D7.3 – Recommendations to FIWARE and FI-PPP M24 Version
<table>
<thead>
<tr>
<th>VERSION</th>
<th>DATE</th>
<th>NOTES AND COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1/3/2015</td>
<td>TABLE OF CONTENTS AND RESPONSABILITIES</td>
</tr>
<tr>
<td>0.2</td>
<td>19/03/2015</td>
<td>UPDATE TO DOCUMENT STRUCTURE</td>
</tr>
<tr>
<td>0.3</td>
<td>22/03/2015</td>
<td>NEW CHAPTER 2.1 (VTT) AND 2.2 (POLIMI)</td>
</tr>
<tr>
<td>0.4</td>
<td>8/4/2015</td>
<td>CHAPTER 9.1 ANSWERS TO GE QUESTIONNAIRE (POLIMI), CHAPTER 3 APPLICATIONS/SERVICES AND DATA DELIVERY CHAPTER (ENG), CHAPTER 4 CLOUD HOSTING CHAPTER (PTIN),</td>
</tr>
<tr>
<td>0.5</td>
<td>20/04/2015</td>
<td>IVLAB CONTRIBUTION IN CHAPTER 1, CHAPTER 6 SECURITY CHARTER, CHAPTER 7 REVISED FIPPP RECOMMENDATIONS</td>
</tr>
<tr>
<td>0.6</td>
<td>26/04/2015</td>
<td>CONTRIBUTIONS FROM NTUA AND TXT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FINAL COMMENTS FROM WORKING TEAM</td>
</tr>
<tr>
<td>0.9</td>
<td>15/06/2015</td>
<td>REVIEWED SECTIONS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 CLOUD HOSTING</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7 INTERFACE TO NETWORKS AND DEVICES (I2ND)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 SECURITY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 APPLICATIONS/SERVICES AND DATA FRAMEWORK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11.1.3 SES ASSESSMENT NTUA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9 ADVANCED WEBUI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11 ASSESSMENT TO GES FROM ENG, ITINNO, ATOS</td>
</tr>
<tr>
<td>0.95</td>
<td>26/07/2015</td>
<td>NTUA REVISED CHAPTER 11.1.3 SES ASSESSMENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REVISED CHAPTER SECURITY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REVISED CHAPTER ADVANCED WEBUI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REVISED CHAPTER I2ND PTIN</td>
</tr>
<tr>
<td>0.96</td>
<td>10/08/2015</td>
<td>REVISED MAPPING OF GE ADOPTION ATOS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RELEASED FOR PEER REVIEW</td>
</tr>
<tr>
<td>0.97</td>
<td>31/08/2015</td>
<td>SG FEEDBACKS ⇒ LIST OF RECOMMENDATIONS IN EXEC SUMMARY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SECURITY – ITINNO – NO AMEND</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ADVANCED WEBUI – DFKI – REC # 36</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CLOUD HOSTING – PTIN CHAPTER 3.1 3.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PEER REVIEW COMMENTS FROM EVA COSCIA –</td>
</tr>
<tr>
<td>0.98</td>
<td>04/09/2015</td>
<td>PEER REVIEW COMMENTS FROM MAURO ISAJA-ENG AND OUTPUT FROM INNSBRUCK MEETING</td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td>RELEASED TO EC</td>
</tr>
</tbody>
</table>
EXECUTIVE SUMMARY .................................................................................................................................. 4
1. DOCUMENT SCOPE AND DOCUMENT STRUCTURE ............................................................................. 7
2. INTRODUCTION ......................................................................................................................................... 9
3. CLOUD HOSTING ....................................................................................................................................... 10
   3.1 IaaS Resource Management GE .................................................................................................................. 11
   3.2 Self Service interfaces ................................................................................................................................ 12
   3.3 Questionnaire analysis and recommendations ................................................................................................. 12
4. DATA/CONTEXT MANAGEMENT .............................................................................................................. 18
5. APPLICATIONS/SERVICES AND DATA FRAMEWORK .............................................................................. 19
   5.1 Repository GE ............................................................................................................................................... 24
   5.2 Marketplace GE ........................................................................................................................................... 25
   5.3 ApplicationMashup GE ................................................................................................................................. 26
   5.4 LightSemanticComposition GE .................................................................................................................... 27
   5.5 Mediator GE ................................................................................................................................................. 28
6. INTERNET OF THINGS (IOT) SERVICES ENABLEMENT ............................................................................. 30
7. INTERFACE TO NETWORKS AND DEVICES (I2ND) .................................................................................. 31
   7.1 Experience from trials ..................................................................................................................................... 32
   7.2 Questionnaire analysis and recommendations ............................................................................................... 32
8. SECURITY ...................................................................................................................................................... 34
   8.1 FITMAN security GE selection ........................................................................................................................ 34
   8.2 Security GE evolution .................................................................................................................................... 35
   8.3 GE25 Security.IdentityManagement ........................................................................................................... 37
   8.4 GE26 Optional_Security_Enablers.DBAnymizer ............................................................................................. 42
   8.5 Enlarging the chapter .................................................................................................................................... 43
   8.6 Cloud security ............................................................................................................................................. 44
9. ADVANCED WEBUI ..................................................................................................................................... 46
   9.1 3D-UI GE (3D-UI-XML3D) .......................................................................................................................... 46
   9.2 Synchronization GE (FiVES) ......................................................................................................................... 47
   9.3 GIS Data Provider - Geoserver/3D ................................................................................................................ 49
   9.4 POI Data Provider / Publish/Subscribe Context Broker - Orion ................................................................. 49
   9.5 Augmented Reality ....................................................................................................................................... 49
   9.6 Virtual Characters ....................................................................................................................................... 50
   9.7 Advanced Middleware (legacy) ................................................................................................................... 50
10. FI-PPP RECOMMENDATIONS ................................................................................................................ 51
11. ANNEX ....................................................................................................................................................... 52
   11.1 Data Collection and Data Analysis methodology .......................................................................................... 52
   11.2 Answers to GE's Questionnaires ................................................................................................................ 57
   11.3 Answers to SE's Questionnaires ................................................................................................................ 57
EXECUTIVE SUMMARY

From the analysis conducted during the adoption and implementation phases until M24 and further a broad set of comments and suggestions emerged from Technical implementation teams of the FITMAN platforms. Some of them are configured as Lesson Learnt from the implementation, while others have been structured as recommendations to FIWARE. A first set of 37 recommendations and lesson learnt came from the implementation experience according the following “Table 1 Recommendation per Chapter at M24”:

<table>
<thead>
<tr>
<th>Chapter</th>
<th># Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud Hosting</td>
<td>1 (+5 lesson learnt)</td>
</tr>
<tr>
<td>Apps</td>
<td>10</td>
</tr>
<tr>
<td>I2ND</td>
<td>1</td>
</tr>
<tr>
<td>Security</td>
<td>8</td>
</tr>
<tr>
<td>WebUI</td>
<td>9</td>
</tr>
<tr>
<td><strong>Total #</strong></td>
<td><strong>34</strong></td>
</tr>
</tbody>
</table>

Table 1 Recommendation per Chapter at M24

The recommendations regarding Data/context Management Chapter and IOT Chapter will be synthesized and formalized in the further release of the deliverable, D7.4 at M30. In this last edition at M30, we will also report about further tests and experimentations still on-going and affecting specifically the BigData Analysis (experimentations with COSMOS GEs in Ch. 4) and the IoT architecture (new IoT Discovery and MR COAP components now integrated in our reference architecture IOT for Manufacturing in Ch. 6. Also including the Data Visualisation SpagoBI GE of the Applications/Services and Data Delivery chapter).

In the following table “Table 3 Recommendations List” we have the summary of the recommendation. For each recommendation it is reported the reference chapter and the page in this documents where it is described in detail. Recommendations in “Table 3 Recommendations List” have been classified in 4 groups in “Table 2 Recommendations per Class”:

- Functionalities: This class address the needs to extend or change the way a given GE works. This could allow a better support to Manufacturing Processes and ease the implementation process.
- Strategy: This class encompass considerations related to the way given GEs in Chapters should evolve or the positioning strategy advised to ensure a broader adoption in Manufacturing industry.
- Support: This class address the issues encountered during the implementation phases and it includes documentation, support, etc.
- Lesson Learnt: This class include experiences gathered during the Trials experimentations and evaluations.

<table>
<thead>
<tr>
<th>Class</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functionalities</td>
<td>19</td>
</tr>
<tr>
<td>Strategy</td>
<td>10</td>
</tr>
<tr>
<td>Support</td>
<td>3</td>
</tr>
<tr>
<td>Lesson Learnt</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 2 Recommendations per Class
In the following “Table 3 Recommendations List”, the recommendations and lesson learnt are listed per chapter. In the column Page, it is reported the page number in this document where the recommendation/lesson learnt is described in detail.

<table>
<thead>
<tr>
<th>#</th>
<th>Chapter</th>
<th>Short Description</th>
<th>Page</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>#01</td>
<td>Cloud Hosting</td>
<td>Definition of a cloud path for Manufacturing Industry</td>
<td>13</td>
<td>Lesson Learnt</td>
</tr>
<tr>
<td>#02</td>
<td>Cloud Hosting</td>
<td>Support issues differentiation of early adopters and commercial customers</td>
<td>14</td>
<td>Lesson Learnt</td>
</tr>
<tr>
<td>#03</td>
<td>Cloud Hosting</td>
<td>Integration issues and Roadmap definition</td>
<td>15</td>
<td>Strategy</td>
</tr>
<tr>
<td>#04</td>
<td>Cloud Hosting</td>
<td>Proper addressing of Security issues towards stakeholders (e.g. CIO)</td>
<td>15</td>
<td>Lesson Learnt</td>
</tr>
<tr>
<td>#05</td>
<td>Cloud Hosting</td>
<td>Business model issues for exploitation of developed SW assets</td>
<td>16</td>
<td>Lesson Learnt</td>
</tr>
<tr>
<td>#06</td>
<td>Cloud Hosting</td>
<td>Provide a service continuity, high availability and resilience framework as well as tools for disaster recovery and data replication.</td>
<td>16</td>
<td>Lesson Learnt</td>
</tr>
<tr>
<td>#07</td>
<td>Apps</td>
<td>Adoption in FIWARE of a two-lane release cycle (Linux-like)</td>
<td>22</td>
<td>Strategy</td>
</tr>
<tr>
<td>#08</td>
<td>Apps</td>
<td>DataVizGE Data Delivery with Manufacturing Focus</td>
<td>23</td>
<td>Strategy</td>
</tr>
<tr>
<td>#09</td>
<td>Apps</td>
<td>Repository GE provides generic web UI / Web Based LinkedUSDL editor</td>
<td>25</td>
<td>Functionalties</td>
</tr>
<tr>
<td>#10</td>
<td>Apps</td>
<td>Direct use of ApplicationMashup without needs for Marketplace + Store GE</td>
<td>26</td>
<td>Functionalties</td>
</tr>
<tr>
<td>#11</td>
<td>Apps</td>
<td>Fine tune for access rights via ACL</td>
<td>27</td>
<td>Functionalties</td>
</tr>
<tr>
<td>#12</td>
<td>Apps</td>
<td>ApplicationMashup Support for some server-side web application engines (e.g. PHP)</td>
<td>27</td>
<td>Functionalties</td>
</tr>
<tr>
<td>#13</td>
<td>Apps</td>
<td>ApplicationMashup Support for LDAP and ActiveDirectory</td>
<td>27</td>
<td>Functionalties</td>
</tr>
<tr>
<td>#14</td>
<td>Apps</td>
<td>LightSemanticComposition GE Fully automated design environment</td>
<td>28</td>
<td>Functionalties</td>
</tr>
<tr>
<td>#15</td>
<td>Apps</td>
<td>LightSemanticComposition GE - new property editor in the BP design environment</td>
<td>28</td>
<td>Functionalties</td>
</tr>
<tr>
<td>#16</td>
<td>Apps</td>
<td>Mediator GE - Add a server configuration option, enabling the logging to a local file of the service payload of every call</td>
<td>29</td>
<td>Functionalties</td>
</tr>
<tr>
<td>#20</td>
<td>I2ND</td>
<td>Include components for implementing SDN and NFV functionality</td>
<td>32</td>
<td>Functionalties</td>
</tr>
<tr>
<td>#21</td>
<td>Security</td>
<td>Security.IdentityManagement GE - A mechanism should be added to the IDM portal interface and API so that an user can remove a web application from the list of authorised applications</td>
<td>40</td>
<td>Functionalties</td>
</tr>
<tr>
<td>#22</td>
<td>Security</td>
<td>Security.IdentityManagement GE - The IDM should implement OpenID Connect to provide more robust authentication and consideration should be given to implementing two-factor authentication</td>
<td>40</td>
<td>Functionalties</td>
</tr>
<tr>
<td>#</td>
<td>Category</td>
<td>Description</td>
<td>Recommendation</td>
<td>Type</td>
</tr>
<tr>
<td>-----</td>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------</td>
<td>--------</td>
</tr>
<tr>
<td>#23</td>
<td>Security</td>
<td>Security.IdentityManagement GE - Better documentation on how to configure the IDM software is needed</td>
<td>41</td>
<td>Support</td>
</tr>
<tr>
<td>#24</td>
<td>Security</td>
<td>Security.IdentityManagement GE - The IDM software would be more generally useful as a stand-alone component if most of the FIWARE branding, messages and analytics were removed or easily configurable</td>
<td>42</td>
<td>Strategy</td>
</tr>
<tr>
<td>#26</td>
<td>Security</td>
<td>FIWARE should consider adding the SAML tool to the security chapter</td>
<td>44</td>
<td>Strategy</td>
</tr>
<tr>
<td>#27</td>
<td>Security</td>
<td>For public clouds, the security features should be clearly enumerated to increase confidence and uptake</td>
<td>44</td>
<td>Strategy</td>
</tr>
<tr>
<td>#28</td>
<td>Security</td>
<td>For software that can be used to create a private cloud and for public cloud systems, the security features should be clearly explained to increase confidence and uptake</td>
<td>45</td>
<td>Strategy</td>
</tr>
<tr>
<td>#29</td>
<td>WebUI</td>
<td>D-UI GE Improve the documentation of specific features of XML3 and XFlow</td>
<td>46</td>
<td>Support</td>
</tr>
<tr>
<td>#30</td>
<td>WebUI</td>
<td>3D-UI GE XML3D should provide a standard binary format for assets</td>
<td>46</td>
<td>Functionality</td>
</tr>
<tr>
<td>#31</td>
<td>WebUI</td>
<td>3D-UI GE XML3D should provide a way to pick generic vertex attributes</td>
<td>47</td>
<td>Functionality</td>
</tr>
<tr>
<td>#32</td>
<td>WebUI</td>
<td>Synchronization GE Improving the bi-directional synchronization of XML3D scenes with FiVES</td>
<td>48</td>
<td>Functionality</td>
</tr>
<tr>
<td>#33</td>
<td>WebUI</td>
<td>Synchronization GE Add XML3D node annotations to mark which parts of the scene will be kept in synch by FiVES directly in the XML3D scene</td>
<td>48</td>
<td>Functionality</td>
</tr>
<tr>
<td>#34</td>
<td>WebUI</td>
<td>GIS Data Provider - Geoserver/3D We suggest to improve the GE to become feature complete</td>
<td>49</td>
<td>Functionality</td>
</tr>
<tr>
<td>#35</td>
<td>WebUI</td>
<td>GIS Data Provider - Geoserver/3D We suggest to strengthen the efforts of the POI WG</td>
<td>49</td>
<td>Functionality</td>
</tr>
<tr>
<td>#36</td>
<td>WebUI</td>
<td>Augmented Reality We strongly suggest to revisit this GE implementation making it more maintainable and improvable utilizing open Development environment and tools</td>
<td>50</td>
<td>Strategy</td>
</tr>
<tr>
<td>#37</td>
<td>WebUI</td>
<td>Advanced Middleware (legacy) - FI-WARE to spend efforts in continuing previous work to ensure the provided reference provides support for more than one platform (in particular C# and JavaScript)</td>
<td>50</td>
<td>Support</td>
</tr>
</tbody>
</table>

Table 3 Recommendations List

As stated above recommendations for Data/Context Management and IoT Chapters and FI-PPP Recommendations will be included in D7.4 at M30.
1. Document Scope and Document Structure

This document is intended to consolidate former deliverables D7.2-D7.6 including recommendations to FIWARE Chapters from the “FI for Manufacturing Industry” viewpoint, as stated in the DoW.

Considering the different FIWARE chapters in isolation, as previously foreseen in the DOW, risked to create artificial walled gardens especially from a Use Case perspective, where different GEs and SEs belonging to different Chapters are integrated in the Trials architectures and sometimes also not clearly distinguishable. The experience gained during the project suggested therefore to consolidate these documents into the present D7.3, in order to ensure a consistent synthesis exercise of the feedbacks to different FIWARE Chapters.

The current document is the M24 issue, intended to be complemented in the M30 revision.

In the course of the last year (M18-M30) of the project, FITMAN made significant improvements in the GE composition of its trial architectures, following the evolution of FIWARE catalogue specifications, reference architectures and reference implementations. Such an evolution was particularly disruptive in Ch. 4 Data/Context Management and Ch. 6 Internet of Things Service Enablement, where not just new GE implementations were introduced but also new open specifications and new reference architectures developed.

FITMAN IT specialists (architects and developers) just in the last months had in fact the opportunity to test and experiment the new evolutions of both Chapters. For this reason, the recommendations regarding these two chapters will be synthesize and formalized in the further release of the deliverable, D7.4 at M30. In this last edition at M30, we will also report about further tests and experimentations still on-going and affecting specifically the BigData Analysis (experimentations with COSMOS GE in Ch. 4) and the IoT architecture (new IoT Discovery and MR COAP components now integrated in our reference architecture IOT for Manufacturing in Ch. 6. Also including the Data Visualisation SpagoBI GE of the Applications/Services and Data Delivery chapter).

This deliverable D7.3 Recommendations to FIWARE is structured as follows:

Section 3. Cloud Hosting – the fundamental layer which provides the computation, storage and network resources, upon which services are provisioned and managed.

Section 4. Data/Context Management – the facilities for effective accessing, processing, and analysing massive volume of data, transforming them into valuable knowledge available to applications. (Not addressed in this release)

Section 5. Applications/Services and Data Framework – the infrastructure to create, publish, manage and consume FI services across their life cycle, addressing all technical and business aspects.

Section 6. Internet of Things (IoT) Services Enablement – the bridge whereby FI services interface and leverage the ubiquity of heterogeneous, resource-constrained devices in the Internet of Things. (Not addressed in this release)

Section 7. Interface to Networks and Devices (I2ND) – open interfaces to networks and devices, providing the connectivity needs of services delivered across the platform.

Section 8. Security – the mechanisms which ensure that the delivery and usage of services is trustworthy and meets security and privacy requirements.

Section 9. Advanced WebUI – new user input and interaction capabilities, such as interactive 3D graphics and immersive interaction with the real and virtual world.

Section 10 FI-PPP Recommendations will be issued at M30.
In Annex 11 the data collection approach and the files with individual answers to the questionnaires issued to IT Partners regarding GEs (11.2 Answers to GEs Questionnaires) and to SEs owners (11.3 Answers to SEs Questionnaires) are reported.
2. Introduction

The evaluation on the GEs reported in this deliverable takes into account not only the GEs selected in FITMAN WP1 at Month 10, but also how the FIWARE catalogue evolved along time and how these new features could be exploited in the Manufacturing domains.

Namely Task 1.3: FI-WARE Generic Enablers selection for FITMAN executed initially a selection of GEs in two rounds. First round producing D1.3 “FI-WARE Generic Enablers Selection for FITMAN” issued on Jul 2013, identified 15 GEs. A second round, producing D1.3 FI-WARE Generic Enablers final selection for FITMAN - Second & final iteration issued in Dec 2013 identified 11 more GEs.

At the end, T1.3 FI-WARE Generic Enablers final selection, identified 26 candidates for their integration into the Smart, Virtual and Digital Factory platforms that FITMAN was implementing.

As stated in the following chapters from 3 to 9, along with the implementation of the Trials till today, the FIWARE catalogue was evolving and some existing GEs substantially improved, some others were suppressed, re-focussed or merged with others, some others newly introduced in the catalogue.

At the time of the issue of this deliverable, the GEs integrated and deployed by the 10 FITMAN trials are reported in the following Table 4 GEs adopted per Trial.

Some additional GEs (i.e. COSMOS PROTON SPAGOBI IOTDISC COAP) are currently under study, testing and possibly integration in some FITMAN Trials platforms: we will report about this on-going activity in the last version of this deliverable at M30.

<table>
<thead>
<tr>
<th>Trial ID</th>
<th>Trial Name</th>
<th>Adopted GEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VW</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>TRW</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>AW</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>Whirpool</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>PIACENZA</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>APR</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>Cons</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>TANET</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>COM+</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>AIDIMA</td>
<td>2</td>
</tr>
</tbody>
</table>

Please refer to chapter from 3 to 9 of this deliverable for a comprehensive description of the experimental activities that took place all along the project in evaluating possible utilization of available GEs and possible alternatives suggested or motivation to justify different solutions. Each chapter is also providing comments about the evolution of the FIWARE Catalogue.

Another version of this deliverable will be released at M30 collecting the final experiences from the project and the experiences from the FML, FITMAN Innovation Lab, the vehicle for the exploitation of FITMAN, providing final recommendations to the whole FI PPP to be a valid contribution to H2020 and subsequent FI PPP initiatives.
3. Cloud Hosting

As a preamble note, the word “Cloud Hosting” designates (in FIWARE terminology) the compute, storage, network and managed services commonly provided by a Cloud Service Provider.

FIWARE developed an ecosystem of software components, designated by Generic Enablers (GEs), to address the cloud services need. It strongly incorporates the cloud service models of Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS), in a perspective of providing everything as a service (XaaS).

![Fig. 1 - Everything as a Service (XaaS) stack model](image)

According to the commonly accepted definition of “Cloud Computing” from NIST, the FIWARE Cloud Hosting GE defines a true cloud model (available at the Fi-Lab) in the sense that they present the following characteristics, along with the facilities of a public delivery model:

- multi-tenancy isolation,
- managed scalability,
- measurable services,
- on demand self-service,
- a pay-per-use model (also known as pay-a-you-grow)
- a broad network access.

Completed by an offer of shared pool of resources:

- compute services (ex: vCPU, VRAM, vDSK),
- network services (ex: reserved bandwidth, load balancing, firewall),
- storage services (ex: attach disk, mount Volume),
- image repository (ex: WS2012std_x64.img, CentOS7x64.img, Ubuntu1504x64.img)

This chapter analyses the compiled responses to the Cloud Hosting questionnaire provided by the trails that experimented the usage of FIWARE Cloud Hosting, namely

- GE9 – Cloud.DCRM (Data Center Resource Management)
- GE10 – Cloud.ServiceManagement
GE11 – Cloud.ObjectStorage
GE12 – Cloud.SelfServiceInterface
GE13 – Cloud.SDC (Software Deployment and Configuration)
GE14 – Cloud.PaaS (Platform as a Service)

As reported in questionnaires, only 2 Cloud GE’s were used on trials:
- **DC Resource Management**
- **Self Service Interface** (aka, “Cloud Portal”)

Some could think that there are an objective interpretation for the low usage of Cloud GE’s, and that would emerge from the questionnaire’s answer, that claimed for incomplete or missing functionality, and lack of support, stability and scalability. However, one negative reason is not representative and cannot be generalized for all trials. On that context, other explanations should be found.

The two main reasons for the reduced adherence to FIWARE Cloud’s GE on FITMAN trials are:

- The cloud GE’s are strongly based on OpenStack modules. Despite the openness and integration potential of OpenStack, with broad support for the major manufacturers, the true is OpenStack projects are passing through a rapid evolution, with short time between stable releases. Similar to OpenStack, the Cloud GE’s are emergent platforms, and sometimes the ICT department doesn’t have the confidence enough to keep it running on production environments. Their time is yet to come but briefly.

- For security reasons, some partners preferred to run the software on top of traditional IT instead of manage them as cloud resources.

Those two issues lead to Recommendation #4 ahead.

To better understand the environment and functionalities of GE’s used on Cloud Hosting trials, let’s make a previous characterization.

### 3.1 IaaS Resource Management GE

The **IaaS Resource Management** Generic Enabler, also called “IaaS Resource Management”, is based on OpenStack core modules like Compute (codename ‘nova’), Image (‘glance’), Block Storage (‘cinder’) and Networking (‘neutron’), and introduces extended capabilities.

From the tenant point-of-view, it offers the same capabilities of OpenStack, such as the provisioning of virtual machines (VM), image catalogue management, network and storage management, and resource monitoring, but also brand new extended capabilities like VM life cycle management, resiliency of VM persistent data, guarantees of resource allocation and secure access to the VM.

From the cloud service provider point-of-view, and in addition to multi-tenancy isolation, DCRM also provides extended capacities like
- Resource optimization and over-commit handling
• Capacity management and admission control
• Automation of regular operations tasks
• Infrastructure and stack resiliency

Similar to all the others Generic Enabler, the FIWARE IaSRM GE, it is open source. The licence is Apache 2.0. The software can be downloaded by anyone from GitHub. The “IaaS Resource Management” has a public open specification and a public RESTful API for easy integration with third parties.

### 3.2 Self Service interfaces

The Self Service interfaces Generic Enabler provides a web console similar to OpenStack Dashboard (codename ‘horizon’) to manage services and resources in a centralized portal. It also provides a set of scripts to execute common tasks through a command line toolkit.

Similar to all the others GE’s, it’s available under explicit Terms and Conditions:

- for experimentation/testing purposes at the FI-PPP scope (under FI-PPP Collaboration Agreement)
- as experimental instances at the FI-Lab or downloading (under FI-LAB Terms and Conditions)
- to external usage as open source under MIT license

The “Self Service Interfaces” has a public open specification.

### 3.3 Questionnaire analysis and recommendations

The first question “Q2.1 – cloud hosting assessment header”, common to all the chapters, allows you to assess the degree of usage of FIWARE (GE) and FITMAN (SE/OC) software components on the trials.

<table>
<thead>
<tr>
<th>Question</th>
<th>#/Total</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>my experimentation involves two or more cooperating GEs from the same chapter</td>
<td>2/10</td>
<td>20%</td>
</tr>
<tr>
<td>the GEs have been integrated into my experimentation site as standalone GEs components</td>
<td>1/10</td>
<td>10%</td>
</tr>
<tr>
<td>I have not been using any GE from the chapter at all</td>
<td>7/10</td>
<td>70%</td>
</tr>
</tbody>
</table>

The answers lead to the interpretation: in fact only 20% of the trials made use of cloud hosting GE’s

- 70% did not use at all the Cloud.GE’s
- one of them decided to run the Cloud.GE on premises: “The experimentation was delivered on local internal server and due to internal ICT policies” [sic.]
- as a matter of fact, only 2 trails in 10 used hosted Cloud.GE’s

The root-cause analysis for such lack of adherence to FIWARE Cloud GE is the same as explained previously:

- emerging technologies not enough stable yet
- security concerns over critical knowledge and innovative solutions and products
The previous table leads also to the first set of recommendations:

**#1 Cloud Hosting Lesson Learnt:** Definition of a cloud migration strategy and implementation path for Manufacturing Industry and SMEs in particular

Considering the universal cloud adoption tendency, with benefits in efficiency (normalization, consolidation and optimization on technologies and processes) and effectiveness of manufacturing critical business processes (cost and time reduction, product quality, digital innovation) FIWARE should not just provide the technological infrastructure, but also devote efforts to identify and support the development of the cloud strategy and implementation path.

The second common question “**Q2.2 - feedback**”

<table>
<thead>
<tr>
<th>Question</th>
<th>Trial#2</th>
<th>Trial#3</th>
<th>Trial#4</th>
<th>#/Total</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y - Is the platform self-consistent N - how is it not self-consistent</td>
<td>No resp.</td>
<td>Y</td>
<td>Y</td>
<td>2/2</td>
<td>100%</td>
</tr>
<tr>
<td>Y - Is the platform complete in terms of functionality N - what is missing in terms of functionality</td>
<td>No resp.</td>
<td>Y</td>
<td>N</td>
<td>1/2</td>
<td>50%</td>
</tr>
<tr>
<td>Y/N - Did they select the right GE's</td>
<td>No resp.</td>
<td>Y</td>
<td>Y</td>
<td>2/2</td>
<td>100%</td>
</tr>
<tr>
<td>Y/N - Did they forget any de-facto or de-jure standard diffusely adopted in the community</td>
<td>No resp.</td>
<td>N</td>
<td>N</td>
<td>2/2</td>
<td>100%</td>
</tr>
<tr>
<td>Y/N - Did they forget any key technology / application domain</td>
<td>No resp.</td>
<td>N</td>
<td>N</td>
<td>2/2</td>
<td>100%</td>
</tr>
</tbody>
</table>

The previous table doesn’t enable clear conclusion, considering the universe of 10 trials and 6 Cloud GE’s:

- both agree with the software components consistency, the right choice and *de facto* compliance, i.e., a very positive feedback
- one of them refers some lack of support, stability and scalability, namely on the PaaS Cloud.GE – “full Platform-as-a-Service support was still missing; the PaaS GE was deployed on FI-LAB (didn’t work properly, though) but without dynamic scalability support” [sic.]

The question “**Q2.3 - GE assessment**” is chapter specific.

<table>
<thead>
<tr>
<th>Question</th>
<th># used</th>
<th>Percentage Used</th>
<th>Percentage NUsed</th>
</tr>
</thead>
<tbody>
<tr>
<td>GE9 – Cloud.DCRMResourceMgmt</td>
<td>1/10</td>
<td>10%</td>
<td>9/10</td>
</tr>
<tr>
<td>GE10 – Cloud.SM</td>
<td>0/10</td>
<td>0%</td>
<td>10/10</td>
</tr>
<tr>
<td>GE11 – Cloud.ObjectStorage</td>
<td>0/10</td>
<td>0%</td>
<td>10/10</td>
</tr>
<tr>
<td>GE12 – Cloud.SelfServiceInterface</td>
<td>2/10</td>
<td>20%</td>
<td>8/10</td>
</tr>
<tr>
<td>GE13 – Cloud.SDC</td>
<td>0/10</td>
<td>0%</td>
<td>10/10</td>
</tr>
<tr>
<td>GE14 – Cloud.PaaS</td>
<td>0/10</td>
<td>0%</td>
<td>10/10</td>
</tr>
</tbody>
</table>

The previous table reminds us that conclusions should be taken carefully because of the reduced number of samples.

The last set of questions about Cloud GE’s usage experience, reports a very positive appreciation but suggesting a few insights and opportunities for improvement (ex, support services).
The previous table leads to potential issues. It should always be stressed that the used case in cloud hosting trials did not produce enough information to make a deep analysis, and so (for this reason) it’s difficult to identify eventual lack of functionality, scalability or inadequate usage. Nevertheless, there are generic issues at the Cloud Hosting scope that can be raised to provide a positive contribution to FIWARE.

**#2 Cloud Hosting Lesson Learnt:** Set-up and provide differentiated levels of support for early adopters and commercial customers

An enterprise will easily add a software component to its application ecosystem if some kind of assistance is provided in case of bug reporting, mal-function or major version evolution needs. Despite bug correction is not ruled by the same service level compromises, +99,9% SLA is typical to recover service on a Cloud Service Providers. So software manufacturers must assure a support channel to respond.

A web support channel is the preferred method, with capacity to track the status of open issues and list historical issues and response time for SLA accomplishment. A hotline or email address can be consider as a complement to the web channel and not its substitute.

As a suggestion for improvement, and considering a basic service operation, an ITIL compliant tool should be activated to submit Service Request issues and Incident Management issues. There are several open source tools on the market, like OTRS and Atlassian JIRA Service Desk, among others.
It will be of particular importance during the current FIWARE technology dissemination phase, once it brings confidence to early adopters, especially at the Cloud Service Provider level, in its role of a business partner or as a client of a software solution.

**#3 Cloud Hosting Recommendation**: Develop a compatibility roadmap for future evolutions of the OpenStack framework.

The “Data Centre Resource Management” GE and the “Self Service Interface” GE used on trials exposes both a RESTful API that is extended from the native OpenStack API v2.0.

![FIWARE Cloud.DCRM extended capabilities](image-url)

Current OpenStack Juno release contain 11 core modules in stable 2.0 version. The next OpenStack Kilo release, announced on April 2015, adds 5 more core modules. Considering the need for evolution at the OpenStack component layer, due to new functionalities and enhanced security and stability, there is a risk of having installed v.2.0, v2.1 and v.3 of OpenStack modules on the same platform. This fact can lead to interoperability issues with FIWARE Cloud GE due to adaptation changes.

As a suggestion for enhancement, it should be clearly published the version compatibility supported by the FIWARE Cloud GE and its roadmap to follow the OpenStack evolution.

**#4 Cloud Hosting Lesson Learnt**: Set-up proper education and awareness actions for Manufacturing Industry CIOs, to clearly explain the fundamentals of cloud architecture, including its security risks and avoid false pre-judicial oppositions.
Due to the internal ICT rules and policies, enterprises tend to avoid the cloud services adoption at the first sight. To acquire confidence and facilitate the migration path to the cloud, then the service owners (cloud service providers, cloud service brokers and cloud developers) should promote several initiatives to explain and de-mystify security obstacles, namely explaining the fundamentals of cloud technologies to the potential adopters, in order to unmount false assumptions.

Security issues can be taken at several layers namely at the network, systems and application level, but also as a holistic approach extending to persons and processes.

Enterprises can also make a fair exercise of comparing its own internal security infrastructure and the technical degree of its own skilled employees, with the equivalent from an external cloud service provider. Most of the times it is recognized a serious lack of technical competences and with a legacy infrastructure. This awareness can facilitate the cloud hosting adoption.

At the application development level there are also requirements to enforce security: a customer database with column level encryption can hide sensible data to cloud team that operates the tenant platform.

As a resume, there is opportunity for training the future customers (ex, build secure applications) but also to define operational rules to the future Cloud Hosting Service Providers.

**#5 Cloud Hosting Lesson Learnt**: Address Business model issues

In the immediate project phase, where cloud hosted solutions are envisaged, it could be up to the FITMAN project to help defining the right offer, even considering that's a product or solution definition.

In order to promote the adoption and usage of FIWARE GE (and FITMAN SE) by the community and by the enterprises, several entry level business scenarios can be elaborated with the goal of dissemination of FITMAN/FIWARE results.

Considering an eventual business case for the deployment of several GE/SE to a particular industrial sector (whatever) in a cloud hosting environment, it is clear that that no royalties are due and all GE are free. Nevertheless you have to define clear SLA for commercial-grade support.

**#06 Cloud Hosting Lesson Learnt**: Provide a service continuity, high availability and resilience framework as well as tools for disaster recovery and data replication.

To industrial manufacturing (as to all other sectors) service failure is a synonym of economic losses. In order to avoid such drawback, redundant topologies should be preferred like:

- High-Availability infrastructure
- Disaster Recovery practices
- Geographic Replication
HA is addressed by “DC Resource Manager” and “Policy Manager” GE’s. However, DR and GeoReplication wasn’t foreseen but can be expressed to the Cloud Service Provider as a service requirement, in order to have more resilient FIWARE solutions.

In addition to local datacentre replication services for disaster recovery purposes, actually the main cloud service operators implemented the concept of “Region”, where data is spread between two or more data centres, typically at distance superior to 350Km. Cloud geo-replication can rise a sovereignty issue if data is supposed to be stored out of border: most countries have strong legal prescriptions to not authorize national data stored in foreign countries. Fortunately for most multinational firms acting in a globalized world, this capacity can be taken as an advantage, since data can be handled in a more efficient way especially in real time services where network latency represent a drawback, preventing the normal function of an application or software solution.
4. Data/Context Management

The Data/Context Management Chapter feedback will be included in D7.4 M30, due to on-going testing and experimentations of some of its fundamental components, the Big Data Analysis GE in particular.
5. Applications/Services and Data Framework

The Applications, Services and Data Delivery Chapter, also known in short as the Apps Chapter, is less internally cohesive if compared with other FIWARE Chapters: its reference architecture is partitioned into three distinct frameworks, each composed of a subset of collaborating Generic Enablers: Business and Marketplace Support\(^1\) Service Composition / Interoperability\(^2\) and Data Analysis and Visualization\(^3\).

Figure 1, extracted from the FIWARE Wiki\(^4\), shows how this vision can be translated into a complete business ecosystem platform. In the use case depicted below, digital assets are wrapped into software modules by a Developer. Once a module is made available, its terms and conditions of use are stored into a Repository GE, while its description is published as a commercial offering to an online Store GE. A Consumer will then leverage a Marketplace GE to gain access to the digital asset, and a Composer will build a User front-end by wiring the relevant software module (e.g., a web widget) to an ApplicationMashup GE instance running on a Service Provider’s infrastructure. Any revenue streams originating from these interactions (e.g., pay-per-use commercial agreements) are tracked by means of the Revenue Settlement and Sharing System GE.

---

1 I.e., Store, Marketplace, Repository, Registry, BusinessModeler, BusinessCalculator and RSS
2 I.e., ApplicationMashup, ServiceMashup, ServiceComposition, LightSemanticComposition and Mediator
3 This only contains the DataViz GE, which at the time of writing is still not defined as a FIWARE Open Specification but is nonetheless available as an open source component (SpagoBI by ENG) in the FIWARE Catalogue
The use case described above is just an example, as the Apps Chapter’s GEs are flexible enough to be assembled into a variety of configurations supporting different scenarios and domains. Most of them are also perfectly suitable as standalone utility components of non-FIWARE architectures.

In FITMAN, the concept of business ecosystem platform has been implemented in the Virtual Factory scope: FITMAN’s Virtual Factory baseline platform\(^5\) included several of the Apps Chapter GEs. Starting from there, Piacenza, APR, COMPlus and TANet have built their custom VF platform, tailored to the specific needs of their scenarios. Besides business ecosystems, GEs from the Apps Chapter also found some usage in Smart and Digital Factory scenarios like those in TRW, AgustaWestland, Consulgal and AIDIMA. In the following paragraphs we briefly recap FITMAN’s experience with the Apps Chapter, from GE selection to GE integration into the Trial sites.

One of the main objectives of the FITMAN project is to assess the suitability, openness and flexibility of FIWARE Generic Enablers in the context of manufacturing. To this goal, in FITMAN we first selected a subset of GEs from the FIWARE Core Platform: candidate Generic Enabler implementations (GEi) defining FITMAN’s toolbox for Trial experimentations. This selection process was done in the scope of task T1.3 – “FIWARE Generic Enablers final selection” – which was active from M1 until M9 (April – December 2013). The process was based on functional requirements coming from the ten FITMAN industrial Trials, and on a careful assessment of the information from FIWARE that was available at the time – i.e., April/December 2013. In particular, the GE’s Open Specification, the reference implementation’s roadmap (if any) and its terms and conditions of use. A preliminary experimentation of the existing implementations (installation, configuration, UI and/or API walkthrough) also contributed to the selection criteria. The final result of this process was a portfolio of 26 GEIs, 8 of them belonging to the Apps Chapter, that we identified as the \textit{FIWARE Generic Enabler final selection for FITMAN}\(^6\). In the following table (Table 5) we summarize these initial findings with respect to the Apps Chapter: which GEIs were evaluated, the evaluation outcome (i.e., \textit{go} vs. \textit{no-go}) and its rationale.

<table>
<thead>
<tr>
<th>Open Specification</th>
<th>Implementation</th>
<th>Provider</th>
<th>Sel.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketplace</td>
<td>ref. impl.</td>
<td>SAP</td>
<td>Y</td>
<td>Core enabler for Virtual Factory scenarios</td>
</tr>
<tr>
<td>Repository</td>
<td>ref. impl.</td>
<td>SAP</td>
<td>Y</td>
<td>Core enabler for Virtual Factory scenarios</td>
</tr>
<tr>
<td>Registry</td>
<td>ref. impl.</td>
<td>SAP</td>
<td>Y</td>
<td>Utility (nice-to-have) enabler for Virtual Factory scenarios</td>
</tr>
<tr>
<td>Store</td>
<td>WStore</td>
<td>UPM</td>
<td>N</td>
<td>No matching requirements from FITMAN Trials</td>
</tr>
<tr>
<td>RSS</td>
<td>ref. impl.</td>
<td>TID</td>
<td>N</td>
<td>No matching requirements from FITMAN Trials</td>
</tr>
<tr>
<td>BusinessCalculator</td>
<td>ref. impl.</td>
<td>iMinds</td>
<td>Y</td>
<td>No matching requirements from FITMAN Trials, but useful within the FITMAN project itself (exploitation planning)</td>
</tr>
<tr>
<td>BusinessModeler</td>
<td>ref. impl.</td>
<td>iMinds</td>
<td>Y</td>
<td>No matching requirements from FITMAN Trials, but useful</td>
</tr>
</tbody>
</table>

\(^5\) See deliverable D1.4 “FIWARE Platform Instantiation for FITMAN” (v2.0, released by M12), section #4.

\(^6\) See deliverable D1.3 “FIWARE Generic Enablers Selection for FITMAN” (v2.0, released by M9).
<table>
<thead>
<tr>
<th>ApplicationMashup</th>
<th>Wirecloud</th>
<th>UPM</th>
<th>Y</th>
<th>Core enabler for Digital Factory scenarios</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceMashup</td>
<td>Mashup Factory</td>
<td>Deutsche Telecom</td>
<td>N</td>
<td>Implementation only accessible as-a-Service from owner’s premises; no technical roadmap available</td>
</tr>
<tr>
<td>ServiceComposition</td>
<td>Ericsson Composition Engine</td>
<td>Ericsson</td>
<td>N</td>
<td>Implementation will not be available in the future, as the owner is in the process of withdrawing its support</td>
</tr>
<tr>
<td>LightSemanticComposition</td>
<td>COMPEL</td>
<td>ATOS</td>
<td>Y</td>
<td>Core enabler for Virtual Factory scenarios</td>
</tr>
<tr>
<td>Mediator</td>
<td>Mediator_TI</td>
<td>Telecom Italia / Thales</td>
<td>Y</td>
<td>Core enabler for Virtual Factory scenarios</td>
</tr>
</tbody>
</table>

Table 5 Outcome of the GE selection process in FITMAN by December 2013: Apps Chapter

Since then, the FIWARE Core Platform has evolved: while the original Open Specifications remained mostly untouched, some significant changes in the FIWARE Catalogue did occur. Below is a list of those changes that were FITMAN was concerned, followed by a very brief discussion of their impact. We will then summarize the lessons we learned during the last two years of collaboration with the FIWARE project, and try to formulate some constructive recommendations.

1. Mediator, LightSemanticComposition, Registry, BusinessCalculator and BusinessModeler are not available any more as an officially endorsed FIWARE GEi.\(^7\)
2. A new DataViz GEi was introduced, filling a gap in the Apps Chapter’s reference architecture (see Figure 1).
3. All GEis listed in the Catalogue are now distributed under an open source license.
4. Marketplace and Repository reference implementations\(^8\), originally developed by SAP, are now in charge of CoNWeT Lab (UPM) for support and further development\(^9\).

With respect to the first point, only the disappearance of Mediator/Mediator_TI and LightSemanticComposition/COMPEL from the Catalogue proved to be a concrete issue for the FITMAN project, as the remaining “orphaned” GEi were just marginally adopted in any of our ten Trials. Regarding COMPEL, it is worth noting that there’s one FITMAN SE which depends on it\(^10\); however, this is really a minor problem, as the SE owner\(^11\) is also a core FITMAN partner, so no lack of support is expected in the foreseeable future.

The second and third point of the list are obviously good news, and have been received positively in FITMAN. While open source licensing was already one of the requirements for GEi adoption, the appearance of the DataViz GE may enable new interesting business opportunities.

\(^7\) The same is true for ServiceMashup and ServiceComposition, however these GE were not included in FITMAN's final selection in the first place
\(^8\) Now named WMarket and Repository RI, respectively.
\(^9\) The same is true for RSS, originally developed by TID, which was not included in FITMAN's final selection
\(^10\) FITMAN-CBPM is an extension of COMPEL
\(^11\) COMPEL is provided by ATOS
scenarios, most likely in the Digital and Virtual Factory domains. More on this in the “lessons learnt” below.

Finally, the fourth point was seen in FITMAN as a positive outcome as well. While the implementation quality of the SAP-owned GEis was excellent, their move under the UPM umbrella will probably boost the active development of the code base. Moreover, having several collaborating GEis now managed by the same organization is likely to improve their mutual integration in the long run – and the internal cohesiveness of the entire Apps Chapter as well.

Overall, after two years of experimentation in FITMAN, there are a few lessons learnt that are worth mentioning.

Adoption of a two lane release cycle to consider the needs of support by developers and by commercial users. The FIWARE Core Platform was still a moving target when FI PPP Phase II, and FITMAN as part of it, started. FITMAN is a big project with many stakeholders and a complex management process: an agile approach to design and development was not enough to keep us at pace with the continuous evolution of FIWARE. Working with industrial partners on production processes required that we define a stable baseline environment at the very beginning. To this goal, our platform included a snapshot of the Apps Chapter taken at M12 (March 2014). As of today, the Apps Chapter has undergone several changes that are not reflected in FITMAN systems. In particular, as explained above, some GEis that are currently part of the FITMAN architecture have no officially supported implementation available any more. Having learned this lesson the hard way we issue this recommendation:

#7 Apps Recommendation We would advocate the adoption in FIWARE of a two-lane release cycle, similar to what happens with the Linux kernel and some Linux distributions: a bleeding edge branch maintained in parallel with a stable one. The latter should have a slower evolution, with a guaranteed supported lifetime of at least two years.

We also found it should be really helpful the development of a data delivery, business intelligence and advanced visualisation facility to allow management of Open Data in Virtual Factories. The Apps Chapter’s reference architecture, in its original form, lacked the “data delivery” part. This is reflected in the Open Specifications, where the DataViz GE is still missing at the time of writing, while a reference implementation does exist in the FIWARE Catalogue. This was quite unfortunate, as the unavailability of this feature set in the beginning may have prevented some of our Trials from including data delivery in their business scenarios. The lessons we learned here is that we should have cooperated more closely with the FIWARE team, lobbying in order to push our requirements on top of the development backlog. However, now that this new opportunity has emerged, in the last

---

12 In the Apps Chapter, UPM now owns five over six GEi: Wirecloud, WStore, WMarket, Repository RI and RSS RI
13 Task T1.3 “FIWARE Generic Enablers final selection for FITMAN” was originally planned to end by M4, but a second iteration of it was run up to M9 so that 2nd and 3rd release GE could be included in the selection
14 This recommendation originates in the Apps Chapter, but obviously applies to all Chapters
15 As a matter of fact, this Chapter was previously named “Applications/Services Ecosystem and Delivery Framework”
17 SpagoBI by Engineering, see http://catalogue.fiware.org/enablers/data-viz-spagobi
months of the FITMAN project we are going to revise our Trials platforms\(^\text{18}\) integrating the DataViz GE. As a recommendation to the FIWARE team:

**#08 Apps Recommendation** we want to stress that data delivery is not only a matter of Open Data in Smart City scenarios – which is the main focus of the current DataViz GE\(^\text{19}\) – and that manufacturing ecosystem may need some means to monetize valuable data sets in a similar way as applications and services

Having described the relationship of the FITMAN project with FIWARE’s Apps Chapter as a whole, in the following paragraphs we report the hands-on experience our developers had with individual Generic Enablers GEs while building and running their own Trial site. The objective of this technical-oriented review is to contribute to the FIWARE Core Platform’s evolution, by identifying any soft spots (possibly also providing some concrete suggestion for improvement), and highlighting those features that contributed the most to FITMAN’s success story. Only GEs that were integrated in at least one Trial site are considered here, using feedback from those developers who actually did the job. A questionnaire was used to collect the raw feedback\(^\text{20}\) and its results have been further processed in order to present a clear and consistent view of every assessed component.

For each GE we start with a simple scorecard: a coarse-grained assessment of four key aspects – i.e., ease of deployment, ease of use, overall engineering and technical documentation. Scores are in the following range: 0 “Insufficient”, 1 “Below average”, 2 “Average”, 3 “Above average”, 4 “Excellent”. These scores represent, at the topmost level, the perceived quality of the GE: they are the result of a subjective judgment, and as such they are not substantiated by any evidence here. They are reported because we think they are a simple but useful indicator of the GE’s overall value from a developer’s perspective.

We then go into more detail with six more specific questions we asked to the same people. The answers to each question have been edited and consolidated here as one single essay, which may optionally include one or more recommendations to the GE owner(s). The topics are the following:

- **Implementation**
  
  Survey question: *Was the implementation the best choice to accomplish to the GE goal?*

  Here we discuss to what extent the GEi implements the GE’s Open Specification

- **Functionality**
  
  Survey question: *On the basis of your experience in developing the solutions for the Trials and of your general knowledge of the FI and ICT for Manufacturing domain, are there any further functionalities you would expect from the GE?*

  The goal is to identify features that would meet some of the Trial’s requirements, but are missing from the GE’s Open Specification

- **Security**
  
  Survey question: *As far as your platform is concerned, do you find that this implementation comes with proper support for security (i.e., access control, data confidentiality)?*

- **Openness**

---

\(^{18}\) Data delivery within a business ecosystem is mostly a concern of Trials in the Virtual Factory domain, so the FITMAN Virtual Trials Platforms are the most likely candidates for this expansion

\(^{19}\) See the GE roadmap: [http://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/Materializing_Applications/Services_Ecosystem_and_Delivery_Framework_in_FI-WARE#Data_Visualization](http://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/Materializing_Applications/Services_Ecosystem_and_Delivery_Framework_in_FI-WARE#Data_Visualization)

\(^{20}\) See section #2.2 “GE Assessment”
Survey question: Where interfaces supporting the right protocols/standards to make the integration easy or something was missing?

- **Integration**
  Survey question: What is your overall experience in integrating this GE with your Trial’s platform (i.e., legacy systems / Trial-specific components / other GEs)?

- **Support**
  Survey question: Did you find enough support (experts, call center, forum, ...)?

As a general comment, we can safely say that assigned scores seem – not surprisingly, being a subjective assessment – to have been strongly influenced by the developers’ background and by their previous experience (or lack of) with similar software. The former point is about expectations: some users will happily reverse engineer any piece of software in order to better understand how it works, while others are more comfortable with components that can be deployed and run nearly off-the-shelf, with the support of detailed user manuals. The latter point makes the difference in the learning curve: in some cases (e.g., Mediator) we had developers who were already proficient with the underlying technology (site #6), while others (site #3) had to start from scratch. Obviously, this cultural gap is responsible for the sometimes striking difference in votes expressed for the same GE (e.g., Marketplace). That said, the more votes are contributed, the more significant is the final GE average score: some of the scorecards presented below (e.g., LightSemanticComposition) are not really significant with respect to this criterion.

### 5.1 Repository GE

The **reference implementation** of this GE$^{21}$ was originally developed by SAP and is currently owned by CoNWeT Lab (UPM). This is a *headless* component: it provides a simple REST-based web API for storing and sharing service and application descriptors on the cloud. In the FITMAN context, this GEi is also a key component of the Collaborative Asset Management Specific Enabler (FITMAN-CAM SE).

<table>
<thead>
<tr>
<th>GEi Scorecard</th>
<th>Site #1</th>
<th>Site #2</th>
<th>Site #3</th>
<th>Site #4</th>
<th>Site #5</th>
<th>Site #6</th>
<th>Site #7</th>
<th>Site #8</th>
<th>Site #9</th>
<th>Site #10</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of Installation</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>2</td>
<td>3</td>
<td>2,5</td>
<td>3</td>
<td>2,5</td>
<td>3</td>
<td>2,5</td>
<td>3</td>
<td>2,5</td>
<td>3</td>
<td>2,5</td>
</tr>
<tr>
<td>Engineering</td>
<td>3</td>
<td>n/a</td>
<td>3</td>
<td>2,5</td>
<td>3</td>
<td>2,5</td>
<td>3</td>
<td>2,5</td>
<td>3</td>
<td>2,5</td>
<td>3</td>
</tr>
<tr>
<td>Documentation</td>
<td>2</td>
<td>3</td>
<td>2,5</td>
<td>3</td>
<td>2,5</td>
<td>3</td>
<td>2,5</td>
<td>3</td>
<td>2,5</td>
<td>3</td>
<td>2,5</td>
</tr>
</tbody>
</table>

*Editor’s notes* – Installation and use scores from site #8 are mostly influenced by a perceived lack of proper documentation. Developers from site #9 didn’t assess the GEi’s source code.

<table>
<thead>
<tr>
<th>GEi Assessment</th>
<th>Implementation</th>
<th>No significant issues to report.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functionality</td>
<td>- The headless nature of this GE implies that all interactions happen by means of a REST client which, according to FIWARE’s architecture, is out of the scope the platform and belongs to the custom solution built on top of it. This approach, which is correct, leaves however something to be desired: basic administration of repository’s content (i.e., CRUD operations of files) is a <em>generic</em> task that might be easily accomplished through a <em>generic</em> web UI. This UI might be an optional component included in the standard distribution – similarly to the Price Simulation Tool in the Marketplace GE. The minimum set of operations supported should be the following: list collections, create collection, delete collection, list resources, create (upload) resource, retrieve (download) resource, delete resource. Note that we are aware of the existence of the HTML representation for collection and resource metadata in the current API, however in our opinion this kind of basic (read-only) hyperlinked browsing is not enough for admin purposes.</td>
<td>21 <a href="http://catalogue.fiware.org/enablers/repository-repository-ri">http://catalogue.fiware.org/enablers/repository-repository-ri</a></td>
</tr>
</tbody>
</table>
Going even further on the same line of the previous point, an integrated, web-based LinkedUSDL editor would be extremely useful. We are aware of the existence of an open source implementation from SAP (see [https://github.com/linked-usdl/usdl-editor](https://github.com/linked-usdl/usdl-editor)) – to the extent that we actually tested it, and besides some minor bugs we found it quite functional. The missing link is just a way of loading / saving LinkedUSDL files from/to the Repository GE. Reading the code it seems like this feature was indeed on the roadmap, but development is not active any more (last source code update on GitHub was on 02/12/2013). We think that, given the new ownership of the Repository reference implementation and the openness of the LinkedUSDL editor license (MIT-style), the latter might become an integral part of the former in some future release.

---

### 5.2 Marketplace GE

The implementation is **WMarket**[22](http://catalogue.fiware.org/enablers/marketplace-wmarket), originally developed by SAP and currently owned by CoNWeT Lab (UPM). This multi-faceted GEi merges together six collaborating but distinct sub-components: Offerings, Registration, Search, Review and Recommendation APIs, plus the Pricing Simulation Tool web UI. Overall, it provides basic but complete functionality for publishing and retrieving service/application offerings within a business ecosystem. In the FITMAN context, some of the source code of this GEi has been included into the Collaborative Asset Management Specific Enabler (FITMAN-CAM SE).

#### GEi Scorecard

<table>
<thead>
<tr>
<th>GEi Scorecard</th>
<th>Site #1</th>
<th>Site #2</th>
<th>Site #3</th>
<th>Site #4</th>
<th>Site #5</th>
<th>Site #6</th>
<th>Site #7</th>
<th>Site #8</th>
<th>Site #9</th>
<th>Site #10</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ease of Installation</strong></td>
<td>n/a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td><strong>Ease of Use</strong></td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2,66</td>
</tr>
<tr>
<td><strong>Engineering</strong></td>
<td>4</td>
<td>3</td>
<td>n/a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3,5</td>
</tr>
<tr>
<td><strong>Documentation</strong></td>
<td>4</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2,66</td>
</tr>
</tbody>
</table>

**Editor’s notes** – In the context of site #2, no installation of this GEi was required as a cloud-based instance was used instead. Installation and use scores from site #8 are mostly influenced by a perceived lack of proper documentation. Developers from site #9 didn’t assess the GEi’s source code.

#### GEi Assessment

<table>
<thead>
<tr>
<th>GEi Assessment</th>
<th>Site #1</th>
<th>Site #2</th>
<th>Site #3</th>
<th>Site #4</th>
<th>Site #5</th>
<th>Site #6</th>
<th>Site #7</th>
<th>Site #8</th>
<th>Site #9</th>
<th>Site #10</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Implementation</strong></td>
<td>No significant issues to report.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Functionality</strong></td>
<td>No significant issues to report.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>No significant issues to report.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Openness</strong></td>
<td>No significant issues to report.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Integration</strong></td>
<td>No significant issues to report.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Support</strong></td>
<td>No significant issues to report.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Editor’s comments** – It must be stressed that our developers did their experimentation on v2.2, while the current release is v3.2. According to the online documentation, the latest version

---

significantly improves over the v2.* branch – at least for what functionality is concerned. Also, the hard-coded configuration parameters problem seems to have been solved in WStore at the time of writing.

5.3 **ApplicationMashup GE**

The implementation is **Wirecloud** by UPM\(^23\). This GEi is a mashup\(^24\) platform for web applications, aimed at users without programming skills. Mashups integrate front-end widgets with back-end components (e.g., data sources), and are defined with drag-and-drop operations in a web-based configuration environment. Quoting from the FIWARE Catalogue: “*(Mashups) typically serve a specific situational (i.e. immediate, short-lived, customized) need, frequently with high potential for reuse.*” This component was not designed for being deployed by the end user, but rather as part on the FIWARE infrastructure – i.e., run by a platform provider. In the FITMAN context, the last point was not always relevant: each Trial site was also responsible for building their own FIWARE infrastructure, so developers often played also the role of platform providers. In this sense, FITMAN’s experimentation was able to assess this GE also from both the provider’s and the user’s point of view. Out of the six involved Trials, only one (\#2) ran the GE on a third-party cloud.

<table>
<thead>
<tr>
<th>GEi Scorecard</th>
<th>Site #1</th>
<th>Site #2</th>
<th>Site #3</th>
<th>Site #4</th>
<th>Site #5</th>
<th>Site #6</th>
<th>Site #7</th>
<th>Site #8</th>
<th>Site #9</th>
<th>Site #10</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of Installation</td>
<td>n/a</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ease of Use</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engineering</td>
<td>4</td>
<td>3</td>
<td>n/a</td>
<td>3</td>
<td>n/a</td>
<td>3</td>
<td>3.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Documentation</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3.66</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Editor’s notes* – In the context of site #2, no installation of this GEi was required as a cloud-based instance was used instead. Developers from sites #5 and #9 didn’t assess the GEi’s source code.

<table>
<thead>
<tr>
<th>GEi Assessment</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Generally speaking, this implementation fully meets the GE’s specification: it provides its users with the capability of quickly creating dashboard-like web application by assembling predefined components from a toolbox. Moreover, developing and integrating custom components proved to be an easy task for experienced programmers. That said, we also realized that UIs created this way might be less appealing compared with custom-built HTML/JavaScript applications, which can achieve (at least potentially) much better results in terms of interactivity, responsiveness, look &amp; feel and user friendliness. In our experience, application mashups are best suited for do-it-yourself rapid prototyping and for internal use within organizations: we would suggest that this somehow limited scope is made more clear in the GE’s description. [source: developers from site #2]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• It would be nice to have a way to share widgets without requiring a Marketplace + Store GE installation just for this purpose. In other words, there are scenarios beyond the <em>application/service business ecosystem</em> (e.g., application rapid prototyping) where the ApplicationMashup GE might still be very useful. In such scenarios, developers - typically belonging to different teams / BU's within the same large organization - could share their components by direct interaction. E.g., User A sends a packaged widget to User B via email; User B installs the widget on her own ApplicationMashup instance by clicking on an “upload and add to workspace” button on some admin page. [source: developers from site #3]</td>
</tr>
</tbody>
</table>

\(^{23}\) [http://catalogue.fiware.org/enablers/application-mashup-wirecloud]

\(^{24}\) In this context, a mashup is a composite web page which us dynamically assembled from multiple standalone UI components.
mechanism could be applied to widgets; however, toggling the visibility of page areas might seriously disrupt the page layout, so a tab-level ACL would probably be a better way of doing it. This added flexibility would significantly increase the GE’s scope beyond application rapid prototyping (see also the Implementation section above). [source: developers from site #10] #11

**Apps Recommendation – Fine tune for access right via ACL**

- In the present implementation, only static content (i.e., HTML, CSS, JavaScript code) is allowed in a widget, as the entire GE concept is about running the web applications on the client side. However, we would very much appreciate the support for some server-side web application engine, in particular PHP. [source: developers from site #9] #12

**Apps Recommendation – ApplicationMashup Support for some server-side web application engine (e.g. PHP)**

**Security**

- No significant issues have been reported. See also the Openness section below for some statements that are also related to Security.

**Openness**

- The GEi is based on OAuth2 for managing authorization, which is good for interoperability by and large. However, in the Enterprise world LDAP and ActiveDirectory are the by far the most common technologies. Direct support for these is missing in the current implementation. [source: developers from site #10] #13

**Apps Recommendation – ApplicationMashup Support for LDAP and ActiveDirectory**

**Integration**

- No significant issues have been reported. See also the Openness section above for some statements that are also related to Integration.

**Support**

- We contacted a couple of times the team supporting this GEi. Though they were offering some solutions the overall score for the support is poor: the contact point was not clear, we usually had redirection to other staff members, we didn’t manage to have a technical telco for fixing runtime errors and sometimes we had no answer of the problems we reported. [source: developers from site #10]

**Security**

**Openness**

**Integration**

**Support**

**Editor’s comments** – Despite the quite long list of reported issues and recommendations, the acceptance of this component was really good: adopted by six Trials, this is the most successful GE in FITMAN by far – not only in the scope of the Apps Chapter, but across the entire FIWARE Core Platform. The number of development teams that were actively involved in experimentation is the reason why so much feedback was collected. Most of the feedback is about requests for the extension of the Wirecloud GEi, adding features like server-side scripting, ActiveDirectory support, etc. The only complaint was about support, but it should also be noted that it came from one single team over six, while the remaining developers didn’t have any issue with respect to this.

### 5.4 LightSemanticComposition GE

The implementation is COMPEL by ATOS. The GE is a web-based platform for the design, execution and monitoring of Business Processes (BP). Based on open standards like Business Process Model and Notation (BPMN) 2.0, it enables the composition and orchestration of web services that are semantically annotated: business modellers can work autonomously from service providers, using an online repository as a common knowledge based where available web services are described and from which they can be “embedded” into an operational workflow. In the FITMAN context, this GEi is also a key component of the Collaborative Business Process Management Specific Enabler (FITMAN-CBPM SE).

**GEi Scorecard**

<table>
<thead>
<tr>
<th>GEi Scorecard</th>
<th>Site #1</th>
<th>Site #2</th>
<th>Site #3</th>
<th>Site #4</th>
<th>Site #5</th>
<th>Site #6</th>
<th>Site #7</th>
<th>Site #8</th>
<th>Site #9</th>
<th>Site #10</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of Installation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Ease of Use</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Engineering</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Documentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

**Editor’s notes** – Having feedback from one single Trial site makes this scorecard less significant than the others.
• In the current implementation, the XML file resulting from the BP design phase must still be manually edited in order to finalize some technical details of the executable process. This requires some non-trivial technical skills, so that the target user (the business modeller) is not entirely autonomous from developers, as was the main objective of this GE. A fully-automated design environment would be highly desirable. **#14 Apps Recommendation** - LightSemanticComposition GE Fully automated design environment

**Functionality**

• In order to have a “service task” to automatically invoke a web service, it is necessary to write some Java code. This is no different in similar BP execution engines on the market; however, in order to exploit the full potential of this component, a smarter way would be highly desirable. Our proposal is to add a new property editor in the BP design environment, capable of mapping process-level properties to service call arguments as defined in the WSDL descriptor of the target web service. To keep this task simple for the BP designer, this editor might support only “flat” argument lists – i.e., primitive scalar values – but it would be a big improvement anyway. **#15 Apps Recommendation** - LightSemanticComposition GE - to add a new property editor in the BP design environment, capable of mapping process-level properties to service call arguments as defined in the WSDL descriptor of the target web service

<table>
<thead>
<tr>
<th>Security</th>
<th>No significant issues have been reported.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Openness</td>
<td>No significant issues have been reported.</td>
</tr>
<tr>
<td>Integration</td>
<td>No significant issues have been reported.</td>
</tr>
<tr>
<td>Support</td>
<td>No significant issues have been reported.</td>
</tr>
</tbody>
</table>

**Editor’s comments** – A more general issue, as already mentioned in the Chapter’s introduction, is that this GEi is not officially endorsed by FIWARE anymore. This might not be a showstopper in FITMAN, given that the software – an extension of the Activity BPM platform – has an open source license and is owned by ATOS, which is a core FITMAN partner. However, we feel that bringing the GEi back under the FIWARE umbrella would improve its chances to achieve a higher maturity level.

### 5.5 Mediator GE

The implementation is Mediator_TI by Telecom Italia and Thales. The Mediator GE is an interoperability middleware providing mediation services. A mediation service is a chain of mediation tasks between a service producer and a service consumer, each task being a translation between two different data formats or protocols. In particular, the Mediator_TI GEi can expose a REST web service as a SOAP web service, publish legacy text-based messages as a web service with an XML payload, transform the XML payload of a web service. It is powered by well-known open source technologies like WSO2 Application Server and Apache Camel.

<table>
<thead>
<tr>
<th>GEi Scorecard</th>
<th>Site #1</th>
<th>Site #2</th>
<th>Site #3</th>
<th>Site #4</th>
<th>Site #5</th>
<th>Site #6</th>
<th>Site #7</th>
<th>Site #8</th>
<th>Site #9</th>
<th>Site #10</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of Installation</td>
<td>3</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3,5</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>3</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3,5</td>
</tr>
<tr>
<td>Engineering</td>
<td>4</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Documentation</td>
<td>3</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3,5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GEi Assessment</th>
<th>No significant issues have been reported.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation</td>
<td></td>
</tr>
<tr>
<td>Functionality</td>
<td>• Give the complexity of mediation services, runtime tracing is of paramount importance in order to spot problems and errors. In particular, payload tracing enables developer to <em>eavesdrop</em> on the conversation between mediation tasks, understanding how the different steps of a mediation service contribute to the overall result. With respect to this, the current implementation leaves something to be desired. We suggest that a server configuration option is added, enabling the</td>
</tr>
</tbody>
</table>

logging to a local file of the service payload of every call

<table>
<thead>
<tr>
<th>GE</th>
<th>Recommendation – Mediator GE - A server configuration option is added, enabling the logging to a local file of the service payload of every call</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>No significant issues have been reported.</td>
</tr>
<tr>
<td>Openness</td>
<td>No significant issues have been reported.</td>
</tr>
<tr>
<td>Integration</td>
<td>No significant issues have been reported.</td>
</tr>
<tr>
<td>Support</td>
<td>No significant issues have been reported.</td>
</tr>
</tbody>
</table>

Editor’s comments – Similarly to COMPEL, this GEi have been recently removed from the FIWARE Catalogue: same considerations apply here. We would also like to bring some attention to the fact that currently the Composition / Interoperability framework has one single GEi in the FIWARE Catalogue – i.e., Wirecloud – and this, in our opinion, undermines its ambition as “framework” in the first place.
6. Internet of Things (IoT) Services Enablement

The Internet of Things Service Enablement Chapter feedback will be included in D7.4 M30, due to on-going testing and experimentations of the IoT reference architecture and its evolution to the IoT for Manufacturing platform, synthesizing aspects of Smart-Digital-Virtual Factories of the Future and including additional GEs as the Data Visualisation – SpagoBI GE.
7. Interface to Networks and Devices (I2ND)

A primordial note that must be stressed is that I2ND architecture has changed radically from I2ND Release 3 (May 2015) to R4 (June 2015), indicating that it is yet an ongoing working project. Considering this, the document will make a brief overview of I2ND architecture until R3 and comment the evolution impact on FITMAN.

Nowadays, it’s commonly accepted that the main drivers that push the ICT community to new challenges are mobility, consumerization (BYOD), cloud, big data, and security.

The transformational shift that is occurring, creates impact over all the ICT agents, namely on Network Service Providers, because they have to adapt its resources to new traffic patterns: for example, the traditional client-server traffic (aka northbound traffic) is shifting to client-client traffic (aka southbound traffic) originated on collaboration services with strict QoS requirements like IP telephony, video-conferencing, desktop sharing.

The goal of I2ND is to provide an open and standardized platform to enable transparent access of heterogeneous devices to a variety of physical network, services and content, provided by a broad range of networks, services and applications.

To address the complexity of the problem, a modular architecture was designed integrating 4 FIWARE Generic Enablers:

**Cloud Edge** (CE) – The CE GE acts as a proxy between the home network and the cloud service provider.

**Connected Device Interfacing** (CDI) – The CDI GE is an add-on installed on the connected device that exploits its capacities, resources, content and contextual information to interact with network services and applications.

**Network Information and Control** (NetIC) – The NetIC GE pretends to provide a generic interface to control and manage open networks, independent of the technology or implementation.
By the time this chapter was written, there is an implementation of the NetIC GE publish at the FIWARE Catalogue as OFNIC (OpenFlow Network Information and Control). OFNIC is an open source reference implementation of the FIWARE NetIC Generic Enabler. It is based on the NOX, the first SDN controller, and relies on the Openflow protocol to send commands to network elements like OpenFlow switches. The last version of OFNIC (R3.3) was released on August 2014, however it was not used at any FITMAN trial. During the same period, other open source SDN Controllers arrived with drivers to main manufacturers.

Service, Capability, Connection and Control (S3C) – The S3C GE is the central control and management entity in I2ND. It provides an adaption layer between the targeted network control layer (ex, WiFi, WiMAX, UMTS LTE, SIP) and all possible applications and services. The S3C architecture integrates with OpenEPC (Evolved Packet Core), an open source software component to provide transparent convergence in fixed and mobile network through all-IP communications.

7.1 Experience from trials

As explained previously, FIWARE I2ND is being developed: the architecture has changed recently (along the evolution from Release 3 to Release 4). In this context, none of the four I2ND GE’s were used on trials.

7.2 Questionnaire analysis and recommendations

The objective of D7.3 is to express a set of recommendations to FIWARE based on experience gained on trials. However none of the I2ND GE were used until now, for the reason that there are no data to analyze the trial’s feedback and by consequence the V&V methodology cannot be applied in finding improvement actions.

However, considering the fundamental role of I2ND, and the need for a generic adaptation layer to abstract network idiosyncrasy, the I2ND platform should be present to play its function and serve other GE’s namely IOT, Cloud Hosting, Data and Security.

The response must arrive from within the consortium by steps:

- validate I2ND specification and architecture, at the light of general acceptance and stability of open source technologies in SDN and NFV,
- analyze the impact of its incorporation in the FIWARE ecosystem
- join efforts and set a deadline to implement I2ND functions as broad as possible into a viable term

#20 I2ND Recommendation : Include components for implementing SDN and NFV functionality

Being a core architectural component, enforce the implementation, test and deployment of the I2ND family of Generic Enabler, taking into account the current evolution of open source technologies in SDN and NFV.

By the time this chapter was revised, a brand new release of I2ND architectural design came to the light: R4 (June, 3, 2015). The architectural re-design encompasses 3 different domains:

- Software Defined Networks (SDN)
- Robotics
- Advanced integration middleware

I2ND R4 is implemented by 4 GE’s with prepared interfaces to two open source components (OpenStack and OpenDaylight) commonly accepted as de facto standard platforms:

- **Netfloc** (Network Flows for Cloud) – to build value-added services (e.g., QoS, Load balancers, resilient ad-hoc solutions) on private cloud networks
- **NetIC** (NETwork Information and Control) towards Open Networks
- **Robotics** – managing the communications, shared behaviours, synchronized actions and distributed intelligence of robots through robotics platform components and interfaces towards other FIWARE GEs
- **Advanced Middleware** – for developers to build highly efficient and secure distributed applications on top of broad networking technologies

![Diagram of conceptual interfaces between I2ND R4 chapter](image)

**Figure 4 - conceptual interfaces between the I2ND R4 chapter**

The interface to others GE’s has now support to “Cloud” and “WebUI” (being added to the already supported “IoT”, “Data” and “Security”).

As a global statement, and considering the current development status, there is a strategic redirect to consider and adapt to the new paradigms like Mobility and Cloud.

The built-in Interfaces to the commonly accepted OpenStack cloud management platform and to OpenDayLight (ODL), that is becoming a must to telco operators, are also very positive signs to FITMAN evolution.
8. Security

The FI-WARE catalogue security GEs have changed markedly over the period of the FITMAN trials experimentation. Firstly we describe FITMAN’s selection of security GEs relevant to the project and then the state of the security GEs now. The subset of the FI-WARE GEs chosen by the FITMAN project includes just two security GEs which are discussed in detail in sections 8.3 and 8.4 below. This is then followed by a proposal for an addition to the FIWARE Security Chapter and finally a brief summary of security issues related to the use of cloud resources which expands on recommendation #4 Cloud Hosting Recommendation #4.

Cloud Hosting Lesson Learnt: Set-up proper education and awareness actions for Manufacturing Industry CIOs, to clearly explain the fundamentals of cloud architecture, including its security risks and avoid false pre-judicial oppositions in Chapter 3.

8.1 FITMAN security GE selection

The FITMAN project conducted two evaluations of the FI-WARE security GEs, first around July 2013 (see first iterations of D1.2 and D1.3) and secondly around December 2013 (D1.2 and D1.3 second iterations).

At the time of the second evaluation, there were five security GEs in the catalogue of which two (DBAnonymizer and IdentityManagement) were selected by FITMAN for use in the project. A list of the GEs (the names of which, at the time were prefixed with “FIWARE.OpenSpecification.Security”) and a summary of the comments from D1.3 follow:

- **AccessControl**
  - This Generic Enabler is considered as out of scope for FITMAN purposes due to its lack of support for the open source version of IDM. It requires the use of the GCP GE (IDM Implementation) which is not going to be used in FITMAN (because the IPR terms are not acceptable).

- **Content-based Security & Compliance**
  - The GE has been designed to be used with other Optional Security GEs (DBAnonymizer, SecureStorageService, ContentBasedSecurity, MalwareDetectionService and AndroidFlowMonitoring). For FITMAN purposes, DBAnonymizer is the unique Optional Security GE that could meet FITMAN requirements. In FITMAN the DBAnonymizer does not need to be continuously monitored as it can work in an offline manner. From this point of view, it can be considered that C-bS&C is not suitable for FITMAN purposes.

- **DBAnonymizer**
  - This GE uses K-Anonymity algorithms to evaluate or propose anonymization policies. In FITMAN, the TRW trial needs to document compliance with Spanish and EU privacy policies. This GE can substantially contribute to assuring that the utilised anonymization algorithms and anonymised data actually comply with the privacy objectives of the trial. Any other trial that has to manage personal information and, to some extent, anonymize it can get benefits from using this GE.

- **IdentityManagement**
  - Federated authentication and authorization is a key functionality of a commercial-grade cloud environment. Any FITMAN cloud provider should benefit from the adoption of such a component. This downloadable, free and...
open source implementation from UPM (KeyRock) is actually required by the Cloud Portal GE.

- Privacy
  - This GE is more related for issues like online shopping, travel booking, e-banking, e-government, etc. where privacy authentication is a main topic. So, it can be considered as out scope for the manufacturing sector.

In summary, the new (“KeyRock”) IDM component from UPM (under the AGPL v3 licence) was considered useful for cloud providers and the DBAnonymizer (then under a BSD licence) useful for TRW.

8.2 Security GE evolution
The analysis for this deliverable was carried out in May-2015. At that point the FI-WARE catalogue contained five security GEs:

- Identity Management – KeyRock (as seen in Dec 2013)
  - Identity Management covers a number of aspects involving users’ access to networks, services and applications, including secure and private authentication from users to devices, networks and services, authorization & trust management, user profile management, privacy-preserving disposition of personal data, Single Sign-On (SSO) to service domains and Identity Federation towards applications.

- Authorization PDP – AuthZForce (formerly “Access Control” GE)

- PEP Proxy – Wilma
  - The PEP Proxy GE allows you to secure your back-end services adding authentication and authorization based on FIWARE account.

- Security Monitoring
  - Security monitoring is a suite of services for risk analysis, security visualization, decision making support and technical forensics.

- Content Based Security - CBS
  - Content Based Security (CBS) is an application layer security approach that provides a means to protect the confidentiality and integrity of information and to control access to it.

Compared to December 2013, just two GEs are the same: the Identity Management GE and the Authorization PDP (formerly Access Control). The other GE which was considered to be useful for FITMAN, the DBAnonymizer, is still available from Github but not from the FIWARE catalogue.

The Authorization PDP GE provides both a PDP (policy decision point) and a PAP (policy administration point) as defined in XACML. That is, it provides APIs for both making access control decisions according to a XACML policy and for creating and updating those policies. The APIs in this implementation are RESTful APIs. The XACML architecture separates the software for making authorisation decisions (the PDP) from the

---

28 FI-WARE catalogue, security section: http://catalogue.fiware.org/chapter/security
29 DBAnonymizer software (formerly a GE) on Github: http://fiware-security-sap.github.io/fiware-dba/
software/resource that is being protected and from the software that enforces those decisions, known as the PEP (policy enforcement point). The PEP effectively sits between the end-user and the software/resource being protected, intercepts the request and checks with the PDP whether the user is permitted or not.

The PDP GE provides Attribute-Based Access Control. This means control based on the following types of factor:

- **Subject attributes:** the subject is an actor (human, program, device, etc.) requesting access to a resource; attributes may be user ID, Organization, Role, Clearance, etc.
- **Resource attributes:** the resource is a passive entity (from the access control perspective) on which subject requests to act upon (e.g. data but also human, device, application, etc.); resource attributes may be resource ID, URL, classification, etc.
- **Action attributes:** the action is the action that the subject requests to perform on the resource (e.g. create, read, delete); attributes may be action ID, parameter A, parameter B, etc.
- **Environment attributes:** anything else, e.g. current time, CPU load of the PEP/PDP, global threat level, etc.

The PEP Proxy GE architecturally sits between an end user and a service and, through communicating with the PDP determines whether or not to let a request through to the service. The PEP Proxy is able to integrate with the IDM GE using OAuth2 to obtain the identity of the user trying to access the service or resource. It then passes that identity information to the PDP to determine whether to allow or deny access.

![Diagram](image.png)

*Figure 5. Architecture of a web application protected using the IDM ("FIWARE Account"), PEP and PDP. The flow implicit in the diagram starts with a request from the end-user to the “Web App” and then proceeds through the arrows from top to bottom.*
The combination of these three GEs is an improvement over the situation in December 2013 when FITMAN selected which GEs to use. At that time the Access Control GE mandated the use of a closed source IDM rather than the KeyRock solution and was on that basis rejected.

Taken together, the IDM, PDP and PEP now available from the FIWARE catalogue would be a set of GEs potentially of use to manufacturers deploying secure web services (though note comments below on the IDM).

The Security Monitoring GE was actually present in the catalogue at the time of the first FITMAN GE evaluation, was missing at the time of the second evaluation described above and is now back in the catalogue. When evaluated, the complexity of the FITMAN trials’ IT infrastructures was not considered great enough to justify the use of this component which analyses potential attack paths for complex deployments.

The Content Based Security component (from Thales) is a different piece of software to the “Content-based Security & Compliance” GE (from Atos) previously evaluated, although they have similar names. The current component provides services to protect (encrypt) and decrypt pieces of data and uses the Authorization PDP to determine who should be able to decrypt pieces of data based on their roles. This provides a complex and fine-grained data protection system that is unlikely to be applicable to the fairly simple data-protection scenarios encountered in manufacturing.

8.3 GE25 Security.IdentityManagement

The KeyRock Identity Management (IDM) GE from UPM provides an authentication service linked to the PDP and PEP GEs. According to the documentation\(^{31}\) it supports parts of the FIWARE IDM specification\(^{32}\): namely OAuth 2.0. In addition to the FI-WARE specification it supports the IETF SCIM 2.0 standard for simple cross-domain identity management\(^{33}\) which provides schemas for describing people and organisations.

Identity Management in general comprises a multitude of aspects involved with users’ access to networks, services and applications, including secure and private authentication from users to devices, networks and services, authorisation management, user profile management, single sign-on (SSO) to service domains and identity federation.

8.3.1. General points

The IDM GE is mostly an OAuth 2.0 service and this is proposed to be used for authentication by other web applications. This leads to the first issue which is (as stated on the OAuth website\(^{34}\)) that “OAuth 2.0 is not an authentication protocol”, it is in fact an authorization protocol.

---


\(^{33}\) SCIM: http://www.simplecloud.info/

\(^{34}\) OAuth website: http://oauth.net/articles/authentication/
The classic use case for OAuth is that of someone storing photos on a cloud service which is secured through some sort of login mechanism such as a username and password. They want to have some of the photos printed by a print shop and could give the print shop their username and password to let the shop log in and access the photos. Doing so would of course give the shop full control over the user’s account and so is undesirable. Instead, the user can use OAuth to delegate access to the print shop, authorising the shop to just be able to read the photos.

Many of us will also be familiar with the use of OAuth on Facebook when for instance an app requests access to your timeline or your public profile and keep that authorisation until you remove it. Again, this is preferable to giving an app your username and password (and therefore control over your entire account) once or repeatedly.

To use a web application which integrates with the IDM GE, a user must first create an account in the IDM using the IDM portal. This is done with a standard web-form where the user enters their name, email address and password and answers a CAPTCHA question (Figure 6. Registration form for the IDM GE.). They must also accept the FIWARE Lab terms and conditions. The IDM stores this information (with the password salted and encrypted) in a database. Once registered, the user is then able to log in to the IDM to manage their profile.

Upon logging in to the IDM portal with a web browser, the browser is given a “session cookie” which is held for a short time by the browser (in the standard way) so that the user can interact with the IDM portal for a while without repeatedly authenticating themselves.

For a web application to use the IDM as a means of authentication, the web app must also be registered with the IDM by an IDM administrator. Registering the application creates and displays what is called a “client ID” and a “client secret” which must then be added to the web app’s configuration so that it can later identify itself to the IDM and be trusted.

When a user comes to use the web app and the web app needs to authenticate them, the web app uses the OAuth protocol to communicate with the IDM and requests access to the user’s profile information (e.g. their name and email address). At this point, the user is redirected in their web browser to the IDM portal and if the user is not logged in (and therefore does not have a valid session cookie) then they are asked to authenticate themselves with their email address and password. Once they are authenticated the user is asked to allow the web app access to their profile information.

35 Where “OAuth” is written, “OAuth 2.0” can be assumed. The OAuth 1.0 spec is not generally used any more.
36 OAuth on Facebook: https://developers.facebook.com/docs/reference/dialogs/oauth
Assuming that the user permits the web app access, then technically at this point the web app will receive an “access token” and a “refresh token”. The web app is then free to use the access token to retrieve the user’s profile information at any time. This is the equivalent of an app on Facebook being able to access your profile information at any time without being authorised again and again. The access token will expire after a set time but then, through the OAuth protocol, the web app is able to use the refresh token to request a new access token. If, in the meantime, the user had decided to deny access to their profile from the web app then the web app would not be given a new access token.

Authenticating a user in this way is not ideal. Authentication is more associated with making sure that the user or entity interacting with an application is both who they claim to be and that they are actually present. Security is moving towards multi-factor authentication where “factors” are items from multiple classes: things the user knows (such as a password), things that the user owns (such as a mobile phone) and things that the user is (such as a fingerprint or other biometric). To confirm that the user is present then they are challenged to present one or more of these factors when they log in and again if there has been a period of inactivity or if they are requesting a sensitive operation (such as a transfer of money in an online banking application).

In contrast, the integration of the IDM with a web app requires single factor authentication at the point of the user permitting the web app access to their profile and subsequently does not require further confirmation that the user is in fact present. If it chose to do so, the web app would be able to refresh its access token indefinitely. More concerning is that there is nothing in the web interface (or the documented REST API) to show to a registered user of the IDM which web apps they have authorised and no way to remove a web app’s authorisation.
This leads to the following recommendations:

**#21 Security Recommendation:** Security.IdentityManagement GE - A mechanism should be added to the IDM portal interface and API so that a user can remove a web application from the list of authorised applications.

Ideally though, the IDM should move beyond this weak authentication mechanism and implement something more robust. A common way of doing this is to use OpenID Connect along with OAuth 2.0 as used by Google, Microsoft and Paypal amongst others. OpenID Connect is in fact part of the FIWARE IDM specification.

**#22 Security Recommendation:** The IDM should implement OpenID Connect to provide more robust authentication and consideration should be given to implementing two-factor authentication.

There are a variety of libraries that could be used to help with implementing OpenID Connect, listed at the OpenID website including a complete Python implementation of OpenID Connect and OAuth 2.0 (the IDM is implemented in Python).

The Google Authenticator or other open source client apps can be used to provide two-factor authentication (using an HMAC one-time password or “HOTP”) via a smartphone. There are many server-side implementations including, for instance, one in Python.

8.3.2. Experience from trials

The uptake of the Security.IdentityManagement (IDM) GE was low even though every trial expressed requirements for IDM.

Of course, while the low use of the GE limits the experience of FITMAN partners it is not necessarily a reflection of the GE’s quality: AugustaWestland for instance report that the “existing internal procedures and authorizations do this functionality”. The AIDIMA trial uses the built-in IDM (via Active Directory) of the Virtual Obeya system and so did not have any need to try the GE. Other trials may be at an experimental stage and more focussed on the main business requirements rather than IDM.

The IDM GE is provided on Github as a downloadable component with good installation instructions for local installations. It was as a local installation that the IDM was tried by some FITMAN partners.

The feedback from INNOVALIA (supporting the TRW trial) was very positive but does not result in any specific recommendations. It is summarised below:

1. Was it easy to install? *Rank 4, as it offers the one time login function as we needs.*
2. Was it easy to use? *Yes, the deployment process is clear.*

---

37 OpenID Connect: [http://openid.net/connect/](http://openid.net/connect/)
38 OpenID Connect libraries: [http://openid.net/developers/libraries/#connect](http://openid.net/developers/libraries/#connect)
39 Python OpenID Connect implementation: [https://github.com/rohe/pyoidc](https://github.com/rohe/pyoidc)
40 Google Authenticator: [https://en.wikipedia.org/wiki/Google_Authenticator](https://en.wikipedia.org/wiki/Google_Authenticator)
41 Python HOTP / TOTP library: [https://github.com/tadeck/oneetimepass](https://github.com/tadeck/oneetimepass)
3. How could we judge the quality of the GEs from a software engineering viewpoint?  
   Rank 4, it works as described.
4. Was the overall documentation complete and clear enough?  
   Rank 4, the documentation are complete and easy to access.
5. Proper support for security? Yes.
6. Was the implementation the best choice to accomplish to the GE goal? Yes.
7. Do you know of some non-FIWARE Open Source component that could have been used to implement this GE, or as the basis for its implementation? No.
8. Were interfaces supporting the right protocols/standards to make the integration easy or something was missing? Yes.
9. Did you find enough support (experts, call center, forum, …)? Yes.
10. Are there any further functionalities you would expect from the GE? No.
11. What is your overall experience? 
   It is very useful for our goal to realize the onetime login for different domains.

In contrast, Softeco (supporting the Piacenza trial) tried to use the IDM GE to integrate with the Application Mashup GE but ultimately could not use it because of some problems. An issue was opened by Softeco in Nov-2014 in the GE’s GitHub repository but was only (partially) answered in Apr-2015. The main points of the issue were:

1. Emails which should be sent to newly registered users were not being sent properly and therefore not received.
2. The content of the email (as seen in the logging) has links requesting the user accepts the “FIWARE Lab Terms and Conditions” and has “from” and “reply-to” fields pointing to the fi-ware.org domain.

Two recommendations result from Softeco’s experience:

**#23 Security.** Better documentation on how to configure the IDM software is needed (part of the issue raised with the GE developers was how to configure the email system which should have been covered by the documentation).

The GE contains messages and branding which relate to FI-LAB and FI-WARE, for instance:
- the main page in the local installation appear identical to the FIWARE Lab page with all the same links to other parts of the FIWARE site;
- a link to the FIWARE Lab Terms and Conditions (which will not apply for a stand-alone installation) is on the registration page;
- the email sent to a newly registered user has links and email addresses relating to FIWARE;
- the web pages generated by the software include Google Analytics code which reports back to the FIWARE account.

Of course, there is nothing wrong with messages stating that the software is a FIWARE component but in practice, when used in standalone mode as an authentication system, it is appropriate to identify the authentication system being used (i.e. who is checking the credentials) more prominently than the source of the authentication software. Therefore we have this recommendation:

---

#24 Security Recommendation: The IDM software would be more generally useful as a stand-alone component if most of the FIWARE branding, messages and analytics were removed or easily configurable.

Apart from these specific recommendations for the IDM GE, the experience of FITMAN demonstrates a need for FIWARE security GEs to support existing (“legacy”) systems deployed in manufacturing (and elsewhere). Both the AugustaWestland and AIDIMA trials mention using existing systems and it is unlikely that a manufacturing company has no IDM system already in place.

#25 Security Recommendation: Support for legacy IDM systems, i.e. ensuring that control points for authentication and authorization support tokens from existing legacy IDM, or that token exchange services are available to convert legacy IDM tokens so they can be understood by authentication and authorisation components protecting FIWARE-hosted resources.

The SAML 2.0 standard (as described in the IDM GE specification) can be used in some cases to let a user log in to a web application using a legacy Active Directory identity provider. The OneLogin SAML Python Toolkit 43 could be considered as it uses the same language as the IDM GE implementation.

8.4 GE26 Optional_Security_Enablers.DBAnonymizer

Database owned by organisations often have confidential and even sensitive personal data in them which an organisation must be careful not to disclose unintentionally. When providing data to a third party, databases are often anonymised by removing names or social security numbers, making postcodes or ages more course-grained and so forth. The DBAnonymizer GE can performs two main functions in this area:

1. evaluation of a database along with a disclosure policy (i.e. which fields to hide) to produce a re-identification risk score.
2. anonymization of a database according to a disclosure policy.

The software works by taking a MySQL database dump file and an XML policy file as input.

Ultimately, none of the FITMAN trials used the DBAnonymizer GE and so the extent that the project can make recommendations is limited. The table below shows which of the 10 trials had some connection to this GE:

<table>
<thead>
<tr>
<th>Trial</th>
<th>Expressed requirements</th>
<th>Tried the GE</th>
<th>Used the GE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VW</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRW</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

FhG IPK (supporting the VW trial) still consider the DBAnonymizer GE to be of “high interest” but as the decision was taken to use a private cloud in the VW trial, it was not felt necessary to further secure the data in the trial’s database by anonymizing it.

INNOVALIA (supporting the TRW trial) performed a quick evaluation of the component and found it to have an evaluation function for assessing the re-identification risk associated with the partial disclosure of a dataset but not the anonymization function required by the trial. The

43 Python SAML library: https://github.com/onelogin/python-saml
function is confirmed to be present in the component now, so this requirement has already been met – just not in time for the component to be actually used in the trial system.

8.5 **Enlarging the chapter**

As described above, the security chapter currently contains five GEs:

- a PDP (combined with a PAP), and a PEP which integrate with each other and with the OAuth 2.0 based IDM;
- a Security Monitoring GE;
- a Content Based Security GE for fine-grained data protection (through encryption).

FIWARE may wish to consider adding in the SAM tool\(^{44,45}\) used in FITMAN, which takes a model of a system (e.g. a set of objects within a computer program or a set of machines on a network) and attempts to verify certain security properties about the system, by exploring all the ways access rights can propagate through the system.

It is designed to handle dynamic systems (e.g. systems containing factories which may create new objects at runtime, and/or object interactions may arise dynamically at run-time using late service binding) and systems where behaviour of some of the objects is unknown or not trusted.

The modelling is done manually using a Java-like syntax to define the actors in the system, their knowledge/state and possible interactions. The model is then evaluated and explored using a GUI which displays a graph of the objects in the system and the references they hold to each other and highlights any unexpected possible invocations (indicating security breaches).

A typical approach is to model a system initially with most objects having defined (trusted) behaviour, and then to explore the effects of making them undefined or of adding additional actors with undefined behaviour into the system.

The model will tell you:

- what behaviours must be ensured for components you own;
- what behaviours you require of other parties you rely on.

Having a formal model (rather than simply relying on the programmers’ and administrators’ intuitions) is useful because:

- it reduces the chance of mistakes;
- it makes assumptions explicit.

For example, if a security property could be enforced by adding a restriction in either of two components being developed, each component developer might assume it would be added at the other point. Modelling the whole system forces us to make that choice and document it.

All the safety properties that are checked when building the initial system can be automatically rechecked when the system changes. When safety properties are checked manually when writing code (or deploying systems), changes to the system later can make the assumptions behind those checks invalid.

---

\(^{44}\) SAM tool: https://github.com/it-innovation/sam

\(^{45}\) SAM tool paper: http://eprints.soton.ac.uk/372445/
The FIWARE security chapter does include the Security Monitoring GE which superficially has some overlap with SAM, but whereas the Security Monitoring GE looks for vulnerabilities in deployed systems, the SAM tool has separate defined model which can be used at design time and to evaluate a deployment. With a correct model of a system, SAM can prove whether a vulnerability exists or not.

We therefore have another recommendation:

**#26 Security Recommendation.** FIWARE should consider adding the SAML tool to the security chapter to provide additional useful functionality.

### 8.6 Cloud security

Manufacturing companies and enterprises have legitimate concerns about placing data in cloud services which are not under their direct control. A manufacturer will expect to be the target of theft of trade secrets and intellectual property via corporate espionage.

Cloud systems broadly are of two types, both providing compute and service virtualisation: public clouds which have multiple tenants often sharing the same underlying physical resources, and private clouds which are generally hosted within the enterprise itself solely for their use but can also mean a segmented area of a third-party provider not shared with another organisation.

In FITMAN, we have two informative cases:

1. The high product data security policy of Volkswagen meant that it was not permitted to use the public cloud. Instead, it was necessary to set up and host the private cloud infrastructure at FhG IPK with access restricted to VW and IPK only. Furthermore, the security of the private cloud installation was tested by IPK to ensure that the GE and SEs used did not attempt to connect to external services.
2. The TRW trial which handled personal data relating to employees found that in order to comply with European and Spanish Directives related to data protection they could not even store the data in the TRW private cloud and instead stored the data locally in the TRW facility.

Both of these experiences, and the particular concerns of IP theft in the manufacturing sector lead to two final security recommendations which, if followed, could lead to more confidence in and higher utilisation of cloud systems (both public and private):

**#27 Security Recommendation.** For public clouds, the security features should be clearly enumerated to increase confidence and uptake. The data-protection and service management policies that the cloud service is compliant with should be described, for instance:

- ensuring effective governance and process:
  - ITIL service management practice
  - ISO/IEC 20000 for IT service management (with 20000-7 specific to cloud computing and 20000-11 in relation to ITIL both being written)
  - SSAE 16 audit standard for service organisations (an update of the older SAS 70 standard) oriented towards accountability and financial activities
- ensuring information security:
  - ISO/IEC 27000 series
- the location and legal jurisdiction of the service
whether an organisation’s virtual machines are deployed on physical hardware shared with other organisations or whether there is any form of physical isolation

#28 Security Recommendation. For software that can be used to create a private cloud and for public cloud systems, the security features should be clearly explained to increase confidence and uptake, for instance:

- whether it connects to any external services and if so, for what purpose
- the protocols being used to identify users (including federation with existing systems deployed in manufacturing):
  - Such as ADFS2, LDAP, SAML 2.0, OAuth 2.0, etc.
- the protocols being used to verify a service’s identity and secure data being transferred:
  - Such as HTTPS, SFTP or FTPS, VPN, etc.
  - The encryption technology being used
9. Advanced WebUI

Two FI-WARE GEs (3D-UI-XML3D and FiVES) were adopted from the Advanced Web-based User Interface (Web UI) chapter for the development of the two specific enablers 3D-Web-Viewer and DyVisual. Additionally to these two GEs we also give feedback on 5 additional GEs from the Web UI chapter.

9.1 3D-UI GE (3D-UI-XML3D)

XML3D is an extension to HTML5 for declarative 3D content represented as a scene graph like structure inside the DOM. All nodes within this graph are also nodes in the Web site’s DOM tree representation and can be accessed and changed via JavaScript like any other common DOM elements as well. On these DOM nodes, HTML events can be registered similar to known HTML elements. Resources for mesh data can be stored externally in any kind of external format (e.g. JSON, XML or binary) and referenced by URLs. XML3D is designed to work efficiently with modern GPUs and Graphics API (such as OpenGL/WebGL) but still tries to stay independent of the rendering algorithm. In addition to XML3D, Xflow allows to combine the scene graphs with dataflows. Xflow is a declarative data flow representation that was designed for complex computations on XML3D elements. These computations include for example skinned meshes and key frame animations.

<table>
<thead>
<tr>
<th>Trial</th>
<th>Expressed requirements</th>
<th>Tried the GE</th>
<th>Used the GE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VW</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TRW</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Whirlpool</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>AIDIMA</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

XML3D has been successfully tested and applied in the trials of VW, TRW, Whirlpool, and AIDIMA. In all cases 3D models are visualized using a standard Web browser. In some cases dynamic aspects are important which make use of the XFlow extension of XML3D. In the TRW use case a virtual avatar is animated to display poses which are ergonomically problematic.

Recommendations from the experience of the deployment of the GE in the FITMAN trials:

#29 WebUI Recommendation 3D-UI GE **Improve the documentation of specific features of XML3 and XFlow**: For non experts it is difficult to find out how specific features of XML3d/XFlow can be used in a specific application context. Improving the documentation would significantly speed-up the development of applications using XML3D/XFlow for system engineers who are not familiar with XML3D/XFlow.

#30 WebUI Recommendation : 3D-UI GE **XML3D should provide a standard binary format for assets**: XML3D provides XML encodings for meshes and assets and JSON and binary encodings (based on Blast) for meshes. The xml3d.js implementation provides a plug-in system to register additional mesh or asset formats. However, there is no general recommended way to encode assets binary, favourably in a streamable Blast container. It would help to have - similar to the established XML and JSON encodings - a standard binary format for assets.
#31 WebUI Recommendation: 3D-UI GE XML3D should provide a way to pick generic vertex attributes: XML3D has a generic vertex attribute system, which we can use to map application data to the GPU, for instance deviations from laser scans. Thus, the application data can be mapped to colours on the GPU, either via custom GPU shaders or using shade.js. However, in the opposite direction (derive the application data from the rendering), pick functionality is only available for vertex attributes with a fixed semantic (position, normal) and for the fragments’ colours. We can use the colours to inverse the mapping functionality; however, this operation suffers from information loss for discrete mappings. If possible, it would be helpful to offer a generic way to pick vertex attributes.

The following table summarizes the evaluation of XML3D according to the FITMAN V&V methodology technical indicators for software components:

<table>
<thead>
<tr>
<th>Technical Indicator</th>
<th>Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correctness</td>
<td>High</td>
<td>No defects were detected in the application of the GE.</td>
</tr>
<tr>
<td>Ease of Application</td>
<td>Level 2</td>
<td>Amount of effort involved depends on the complexity of the application.</td>
</tr>
<tr>
<td>Efficiency</td>
<td>High</td>
<td>Efficiency was well-suited for the selected applications.</td>
</tr>
<tr>
<td>Interoperability maturity</td>
<td>Level 3</td>
<td>XML3D builds on WebGL which is an accepted standard.</td>
</tr>
<tr>
<td>Openness</td>
<td>Level 2</td>
<td>XML3D is available open source.</td>
</tr>
<tr>
<td>Reliability</td>
<td>High</td>
<td>No problems were experienced in the prototype development.</td>
</tr>
<tr>
<td>Sustainability</td>
<td>Medium</td>
<td>Modifications need expert knowledge. However, there is community which can give support.</td>
</tr>
</tbody>
</table>

9.2 Synchronization GE (FiVES)

DFKI’s implementation of a synchronization server (FiVES) was adopted for FITMAN. It is used for synchronising the views of different users, which might reside at different geographical locations. The visualisation infrastructure combines several GE implementations, with FiVES as multi-user synchronisation framework, including KIARA as transport layer and XML3D as web-based 3D visualisation approach. Besides browser-based clients (e.g. Firefox or Google Chrome) other client frameworks can be supported by implementing the required client plugins.

<table>
<thead>
<tr>
<th>Trial</th>
<th>Expressed requirements</th>
<th>Tried the GE</th>
<th>Used the GE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VW</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TRW</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Whirlpool</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>AIDIMA</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

The synchronization server FiVES was used to allow multiple users to view simultaneously models and animations in a synchronized manner in the VW, TRW, Whirlpool, and AIDIMA trials. For the TRW use case a group of experts might analyse specific poses displayed by the avatar. In the Whirlpool case deviation maps might be analysed by a group of experts.
Application of FiVES in the use case trials is from an end user point of view straightforward and without any complications. For system engineers, developing applications that use FiVES significant effort is necessary to get familiar with the GE. However, this is mainly true in combination with XML3D because the XML3D clients need to be adapted with respect to the synchronization of dynamic aspects of the scene when animations are concerned. Synchronization of modifications to the scene which are not animated basically come for free. However, if the manipulations of the scene exceed the features which directly supported by FiVES significant development effort is necessary.

Recommendations from the experience of the deployment of the GE in the FITMAN trials:

#32 WebUI Recommendation: Synchronization GE Improving the bi-directional synchronization of XML3D scenes with FiVES:
As stated in the previous section, extending the feature set of a FiVES application currently needs expert knowledge on both server and client side. It is in particular not possible to set up a synchronized 3D scene without implementing client logic in JavaScript. This is also reflected in the provided FiVES software repository by the number of additional JavaScript files that are provided as plugins. While this approach can be considered as a good way to extend the functionality, it makes it hard for a developer who starts working with FiVES to capture a complete image where in the code which part of the FiVES feature set is implemented.

More specific, this also relates to most basic functionality like scene setup. Adding objects to a synchronized XML3D application is currently only provided in terms of a JavaScript API that demands the developer to create FiVES objects which provide client code that turns into XML3D objects. The exact way in which this is achieved is not directly transparent for the developer.

#33 WebUI Recommendation: Synchronization GE Add XML3D node annotations to mark which parts of the scene will be kept in synch by FiVES directly in the XML3D scene.
This feature would bring two benefits: Firstly, a developer could define the XML3D scene beforehand in the usual XML3D mark-up. By this, a FiVES client could make use of the advantages of the declarative approach of XML3D, which is currently hidden behind the dynamic JavaScript-based scene creation. Secondly, the declarative approach would transfer to the FiVES synchronisation mechanism. Instead of subscribing to FiVES scene updates and handling them programmatically in JavaScript code, a developer could declare directly in the 3D scene mark-up which parts of the scene graph, and with that, which node attributes in particular, should be kept in synch.

The following table summarizes the evaluation of FiVES according to the FITMAN V&V methodology technical indicators for software components:

<table>
<thead>
<tr>
<th>Technical Indicator</th>
<th>Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correctness</td>
<td>High</td>
<td>No defects were detected in the application of the GE.</td>
</tr>
<tr>
<td>Ease of Application</td>
<td>Level 2</td>
<td>Amount of effort involved depends on the complexity of the application.</td>
</tr>
<tr>
<td>Efficiency</td>
<td>High</td>
<td>Efficiency was well-suited for the selected applications.</td>
</tr>
<tr>
<td>Interoperability maturity</td>
<td>Level 2</td>
<td>To our knowledge not international standards exist.</td>
</tr>
<tr>
<td>Openness</td>
<td>Level 2</td>
<td>FiVES is available open source.</td>
</tr>
<tr>
<td>Reliability</td>
<td>High</td>
<td>No problems were experienced in the prototype development.</td>
</tr>
</tbody>
</table>
9.3 GIS Data Provider - Geoserver/3D
The GIS Data Provider GE provides terrain and building data in various formats. It delivers geospatial data compliant with the 3D-UI-XML3D GE to be used in location related applications, such as Smart City applications and citywide games. The overall impression of the GE is satisfactory. Its implementation extends the GeoServer, which is a well-established open source software with a large community and thus can be considered mature. It is providing the essential, but not yet complete, set of functionalities and good documentation. The GE capabilities are utilized for mixed reality applications in citywide gaming scenarios incorporating also the Synchronization FiVES GE.

#34 WebUI Recommendation: GIS Data Provider - Geoserver/3D
We suggest to improve the GE to become feature complete (e.g. support for re-projection of CRS (coordinate reference systems) is crucial for most applications and is still missing). Moreover, the further development should be aligned with existing OGC standards, such as the 3D Portrayal Service.

9.4 POI Data Provider / Publish/Subscribe Context Broker - Orion
The POI Data Provider GE (POIDP GE) specifies an extensible and standardized format for describing points of interest as well as service interfaces to query them. However, in terms of functionality the GE performs rather poor. It introduces an odd REST dialect, specifies a return format with no capability for ordering, the API provides no capability for pagination and no out of the box support for social interactions on POIs (likes, ratings, comments, check-ins). Furthermore, authentication is not included in the specification. Finally, the provided reference implementation of the POIDP GE is PHP based and rather amateurish due to a low code quality, erroneous installation and setup, weird sample data, and weak unit tests. In addition to the POIDP GE, FIWARE offers the Publish/Subscribe Context Broker GE that covers very similar use cases with a much more mature implementation. Considering this, the FIWARE POI WG currently elaborates how to converge to an enhanced common approach including an API, data model and transfer format.

#35 WebUI Recommendation: GIS Data Provider - Geoserver/3D
We suggest to strengthen the efforts of the POI WG, which specifically targets the usage of enablers across chapters and thereby nicely advances towards the overall vision of FIWARE. Moreover, it covers a hot topic of Phase III participants. Depending on the future work of the WG, we expect a redefinition of the POIDP GE and a revision of the Publish/Subscribe Context Broker GE.

9.5 Augmented Reality
The Augmented Reality (AR) GE provides a high-level service to applications that implements typical AR functionality. Deploying this GE in an application is rather hard and results in a huge overhead of dependencies since the implementation is a cross-compiled C/C++ library (using Emscripten). Thus, the implementation seems to be neither maintainable nor improvable in terms of adding further tracking approaches. The inactivity in the respective repository gives strong support to this impression.

Although the provided GE is not really valuable, the general AR GE topic is highly important and relevant, especially given the high interest in AR today. A better implementation of this AR GE would be highly desirable.
#36 WebUI Recommendation: Augmented Reality

We strongly suggest to revisit this GE implementation and rectify the mentioned issues, making it more maintainable and improvable utilizing an open development platform.

9.6 Virtual Characters

This GE is supposed to provide a basic infrastructure that enables applications to easily create, animate, and interact with virtual characters. Although the technology is mature and widely used as it is a part of the realXtend framework, the provided implementation is not compatible with the 3D-UI-XML3D GE, which we consider as the core element of the WebUI chapter.

WebUI Recommendation #9: We suggest to reconsider the intention of this GE and align its implementation with the overall aim of the WebUI chapter.

9.7 Advanced Middleware (legacy)

The Advanced Middleware GE provides a convenient interface across different platforms and architectures to perform efficient, scalable, and secure communication between software components and is used within the Pervasive Games Platform. The provided libraries are used to update game settings and 3D scenes in multiplayer scenarios across different client platforms in conjunction with the Synchronization FiVES GE.

Support for a generic binary protocol is missing, although it could be easily added due to the extensible design of the enabler. The documentation is good but hard to find since this GE implementation is meanwhile no longer advertised via the FIWARE catalogue.

This GE is undergoing a complete reengineering and thereby seems to drop crucial features of the reference implementation, such as the support of different platforms and architectures (except Java). This new version of the GE will not be of much interest for new FI applications. Its functionality is readily available already in Java (actually several similar implementations are broadly available).

#37 WebUI Recommendation: Advanced Middleware (legacy) - We suggest to reconsider the strategic alignment of this GE and its implementation. We strongly encourage FI-WARE to spend efforts in continuing previous work to ensure the provided reference implementation remains well integrated with the relevant WebUI enablers and provides support for more than one platform (in particular C# and JavaScript).
10. FI-PPP Recommendations

To be Released in the M30 deliverable D7.4.
11. Annex

In this sections are reported the methodology utilized for data collection (coming from WP2 and based on experiences gained in D7.1) and the results collected via questionnaires.

11.1 Data Collection and Data Analysis methodology

In this sections are reported the methodology utilized for data collection (coming from WP2 and based on experiences gained in D7.1) and the results collected via questionnaires. Data Collection and Data Analysis methodology describes how information for the Deliverable D7.3 has been collected and what have been the sources.

Section 11.1.1 describes what kind of information has been collected as a part of deploying the FITMAN V&V Methodology. The focus is on Technical Indicators for software components.

In section 11.1.2 GEs Assessment it is presented the structure of the questionnaire about GEs distributed to the IT providers of the 10 trials. The individual answers are listed in 11.2 Answers to GEs Questionnaires. The questions are grouped and consolidated answers to questions related to GEs are grouped according the following sequence, according with the “by Chapter” according the FIWARE define grouping:

1. Applications/Services Ecosystem and Delivery Framework;
2. Cloud Hosting;
3. Internet of Things (IoT) Services Enablement & Data/Context Management;

In section 11.1.3 SEs assessment are grouped questions related to the SEs identified in the first round and from Open Calls. Here are requested to answer both the adopters in the trials and the SEs owners.

11.1.1 FITMAN V&V Method contribution for recommendations to FIWARE

The FITMAN V&V method offers significant information to be utilized for the recommendations to FIWARE. Relevant information is mainly drawn from assessing the technical indicators for the software components and from the technical trial journals.

11.1.1.1 Technical indicators

In the FITMAN V&V method there are two kinds of technical indicators (see FITMAN WP2 deliverables): for the software components and for the trial solutions. For recommendations to FIWARE, some valuable information can be drawn from the results of assessing the Generic Enablers (GEs) and the Specific Enablers (SEs) used and elaborated in FITMAN. The purpose of the assessment has been to obtain information concerning e.g. functionalities and features of the components. Technical indicators, their definition and the assessment scale are presented in the following table.
## Table 6 FITMAN V&V methodology technical indicators for software components

<table>
<thead>
<tr>
<th>Technical Indicator</th>
<th>Definition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Correctness</strong></td>
<td>The degree to which the software is free from defects in its specification and implementation</td>
<td><strong>High</strong> (no defects detected) &lt;br&gt; <strong>Medium</strong> (relatively minor defects detected) &lt;br&gt; <strong>Low</strong> (severe defects detected)</td>
</tr>
<tr>
<td><strong>Ease of Application</strong></td>
<td>A measure of the applicability of the software in the particular environment in terms of amount of work and extra actions or means</td>
<td><strong>Level 0</strong>: No applicability in our environment without extra applying actions or means &lt;br&gt; <strong>Level 1</strong>: Applicable with significant amount of work &lt;br&gt; <strong>Level 2</strong>: Applicable with some amount of work &lt;br&gt; <strong>Level 3</strong>: Easily applicable in our environment</td>
</tr>
<tr>
<td><strong>Efficiency</strong></td>
<td>The capability of the software to provide appropriate performance, relative to the amount of resources used</td>
<td><strong>High</strong>: (performance beyond expectations, with reasonable resources consumption) &lt;br&gt; <strong>Medium</strong>: (expected performance, even with small increase of resources utilization) &lt;br&gt; <strong>Low</strong>: (Marginally accepted performance and/or overuse of available resources)</td>
</tr>
<tr>
<td><strong>Interoperability maturity</strong></td>
<td>The capability of the software to interact with other systems</td>
<td><strong>Level 0</strong>: Isolated Approach (No API exposing the GE / SE functionalities) &lt;br&gt; <strong>Level 1</strong>: Baseline Unified Approach (International Standard exists) &lt;br&gt; <strong>Level 2</strong>: Open Unified Approach (No International Standard exists) &lt;br&gt; <strong>Level 3</strong>: Standardized Integrated Approach</td>
</tr>
<tr>
<td><strong>Openness</strong></td>
<td>The degree to which developers may access the software for free with specified rights</td>
<td><strong>Level 0</strong>: Open specifications – Developers can view &amp; study the requirements posed and implement them as they wish &lt;br&gt; <strong>Level 1</strong>: Enablers as a Service – Developers can utilize software provided as a service through open interfaces &lt;br&gt; <strong>Level 2</strong>: Releasing code as open source - Developers can inspect, download, run and improve the open source code according to their needs. &lt;br&gt; <strong>Level 3</strong>: Consulting with the use cases about their needs and collaboratively contributing to the source repository, design documents, and bug reports</td>
</tr>
<tr>
<td><strong>Reliability</strong></td>
<td>The capability of the software to maintain a specified level of performance when used in the factory settings</td>
<td><strong>High</strong>: (totally stable performance) &lt;br&gt; <strong>Medium</strong>: (marginal performance divergences) &lt;br&gt; <strong>Low</strong>: (the performance of the software varies significantly depending on the conditions)</td>
</tr>
<tr>
<td><strong>Sustainability</strong></td>
<td>How easy it is to maintain the software and apply required modifications for adapting to changes in the operating environment</td>
<td><strong>High</strong>: (software maintenance and modification are very easy) &lt;br&gt; <strong>Medium</strong>: (software maintenance requires reasonable effort) &lt;br&gt; <strong>Low</strong>: (software maintenance is very complex and modifications difficult to be applied)</td>
</tr>
</tbody>
</table>
11.1.1.2 Trial Journals

Purpose of the trial journals in the FITMAN V&V methodology is to collect unstructured data from the trials’ instantiation processes. The trial journal includes two parts: business and technical journal. Especially the technical journal offers valuable information for recommendations to FIWARE. The technical journal includes e.g. information concerning implementation issues encountered in the implementation of the trial system, and information of the operational resilience of the Trial (e.g. major bugs, blocking errors, etc.).

Each trial has written important information in the trial journal. The following types of contribution can be found in the trial journal to be utilized in recommendations:

- Practical description of implementation phases and reported main observations in each phase
- Observations related to the components, their practicality and adaptation needs
- Reporting of experienced challenges and problems with different components, usually including also solutions that have been found, often also with effort of several stakeholders
- Problems related to the documentation and support of the components, and some related recommendations
- Observed issues related to various subjects, e.g. security, and description of how these issues have been solved
- Description of the most important operational issues to be managed related to the components and their use (e.g. related ethical issues)
- Some trials have written lessons learnt observations, which in many cases can be translated into recommendations
- Good benchmarks of some trials of e.g. overcoming challenges
- User experiences in general.

The data from trial journals are reported in deliverable D7.1 FITMAN Smart-Digital-Virtual Factory Trials Experiences release at M24.

11.1.2. GEs Assessment

The questions related to GEs are grouped according the following sequence, according with the “by Chapter” according the FI-WARE define grouping:

1. Applications/Services Ecosystem and Delivery Framework;
2. Cloud Hosting;
3. Internet of Things (IoT) Services Enablement & Data/Context Management;

For each of the four chapters a preliminary assessment is focus on understanding at what level the GEs of the chapter have been utilized, if they have been utilized as a collaborative set of functions or as specific component. These are the questions:

Chapter Assessment Header

1. Yes: my experimentation involves two or more cooperating GEs from the same chapter → please go to Chapter Feedback and then please assess each GE you utilized;

2. Yes: the GEs have been integrated into my experimentation site as standalone GEs components → please go to the sections of utilized GEs of this chapter;
3. No: **I have not been using any GE from the chapter at all** → please go through each GE in this chapter and articulate the reason you do not use GEs.

In case of positive answer to question 1, the GEs have been utilized as organic set of components belonging to the chapter. In such case the questionnaire investigates how the Chapter has a whole was utilized. The section called Chapter Feedback contains the following questions:

**Chapter Feedback**

Provide your feedback about the experience in using the chapter as a whole.

- Is the platform self-consistent?
  - Yes
  - No: how is it not self-consistent?
- Is the platform complete in terms of functionality?
  - Yes
  - No: what is missing in terms of functionality?
- Did they select the right GEs?
- Did they forget any de-facto or de-jure standard diffusely adopted in the community?
- Did they forget any key technology/application domain?

After the initial assessment of the Chapter, a detailed set of questions for each GE is asked. In case GEs have been utilized, only the questions for them is required. If no GE of the chapter has been utilized, a survey for each GE of the chapter is required. The reason for that is because we need to really understand why none of the proposed GEs were suitable and how the platform can be updated to make it compliant with the users’ needs.

Here following the set of questions asked for each GE:

Did you use this GE?

- Yes, we are using it now;
- No, we tried to but did not use it in the end (please explain what was the problem and what you used instead);
- No, we did not try it, we used something else for this functionality (please specify why you took this choice and what you used);
- No, we did not need this functionality.

*(If the answer is Yes please go next questions; if No, please jump to the next GE in the Chapter after giving rationale for the choice)*

Provide your feedback about the experience in **using** this GE.

1. Was it easy to install? Rank from 1-Low to 4 High + Comments
2. Was it easy to use? Rank from 1-Low to 4 High + Comments
3. How could we judge the quality of the GEs from a software engineering viewpoint? Rank from 1-Low to 4 High + Comments
4. Was the overall documentation (the GE open specs and the implementation's manual) complete and clear enough? If not, please elaborate on your failed expectations. Rank from 1-Low to 4 High + Comments
5. As far as your platform is concerned, do you find that this implementation comes with proper support for security (i.e., access control, data confidentiality)? In not, please elaborate on your failed expectations.
6. Was the implementation the best choice to accomplish to the GE goal?
7. Do you know of some non-FIWARE Open Source component that could have been used to implement this GE, or as the basis for its implementation? If yes, please elaborate on the expected advantages of the alternate solution(s).
8. Where interfaces supporting the right protocols/standards to make the integration easy or something was missing?
9. Did you find enough support (experts, call center, forum, …)?
10. On the basis of your experience in developing the solutions for the Trials and of your general knowledge of the FI and ICT for Manufacturing domain, are there any further functionalities you would expect from the GE?
11. What is your overall experience in integrating this GE with your Trial's platform (i.e., legacy systems / Trial-specific components / other GEs)?

The answers to the questionnaires, compiled by technology partners that assessed/utilized available GEs are reported in attached FITMAN D7.3 GE-Questionnaires.docx.

11.1.3. SEs assessment
Aiming at collecting in a structured way the experiences of Specific Enablers’ owners who built their enablers on the FIWARE Generic Enablers (GEs) or developed new Specific Enablers (SEs) from scratch in order to cover specific needs of the manufacturing domain that were not addressed by FIWARE, a questionnaire was structured. The questionnaire comprised of a specific number of questions that needed to be filled in order to capture the developers’ experiences of the FIWARE GEs.

- questions needed to be completed, where the answer was given in the form of ranking from 1-Low to 4-High.
- The stakeholders were also asked to provide additional comments if needed, in order to complement their experience in using each GE.

The questionnaire given to Specific Enablers owners is described in detail below:

1. In case your SEs builds on top of GEs, please provide your feedback about the experience in using each GE. (Please repeat the questions for all GEs you have reused in your SE)
   I. Was it easy to install? [Rank from 1-Low to 4 High + Comments]
   II. Was it easy to use? [Rank from 1-Low to 4 High + Comments]
   III. How could we judge the quality of the GEs from a software engineering viewpoint? [Rank from 1-Low to 4 High + Comments]
   IV. Was the overall documentation (the GE open specs and the implementation's manual) complete and clear enough? If not, please elaborate on your failed expectations. [Rank from 1-Low to 4 High + Comments]
   V. As far as your platform is concerned, do you find that this implementation comes with proper support for security (i.e., access control, data confidentiality)? In not, please elaborate on your failed expectations.
   VI. Was the implementation the best choice to accomplish to the GE goal?
   VII. Do you know of some non-FIWARE Open Source component that could have been used to implement this GE, or as the basis for its implementation? If yes, please elaborate on the expected advantages of the alternate solution(s).
   VIII. Were the provided interfaces supporting the right protocols/standards to make the integration easy or something was missing?
IX. Did you find enough support (experts, call center, forum, …)?
X. On the basis of your experience in developing your SE and of your general knowledge of the FI and ICT for Manufacturing domain, are there any further functionalities you would expect from the GE?

11.2 Answers to GEs Questionnaires
Please refer to “FITMAN D7.3 GE-Questionnaires.docx” for the details of the answers to the questionnaire from IT partners working with GEs.

11.3 Answers to SEs Questionnaires
Please refer to “FITMAN D7.3 SE Questionnaires-v2.docx” for the details of the answers to the questionnaire from IT partners providing SEs.