unit that performs one or more functions and that communicates and interoperates with other components through common interfaces.

Source System

Container of unstructured, semi-structured or structured data and/or a provider of functions in terms of services. The source systems are of very heterogeneous nature and contain information in a variety of types and formats.

Support Service

Service that facilitates the operation of an architecture or system oriented service, e.g. providing an added value by combining the usage of Info-Structure Services.

System [ISO/IEC 10746-2:1996]

Something of interest as a whole or as comprised of parts. Therefore a system may be referred to as an entity. A component of a system may itself be a system, in which case it may be called a sub-system.

Note: For modelling purposes, the concept of system is understood in its general, system theoretic sense. The term "system" can refer to an information processing system but can also be applied more generally.

System User

Provider of services that are used for an application domain as well as IT architects, system developers, integrators and administrators that conceive, develop, deploy and run applications for an application domain.

Terminal

Terminals are a mobile device that is capable of running mobile services and/or mobile applications.

Use case

A common definition of use cases is the one described by Jacobson (Jacobson et al (1995) [34]): "*When a user uses the system, she or he will perform a behaviourally related sequence of transactions in a dialogue with the system. We call such a special sequence a use case*". In Other words, a use case is a textual presentation or a story about the usage of the system told from an end user's perspective.

The use cases provide some tools for people, with different skills (e.g. software developers and non-technology oriented people), to communicate with each other. The use

cases are general descriptions of needs or situations that often are related to basic scenarios and that are independent of the technologies and implementations of the underlying system.

User

Human acting in the role of a system user or end user of the service and system.

WADL

The Web Application Description Language is a machine-readable XML description of HTTP-based web applications (typically REST web services) WADL models the resources provided by a service and the relationships between them. WADL is intended to simplify the reuse of web services that are based on the existing HTTP architecture of the Web. It is platform and language independent and aims to promote reuse of applications beyond the basic use in a web browser.

Web Service

Self-contained, self-describing, modular service that can be published, located, and invoked across the Web. A Web service performs functions, which can be anything from simple requests to complicated business processes. Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service.

W3C Web Service [W3C, http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice]

Software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

4 References

The following references are used as background documents for the preparation of this document. References are categorized standards (i.e. standards and specifications from the consortium working groups or alliances and specifications or drafts standardization bodies) and other documents, publications and technical or scientific books.

[1]	FIspace project. FIspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics. Deliverable D200.1 *"FIspace Design and Release Plan"*, 2014.
[2]	FIspace project. FIspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics. Deliverable D200.2 *"FIspace Technical Architecture and Specification"*, 2014
[3]	FIspace project. FIspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics. Deliverable D200.3 *"FIspace Integrated Release V1"*, 2014
[4]	FIspace project. FIspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics. Deliverable D200.4 *"FIspace Development Progress Report and V1 Updates"*, *2014*
[5]	FIspace project. FIspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics. Deliverable D200.5 *"FIspace Integrated Release V2"*, 2014.
[6]	FIspace project. FIspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics. Deliverable D200.6 *"FIspace Development Progress Report and V2 Updates"*, *2014*
[7]	FIspace project. FIspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics. D200.5 Annex *"FIspace Front-End User Guide"*
[8]	FIspace project. FIspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics. D200.5 Annex *"FIspace SDK User and Developer Guide"*
[9]	FIspace project. FIspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics. Deliverable D200.7 *"FIspace Integrated Release V3"*, 2014.
[10]	FIspace project. FIspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics. D200.7 Annex *"FIspace Front-End User Guide"*
[11]	FIspace project. FIspace: Future Internet Business Collaboration Networks in Agri-Food, Transport and Logistics. D200.7 Annex *"FIspace SDK User and Developer Guide"*

[12]	FIspace Business collaboration web site. http://www.fispace.eu/
[13]	FIspace Developer Documentation web site. http://dev.fispace.eu/doc/wiki/Home
[14]	FIspace Deliverables web site. http://www.fispace.eu/deliverable.html
[15]	FIspace Tutorial web site. http://www.fispace.eu/tutorials.html
[16]	FIspace Front-End Users Information web site. http://dev.fispace.eu/doc/wiki/gui
[17]	FIspace Front-End User Guide web site. http://dev.fispace.eu/doc/wiki/gui/gui-guide
[18]	FIspace App Developer Intro web site. http://dev.fispace.eu/doc/wiki/App%20Developer%20Intro
[19]	FIspace SDK Guide web site. http://dev.fispace.eu/doc/wiki/sdk
[20]	FIspace apps for newbies web site. https://bitbucket.org/fispace/apps/wiki/FIspace%20apps%20for%20newbies
[21]	FIWARE web site. http://www.fi-ppp.eu/projects/fi-ware/
[22]	FIWARE Catalogue of the Generic Enablers (GEs). http://catalogue.fi-ware.org/
[23]	FIWARE community web site. http://www.fi-ware.org/community/
[24]	FIWARE - Catalogue - Application Mashup – Wirecloud web site. http://catalogue.fi-ware.org/enablers/application-mashup-wirecloud
[25]	FIWARE - Catalogue - Store – Wstore web site. http://catalogue.fi-ware.org/enablers/store-wstore
[26]	Eclipse web site. https://www.eclipse.org/ , https://www.eclipse.org/downloads/
[27]	Maven web site. http://maven.apache.org/ , http://maven.apache.org/download.cgi
[28]	RabbitMQ web site. http://www.rabbitmq.com/
[29]	Express web site. http://expressjs.com/
[30]	ACSI EU-project web site. http://www.acsi-project.eu/
[31]	Proton - CEP recorded webinar and tutorial. http://edu.fi-ware.eu/course/view.php?id=58 Proton - CEP user and programmers guide. http://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/CEP_-

	_User_and_Programmer_Guide
[32]	Keycloak JBoss project. http://keycloak.jboss.org/ Keycloak JBoss documentation. http://keycloak.jboss.org/docs
[33]	Powell, D. (Ed.) (1991). Delta-4: A Generic Architecture for Dependable Distributed Computing. Re-search Reports ESPRIT. Project 818/2252 Delta-4 Vol.1. ISBN 3-540-54985-4 Springer-Verlag 1991.
[34]	Jacobson, I., Bylund, S., Jonsson, P., and Ehneboom, S. (1995), “Mod-eling with Use Cases: Using contracts and use cases to build plugable architectures”. Journal of Object Oriented Programming, Vol. 8, No. 2, pp. 18-24.


FIspace
Business Collaboration