

Deliverable D200.7

B2B Core Modules – The manual for business architects

WP 200

| | |
|----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| Project Acronym & Number: | Flspace – 604 123 |
| Project Title: | Flspace: 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: | Flspace-D200.7- B2B_Core_Modules_documentation_for_business_architects v0.4.docx |
| Date: | 27.02.2015 |
| Revision: | 004 |
| Project website address: | http://www.Flspace.eu |

¹ Contributors to Flspace code (“P”) include ATB, UDE, IBM, ATOS, KOC, TOG, AST, NKUA, UPM and LimeTri; contributing persons are listed at <https://bitbucket.org/flspace/profile/members>

The Flspace Project

Leveraging on outcomes of two complementary Phase 1 use case projects (Flnest & SmartAgriFood), aim of Flspace is to pioneer towards fundamental changes on how collaborative business networks will work in future. Flspace will develop a multi-domain Business Collaboration Space (short: Flspace) 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, Flspace 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. Flspace 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)
 Bert Vermeer (deputy coordinator)
 Project Website

e-mail: sundmaeker@atb-bremen.de
 e-mail: bert.vermeer@wur.nl
 Web link: <http://www.flspace.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 Core Modules – The manual for business architects | 09.12.2014 |
| 002 | Update of the contents, update BCM Templates section, How-To sections, minor changes related to rephrasing | 20.01.2015 |
| 003 | Internal review process, checking URL links to the Flspace 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 “ <i>More Information</i> ” Added Flspace 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 | IDE | Integrated Development Environment |
| ACSI | Artifact-Centric Service Interoperation | IDM | Identity Management |
| AdvB | Advisory Board | i.e. | id est = that is to say |
| AJAX | Asynchronous JavaScript + XML | IE | Integration Environment |
| API | Application Programming Interface | IEC | International Electrotechnical Commission |
| App | Software Application | IETF | Internet Engineering Task Force |
| B2B | Business-to-business | I/O | Input / Output |
| B2C | Business-to-Consumer | IoT | Internet of Things |
| BCM | Business Collaboration Module in Flspace | IP | Intellectual Property |
| BCO | Business Collaboration Objects in Flspace | IP (protocol) | Internet Protocol |
| BE | Business Entities | IPR | Intellectual Property Rights |
| BPPC | Business Process Participant Configuration | IPsec | Internet Protocol Security |
| BSS | Business Support Systems | IT | Information Technology |
| CDR | Charging Detailed Records | ITU | International Telecommunication Union |
| CEP | Complex Event Processing | ISO | International Standardization Organisation |
| CSB | Cloud Service Bus | J2SE | Java 2 Platform, Standard Edition |
| CSS | Cascading Style Sheets | JDK | Java Development Kit |
| CSV | Comma-Separated Values | JDT | Related to Eclipse Java Development Tools |
| D | Deliverable | JMX | Java Management Extensions |
| DAO | Data Access Object | JRE | Java Runtime Environment |
| DB | Database | JS | JavaScript |
| DoW | Description of Work | JSON | JavaScript Object Notation |
| EC | European Commission | JSP | Java Server Page |
| EDI | Electronic Data Interchange | JVM | Java Virtual Machine |
| EE | Experimentation Environment | KPI | Key Performance Indicator |
| e.g. | Exempli gratia = for example | LPA | Logistics Planning Application |
| EPA | Event Processing Agent | M | Month |
| EPM | Event Processing Module in Flspace | MTBF | Mean Time Between Failures |
| ESB | Enterprise Service Bus | MVC | Model–View–Controller |
| EU | European Union | OASIS | Organization for the Advancement of Structured Information Standards |
| FIA | Future Internet Assembly | OAuth | Open standard Authentication protocol |
| FI-PPP | Future Internet Public Private Partnership | OMG | Object Management Group |
| FP7 | Framework Programme 7 | OSS | Operational Support Systems |
| GA | Grant Agreement | P2P | Peer-to-peer |
| GE | Generic Enabler | PaaS | Platform as a Service |
| GUI | Graphical User Interface | PDE | Related to Eclipse Java Development Tools |
| HTML | HyperText Markup Language | PE | Production Environment |
| IaaS | Infrastructure as a Service | PIA | Product Information App |
| ICT | Information and Communication Technology | | |

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

Table of Contents

1 Introduction 8

1.1 Scope 8

1.2 Intended audience 10

1.3 General remark 11

2 B2B Core Modules – The manual for business architects 12

2.1 Overview 12

2.2 Getting Started with BCM 13

 2.2.1 Related papers 13

 2.2.2 Before start using the BCM authoring tool 13

 2.2.3 Access to the BCM authoring tool from the SDK 15

 2.2.4 BCM templates 16

 2.2.5 Creating a new application 17

 2.2.6 How to integrate BCM with other components in the platform? 18

 2.2.7 How to package and deploy a new application? 21

2.3 Getting Started with EPM 21

 2.3.1 Related papers 21

 2.3.2 Before start using the EPM authoring tool 21

 2.3.3 Access to the EPM authoring tool from the SDK 24

 2.3.4 EPM templates 24

 2.3.5 Creating a new application 30

 2.3.6 How to integrate EPM with other components in the platform? 31

 2.3.7 How to package and deploy a new application? 34

3 Glossary 36

3.1 Terms and definitions 36

4 References 43

List of Figures

Figure 1: BCM – GSM model.....14
 Figure 2: Greenhouse information model14
 Figure 3: BCM – Greenhouse sequence diagram15
 Figure 4: BCM access from Flspace SDK16
 Figure 5: Example of a BCM Template16
 Figure 6: EPM – Event definition for Greenhouse scenario22
 Figure 7: EPM – Rules definition for Greenhouse scenario.....23
 Figure 8: EPM – Greenhouse sequence diagram23
 Figure 9: EPM access from Flspace SDK24
 Figure 10: EPM Authoring Tools – New Menu25
 Figure 11: EPM Authoring Tools – Created template.....26
 Figure 12: EPM Authoring Tools – Type selection.....26
 Figure 13: EPM Authoring Tools – Template Menu29
 Figure 14: EPM Authoring Tools – Choose template name29
 Figure 15: EPM Authoring Tools – Choose template type30
 Figure 16: EPM Authoring Tools – Fill template values.....30
 Figure 17: EPM Authoring Tools – Save / Export the template.....30
 Figure 18: XSD schema and automatic JAXB-based generation.....32
 Figure 19: Example of CSB producer33
 Figure 20: Example of CSB consumer34
 Figure 21: Proton.properties and port configuration.....35

List of Tables

Table 1: Other Flspace and FIWARE resources.....9
 Table 2: Wirecloud online documentation.....9
 Table 3: Store online documentation10
 Table 4: External development tools references10
 Table 5: Bitbucket collaborative environment for Flspace development.....10

1 Introduction

The aim of this document is to describe the “*B2B Core Modules – The manual for business architects*” presented as a user guideline related to the business collaboration activities. It provides an overview about the B2B core module, describes the steps to “getting started” with the BCM and EPM modules and the corresponding tools.

Online documentation for B2B Core Modules – The manual for business architects: <http://dev.fispace.eu/doc/wiki/b2b>

1.1 Scope

Business Architects are the experts (internal or external to the User organization of the supply chain actor) that are in charge of configuring FIspace for their individual business needs. Particularly they define customized collaborative workflows and connect those workflows with FIspace Apps and backend systems.

The functionality to be provided to users depends on the availability of defined and configured business processes or workflows which represent the business information needs and in which these workflows are connected to the appropriate Apps, backend systems, and IoT devices. Business architects, are in charge of developing and configuring these workflows in FIspace. In particular they will define customized collaborative workflows and connect those workflows with FIspace Apps and backend systems.

The role of Business Architects is to ensure reusability along a number of lines. A farmer for instance may use almost the same business process collaboration for the same crop over time and for different crops and resources without further assistance provided that the instantiation mechanisms allow the same business process model (BPM) to be instantiated for these differing situations. Similarly the same BPM can be instantiated by other farmers who enter into similar co-operations for their own crops.

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 FIWARE.

| 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 Flspace 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 |

| | |
|------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | p/Store - W-Store - User and Programmer Guide |
| FIWARE - Store - W-Store - Store - W-Store - Installation and Administration Guide | https://forge.fiware.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 Flspace 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 Flspace repository home page | https://bitbucket.org/flspace |
| Bitbucket Flspace core component home page | https://bitbucket.org/flspace/core/wiki/Home |
| Bitbucket Flspace Roadmap page | https://bitbucket.org/flspace/core/wiki/roadmap |

Table 5: Bitbucket collaborative environment for Flspace development

1.2 Intended audience

The main interest groups of this deliverable are the participating teams and the responsible partners of Flspace 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 Flspace platform and the underlying components and sub-systems who want to know more about some technical information intrinsic to the Flspace 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 Flspace 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 Flspace 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 Core Modules – The manual for business architects

2.1 Overview

At the heart of the envisaged Flspace 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 Flspace 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](#)). 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 Flspace we apply 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. 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). The EPM in Flspace 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.

These two modules are connected each to the other and to the other Flspace components of the platform via the [Component Service Bus \(CSB\)](#).

2.2 Getting Started with BCM

2.2.1 Related papers

This is a new way of looking at and implementing business processes. So, first step is to get familiar with the essentials of this paradigm. Find below a short list of relevant papers that you can find useful:

- Richard Hull: **Data-Centricity and Services Interoperation**. ICSOC 2013: 1-8
- Fenno (Terry) Heath, David Boaz, Manmohan Gupta, Roman Vaculín, Yutian Sun, Richard Hull, Lior Limonad: **Barcelona: A Design and Runtime Environment for Declarative Artifact-Centric BPM**. ICSOC 2013: 705-709
- Richard Hull: **Towards Flexible Service Interoperation Using Business Artifacts**. EDOC 2011: 20-21
- Richard Hull, Elio Damaggio, Riccardo De Masellis, Fabiana Fournier, Manmohan Gupta, Fenno Terry Heath, Stacy Hobson, Mark H. Linehan, Sridhar Maradugu, Anil Nigam, Piyawadee Noi Sukaviriya, Roman Vaculín: **Business artifacts with guard-stage-milestone lifecycles: managing artifact interactions with conditions and events**. DEBS 2011: 51-62
- Richard Hull, Elio Damaggio, Fabiana Fournier, Manmohan Gupta, Fenno Terry Heath, Stacy Hobson, Mark H. Linehan, Sridhar Maradugu, Anil Nigam, Piyawadee Sukaviriya, Roman Vaculín: **Introducing the Guard-Stage-Milestone Approach for Specifying Business Entity Lifecycles**. WS-FM 2010: 1-24
- First article ever on business artifacts: Anil Nigam, Nathan S. Caswell: **Business artifacts: An approach to operational specification**. IBM Systems Journal 42(3): 428-445 (2003)

2.2.2 Before start using the BCM authoring tool

Before you start working with the authoring tool of the BCM, we encourage you to:

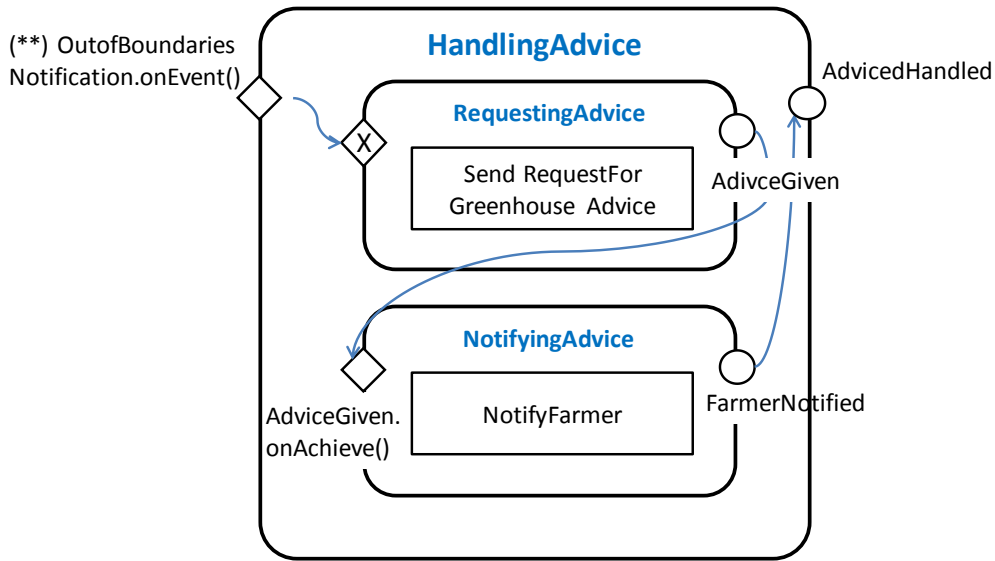
1. Draw the GSM model in PowerPoint or any other graphical tool (pencil and paper are also valid).

Example from the Greenhouse scenario:

Two main business actors are involved in this scenario: the farmer/greenhouse manager and an advisory/expert system enterprise, which provides advisory services to the greenhouse based on the conditions inside the greenhouse. The idea is that the sensors' values of the Greenhouse are forwarded to the Greenhouse Farm Management System (FMIS), where they are contextualized, and afterwards they are forwarded to Flspace. In case of events inside the greenhouse (i.e. sensor values detected out of pre-defined boundaries) a request for actions is sent to the advisory system. The end-user receives the actions from the expert system via the respective Flspace app.

Business Entity (BE): GreenhouseAdvice

GSM model (*):



- (*) SendRequestForGreenhouseAdvice produces the RequestForGreenhouse message
- AdviceGiven is achieved by GreenhouseAdvice.onEvent()
- NotifyFarmer is achieved by NotifyFarmer.onTaskSuccess()
- AdvisedHandled is achieved by FarmerNotified
- (**) emitted by the EPM

Figure 1: BCM – GSM model

Greenhouse Information model:

| Attribute | Type |
|---------------------|-----------------|
| ID | Long |
| farmID | String |
| cropFieldID | String |
| greenhouseAdviceURL | String |
| sensorValues | DT_sensorValues |

| Attribute | Type |
|--------------|--------|
| temperature | Double |
| luminosity | Double |
| airHumidity | Double |
| ph | Double |
| ec | Double |
| co2 | Double |
| soilMoisture | Double |

Figure 2: Greenhouse information model

2. Draw the sequence diagram of the messages to and from the BCM. Example from the Greenhouse scenario (Note that for the sake of simplicity the Cloud Service Bus or CSB is not shown, but all interactions inside FIspace platform are done through the CSB):

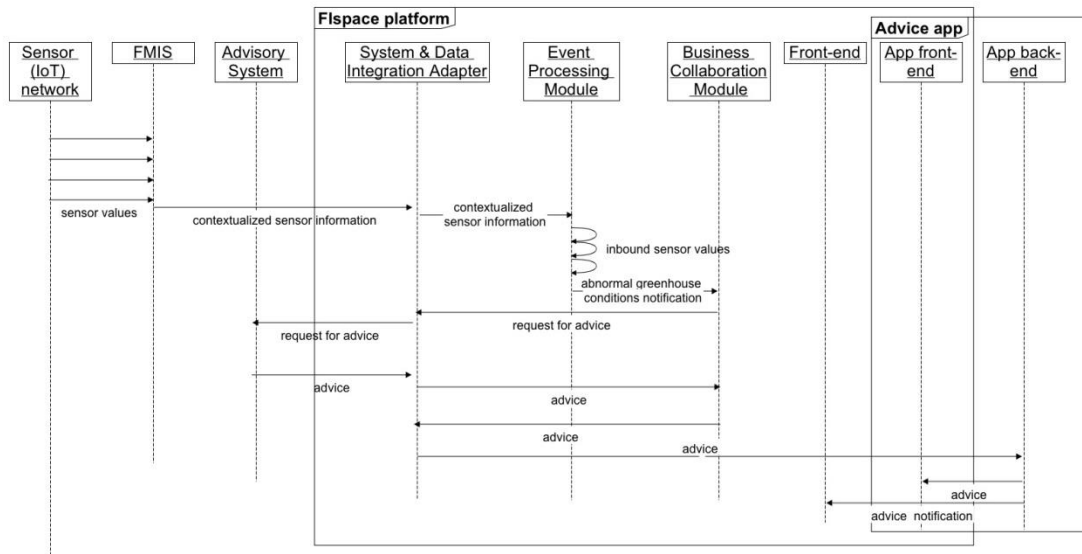


Figure 3: BCM – Greenhouse sequence diagram

3. Relevant documentation

The BCM is based on the open source BizArtifact tool (<http://sourceforge.net/projects/bizartifact/>), which has been extended to cope with Flspace requirements.

The user-manual is available here: <http://sourceforge.net/projects/bizartifact/files/Acsi-I-Hub%20Users%20Guide.pdf/download>

Information about the Artifact-Centric Service Interoperation (ACSI) Project, including publications can be found here: <http://www.acsi-project.eu/>

2.2.3 Access to the BCM authoring tool from the SDK

The design of a new application is done via the BCM authoring tool accessible from Flspace SDK (click on the EPM button, shown in a red circle in Figure 4).

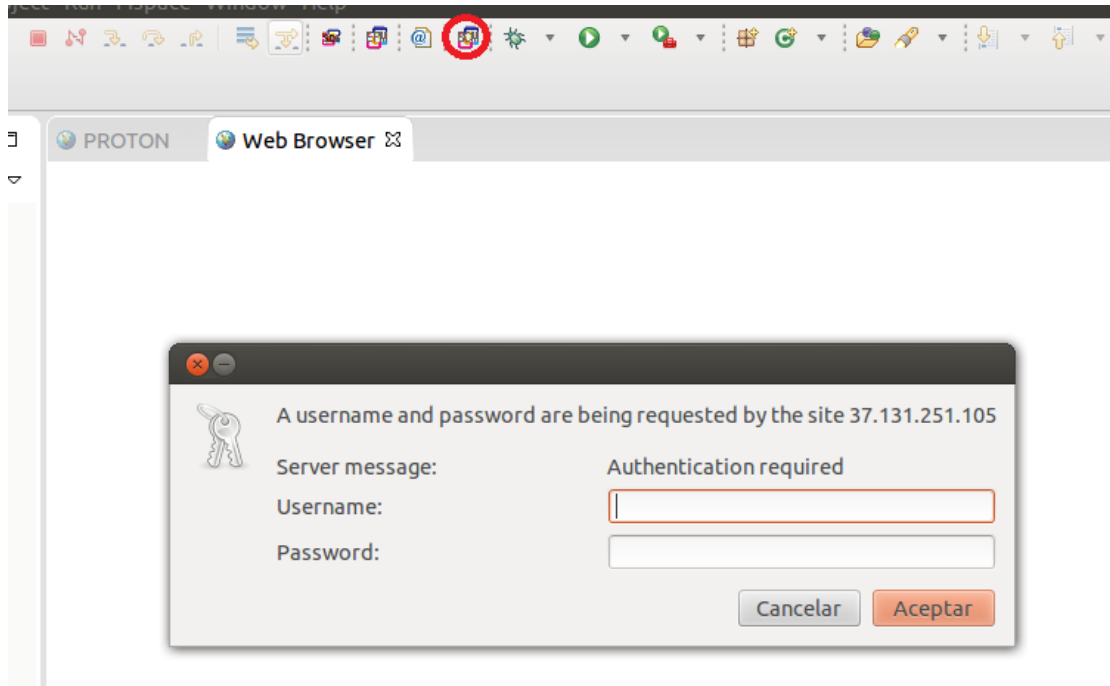


Figure 4: BCM access from Flspace SDK

2.2.4 BCM templates

To make an easier start-up with a new application, the B2B team has created a first generic small process typical of any transport and logistics scenario. The idea is to use this generic process as a template that can be overwritten and enlarged depending on the collaboration in-hand. This is included as an example in the repository.

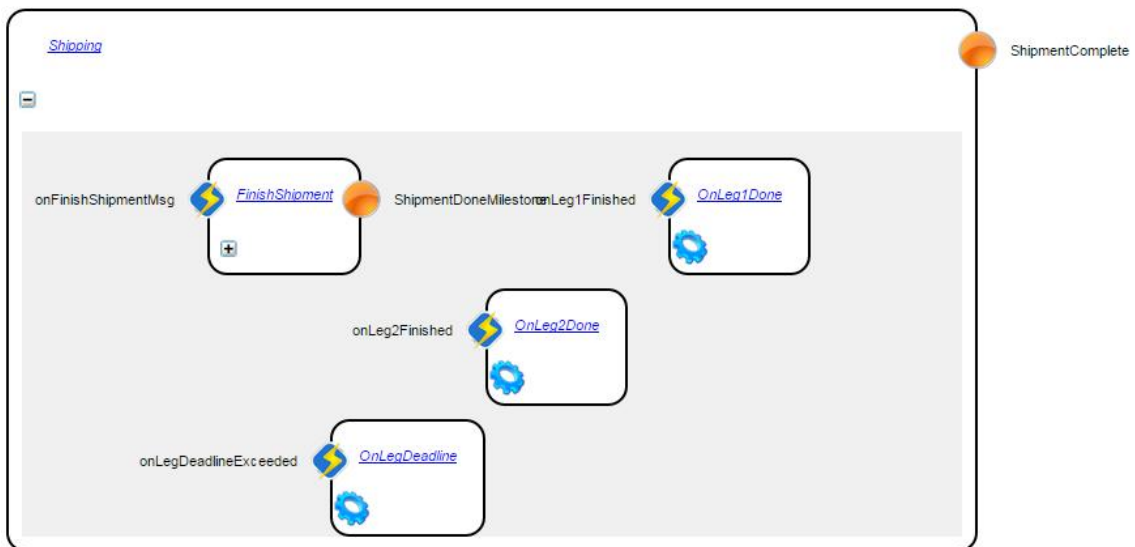


Figure 5: Example of a BCM Template

2.2.5 Creating a new application

There are two parts in creating a new application. Each application has its own CSB-BCM Bridge configuration, which is based on a common CSB-BCM Bridge base and runtime. The base configuration defines the available CSB queues, as well as base services to be called from BCM. This bridge facilitates all communications between ACSI and the other components through the CSB. This is discussed in the next section around integrating BCM with other components.

In BCM, a new application needs to be created from the ACSI Editor. To facilitate this, base ACSI applications are being made available for each domain, and include all the data and message definitions necessary for that domain. These base ACSI applications are released under the eu.fispace.b2b/BCMSchemas artefact (which is a zip file). Their basic structure looks like:

- <Domain>
 - sienna.xml
 - data
 - <Domain>_DataTypes.xsd

To create your “application” called <App>, rename the <Domain> directory and the <Domain>_DataTypes.xsd file, putting in <App> instead of <Domain>. Also, edit the sienna.xml file, where you need to replace *name*="*<Domain>*" with *name*="*<App>*" and similarly edit the <App>_DataTypes.xsd file where *id*="*<Domain>_DataTypes*" should be replaced with *id*="*<App>_DataTypes*". After this, you can zip up the entire directory structure (so that the top-level of the zip contains the <App> directory). This zip can then be imported into ACSI. A short set of instructions is provided at the end of this section for packaging and deploying applications. Some important details for configuring ACSI for your application are:

1. OCL expressions for accessing primitive “lists”; generally if the XML schema for a data type defined a primitive (non compound/complex) field, the ACSI data type will be defined with the primitive field of cardinality “n”. The expression that you should then write to access its value would be <object>.<field>->first().<field>. For example, to access the status field for an incoming event called MyEvent you would write self.MyEvent.status->first().status.
2. OCL expressions for Boolean logic; OCL does not support using parentheses to delineate evaluation precedence for Boolean expressions. Instead, Booleans returned by expressions are themselves objects, for which the operators can be called on them. Thus, in order to write the logic for “a && (b || c)” (using C-style notation) in OCL one would write “a.and(b or c)” or “a.and(b.or(c))”.
3. Correlation keys:
 - i. The event XPATH expression for accessing primitive fields should be of the form “<eventType>InputMessage/<field>[1]”
 - ii. Correlation keys must contain the “businessProcessId” field (but can include other fields as well). It is thus generally recommended to include a “businessProcessId” field of type integer and cardinality (“Max Occurs”) 1 in the information model.
4. Milestones – one should validate that all milestones are defined in the information model as booleans of cardinality (“Max Occurs”) 1. Thus, if you have a stage which has a milestone called “MyMilestone”, one should validate that

there is a boolean field with a “Max Occurs” value of 1 called “MyMilestone” in the information model.

For more details on configuring ACSI, consult the user manual.

2.2.6 How to integrate BCM with other components in the platform?

The base CSB-BCM Bridge listens on the eu.fispace.b2b.in and eu.fispace.epm.bcm queues for incoming messages. It also defines two services at rest/callService/b2bOut and rest/callService/bcm2epm which will send CSB messages out to the eu.fispace.b2b.out and eu.fispace.bcm.epm queues, respectively. In your ACSI application, to send a CSB message to the eu.fispace.b2b.out queue, define an external service with the path <http://localhost:8080/CBBridge/rest/callService/b2bOut> and method POST.

The name of the service doesn't matter, but for convenience, you could call it send<MsgType>. For the service input message, send 2 parameters. The first, should have its Attribute (parameter name) set to destCapabilityTypeId and should be of type int. When invoking the service, you will set this value to an Expression and type in the value of the capability type id that should receive this message (according to the business process template). For the second parameter, set the Attribute (the parameter name) to be exactly the same as the message type, which should be one of the domain messages. For example, define a parameter of type *ReceiveShipmentStatusRequestMessage* and called *ReceiveShipmentStatusRequestMessage*. You would define a similar external service with the <http://localhost:8080/CBBridge/rest/callService/bcm2epm> path to send to EPM.

The primary mode of integration revolves around the application's CSB-BCM Bridge configuration. This configuration deals with defining POJO handlers for dealing with incoming messages. The configuration also supports synchronous request/response mechanisms over CSB. The configuration consists of several files. The primary file which must be present is application-config.xml which is a Spring configuration file.

Configuring the handling of received CSB POJO messages:

Specifically, whether to create a new ACSI Business Entity and/or send an event to ACSI. This includes transforming the received CSB POJO messages to XML. Consider the following example:

```
<!-- Business Process Template Id -->
<!-- The configuration specified in this file will only apply to messages whose business process
is an instantiation of the following id-->
<bean id="businessProcessTemplateId" class="java.Lang.Integer">
    <constructor-arg value="2"/>
</bean>

<!-- marshaller for creating a new shipment -->
<bean id="createMarshaller" class="org.springframework.oxm.castor.CastorMarshaller">
    <property name="mappingLocation" value="CreateShipment_Mapping.xml" />
</bean>

<!-- csbQueueMessagePojoHandlerConfiguration: Map<Class, IPojoHandler>; class is the POJO's class for
which the handler should be used -->
```

```

<!-- IPojoHandler instances will usually be: ChainPojoHandler, CreateBEPojoHandler, PassEventToBCMPo-
joHandler -->
<util:map id="csbQueueMessagePojoHandlerConfiguration" key-type="java.Lang.Class">
  <entry key="eu.fispace.api.Lg.ReceiveShipmentStatusRequestMessage">
    <bean class="com.ibm.cbbridge.impl.ChainPojoHandler">
      <property name="handlers">
        <list>
          <bean class="com.ibm.cbbridge.impl.FilterPojoHandler">
            <property name="filter">
              <bean class="com.ibm.cbbridge.impl.MapPojoFilter">
                <property name="specification">
                  <map>
                    <entry key="getStatus.value">
                      <util:list value-type="java.Lang.String">
                        <value>Announced</value>
                      </util:list>
                    </entry>
                  </map>
                </property>
              </bean>
            </property>
          </bean>
          <bean class="com.ibm.cbbridge.impl.c2b.CreateBEPojoHandler">
            <property name="marshaller" ref="createMarshaller"/>
            <property name="beName" value="Shipment"/>
          </bean>
          <bean class="com.ibm.cbbridge.impl.c2b.PassEventToBCMPojoHandler">
            <property name="eventName" value="ReceiveShipmentStatusRequestMessage"/>
            <property name="marshaller" ref="jaxb2Marshaller"/>
          </bean>
        </list>
      </property>
    </bean>
  </entry>
  <entry key="eu.fispace.api.Lg.ProvideShipmentDetailsRequestMessage">
    <bean class="com.ibm.cbbridge.impl.c2b.PassEventToBCMPojoHandler">
      <property name="eventName" value="ProvideShipmentDetailsRequestMessage"/>
      <property name="marshaller" ref="jaxb2Marshaller"/>
    </bean>
  </entry>
</util:map>

```

The CreateShipping_Mapping.xml file contains:

```
<?xml version="1.0" encoding="UTF-8"?>
<mapping>
  <class name="eu.fispace.api.Lg.ReceiveShipmentStatusRequestMessage">
    <map-to xml="Shipment"/>
    <field name="shipmentId" type="string">
      <bind-xml name="shipmentId" node="attribute"/>
    </field>
    <field name="businessProcessId" type="integer">
      <bind-xml name="businessProcessId" node="attribute"/>
    </field>
  </class>
</mapping>
```

Here we can see handlers being defined for two messages. The first message `ReceiveShipmentStatusRequestMessage` may create a Business Entity in ACSI; but it should only do this when the message instance `getStatus().getValue()` has a value of "Announced". If we were to want to create a new Business Entity for other values too, we'd add them to the list. If we'd want to always create the Business Entity for every message received, we would have just used the `CreateBEPojoHandler` directly, without the `FilterPojoHandler`. Since we want to create a Business Entity whose name and data format can be configured in ACSI, we need to define a specific way to map the message to the Business Entity. To do so, we use a Castor Marshaller with a specified mapping file – in general, fields should be mapped explicitly to attributes if in the ACSI information model it is receiving a simple type and it has cardinality 1. Fields that are used as part of correlation keys should usually be included here, in particular, `businessProcessId` See: <http://castor.codehaus.org/1.2/xml-mapping.html> for more on configuring Castor. After creating the Shipment Business Entity, we want to send it as an event to ACSI; for that we use the `PassEventToBCMPojoHandler`. The marshaller for sending events should always be `jaxb2Marshaller`. We can see for the second message, `ProvideShipmentDetailsRequestMessage`, that we just want to send it as an event. To summarize:

- Use `CreateBEPojoHandler` to create a new Business Entity (and start a new lifecycle) in ACSI. Always use a Castor Marshaller that you define a mapping file for.
- Use `PassEventToBCMPojoHandler` for sending events. Always use the `jaxb2Marshaller`.
- To do both, use the `ChainPojoHandler`; always put the `CreateBEPojoHandler` first so it will create the business entity before sending the event.
- To selectively create a business entity, or selectively send an event, pass the `PojoHandler` as a delegate to a `FilterPojoHandler`. Set the filter to be a `MapPojoFilter`, which takes a map; a key specifies method calls to be made on the message (a getter chain). The value is a list of values which are acceptable. If the message returns a value which is not in the list, then the delegate handler will not be called. Put `<null/>` in the list if you want to accept null. If you have several keys in

the map, then all of them must be accepted or the delegate handler will not be called.

- Expose Business Entity information by implementing CSB Request/Response (with a configurable component name) and querying ACSI for the information.
- Note as well that you need to configure the `businessProcessTemplateId` field; this will ensure that only messages corresponding to that template are handled

As can be readily understood, the other components communicate by sending POJOs through CSB queues and through the CSB Request/Response mechanism.

2.2.7 How to package and deploy a new application?

In order to package a new application, a new zip file should be prepared for the application's bridge configuration. This must consist of an `applicaton-config.xml` in the top-level of the zip. You can include additional files or directories in the zip. Note that all paths listed in your configurations should be relative to the top-level in the zip. The configuration can then be uploaded to the bridge at "`http://<path to bridge>/CBBridge`" – you can delete, or update your configurations here (Note – this is only temporary; work is being done on integrating the management of these configurations within the SDK). You must give a name for the configuration when uploading. In ACSI, an application can be packaged by exporting it to a zip file from the ACSI Editor. This is done by right-clicking on the application and selecting Download. An application can then later be imported into the ACSI Editor. One deploys the application by right-clicking on it in the editor and selecting Run or one of the Redeploy options. It is recommended to see the user's manual for more.

2.3 Getting Started with EPM

2.3.1 Related papers

The EPM terminology and semantics is based on, and follows, the event processing language definition as presented by Etzion O., Niblett P. in **Event Processing in Action**, Manning Publications, 2010.

Another relevant paper that you can find useful:

Zohar Feldman, Fabiana Fournier, Rod Franklin, Andreas Metzger: Proactive event processing in action: **a case study on the proactive management of transport processes** (industry article). DEBS 2013: 97-106

2.3.2 Before start using the EPM authoring tool

Before you start working with the authoring tool of the EPM, we encourage you to:

1. Draw the event definitions and rules in PowerPoint or any other graphical tool (pencil and paper are also valid).

Example from the Greenhouse scenario:

Two main business actors are involved in this scenario: the farmer/greenhouse manager and an advisory/expert system enterprise, which provides advisory services to the greenhouse based on the conditions inside the greenhouse. The idea is that the sensors' values of the Greenhouse are forwarded to the Greenhouse Farm Management System (FMIS), where they are contextualized, and afterwards they are forwarded to

Flspace. In case of events inside the greenhouse (i.e. sensor values detected out of pre-defined boundaries) a request for actions is sent to the advisory system. The end-user receives the actions from the expert system via the respective Flspace app.

Sensor events are received in the EPM that checks whether all the sensor values are in the permitted range. It emits a notification alert whenever **At least one of the sensor values is not in the permitted range and there is no pending Advice for the specific (farmID and cropID)**

Allowed sensor values:

- $10 \leq \text{temperature} \leq 33$
- $5 \leq \text{luminosity} \leq 40$
- $50 \leq \text{airHumidity} \leq 80$
- $5 \leq \text{PH} \leq 7$
- $1.5 \leq \text{EC} \leq 3.5$
- $60 \leq \text{soilMoisture} \leq 90$
- $200 \leq \text{CO}_2 \leq 1000$

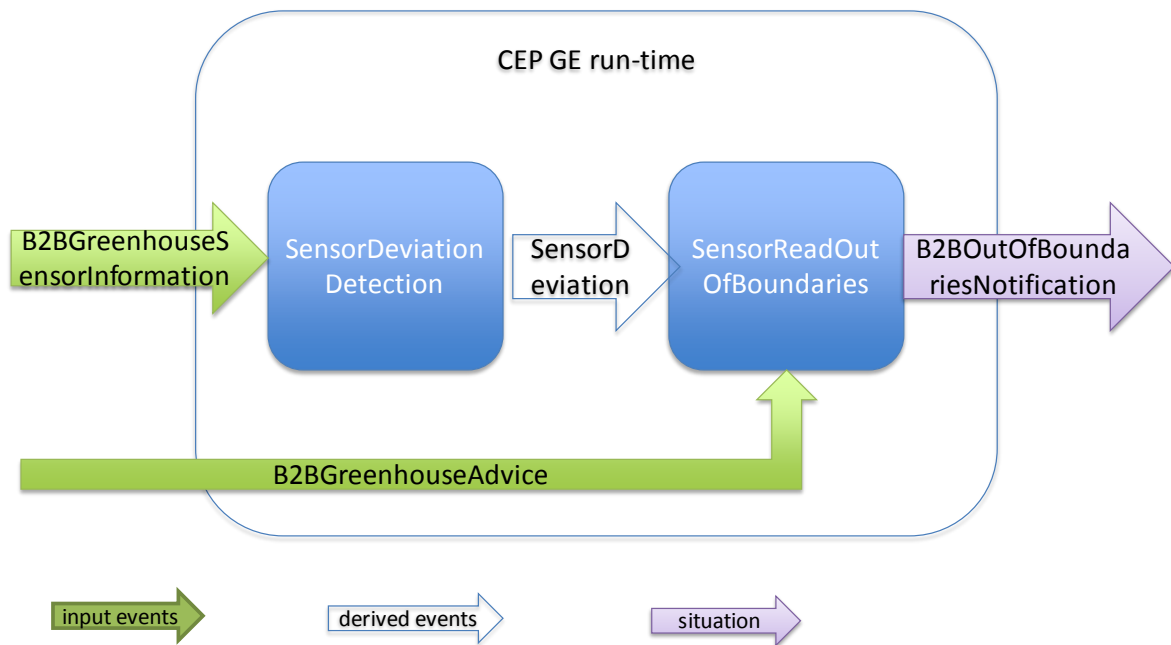


Figure 6: EPM – Event definition for Greenhouse scenario

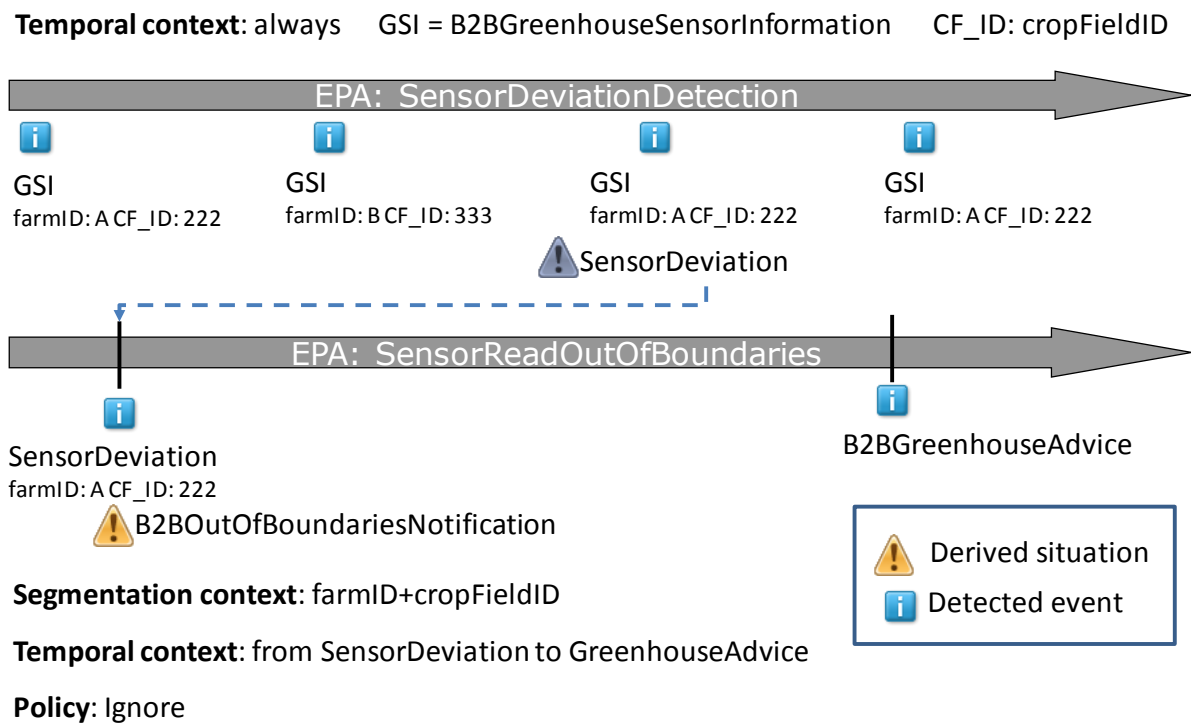


Figure 7: EPM – Rules definition for Greenhouse scenario

2. Draw the sequence diagram of the messages to and from the EPM. Example from the Greenhouse scenario (Note that for the sake of simplicity the Cloud Service Bus or CSB is not shown, but all interactions inside the FIspace platform are done through the CSB):

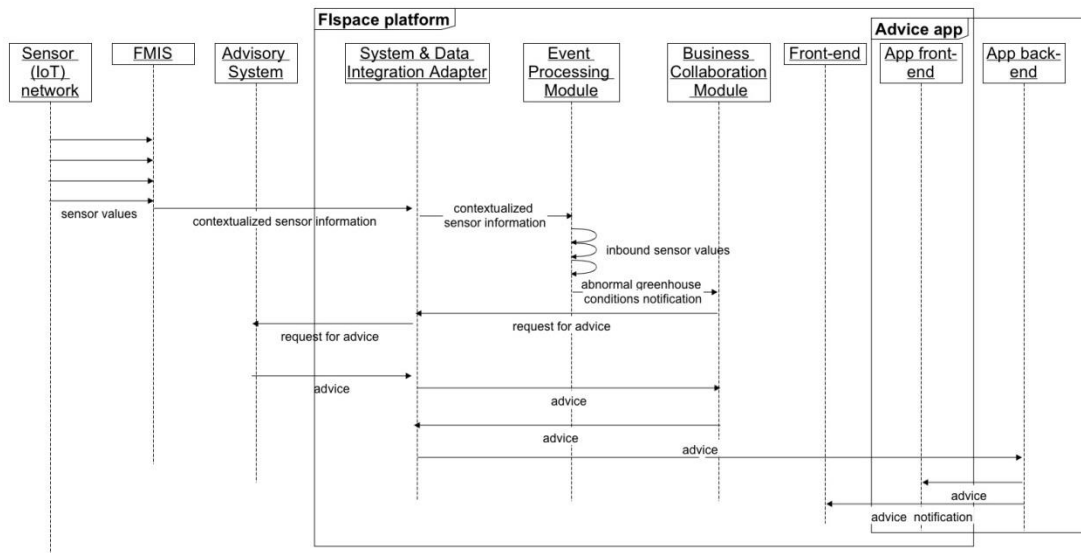


Figure 8: EPM – Greenhouse sequence diagram

3. Relevant documentation

The EPM is based on the open source IBM Proactive Technology Online (Proton) tool <https://github.com/ishkin/Proton>, which has been extended to cope with Flspace requirements.

Designing and authoring EPM applications – tutorials and documentation:

- CEP recorded webinar and tutorial – <http://edu.fi-ware.eu/course/view.php?id=58>
- CEP user and programmers guide – http://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/CEP_-_User_and_Programmer_Guide

Installing, configuring and managing EPM, EPM architectural guide:

- CEP Architecture – <http://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.Data.CEP>
- CEP installation guide – https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/CEP_GE_-_IBM_Proactive_Technology_Online_Installation_and_Administration_Guide

EPM RESTful APIs:

- http://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/Complex_Event_Processing_Open_RESTful_API_Specification

2.3.3 Access to the EPM authoring tool from the SDK

The design of a new application is done via the EPM authoring tool accessible from Flspace SDK (click on the EPM button, shown in a red circle in Figure 9).

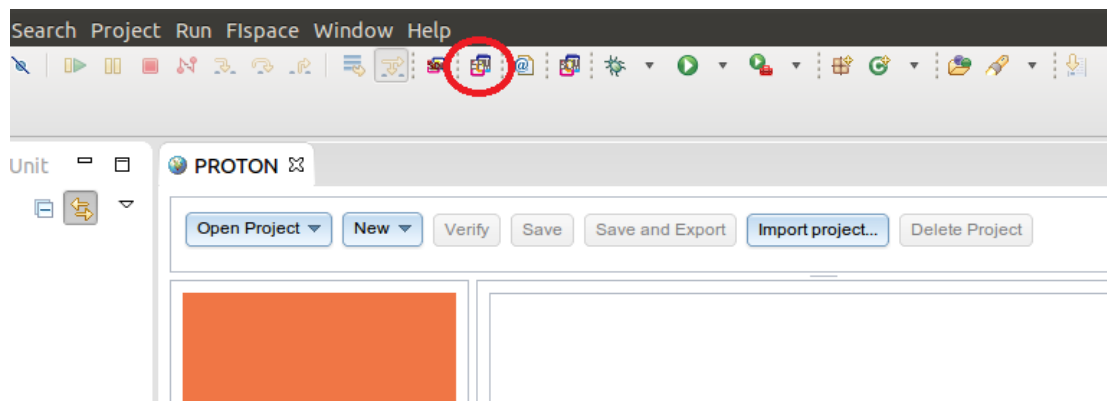


Figure 9: EPM access from Flspace SDK

2.3.4 EPM templates

To make an easier start-up with a new application, the B2B team has created a set of pre-defined event rules for the following event patterns: FILTER, COUNT, and ABSENCE; typically used in any transport and logistics scenario. The idea is to overwrite these generic templates to speed up the creation of a new application.

Templates are based on the idea that some scenarios are very common and could be easily implemented using some kind of common approach. Such scenarios include detecting an absence of event in certain time window, filtering out/in input event and creat-

ing derived event in case of filter evaluation is true, detecting a trend of some value within a time window and counting instances of events within time window.

For the purpose of speeding up the scenario definitions for such common patterns, we provide templates. Templates is based on existing EPN event model (meaning that we assume the events figuring in the template have already been defined by the time template is created), and it will create all the other relevant artifacts (epa with policies, temporal, segmentation and composite contexts).

The general idea is to add a template to the EPN definition project, choose the template type, fill in the template's parameters values according to the type, and then export the definitions into JSON. The JSON created will already include all the template artifacts for the pattern, and can be imported back to authoring tool for additions/changes.

To create a definition using templates the following steps should be followed:

- Relevant events and their attributes should be defined. For example, if you are using a **Filter** template, the relevant input and derived event of the pattern should already be defined before using a template.

The assumption is that the templates would just define part of the application, and all the event model of the application should already exist before using the templates.

What events are relevant is dependent on the template: for **Filter** template it is just the input and the output event, for other templates it is dependent on the template, please see the template description.

- In [EPM Authoring Tool](#) click on the New menu and choose the option of creating a new template:

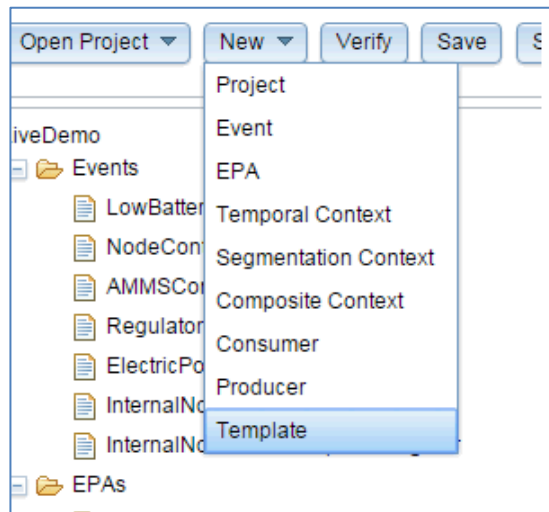


Figure 10: EPM Authoring Tools – New Menu

This will allow specifying the name for the template. The created template will be added to projects artifacts tree:

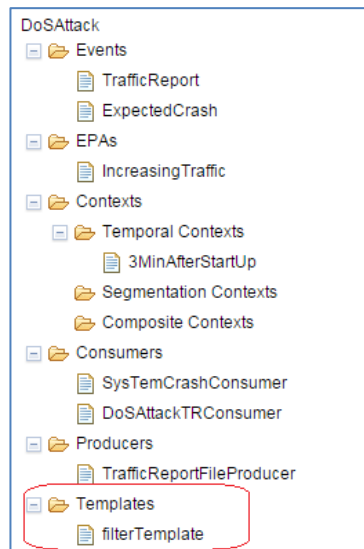


Figure 11: EPM Authoring Tools – Created template

- After the template is created, the template type should be chosen from the drop down box. According to the chosen type, different values should be filled in into template’s parameters:

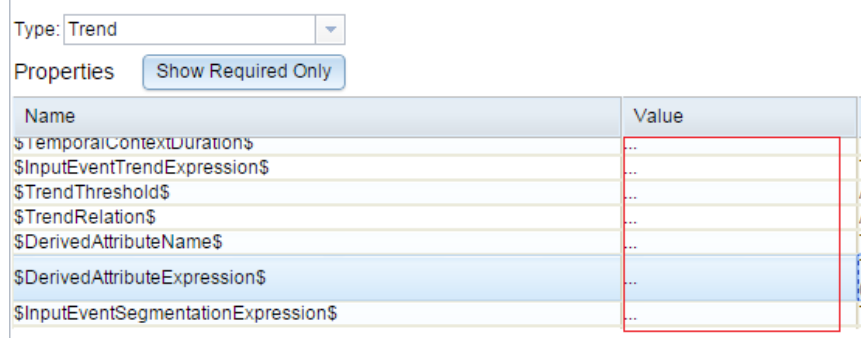


Figure 12: EPM Authoring Tools – Type selection

The following values are relevant for different templates types:

- **Filter:** the template is for filtering out events. It is based on temporal context lasting for the whole lifetime of the application (starting at startup and never ending). Filter creates derived events multiple times, every time the input event passes evaluation.
 - \$InputEvent\$ - the name of the input event which should be filtered out
 - \$FilterExpression\$ - the expression on input events attribute. Only those events for which this expression holds true will cause derivation of output event
 - \$OutputEvent\$ - the name of the derived event to create
 - \$DerivedAttributeName\$ - the name of the attribute of the derived event into which we want to assign some value, usually based on input event attributes

- **\$DerivedAttributeExpression\$** - The expression for the derived attribute. Can be a constant value, or based on an attributes of input event.
- **AbsenceEventInitiator:** the template is for detecting absence of event in a certain time window, started by some initiator event and lasting for N millis. The absence is detected at the end of time window. The absence is detected for a certain segmentation context, for example absence of withdrawal following deposit for certain customer would mean segmentation on customerID.
 - **\$InputEvent\$** - the name of the input event we are monitoring the absence of
 - **\$OutputEvent\$** - The name of the event to create if such absence is detected
 - **\$InitiatorEvent\$** - the name of the event to start the temporal context during which we monitor for absence
 - **\$ContextWindowSize\$** - defines the length of temporal window during which we monitor the absence, in millis, from initiator event.
 - **\$DerivedAttributeName\$** - The name of the attribute in the derived event which is derived if absence is detected
 - **\$DerivedAttributeExpression\$** - The expression for the derived attribute. Can be a constant value, or the partition of the segmentation context.
 - **\$InputEventSegmentationAttributeExpr\$** - The segmentation expression based on the input event attributes for the segmentation context. (for example, if we monitor absence of Withdrawal for certain customer, then it is the Withdrawal.customerID attribute)
 - **\$InitiatorEventSegmentationAttributeExpr\$**-The segmentation expression based on the initiator event attributes for the segmentation context. (For example, if the context is initiated by a first deposit of a customer, than the initiator event is the first deposit, and the expression here is Deposit.customerID)
- **Count:** the template is for counting instances of event in a certain time window, started by some initiator event and lasting for N millis. The count is done within a certain segmentation context, for example counting withdrawals for a specific customer (segmentation by customer id). The count is at the end of time window.
 - **\$InputEvent\$** - the name of the input event we counting
 - **\$OutputEvent\$** - The name of the event to emit
 - **\$InitiatorEvent\$** - the name of the event to start the temporal context during which we calculate count
 - **\$TemporalContextDuration\$** - defines the length of temporal window during which count, in millis, from initiator event.

- `$DerivedAttributeName$` - The name of the attribute in the derived event which will hold the count value.
- `$InputEventSegmentationAttributeExpr$` - The segmentation expression based on the input event attributes for the segmentation context. (for example, if we count the number of Withdrawals for certain customer, then it is the `Withdrawal.customerID` attribute)
- `$InitiatorEventSegmentationAttributeExpr$`-The segmentation expression based on the initiator event attributes for the segmentation context. (For example, if the context is initiated by a first deposit of a customer, than the initiator event is the first deposit, and the expression here is `Deposit.customerID`)
- **Trend:** the template is for searching for a trend for a certain attribute's value of input event in a certain time window, started by some initiator event and lasting for N millis. The trend is searched for within a certain segmentation context, for example counting withdrawals for a specific customer (segmentation by customer id). The trend is reported once, at the moment the trend count reaches the specified threshold.
 - `$InputEvent$` - the name of the input event we are looking for trend over attribute of/expression
 - `$OutputEvent$` - The name of the event to when trend count reaches a certain threshold
 - `$InitiatorEvent$` - the name of the event to start the temporal context during which we look for trend
 - `$TemporalContextDuration$` - defines the length of temporal window during which we look for trend, in millis, from initiator event.
 - `$InputEventTrendExpression$` - the expression over attributes of input event over which we evaluate if trend exists
 - `$TrendTreshold$` - once the number of observations of trend within the value of previously specified expression reaches this threshold, output event will be derived
 - `$TrendRelation$` - specifies what kind of trend we are looking for, whether 'Increase', 'Decrease' or 'Stable'
 - `$DerivedAttributeName$` - The name of the attribute in the derived event where we want to assign value during derivation.
 - `$DerivedAttributeExpression$` - The expression for the derived attribute. It can be a trend count (use 'trend.count' for that), a segmentation partition value a segmentation partition value or a constant value
 - `$InputEventSegmentationAttributeExpr$` - The segmentation expression based on the input event attributes for the segmentation context. (for example, if we looking for trend in the Withdrawals for certain customer, then it is the `Withdrawal.customerID` attribute)

- \$InitiatorEventSegmentationAttributeExpr\$-The segmentation expression based on the initiator event attributes for the segmentation context. (For example, if the context is initiated by a first deposit of a customer, than the initiator event is the first deposit, and the expression here is Deposit.customerID)
- Once the template parameters are filled in, Verify the correctness of the values you provided by pressing the “Verify” button
- In order to add the artifacts (epas and contexts) defined by the template, do **Save and Export**. This will create the JSON representing the project together with template artifacts. This JSON can be imported back into project to see the artifacts in the project tree in order to review/change things.

Note: please take into account that template is not a constant artifact belonging to the project. This is a temporal artifact. That means it is not saved as part of the projects artifacts, and if you do **Save** or **Save and Export** what will happen is that the EPA and contex artifacts defined by the template will be added to the project, and you will see it in the tree under **EPAs** and **Contexts** headers, but the template itself is not saved as part of the project.

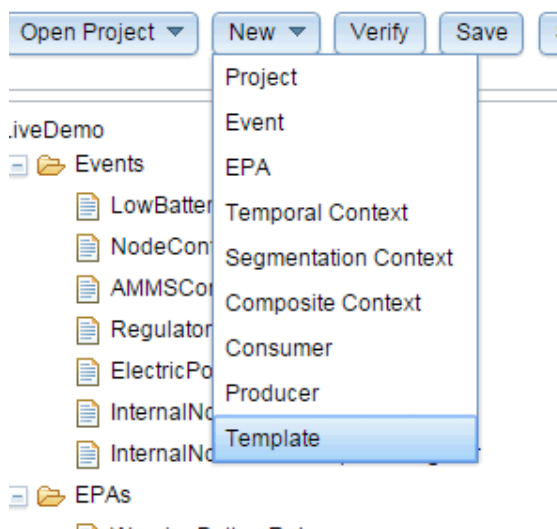


Figure 13: EPM Authoring Tools – Template Menu

And then you choose the name of the template:

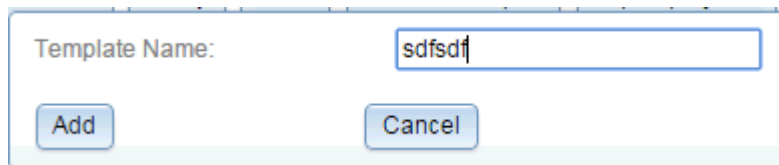


Figure 14: EPM Authoring Tools – Choose template name

And choose the type and fill in the values:

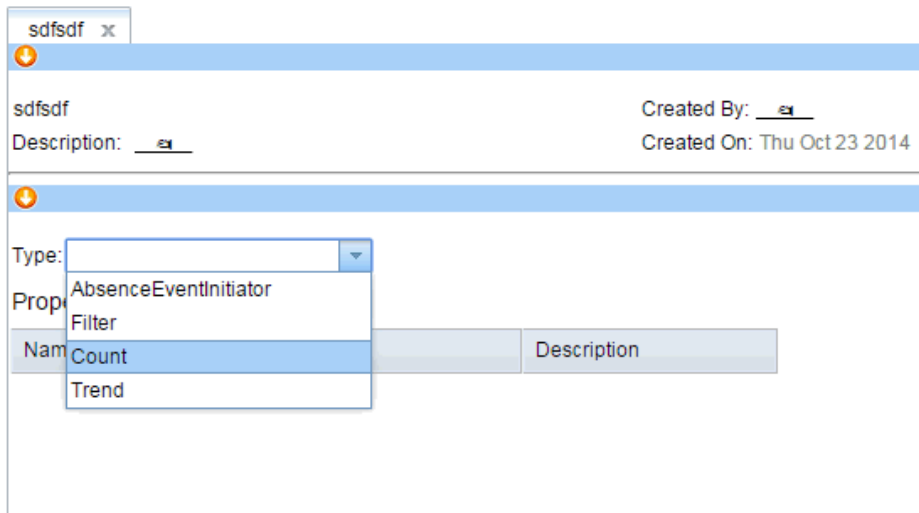


Figure 15: EPM Authoring Tools – Choose template type

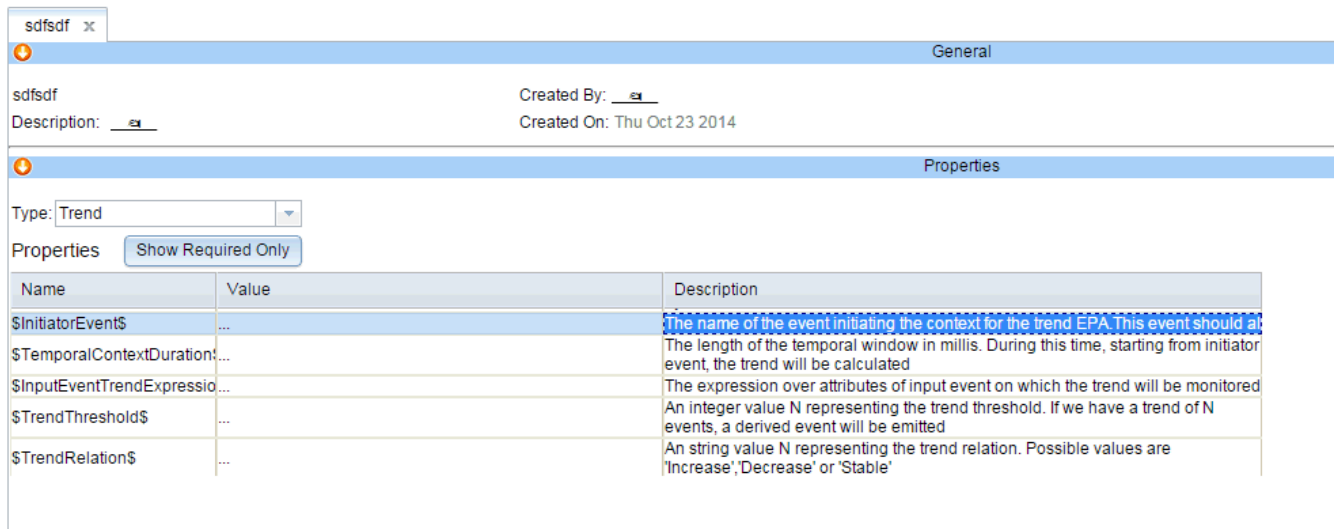


Figure 16: EPM Authoring Tools – Fill template values

After that, this template is placed with as part of artifacts of the project and the tab can be opened or closed.

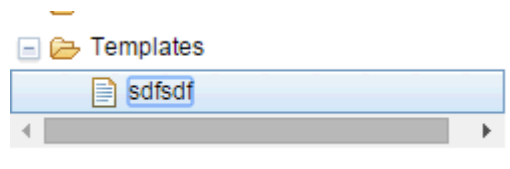


Figure 17: EPM Authoring Tools – Save / Export the template

But the template is not saved as part of the project. So if you open the project a new without doing anything, this information would be lost.

In order to generate the artifacts from template, you do Save and Export-> and then the JSON including all the artifacts is created.

2.3.5 Creating a new application

To create a new EPM application the following steps are to be followed:

- Follow steps 1-2 in [Before starting to use EPM Authoring Tool](#)
- Use the [EPM Authoring Tool](#) to define event processing definitions for the required application
- After designing the application in EPM Authoring Tool, verify that all the definitions are correct by pressing on “*Verify*” button, and then export the definitions to EPM configuration file using the “*Save and Export*” option of the authoring tool. Save the file somewhere in the local file system.

2.3.6 How to integrate EPM with other components in the platform?

EPM is integrated with other components in the platform using a publish-subscribe CSB infrastructure.

Like any other publish-subscribe system, CSB uses the concept of topics to which business messages are published, and to which other components can subscribe to receive those messages.

Therefore to integrate EPM to other components three things have to be specified:

1. The CSB queues to which EPM subscribes to in order to receive incoming events (EPM producer), and queues to which EPM outputs derived events
2. The messages which are passed via CSB to EPM and represent the incoming events and the messages outgoing from EPM to CSB and represent derived events. The following guidelines should be followed while designing messages intended for EPM:
 - a. The name of the message object should match the relevant event type name in EPM JSON configuration file
 - b. The message object should include getters (and for messages created by EPM to represent derived events – setters) for fields whose names match the names of matching attributes within the event type definition. Those fields should also be primitives (or auto-boxed representation of primitives)
 - c. The object message should include a default constructor

Note: The message objects can be generated using JAXB from XSD, as long as the matching field types in XSD are flat and not nested.

See example: The XSD schema and the automatic JAXB-based generation of matching POJO object vrs definition of matching event type in EPM.

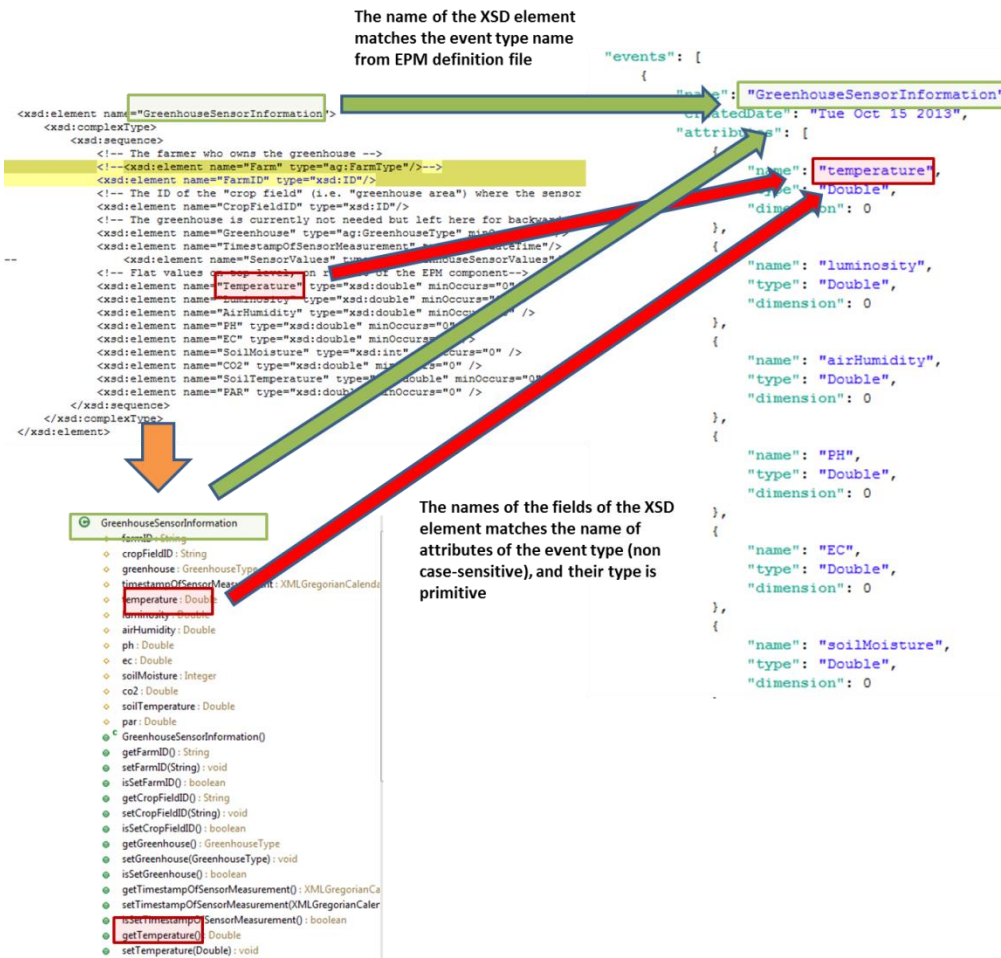


Figure 18: XSD schema and automatic JAXB-based generation

- Once the messages are defined the queues are defined, their specifications should be included in the EPM application definition using a CSB consumer and/or CSB producer.

The CSB producers should be specified as part of the [authoring application on Authoring Tool step](#).

To specify a CSB producer: create a producer using the Authoring Tool and specify the following properties: (other properties can be left as default)

- Type of the producer – CSB
- The name of the CSB queue on which messages will be arriving to EPM
- The path to the jar holding .class files for the business messages which match the EPM event types and which will be arriving to EPM via the CSB queue. The path can be either the path of the file (i.e. fispace-greenhouse-core-api.jar) or a path to a directory holding a number of jars (i.e \fispace-greenhouse-core) in case the message objects used by EPM is defined in several jars, or if there are dependencies on other jars.

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 19: Example of CSB producer

The same goes to specifying CSB consumer: create a consumer using the Authoring Tool and specify the following properties: (other properties can be left as default)

- Type of the consumer – CSB
- The name of the CSB queue on which EPM will write messages intended for other components
- The path to the jar holding .class files for the business messages which match the EPM derived event types and which EPM will emit to the CSB queue .Like in producer, this can be either a path to the jar file, or a path to the directory holding the jar file.

Example of CSB Consumer:

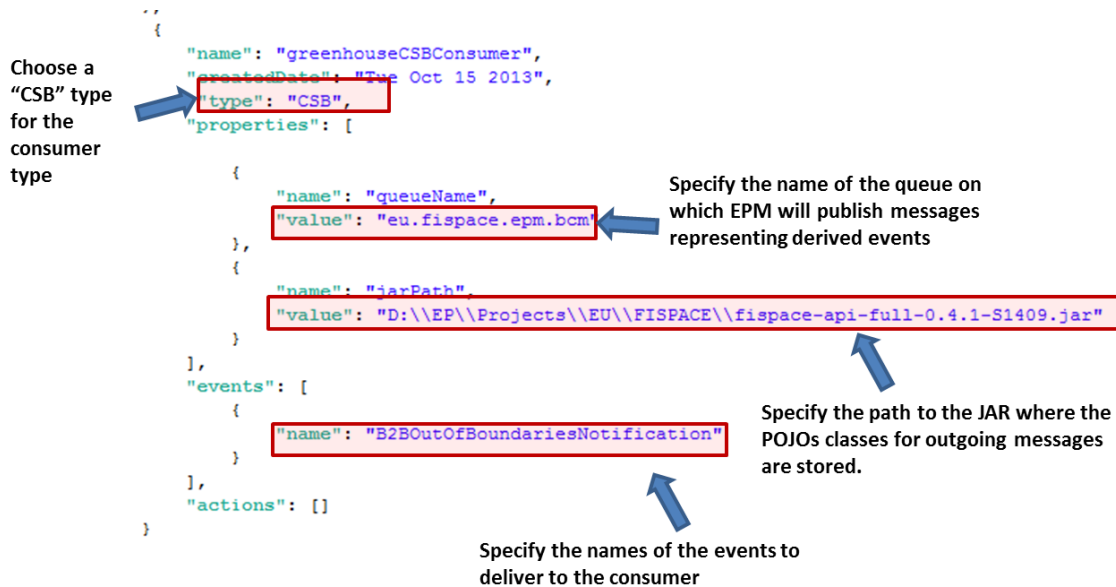


Figure 20: Example of CSB consumer

2.3.7 How to package and deploy a new application?

To create and deploy a new application instance the following should be done:

1. Ensure a CSB node is installed on the same machine as web application server on which EPM is going to be running.
2. The latest WARs for EPM: *ProtonOnWebServer.war* and *ProtonOnWebServerAdmin.war* should be acquired from the binary repository
3. If another instance of EPM already exists on the web server, the WARs should be renamed – preferably to something indicating the particular application scenario (i.e. *ProtonForGreenhouse.war* and *ProtonForGreenhouseAdmin.war* for the greenhouse application)
4. The WARs should be deployed on the web server (for Tomcat – see deployment instructions here: https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/CEP_GE_-_IBM_Proactive_Technology_Online_Installation_and_Administration_Guide#Installing_a_new_Proton_instance)
5. The web server should be restarted to allow the unpackaging of the web application on the server.
6. Stop the web server
7. Access the web applications folder on the web server (for Tomcat this is a "webapps" directory under Tomcat root directory), and configure the following:
 - a. Inside *Proton.properties* (located at the root directory of EPM web application – if the application is named **ProtonOnWebServer.war** – the *Proton.properties* will be located by default under $\${Tomcat\ install\ dir}/webapps/ProtonOnWebServer$ directory) configure the path to the JSON definition file you have exported in a [Creating a new application](#) step.

- b. Also inside Proton.properties, please configure two unoccupied ports (any port can be checked with *netstat -an* option) for EPM to work with.

Example:

```

# metadata file name - a path to the JSON file containing the application metadata
# for example:
# relative path .\\sample\\logistics.json
# Absolute path D:\\ProtonFiles\\logistics.json
#-----
metadataFileName = CSBgreenhouseAdvice.json

# should be either "relative" or "absolute"
metadataFilePathType = relative

# A free port to allocate the input adapter socket server
inputPortNumber = 3002
# A free port to allocate the output adapter socket server
outputPortNumber = 3302
    
```

The path to the EPM JSON configuration file. The path can be relative or absolute. Use escape characters while specifying the path

EPM uses sockets communication to listen to incoming and publish outgoing events. Please configure here unoccupied ports on which EPM can work.

Figure 21: Proton.properties and port configuration

- 8. As explained in [How to integrate EPM with other components on the platform](#), please make sure that the path to the POJO's jar is correct, and that the proper business object classes are located in this jar.

Follow the links for https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/CEP_GE_-_IBM_Proactive_Technology_Online_Installation_and_Administration_Guide#Installing_a_new_Proton_instance and https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/CEP_GE_-_IBM_Proactive_Technology_Online_Installation_and_Administration_Guide#Configuring_an_engine_instance for more information regarding installing and configuring a new EPM instance.

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

| | |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [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_- |

| | |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <u>User and Programmer Guide</u> |
| [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 pluggable architectures". Journal of Object Oriented Programming, Vol. 8, No. 2, pp. 18-24. |

