Deliverable D200.7

Business to business (B2B) 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 **IBM**: Fabiana Fournier, Moti Nisenson, Inna Skarbovsky
Document Identifier:	FIspace-D200.7-FIspace_Integrated_Release_V3-B2B-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 B2B component	09.12.2014
002	Update of contents, added new section related to main features	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 **B2B core component**, the description of the development and implementation of the B2B core components that is part of the FIspace platform.

At the heart of the envisaged FIspace platform reside the Business-to-Business Core Modules. The B2B Core ensures that all information and status updates are provided to each involved stakeholder in real-time. The B2B core allows for the creation, management, execution, and monitoring of collaborative business processes in the FIspace platform. The B2B Core consists of two interrelated components:

- A **Collaboration Engine** that captures, in form of so-called Business Entities, the information that are to be exchanged among collaborating stakeholders along with status and control of the a collaborative business processes. The BCM component is responsible to orchestrate the different processes from different stakeholders and assure the correct sequence of the tasks execution.
- An **Event Processing Engine** that detects and analyses events coming from activities in the collaborative processes or from IoT devices. The Event Processing Module (EPM) component monitors events and detect situations of interest, i.e., situations that require appropriate reactions.
- **Authoring tools**: Both engines will be accompanied by respective authoring tools that allow defining business entities resp. event rules.

The BCM component is responsible to orchestrate the different processes from different stakeholders and assure the correct sequence of the tasks execution. The BCM is based on the entity-centric approach (for more details, please refer to deliverable D400.10). This approach relies on the notion of *entities* (aka, as business entities, artefacts, or dynamic artefacts, or business collaboration objects). These provide a holistic marriage of data and process, both treated as first-class citizens, as the basic building block for modelling, specifying, and implementing services and business processes. A (business) entity is a key conceptual concept that evolves as it moves through a business (or other) process. An entity type includes both a data schema and a lifecycle schema which are tightly linked. The data schema provides an end-to-end conceptual view of the key data for this entity type. The lifecycle schema of an entity type specifies the different ways that an entity instance might evolve as it moves through the overall process. In FIspace we will use the GSM (Guards, Stages, and Milestones) model to specify the lifecycle schema of the business entities.

The Event Processing Module (EPM) component monitors events and detect situations of interest, i.e. situations that require appropriate reactions. The events sources (aka events producers) can be the actual execution of the collaboration (i.e., the BCM), external systems, or sensors. The EPM processes these events and by applying pattern matching derives situations of interest (for a background on event processing refers to Section 2). Examples of situations of interest can be: Missing documentation at a certain point in time, a sensor reading outside a permitted range, a delay in a delivery. In general, we can distinct between situations that result from the actual execution of the process or collaboration and situations that result from external events (i.e., events coming from external systems or sensors).

The EPM in FIspace supports two types of situation detection capabilities: reactive and proactive. Reactive rules analyse past events and derive situations by applying pattern matching over a single or a set of events over time. Proactive rules, on the other hand, relate to situations that are likely to happen in the (near) future. In general, we refer to proactive event-driven computing as the ability to mitigate or eliminate undesired states, or capitalize on predicted opportunities—in advance. This is accomplished through the online forecasting of future events, the analysis of events coming from many sources, and the application of online decision-making processes.

Online documentation for B2B Core: http://dev.fispace.eu/doc/wiki/b2b

1.1 Scope

The aim of this document is mainly to describe and detail the **FIspace B2B 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 web site	http://dev.fispace.eu/doc/wiki/Home
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 B2B – Event Processing Module/Business Collaboration Module (EPM/BCM)

2.1 Overview

At the heart of the envisaged FIspace platform reside the Business-to-Business (B2B) Core Modules. The B2B Core ensures that all information and status updates are provided to each involved stakeholder in real-time. The B2B core allows for the creation, management, execution, and monitoring of collaborative business processes in the FIspace platform. The B2B Core consists of two interrelated and complementary components:

- A Collaboration Engine that captures, in the form of Business Entities, the information to be exchanged among collaborating stakeholders along with status and control of the a collaborative business processes. The BCM component is responsible to orchestrate the different processes from different stakeholders and assure the correct sequence of the tasks execution.
- An Event Processing Engine that detects and analyses events coming from activities in the collaborative processes or from IoT devices. The Event Processing Module (EPM) component monitors events and detect situations of interest in real-time, i.e., situations that require appropriate reactions.

The BCM component is responsible to orchestrate the different processes from different stakeholders and assure the correct sequence of the tasks execution. The BCM is based on the entity-centric approach (for more details, please refer to the outcomes of the ACSI project - Artifact-Centric Service Interoperation (ACSI) [30]).

This approach relies on the notion of business entities (aka, as (dynamic/business) artefacts). These provide a holistic marriage of data and process, both treated as first-class citizens, as the basic building block for modelling, specifying, and implementing services and business processes.

A (business) entity is a key conceptual concept that evolves as it moves through a business (or other) process. An entity type includes both a data schema and a lifecycle schema which are tightly linked. The data schema provides an end-to-end conceptual view of the key data for this entity type. The lifecycle schema of an entity type specifies the different ways that an entity instance might evolve as it moves through the overall process. In FIspace we apply the GSM (Guards, Stages, and Milestones) model to specify the lifecycle schema of the business entities.

BCM tooling: The BCM relies on and extends the BizArtifact open source tool, a.k.a ACSI, (both design and run times)

The Event Processing Module (EPM) component monitors events and detect situations of interest, i.e. situations that require appropriate reactions. The events sources (aka events producers) can be the actual execution of the collaboration (i.e., the BCM), external systems, or sensors.

The EPM processes these events and by applying pattern matching derives situations of interest. Examples of situations of interest can be: Missing documentation at a certain point in time, a sensor reading outside a permitted range, or a delay in a delivery. In general, we can distinct between situations that result from the actual execution of the process or collaboration and situations that result from external events (i.e., events coming from external systems or sensors).

These two modules are connected each to the other and to the other FIspace components of the platform via the Cloud Service Bus (CSB) component (See section of the corresponding document related to the CSB and online documentation in http://dev.fispace.eu/doc/wiki/csb).

EPM tooling: The EPM relies on and extends the CEP GE (Complex Event Processing Generic Enabler), a.k.a Proton, from FI-WARE (both design and run times)

2.2 Main features

We outline below the main features that have been extended to both the BCM and EPM engines. To better understand the modus operandi in each engine, please refer to the “B2B Core Modules documentation for business architects” document.

2.2.1 BCM

BizArtifact tool comprises a run-time engine and a design-time authoring tool using a single declarative programming model (GSM).

The BCM extends the BizArtifact open source tool as follows:

- Templates support
- Integration with the CSB
- Request/response (sync. mechanism) support
- Query API support

2.2.2 EPM

Proton comprises a run-time engine, producers, and consumers with the characteristics and capabilities to develop, deploy, run, and maintain event driven applications using a single declarative programming model.

The EPM extended the CEP GE from FI-WARE as follows:

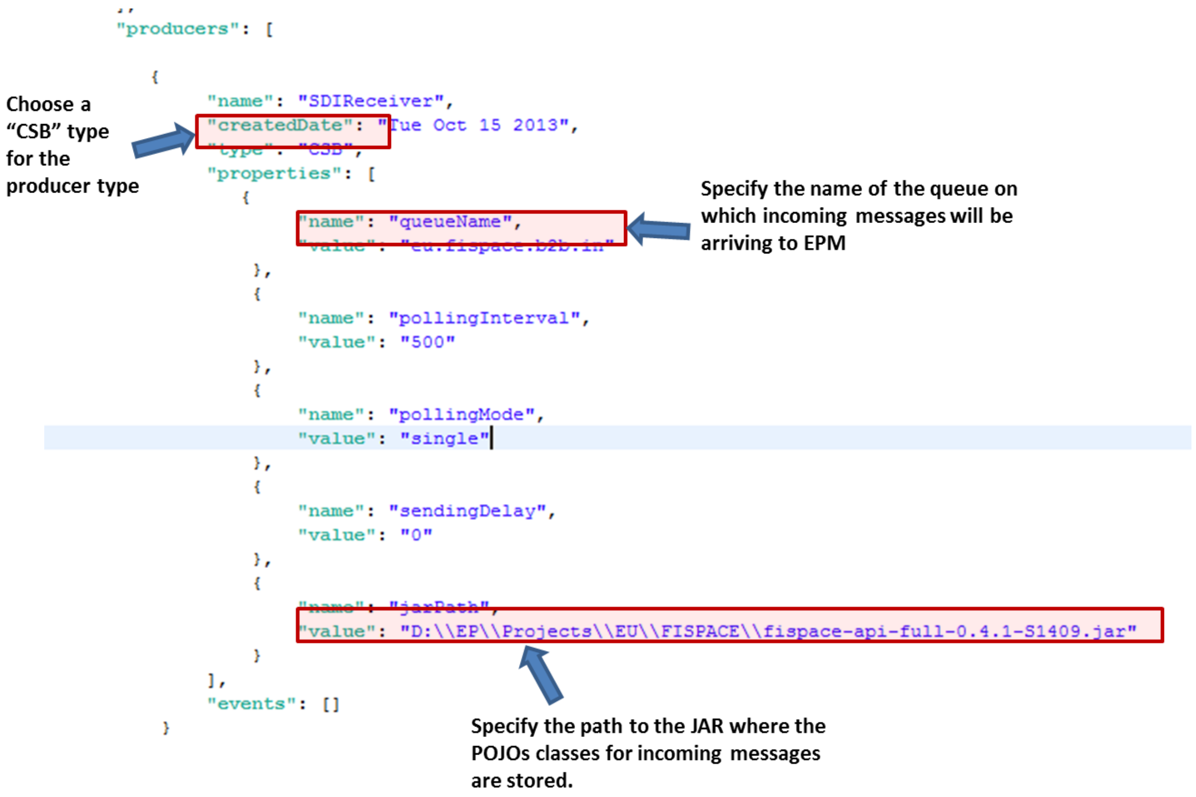
- New operators: TREND (increasing and decreasing functions)
- Templates support
- New adapters for the CSB

2.3 Interfaces / API

The B2B interfaces to other components via asynchronous publish-subscribe mechanism provided by CSB infrastructure.

The EPM receives messages representing input events via CSB queue configured in EPM definition file in CSB producer. It outputs messages representing derived events via CSB output queue, defined in EPM definition file in CSB consumer.

Figure 1 shows an example of CSB producer



```
        ,
        "producers": [

            {
                "name": "SDIReceiver",
                "createdDate": "Tue Oct 15 2013",
                "type": "CSB",
                "properties": [
                    {
                        "name": "queueName",
                        "value": "eu.fispace.b2b.in"
                    },
                    {
                        "name": "pollingInterval",
                        "value": "500"
                    },
                    {
                        "name": "pollingMode",
                        "value": "single"
                    },
                    {
                        "name": "sendingDelay",
                        "value": "0"
                    },
                    {
                        "name": "jarPath",
                        "value": "D:\\EP\\Projects\\EU\\FISPACE\\fispace-api-full-0.4.1-S1409.jar"
                    }
                ],
                "events": []
            }
```

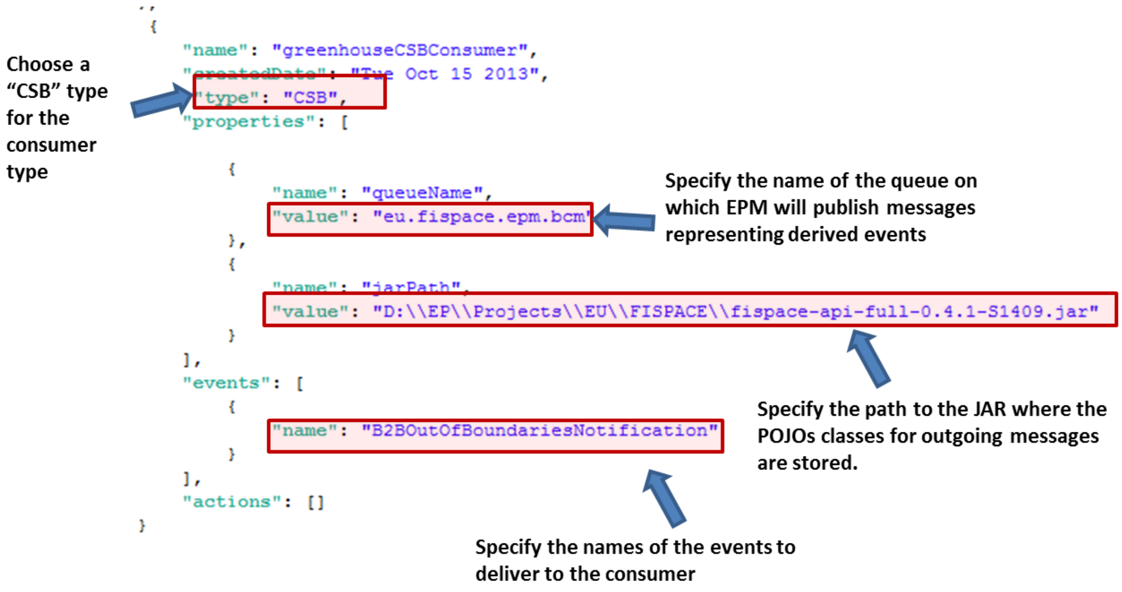
Choose a “CSB” type for the producer type

Specify the name of the queue on which incoming messages will be arriving to EPM

Specify the path to the JAR where the POJOs classes for incoming messages are stored.

Figure 1: Example of CSB producer

Figure 2 shows an example of CSB consumer:



```
    ,
     {
        "name": "greenhouseCSBConsumer",
        "createdDate": "Tue Oct 15 2013",
        "type": "CSB",
        "properties": [

            {
                "name": "queueName",
                "value": "eu.fispace.epm.bcm"
            },
            {
                "name": "jarPath",
                "value": "D:\\EP\\Projects\\EU\\FISPACE\\fispace-api-full-0.4.1-S1409.jar"
            }
        ],
        "events": [
            {
                "name": "B2BOutOfBoundariesNotification"
            }
        ],
        "actions": []
    }
```

Choose a “CSB” type for the consumer type

Specify the name of the queue on which EPM will publish messages representing derived events

Specify the path to the JAR where the POJOs classes for outgoing messages are stored.

Specify the names of the events to deliver to the consumer

Figure 2: Example of CSB consumer

The incoming and outgoing messages representing EPM event types should match EPM definitions: the name of the object should match the event types names, and the structure of the object – its member fields – should match the attribute names of the event type.

The path to the jar with the message objects class files is declared in CSB consumer/producer.

There are no additional synchronous APIs for the EPM component.

The BCM component is accessed through the application’s CSB-BCM bridge. It listens on 2 CSB queues, one for messages from other components (eu.fispace.b2b.in), and one from the EPM (eu.fispace.epm.bcm). BCM can call services defined in the bridge which passes on the messages to the eu.fispace.b2b.out CSB queue. The bridge contains the configuration of these queues, as well as how to transform Java objects to and from XML for communications with BCM. Synchronous APIs (request/response) are detailed in the next section (API operations).

2.3.1 API operations

Request/response operations are made possible through the CSB request/response mechanism. These operations may take place by sending a request to the configured CSB-BCM bridge component name; different applications have different component names, enabling communicating with specific instances. The request object should contain the identifier for the business entity as defined in ACSI for the given application. These requests are then transformed into a request for the business entity from ACSI; which is then sent back as the response object. For example, in the agriculture domain, this would be the request for the AdviceEntityInfo matching the farm id and crop id. It is also possible to define request/response operations where the request originates in ACSI and is sent to a configured component name.

To enable this, the application’s CSB-BCM bridge should be configured with one synchronous service per target component name; this is configured through the bcmSyncServiceHandlers map in the application-config.xml file – the key name gives the service name, and the value is a com.ibm.cbbridge.impl.b2c.SendRequestToCSBXmlHandler bean.

For example, in the section below one can see a test configuration for defining a synchronous service called “loopback” which forwards a request for AdviceEntityInfo to the CSB component eu.fispace.bcm.greenhouse – this will result in the bridge then receiving that request back (as it listens to CSB requests for that component name) and forwarding the request to ACSI and returning back the response:

```
<!-- bcmSyncServiceHandlers: Map<String, IRequestXmlHandler>; string is the serviceId, used as
part of the service path http(s)://<server>:<port>/<webApp>/callSyncService/{serviceId} -->
<!--    IRequestXmlHandler instances will usually be SendRequestToCSBXmlHandler -->
<util:map id="bcmSyncServiceHandlers">
       !-- Use the following for testing the bcm->csb request/response mechanism, this loops
back through the csb->bcm request/response and can be tested with a POSTer -->
    <entry key="loopback">
        <bean class="com.ibm.cbbridge.impl.b2c.SendRequestToCSBXmlHandler">
             <property name="componentName" value =
             "eu.fispace.bcm.greenhouse.loopback"/>
              <property name="respondingComponentName" value =
             "eu.fispace.bcm.greenhouse"/>
             <property name="responseMarshaller">
                   <bean class="org.springframework.oxm.castor.CastorMarshaller">
                   <property name="mappingLocation" value="/WEB-
                   INF/app/B2BAdviceEntityInfo_Mapping.xml" />
                   </bean>
             </property>
             <property name="requestUnmarshaller">
                   <bean class="org.springframework.oxm.castor.CastorMarshaller">
```

```
                    <property name="mappingLocation" value="/WEB-
                    INF/app/B2BServiceRequestAdviceEntityInfo_Mapping.xml" />
                    </bean>
             </property>
        </bean>
    </entry>
</util:map>
```

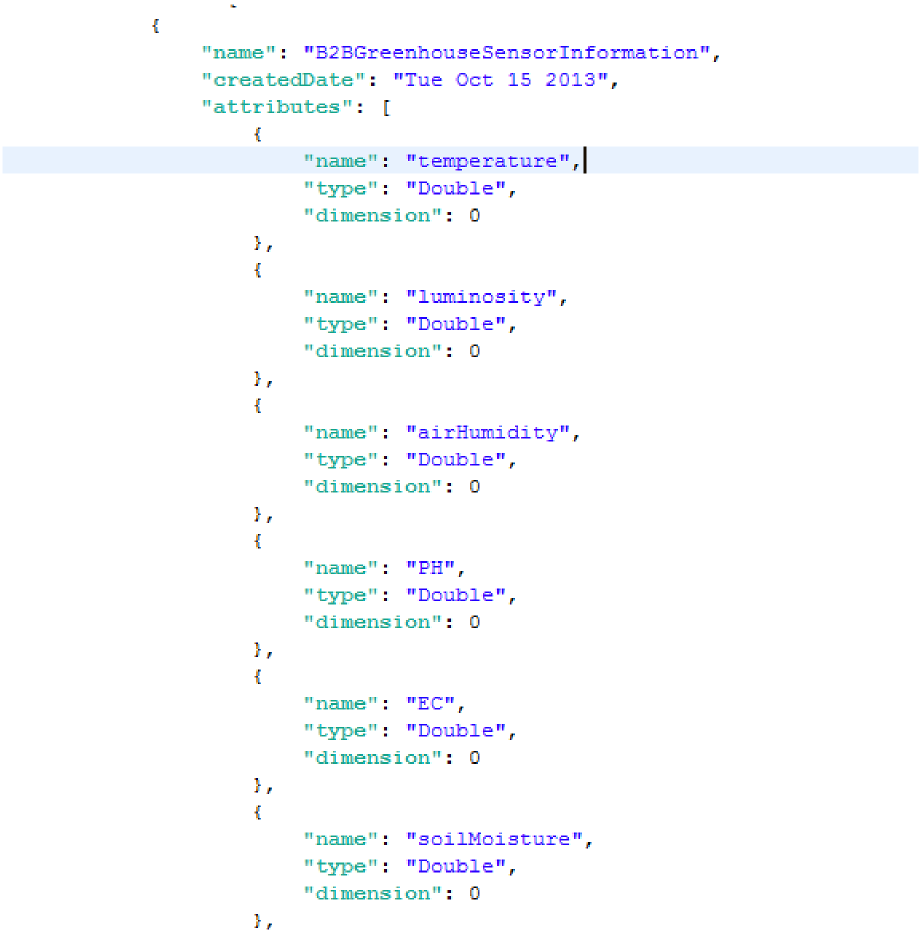
These services are configured inside ACSI as external services, with the appropriate data-type specified to receive the output message.

2.4 Information model

EPM's information model is based on the notion of event types. Event type represent an event message either incoming to the system, or a derived event outgoing from the system.

Event type is represented by event name, and a set of attribute fields, where each field is either a primitive or array of primitives.

Figure 3 shows an example of event type definition in JSON: the name of the event type is "B2BGreenhouseSensorInformation" and it has attributes like "temperature", "luminosity" etc.



```
{
    "name": "B2BGreenhouseSensorInformation",
    "createdDate": "Tue Oct 15 2013",
    "attributes": [
        {
            "name": "temperature",
            "type": "Double",
            "dimension": 0
        },
        {
            "name": "luminosity",
            "type": "Double",
            "dimension": 0
        },
        {
            "name": "airHumidity",
            "type": "Double",
            "dimension": 0
        },
        {
            "name": "PH",
            "type": "Double",
            "dimension": 0
        },
        {
            "name": "EC",
            "type": "Double",
            "dimension": 0
        },
        {
            "name": "soilMoisture",
            "type": "Double",
            "dimension": 0
        },
```

Figure 3: Example of event type definition

The BCM information model consists of data types (which are stored in XSDs), events and artifacts (note: in the ACSI Editor, the term "Business entity" is used in place of artifact). The data types define all the simple or complex types. Events have a name and are represented by incoming and outgoing messages. Artifacts define a hierarchy of instances. The information model can be edited by using the ACSI Editor.

2.5 Interaction model

Interaction model between B2B and other core components are based on the loosely-coupled publish-subscribe mechanism provided by CSB infrastructure.

The core components generate messages, delivered via CSB topics to EPM and BCM components subscribing to those topics. The messages trigger internal processing with-in those components, as a result of which other messages are generated and delivered via CSB infrastructure back to the core components.

2.6 High level composite architecture

Figure 4 shows the high level composite architecture for EPM.

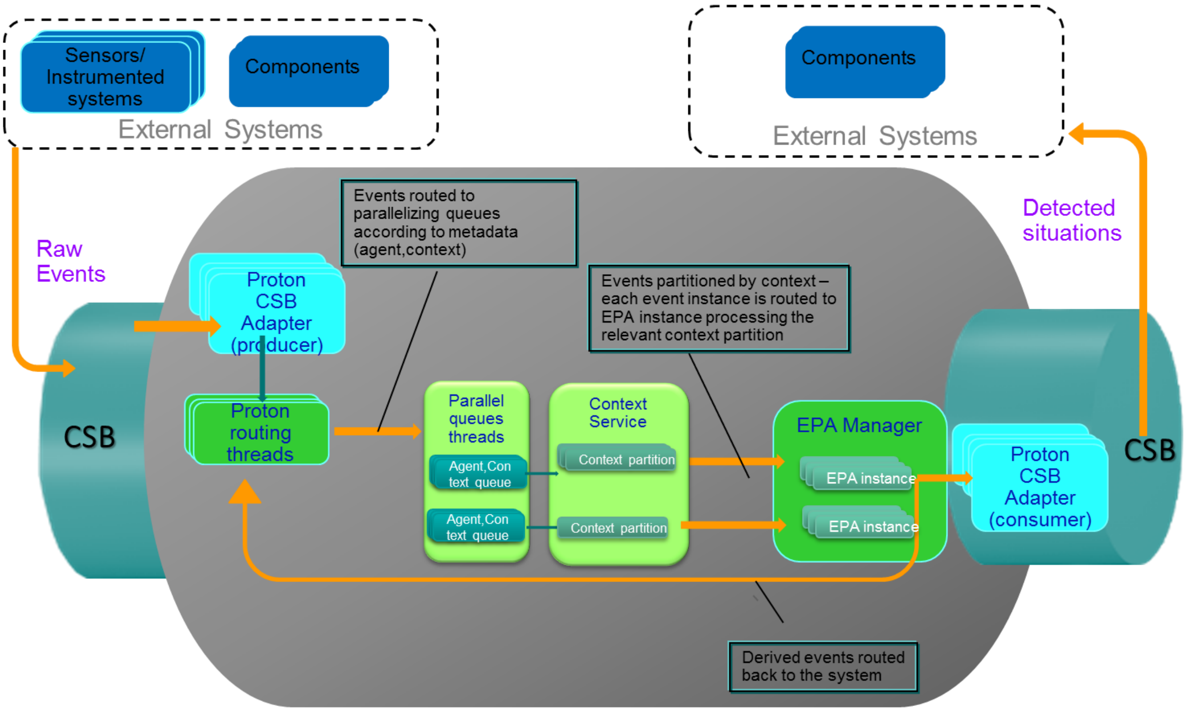

Sensors/ Instrumented systems
Components
External Systems
Components
External Systems
Raw Events
Detected situations
CSB
CSB
Proton CSB Adapter (producer)
Proton routing threads
Events routed to parallelizing queues according to metadata (agent,context)
Parallel queues threads
Agent,Context queue
Agent,Context queue
Context Service
Context partition
Context partition
Events partitioned by context – each event instance is routed to EPA instance processing the relevant context partition
EPA Manager
EPA instance
EPA instance
Proton CSB Adapter (consumer)
Derived events routed back to the system

Figure 4: High level composite architecture for EPM

Figure 5 shows the high level composite architecture for BCM.

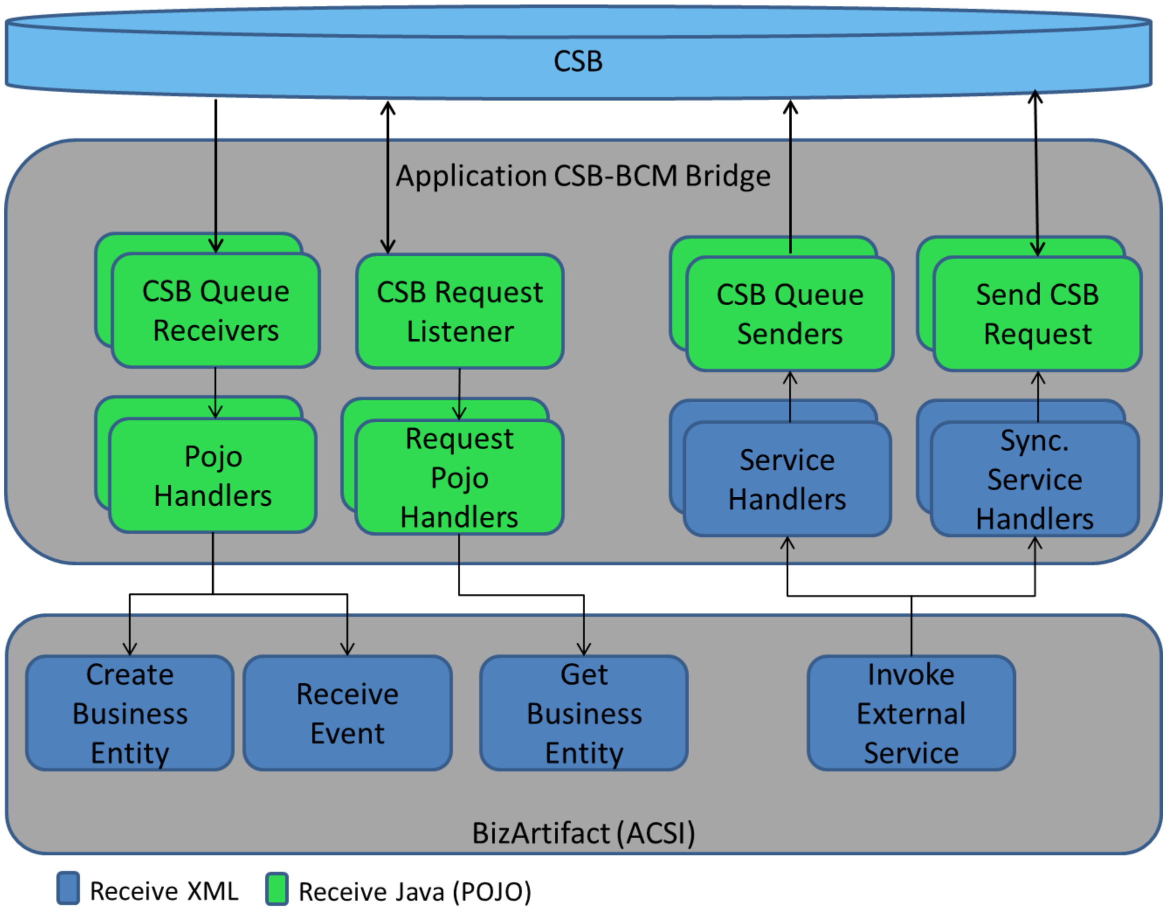


Figure 5: High level composite architecture for BCM

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)] [32]

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 architecture 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) [33]): “*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]	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.
[33]	Jacobson, I., Bylund, S., Jonsson, P., and Ehneboom, S. (1995), “Modeling 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