



D32.3
FI Platform Federation first prototype
M18 Issue

Document Owner:	Jerry Dimitriou – SingularLogic
Contributors:	Markus Heller – SAP
Dissemination:	Confidential
Contributing to:	WP32
Date:	26/03/2013
Revision:	1.0

Table of Contents

1	EXECUTIVE SUMMARY	3
2	INTRODUCTION TO THE PROTOTYPE MODULES AND COMPONENTS.....	3
2.1	FI-WARE ASSETS RELEVANT WITH THE FI PLATFORM FEDERATION SERVICES.....	3
2.1.1	<i>USDL / Linked USDL</i>	3
2.1.2	<i>FI-WARE USDL Repository</i>	4
2.1.3	<i>FI-WARE Marketplace</i>	4
2.1.4	<i>Consumer Marketplace.....</i>	5
2.1.5	<i>IoT Manager.....</i>	5
3	PROTOTYPE MODULES	7
3.1	CONSUMER MARKETPLACE	7
3.2	IOT MANAGER.....	7
4	SUMMARY AND CONCLUSIONS	8
5	REFERENCES.....	8

1 Executive Summary

MSEE WP32 stands to gain from integrating and federating with existing components and services offered by other Future Internet-inspired platforms through reusing, assembling and possibly extending parts of their offerings towards satisfying the needs of the MSEE Use Cases and maximizing the project's applicability and coverage in terms of business domain functions. This integration and virtual federation also allows utilisation, extension and improvement of other ecosystems' services

In this context, deliverable **D32.3 – FI Platform Federation first prototype** aims to depict the current status on the development of the first prototypes of the WP32 "FI Platforms Federation" based on the specification included in the document D32.1 issued at M12 and their relation with other FI-WARE assets like the FI-WARE Marketplace and the FI-WARE USDL Repository, which are described in section 2.1.

The developed modules of the MSEE FI Platform Federation Services developed are the Consumer Marketplace and the IoT Manager which are described in section 2.2.

In section 3, an overview of the modules of the developed services, with their functionalities, is presented while in section 4 a short summary and conclusions is included.

For further information, refer to D32.1 [3] and to the factsheets, provided as annexes of this document, as shown in section 5.

2 Introduction to the prototype modules and components

This section, after a general introduction to the FI Platform Federation services and the FI-WARE assets that are relevant with the services, introduces the software components that have been developed in WP32, that are described in details and from a technical point of view in [1], [2].

2.1 FI-WARE Assets relevant with the FI Platform Federation Services

The *FI-WARE Business Framework* contains a set of GE enablers to support electronic commerce in the Internet ([8]).

In the sequel, we summarize briefly two core parts of the business framework, namely the USDL Repository and the Marketplace GE. For both GE, there are reference implementations available which can be downloaded and used.

2.1.1 USDL / Linked USDL

Two core elements of the FI-WARE Business Framework are the Repository and the Marketplace. Both support the maintenance of USDL models described in *Linked USDL* specification and the trading of services between service providers and service consumers ([8]). The repository and the marketplace introduced later are used to store service descriptions in USDL format.

USDL is a platform-neutral specification language to describe important metadata of services in the form of service descriptions. **Linked USDL** is the second version of the USDL language

which builds on Linked Data principles. The work is supported by several interested companies stemming from earlier USDL efforts within a W3C USDL Incubator Group.

Linked USDL contains a core vocabulary “Linked USDL Core” to describe the most important parts of a service description. Additionally, a set of vocabularies is available for Linked USDL, like Pricing Vocabulary, Security Vocabulary. Using this modeling language, relevant information pieces for a service description can be modeled. Typically, a single file contains one service description modeled in USDL (but other possibilities also are possible).

Within MSEE, support for Linked USDL has been provided by SAP. In this frame, Linked USDL sessions were given to interested MSEE members in telecom sessions on 23.07.2012 and 27.02.2013.

2.1.2 FI-WARE USDL Repository

The FI-WARE USDL Repository ([6], [11]) is used to store service models into a database and is a central location to maintain service descriptions. Service description models in USDL can be stored into the repository, searched for, viewed, and downloaded from the repository. The USDL Repository is defined as a FI-WARE GE and constitutes a part of the Business Framework. A reference implementation is available from FI-WARE ([11]).

A detailed description of the repository GE API is found at [9] and [10], all artifacts in the repository are denoted by unique UR, structure based on a REST style protocol. HTTP content negotiation is possible with e.g. HTML, RDF, and JSON content negotiation types. The repository support RDF input and output types to a certain extent described at [7].

As **data structure**, core objects are maintained, being either a resource or a collection of resources. Resources denote information objects, like a service description (file) for a single service. Collections serve as containers for resources.

The *main operations* for the Repository are: (1) Managing Resources, (2) Managing Collections, (3) Listing Content, (4) Listing Services, (5) Searching the Repository, and (6) Querying the Repository ([see 4]).

2.1.3 FI-WARE Marketplace

The FI-WARE Marketplace ([7],[12]) supports internet-based electronic commerce where multiple provides of a good offer these on a central hub and interested consumers can locate them there and buy them for consumption. Within FI-WARE, the focus is on advertising services and trading them between providers and consumers on the Internet.

The FI-WARE Marketplace is a GE of FI-WARE, and a reference implementation is also available at [12].

The main *data structure* is an offering, here most prominently offerings for services.

The *main operations* for the Marketplace are: (1) User Registration and Directory Services, (2) Offering of services, (3) Discovery and Matching of services, (4) Review and Rating, and (5) Recommendation of services (see [10]).

Repository and Marketplace both can handle Linked USDL service descriptions.

2.1.4 Consumer Marketplace

The purpose of the Consumer Marketplace is to provide an infrastructure for MSEE so that service end-users (consumers, manufacturing ecosystem members) will be able to seek and compare service offerings. The Consumer Marketplace implements the business logic of creating and managing stores, as well as displaying service offerings in them.

In this context, the application provides the initial point of contact between manufacturing Service Providers and their target Service Consumers. Service consumers will be able to seek and compare service offerings from MSEE. The marketplace will then redirect users to the Service Provider's store, where they will be able to communicate with the provider and/or complete the transaction.

The intended audience for the Consumer Marketplace includes:

- Service Providers: they are providers of manufacturing service ecosystem services.
- Service Consumers: end-users of services provided by Manufacturing Service Providers through the Manufacturing Service Ecosystem. Service Consumers can be other organizations (B2B use case) or consumers (B2C use case).
- Administration: managing host platform governance

The key use cases which represent the functional requirements implemented in Consumer Marketplace include:

- Service Provider Registration
- Browsing and administration of Stores
- Browsing and administration of Service offerings
- Service comparison
- Various types of Search within federated Marketplaces
- Federated search

Consumer Marketplace takes advantage of FI-WARE Marketplace RI, the reference implementation of Marketplace related Open Specifications provided as open source by SAP ([5]). Detailed description of the technical implementation and the relationship of Consumer Marketplace to FI-WARE Open Spec APIs is provided in [1].

2.1.5 IoT Manager

The Internet of Things Manager (IoT Manager) is intended as an easy to use gateway for MSEE applications to devices in the MSEE ecosystems. Access to sensors, actuators and embedded devices has been identified in [3] as a key feature of at least three out of four MSEE use cases (Indesit, Ibarmia and Bivolino). The IoT Manager aims to abstract the complexities of accessing devices into an easy to use interface.

The intended audience for IoT Manager are MSEE service developers building applications that require access to devices, for example in order to obtain data from a variety of sensors in the Carefree Washing use case. These users are taking advantage of the IoT Manager's API to retrieve data for further processing. Taking into account the key requirements of pilot use cases for M18 prototypes, the first prototype of IoT Manager focuses on data acquisition and sensors management.

The original IoT Manager architecture was designed in [3] with the intent to reuse Generic Enablers from the Internet of Things Services Enablement Chapter of FI-WARE project. However due to several factors pertaining to FI-WARE (technical architecture redesign, reshaping of FI-WARE partners), access to FI-WARE Generic Enablers was not available for M18; therefore, given the focus on data acquisition for the first prototype, IoT Manager was redesigned to integrate with the Global Sensor Network (GSN) open source project ([4]). The GSN project provides middleware service for the deployment and programming of sensor networks.

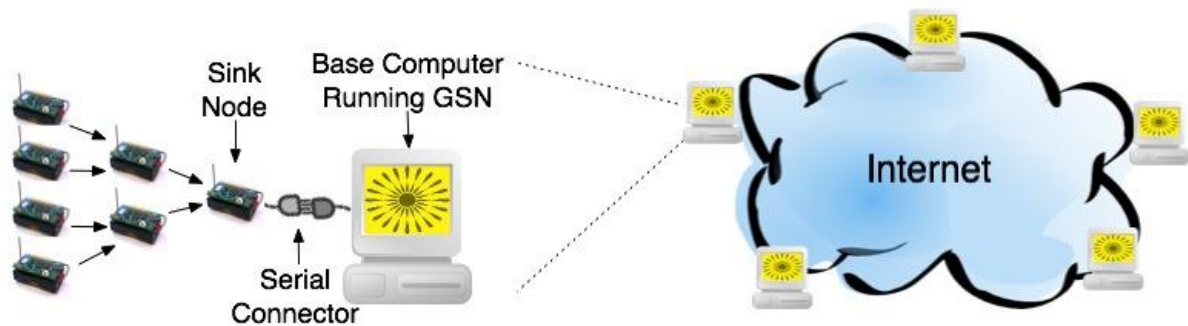


Figure 1 GSN overview

The key concepts in GSN terminology are:

- The key abstraction in GSN is the virtual sensor. Virtual sensors abstract from the implementation details of the data source to sensor data and correspond either to a data stream received directly from sensors or to a data stream derived from other virtual sensors. A virtual sensor can be any kind of data producer, for example, a real sensor, a wireless camera, a desktop computer, or any combination of virtual sensors. A virtual sensor may have any number of input data streams and produces exactly one output data stream (with predefined format) based on the input data streams and arbitrary local processing. The specification of a virtual sensor provides all necessary information required for deploying and using it, including (1) metadata used for identification and discovery, (2) the details of the data streams which the virtual sensor consumes and produces (3) an SQL-based specification of the stream processing (filtering and integration) performed in a virtual sensor, (4) the processing class which performs the more advanced and complex data processing (if needed) on the output stream before releasing it and (5) functional properties related to persistency, error handling, life-cycle, management, and physical deployment.
- Wrappers are software components that wrap the native code required to communicate with a specific sensor type and supply sensors' data streams in a unified API, encapsulating the variations of data access per sensor type. GSN ships with a default set of wrappers for several known sensor network types (e.g. Mica2, RFID readers, Bluetooth devices etc) and more can be added as required.
- Virtual sensors and advanced querying capabilities can be used to setup complex data processing pipelines, visualization as well as data streaming with server-push.

In terms of its REST API, IoT Manager still provides the same functionality originally designed. The IoT Manager design encapsulates the underlying implementation of data retrieval from remote sensors, therefore it will be possible in the future to switch from GSN

implementation to another implementation of its backend services, while IoT Manager's API will remain the same and clients will operate normally.

3 Prototype modules

The Following tables provide the list of the software components that D32.3 delivers in the scope of the MSEE project.

3.1 Consumer Marketplace

Component	Implemented functionalities
FI-WARE Marketplace Client Library	<p>This module interfaces with a single FI-WARE Marketplace RI: it manages authentication & session/cookies with the FI-WARE Marketplace and provides access to its functionality.</p> <p>Detailed technical information for this component is available in [1].</p>
Consumer Marketplace Business Logic	<p>This module provides the implementation of Consumer Marketplace core business logic; it provides federated access to several FI-WARE Marketplace RI instances, via FI-ware Marketplace Client Library module. Consumer Marketplace Business Logic implements all of the core functionality of Consumer Marketplace, including management of stores and offerings across several instances of FI-WARE Marketplace RI.</p>
Consumer Marketplace Web Frontend	<p>This module provides a simple and easy to use web application GUI built on top of the Consumer Marketplace Business Logic module.</p> <p>Detailed technical information for this component is available in [1].</p>
Consumer Marketplace REST API	<p>This module wraps Consumer Marketplace Business Logic module and exposes its functionality via a REST API for consumption by other services.</p>

3.2 IoT Manager

Component	Implemented functionalities
Device Management Module	<p>Device Management module is responsible for interfacing with GSN middleware instances via GSN's SOAP API. It provides an internal API that allows other components of IoT Manager to communicate with one GSN middleware instance and obtain Virtual Sensors information and data streams.</p> <p>Detailed technical information for this component is available in [2].</p>
Auditing	<p>Auditing module wraps all actions occurring on IoT Manager and maintains a complete timestamped log of action and affected entities.</p>
Core Business Logic – Information Management	<p>The core module of IoT Manager manages the federation of GSN instances and Virtual Sensors defined per each instance. It uses the internal interface of Device Management Module to integrate with each federated GSN instance, while its own internal interface provides</p>

Services	support for actions across all federated GSN instances.
Administration web app	<p>The administration webapp is a Java web application with a simple & clean interface for the IoT Manager administrator to:</p> <ul style="list-style-type: none"> • Manage federated GSN instances • Manage Virtual Sensors per each GSN instance • View audit logs <p>The operation of the administration web app is described in [2].</p>
IoT Manager REST API	<p>IoT Manager's REST API provides a programmatic frontend for other MSEE services to take advantage of federated access to sensors data.</p> <p>Detailed technical information for this component is available in [2].</p>

4 Summary and conclusions

This deliverable provides the first prototypical implementation of the **Consumer Marketplace** and **IoT Manager** FI Platform Federation Services. This document is to be considered as an introduction to these prototypes, whereas [1], [2] provide the technical details of the components that have been delivered at month 18. In future developments of MSEE Project, the prototypes may be refined and extended, and FI Platform federation services categories may be further analyzed to identify other services needed from MSEE Use Cases.

5 References

- [1] MSEE D32.3 Factsheet#1.docx – “FI Platform federation Services – Consumer Marketplace”
- [2] MSEE D32.3 Factsheet#2.docx - “FI Platform federation Services – IoT Manager”
- [3] MSEE D32.1a – FI Platform Federation and Architecture – M12 issue
- [4] GSN project on sourceforge - <http://sourceforge.net/projects/gsn/>
- [5] <https://github.com/service-business-framework/Marketplace-RI>
- [6] <http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Repository>
- [7] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.Apps.Repository#Technical_interfaces
- [8] <http://linked-usdl.org/>
- [9] <http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.Apps.RepositoryREST>
- [10] <http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.Apps.Marketplace>
- [11] <http://catalogue.fi-ware.eu/enablers/repository-sap-ri>
- [12] <http://catalogue.fi-ware.eu/enablers/marketplace-sap-ri>