D32.2 – Future Internet Platform Federation Specifications and Architecture – M21 issue

Document Owner:	Charalampos Vassiliou (SINGULAR)
Contributors:	Andreas Friesen (SAP), Markus Heller (SAP), Daniel Oberle (SAP), Ioan Toma (UIBK), Martino Maggio (ENG), Davide Storelli (ENG)
Dissemination:	Public.
Contributing to:	WP 3.2
Date:	15 07 2013
Revision:	1.0

VERSION HISTORY

VERSION	DATE	NOTES AND COMMENTS
0.1	18.6.2013	INITIAL DRAFT STRUCTURE AND TOC
0.4	26.6.2013	ADDITIONS TO DRAFT, RESTRUCTURING
0.5	30.6.2013	FINAL DRAFT
0.6	8.7.2013	SILO INTERNAL REVIEW
0.7	8.7.2013	INTERNAL REVIEW VERSION
1.0	15.7.2013	SUBMISSION VERSION

DELIVERABLE PEER REVIEW SUMMARY

ID	**Comments**	**Addressed (✔) Answered (A)**
1	Executive Summary: Inclusion of details about the services under development	✔
2	Executive Summary: Various amendments	✔
3	Conclusions: Addition of a development roadmap for the services under development	✔
4		

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Executive Summary

Work Package 32 focuses on integrating and federating MSEE solutions with existing components and services offered by other Future Internet-inspired platforms through reusing, assembling and possibly extending parts of their offerings. As a particular case, we aimed at the FI PPP OSS implementations of some generic enablers, but in general we addressed the whole set of FI-inspired projects involved in the FI Assembly movement. These activities aim to better satisfy the needs of the MSEE Use Cases and maximise the project's applicability and coverage in terms of business domain functions.

The present document, D32.2, is the second deliverable of Task T3.2.1 "FI Platform Federation Specification and Design". The first deliverable of the task provided an initial specification for MSEE's FI Platform Federation. The present deliverable updates this specification, based on the experience gained from developing the first prototype, further research, and cooperation with stakeholders involved in the creation of associated MSEE platform components. This specification will focus on the final release of the platform at M24 (D32.4 "FI Platform Federation final prototype").

Deliverable D32.2 "Future Internet Platform Federation Specifications and Architecture" aims to explore ways that Future Internet software assets, delivered through an open platform federation or through re-use and extension, can be leveraged to provide benefits to the MSEE project. As the deliverable at hand describes, such a WP32 federation will leverage some of the FI PPP Core Platform Generic Enablers open source implementations (services, applications, components) that are already in place and at the same time it will also reuse (to the extend this is possible) existing services and software components coming from other open platforms such as those provided by FI Assembly projects or by other successful open-source projects.

In order to identify potential enablers that fit to the MSEE needs, a comprehensive analysis of existing projects in the Future Internet - related domains has been conducted (in the areas of Internet of Services, Internet of Things, Internet of Content/Knowledge and Internet for and by the People) which revealed components that are of interest to MSEE and have a potential to be reused in the context of the project.

This analysis, in conjunction with a set of criteria based on functionalities necessary for the MSEE Use Cases resulted in the definition of two service-based applications to be integrated in the prototypes of this work package:

- The **Consumer Marketplace** is implementing an online marketplace for service offerings, where service consumers will be able to search for, browse and compare offerings by service providers. The Consumer Marketplace will be primarily used in the Bivolino use case allowing manufacturing service suppliers in the ecosystem to advertise their unallocated capacity. However, the Consumer Marketplace App is fairly generic and can be adapted to any use case where there is a need to bring together service consumers and service suppliers.

  The application is based on one of the FI-WARE Open source Generic Enablers Implementations, the Marketplace GE. This Generic Enabler encapsulates the abstract core entities and functions of a marketplace, i.e. stores, offerings, search, browsing, administration and provides a programming interface for accessing and manipulating them. The Consumer Marketplace interfaces with Marketplace GE instances, and provides an API for access from within the MSEE platform, complex services built from the basic functions of the Marketplace GE, as well as a graphical user interface. Furthermore, it can handle several trusted Marketplace GE instances (i.e. several Marketplaces) and provide aggregate functions such as federated search.

- The **IoT Manager** is an application aiming to simplify the process of managing and accessing IoT devices in the field for services and applications in the MSEE platform. The IoT Manager is central to the Ibarmia and Indesit use cases, as it provides an interface to access embedded sensors in Ibarmia and Indesit products and collect data about their state and operation parameters. This data will then be used in both cases to provide services to ecosystem partners and consumers such as proactive maintenance and personalised offerings.

  The application is based on one of the FI Assembly IoT projects (OpenIOT) belonging to the IERC cluster. OpenIOT uses the the Global Sensor Network open source platform as a middleware for managing and connecting with remote sensors. The IoT Manager uses it in a similar way, by interfacing with instances of the GSN middleware as intermediaries to the sensor networks. The application then provides a programming interface to the rest of the MSEE platform for sensor management and data acquisition, as well as a graphical user interface, advanced management features, access control and logging.

The present deliverable updates the analysis included in D32.1 on the basis of research and prototype development work conducted in the framework of WP32. The updated specifications and the requirement analysis included in the deliverable will guide the development of D32.4: “FI Platform Federation final prototype” which is the main targeted outcome of WP32.

1. Introduction

1.1. Context and Purpose of the Deliverable

MSEE stands to gain from integrating and reusing already components from other platforms, by assembling or extending, such modules according to its own needs. This will enable the project to cover a broader spectrum of functionality and applicability by providing a richer assortment of services to its users. As MSEE's background is strongly connected to the Future Internet concept, it is vital to examine how the project will take advantage of FI Core Platform Enablers, as well as other services offered by other open platforms, which will altogether generate added value to the MSEE solution and to its use cases.

In this context, MSEE project deliverable D31.2 "Functional and Modular Architecture of Future Internet Enterprise Systems" has already presented and explained the concept of federation with Future Internet Platforms.

Deliverable D32.2, "Future Internet Platform Federation Specifications and Architecture" has the following distinct objectives:

- To update the analysis of the Future Internet-inspired platforms, given in D32.1, on the basis of recent updates
- To describe the associations of the MSEE project use cases with Future Internet services and functionalities.
- To update the specifications and architecture of the Future Internet Platforms Federation of MSEE and guide the development of its final implementation

The deliverable at hand provides a set of specifications and architectures for MSEE's Future Internet Platforms Federation for the development of the final prototype (released on M24). In the framework of the two-cycle development followed for the implementation of the Future Internet Platform Federation, Deliverable D32.2 updates D32.1 towards the release of the second and final prototype version.

1.2. Relation to other Work Packages and Deliverables

D32.2 builds on the knowledge of D31.1 & D31.2 and elaborates on the concept of Future Internet Platform Federation, by examining in detail the environment and means to achieve it. The following table illustrates the relations of D32.2 to other MSEE deliverables.

MSEE Deliverables	**Relation to D32.2**
D31.1 & D31.2: Functional and Modular Architecture of Future Internet Enterprise Systems	Initial & Updated Architectural considerations about the FI context in which utility services are placed, including a more refined analysis of their use cases.
D52.1 & D52.2: User Requirements Analysis for Virtual Factories & Enterprises in MSEE	The use cases and the analysis of their elements and requirements is used in D32.2
D32.1: FI Platform Federation specifications and architecture	Initial version of D32.2
D32.4: FI Platform Federation final prototype	The prototype will be based on the specification set by D32.2
D33.2: FI Utility Services Specifications and Architecture	Utility services may use functionality offered by Enablers and FI-inspired platforms, which may be mediated by the FI Platform

	Federation
D34.2: Detailed Specifications of next generation ESA Value Added Services	Smart Enterprise Applications may use functionality offered by Enablers and FI-inspired platforms, which may be mediated by the FI Platform Federation

Table 1: D32.2 and its relation to other MSEE deliverables

1.3. Structure of the Document

D32.2 follows a structure which is conceptually organised into three blocks:

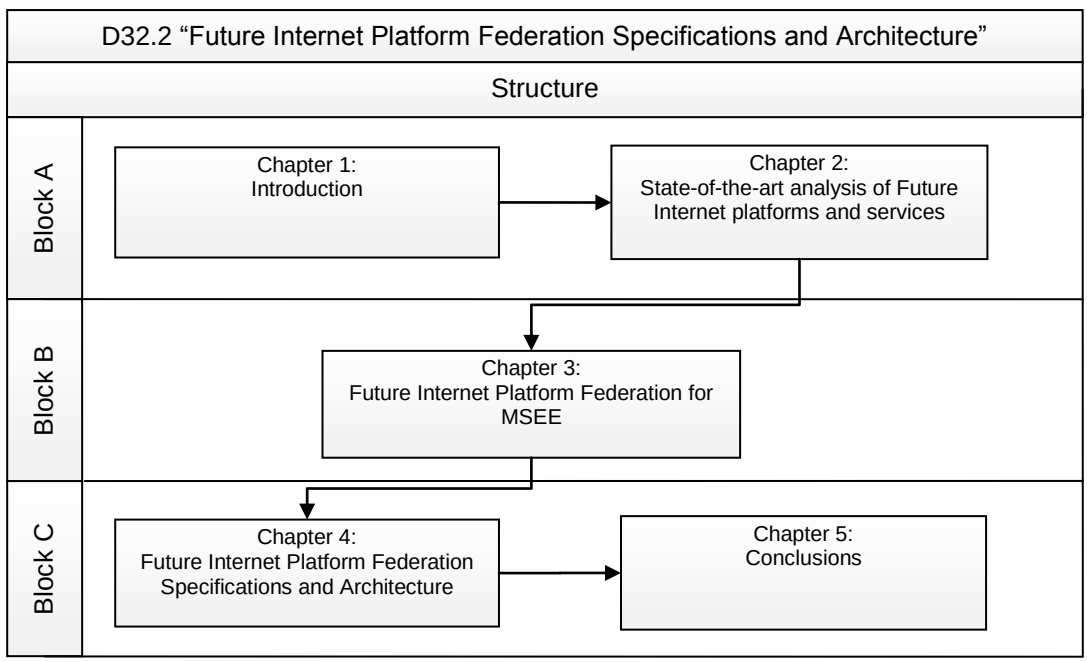

Figure 1: D32.2 Structure

Block A aims to introduce the reader to the purposes of this deliverable, and examine the current state of the Future Internet landscape. Chapter 1 "Introduction" provides an introduction to the context, objectives, associations and structure of this document. Chapter 2 "State-of-the-art analysis of Future Internet platforms and services" provides a review of relevant Future Internet efforts (platforms and services). This chapter investigates platforms and services in the areas of Service Front-ends, Internet of Services, Internet of Things, Internet of Content/Knowledge and Internet for and by the People, focusing on projects that are relevant to the purposes and application domains of MSEE.

Thereafter, **Block B** aims to examine and evaluate FI resources in relation to the needs of MSEE and its use cases. It includes Chapter 3 "Future Internet Platform Federation for MSEE" which contains an account of the methodology used for the examination of appropriate Future Internet assets and the selection of FI-based services for MSEE.

Finally, **Block C** provides the updated specification and architecture of MSEE's Future Internet Platform Federation. This is presented in Chapter 4 "Future Internet Platform Federation Specifications and Architecture" which includes a specification and architecture

view of services that use federation with FI-inspired platforms to provide services to MSEE and its use cases. Lastly, Chapter 5 "Conclusions" concludes the deliverable, summarising results and future work.

An Appendix is included, containing selected material from the initial iteration of this deliverable (D32.1 – M12)

2. State-of-the-art Analysis of relevant platforms and projects

This chapter includes a review of various Future Internet platforms and their services, relevant to the purposes and objectives of MSEE.

2.1. FI platforms and projects

Future Internet platforms are categorised in this chapters in five discrete categories corresponding to the five pillars of the Future Internet[2]:

- Internet of Services
- Internet of Things
- Internet of Contents/Knowledge
- Internet for and by the People
- Future Network Infrastructures

The platforms examined in the following sections have been selected because of their relevance to the domain areas covered by MSEE and its use cases and cover the first four pillars. Future Network Infrastructures are beyond the scope of this deliverable and have not been included.

The following table lists the platforms examined in this section, as well as their potential added value to MSEE:

Domain / Platform	**Potential Added-value**
Internet of Services	
COIN	Implements a platform for Enterprise Interoperability and Collaboration. Supports federation and provides components as open source software.
iSURF	Implements a supply chain management platform using RFID. Source code is available as open source.
SLA@SOI	Implements a framework for Service Level Agreement – aware applications. Components are available as open source.
Service Front-ends (multiple platforms) • SERENOA • EzWeb	Both provide front-ends for services: • SERENOA: adaptable User Interfaces with context-awareness. • EzWeb: service mash-up composition and discovery interface.
Internet of Things	
GSN	Provides a middleware for the deployment and programming of sensor networks.
ASPIRE	Provides a middleware for RFID-based applications, available as open-source.
OpenIOT	Provides a middleware for IoT applications.
CUTELOOP	Provides a platform for RFID and GNSS tracking. Components available as open source.
WEBINOS	Implements a framework for applications running across several platforms, including IoT-enabled devices and mobile terminals. The platform is available to developers.
Internet of Contents/Knowledge	
CONVERGENCE	Provides a platform for management of digital content at any point in the digital content value-chain (from producers to consumers). A demonstrator application is available.
Internet for and by the People	
PERSIST	Provides a platform for the creation of personalised, user- and context- aware services, source code and executables are available.

Moreover, the FI PPP Core Platform has been carefully analysed:

FI PPP	Potential Added-value
FI-WARE	Provides several reusable building blocks (Generic Enablers) providing functionalities relevant to: • Data/Context Management • Internet of Things (IoT) Services Enablement • Applications/Services Ecosystem and Delivery Framework • Security • Cloud Hosting • Interface to Networks and Devices (I2ND)

Table 2: Platforms and projects examined

2.1.1. Internet of Services

2.1.1.1. COIN

COIN[3] is an FP7 funded Integrated Project aimed to develop a platform for services and collaboration in the business domain. The project addressed issues in the fields of Enterprise Collaboration (EC) and Enterprise Interoperability (EI). The project goals were:

- Design and develop a Service Platform to host COIN services for EI and EC and make them available to European enterprises.
- Consolidate and stabilize the ICT results of both EC and EI FP6 research into COIN Baseline Services.
- Extend and enrich its baseline services, with new services in the EC and EI fields, taking under consideration recent advances.
- Support the convergence of the EC and EI research fields.
- Execute 6 realistic test cases in the fields of Aeronautics, Automotive, Aerospace, Pulp & Paper, Healthcare and ICT.

The services provided by the COIN platform are organised in two domains

- Enterprise Collaboration
  - Enterprise Collaboration Baseline Services: Support for the collaboration life-cycle in general
  - Collaborative Product Development Services: services which assist collaborative product development and innovation
  - Collaborative Production Planning Services: services supporting real time inventory management, optimised dynamic planning and scheduling and integration with manufacturing and ERP legacy systems
  - Collaborative Project Management Services: services supporting collaborative project management, focusing on dynamic performance indicators and proactive events management
  - Collaborative Human Interaction Services: services focusing on user interaction with a service platform, supporting collaboration, context-awareness and mobile usage.
- Enterprise Interoperability
  - Enterprise Interoperability Baseline Services: model transformations and semantic reconciliation mechanisms for business documents, business processes and enterprise models

- Information Interoperability Services: services supporting multiple actors involved in collaboration and use the publish/subscribe/negotiate mechanism
- Knowledge Interoperability Services: services supporting semantic representation of Enterprise semantic profiles (e.g. competencies)
- Business Interoperability Services: services supporting high-level generation and model transformation for the creation of business process compositions with multiple business actors exposing internal processes

Potential Contribution to MSEE

COIN is of particular interest to WP32 and to MSEE in general, as it is a mature project and offers many Enterprise Interoperability and Enterprise Collaboration services which are relevant to the manufacturing domain and MSEE in particular. COIN natively supports federation of services[1] [2] and COIN software assets are available as open source downloads[3].

2.1.1.2. iSURF

The iSURF[4] project ("An Interoperability Service Utility for Collaborative Supply Chain Planning across Multiple Domains Supported by RFID Devices") aimd to provide an intelligent collaborative supply chain planning network. The project provided a collaboration environment to SMEs to share information on the supply chain visibility, individual sales and order forecast of companies, current status of the products in the manufacturing and distribution process, and the exceptional events that may affect the forecasts in a secure and controlled way. Moreover, iSURF provided a Service Oriented Collaborative Supply Chain Planning Process Definition and Execution Platform. This platform presented "template" collaborative planning process definitions, enabled customisation of these templates graphically and provided wizards to create executable planning process definitions as an OASIS WS-BPEL[4] package that can be easily deployed in integration with the underlying enterprise planning applications.

iSURF provided also a Semantic Interoperability Service Utility (ISU) for achieving the semantic reconciliation of the planning and forecasting business documents exchanged between the companies according to different standards.

Furthermore, iSURF provided an open source Smart Product Infrastructure (SPI) based on RFID technology using EPCGlobal standards. To achieve maximum benefit from RFID technologies, the supply chain must be supported with Global Data Synchronisation mechanisms to allow partners to share accurate master data reliably and efficiently. To facilitate this, iSURF provided Global Data Synchronisation Service Utility (GDSSU).

[1] http://demos.txt.it/coinmediawiki/index.php/COIN_GSP_federation
[2] http://demos.txt.it/coinmediawiki/index.php/Online_Support
[3] http://demos.txt.it/AISBL/web/guest/coin-platform-service-download
[4] https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel

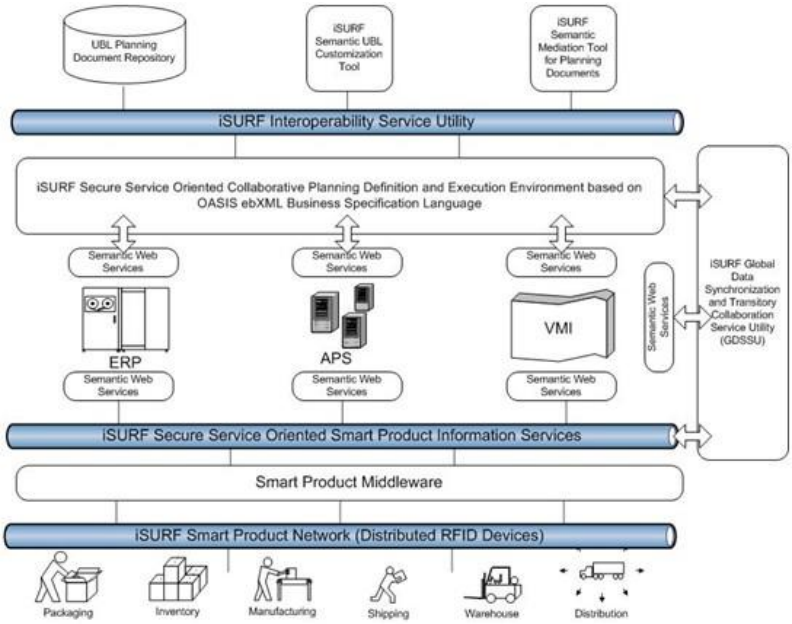

Figure 2: iSurf Architecture[4]

Potential Contribution to MSEE

iSURF can provide tools for implementing IoT services based on RFID-enabled devices in case needed. The project could potentially provide a full stack (from scanners to high-level service composition) for supply-chain management services. iSURF software assets are currently available as open source Java source code[5].

2.1.1.3. SLA@SOI

The SLA@SOI project[5] aimed to research, engineer and demonstrate technologies that can embed ”Service Level Agreement”-aware infrastructures into the service economy. Service Level Agreements are formal agreements and contracts between service providers and service consumers, regarding the contents and level of the services provided.

SLA@SOI achieved the following results:

- documented an SLA-enabling reference architecture suitable for both new and existing service-oriented, cloud infrastructures,
- defined SLA description models,
- released a suite of open-source software, including the SLA@SOI framework of tools and components to help implement SLA-aware solutions, supported by extensive developer documentation,
- contributed to several open standards initiatives including W3C’s USDL and OGF’s OCCI and WS-Agreement,
- published research results extensively.

The project included four industry-led use cases in the areas of ERP Hosting, Enterprise IT, Service Aggregation and eGovernment.

Potential Contribution to MSEE

Service provision modes developed within MSEE are expected to employ SLAs. For this reason, SLA@SOI is of particular interest to MSEE, especially for the open source software

[5] http://isurf.svn.sourceforge.net/viewvc/isurf/

released by the project, which could be adapted to support SLAs in the manufacturing domain. This includes the components of the overall SLA@SOI Framework, SLA Models and SLA Tools which are published under the BSD license and include extensive documentation[6,7].

2.1.1.4. Service-Front Ends

In the context of service architectures and platforms for the Future Internet, service front ends represent the multi-faceted interface between the final user and the complexity of the Internet of Services. Service front ends and interaction enablers are considered not only as end-points of the user experience in service consumption, but also as the means to enable people to become service creators and knowledge providers for other people. User-generated content and service mash-ups are indeed one of the major trends in the Information Society, and interaction systems are progressively evolving toward transparent interfaces between the physical and the digital world. As this research progresses, we are not far away from scenarios in which users are surrounded by the right services and information at the right moment and in the right place thanks to the continuous mediation of context-sensible and collective-intelligence-oriented service front ends.

Building next generation service front-ends for the Internet of Services should follow some fundamental guidelines as stated in [6]:

- ***end-user empowerment*** *– This refers to enhancing traditional user-service interaction by facilitating the selection, creation, composition, customisation, reuse and sharing of applications in a personalised operating environment.*
- ***seamless context-aware user-service interaction*** *– New-generation service front-ends should have the capability to detect, represent, manipulate and use contextual information to adapt seamlessly to each situation.*
- ***end-user knowledge exploitation*** *– This principle aims to exploit users' domain knowledge and collective intelligence to improve service front-ends (e.g., tagging resources).*
- ***universal collaborative business ecosystems*** *– Enterprise systems should incorporate advanced user-centric, context-aware front-ends to enable their employees and other stakeholders to exploit and share their extensive domain expertise and their business knowledge.*

Such guidelines have been interpreted and further developed during the last years in different research projects. Each project addressed some particular aspects of service front ends. In the following paragraphs a brief overview on two such projects is provided.

2.1.1.4.1 SERENOA

Project **SERENOA**[7] (Multidimensional, context-aware adaptation of Service Front-Ends) has the objective of conceiving and developing an open platform that enables the creation of service front ends (SFEs) that are aware of the physical environment in which they are used and in general that are sensible to the context of the user. A context-aware (or context-sensitive) service front end is defined as user interface (UI) that *"exhibits some capability to be aware of the context and to react to changes of this context in a continuous way"*[8]. SERENOA's context model includes a model of the user, of the hardware-software platform (which includes the set of computing, sensing, communication, and interaction resources that

[6] http://sla-at-soi.eu/results/software/
[7] http://sourceforge.net/projects/sla-at-soi/

bind together the physical environment with the digital world), and of the social and physical environment (where the interaction is actually taking place). The result of the adaptation of the UI to the user's preferences, activities, abilities and device is considered to be the improving of the effectiveness of the human-service interaction and of the user's satisfaction with respect to traditional SFEs based on static UIs.

Potential Contribution to MSEE

The SERENOA project can provide tools and software components for adaptable end-user mobile interfaces. Mobile applications in MSEE could potentially use these adaptable interfaces. Source code for project components is currently openly available[8].

2.1.1.4.2 EzWeb

The **EzWeb**[9] project aims at developing the architecture (and its reference implementation) of an open platform which manages the front-end access to Future Internet services and supports the economics of a resource marketplace. Taking advantage of a next-generation SOA architecture, the project addresses all its layers with the ultimate goal of building a front end layer that empowers end-users, allowing them to personalize their operating environment by self-serving from a wide range of valuable resources. Such resources are created as a combination of simple ones through piping, orchestration and mash-up techniques and leveraging semantic and context information. Resources are catalogued and published on a marketplace where users can find the ones that meet their needs and where providers can manage all aspects related to accounting, auditing and payment of the resources. Finally knowledge exchange and exploitation is supported by lightweight techniques for knowledge representation (e.g. tagging) and advanced inference mechanisms.

Potential Contribution to MSEE

EzWeb focuses on creating an environment (including a web UI) where users can browse, access and compose "gadget" representing services. This could provide the basis for a service description and discovery front-end for MSEE. The EzWeb platform is available and freely downloadable[9].

2.1.2. Internet of Things

2.1.2.1. GSN

The GSN (Global Sensor Network) open source project [54] started in 2005 under the coordination of the LSIR Laboratory of EPFL[10]. The project has led to the creation of the GSN middleware service supporting the deployment and programming of sensor networks. The open source platform of GSN is currently used by several labs in EPFL and other universities as a middleware for receiving data from sensor networks and for publishing, processing and sharing these data.

The GSN project itself has been supported by NCCR-MICS[11] and by the Lion project supported by Science Foundation Ireland under grant no. SFI/02/CE1/I131. Currently the project is being backed by research teams in EPFL and DERI with several researcher contributing to the project regularly.

[8] https://forge.morfeo-project.org/scm/browser.php?group_id=151
[9] http://forge.morfeo-project.org/wiki/index.php/Guides_and_Manuals?lng=en
[10] http://lsir.epfl.ch
[11] http://www.mics.org/

GSN utilizes the notion of the virtual sensor which reflects any producer of data. The data may derive either from a real sensor, a camera, a microphone etc or from a computer – in other words, any device that produces, as an output, a structured data stream, can be considered as a virtual sensor for GSN. Notably, a virtual sensor may have many input data streams but only one output stream [55]. By utilizing virtual sensors that collect data from real sensors and any other sources, GSN can operate as a middleware for transferring and processing sensors' data, while managing complementary information in form of metadata. This metadata is extremely useful for the identification and discovery of the data streams. GSN assumes that stream filtering and integration is performed in each virtual sensor using an SQL-based specification.

Thanks to all the above, the GSN middleware provides a very flexible and uniform platform for the fast and easy deployment and integration of heterogeneous sensor networks. According to [57] the design of GSN follows four main design goals, which are Simplicity, Adaptivity, Scalability and Light-weight implementation.

Potential Contribution to MSEE

The open source middleware provided by the GSN project[12] is can be an important infrastructure complementing the work of MSEE. The capabilities of the GSN middleware cover most of the IoT aspects required by the project. The GSN framework has been thoroughly and successfully tested in the framework of OpenIoT FP7 project and it has been proven to offer a stable and mature middleware to be examined as an alternative solution to FIWARE IoT Chapter GEs.

2.1.2.2. ASPIRE

ASPIRE[10] (Advanced Sensors and lightweight Programmable middleware for Innovative Rfid Enterprise applications) was an FP7 project whose main goal was to offer a lightweight, scalable, royalty-free RFID middleware able to provide the main intelligence to RFID's solutions and enabling SMEs to deploy RFID solutions with a significantly lower entry cost and without the need to engage extensively with low-level middleware.

The ASPIRE middleware, through the incorporation of reader and tag virtualisation capabilities, is able to operate with any reader platform regardless of vendor, frequency and supported functionality. Moreover the middleware is designed to implement the EPCglobal standard (driving the RFID domain) guarantying the provision of clean and intelligently processed information to end-user applications.

The middleware has been developed as a set of independent modules namely: Tag Data Translation, Reader virtualisation, Filtering and Collection, Business Events Generator, EPC Information System, Object Naming Service, Business Intelligence, Connectors.

[12] http://sourceforge.net/apps/trac/gsn/

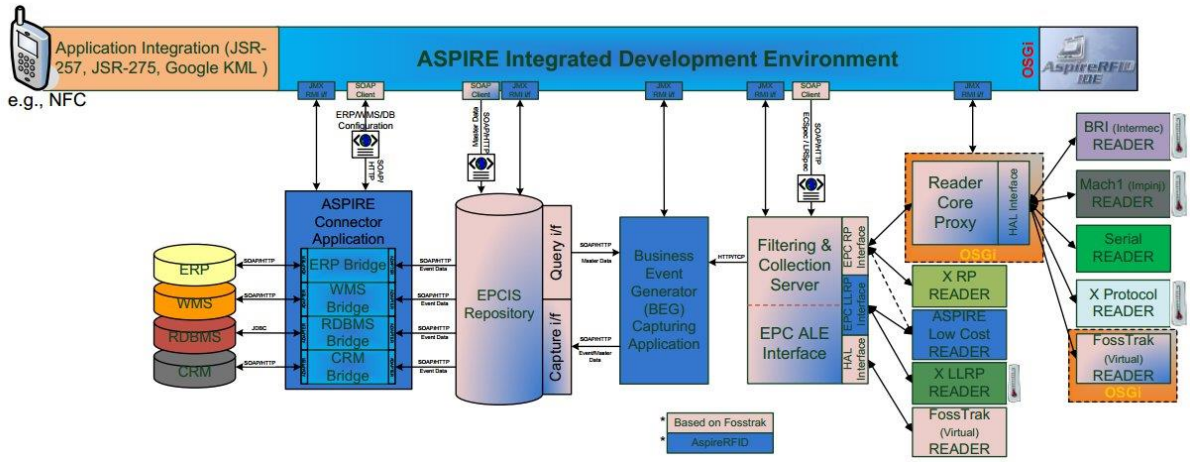

Figure 3: Aspire Middleware Architecture[46]

The goal of these components was to transform RFID reads into information suitable for consumption by human and/or third party software and this goals is achieving applying to the raw data coming from as sensors several hierarchical levels of functionality provided by the various components.

The lower level is represented by the hardware level containing all the required hardware with its proprietary APIs. At a higher level the Hardware Abstraction Layer (HAL) hides the proprietary communication aspects of the hardware from the higher levels. The event level utilizes the abstraction provided by the HAL and processes the streams of data from the hardware level. Event level are less than raw RFID reads, however they are still significant in amount and do not provide business level information. The Filtering and Collection and the Business Events Generator (BEG) components are in charge of leveraging the content of lower level events transforming them in business events through the addition of suitable metadata handled by the BEG component. Business events are then forwarded to a higher hierarchical level, where it is consumed by the Information System (IS) component. The IS component comprises a database which aggregates business events, applies additional business logic and stores business information, which could then be conveyed to the company enterprise IT systems through the use of suitable connectors.

Potential Contribution to MSEE

The ASPIRE middleware provides a complete end-to-end stack for accessing and managing RFID-tagged objects. The middleware creates a chain from low-level RFID scanner access and to high-level applications. MSEE could reuse ASPIRE components to create an IoT service stack for RFID in case needed. The middleware components are available as open-source software[13].

2.1.2.3. OpenIOT

The OpenIoT[11] project investigates ways to formulate and manage IoT based cloud environments (i.e. environments comprising IoT "entities" and resources (such as sensors, actuators and smart devices) and offering utility-based (i.e. pay-as-you-go) IoT services). The project aims to support the creation of cloud-based and utility-based sensing services, using a "Sensing-as-a-Service" paradigm. At the same time, the project will provide an adaptive middleware framework for deploying and providing services into cloud environments.

[13] http://forge.ow2.org/projects/aspire

OpenIoT is leveraging previous research in IoT, including GSN [54] and ASPIRE [10]. The project will explore IaaS, PaaS and SaaS business models for the creation of sensing applications via cloud infrastructures. Also, the project will research self-management of device networks and IoT entities, as well as relevant ontologies and semantic structures.

Potential Contribution to MSEE

The open source middleware provided by the project is also considered as an important infrastructure to complement the work of MSEE. Specifically, the OpenIOT middleware will support flexible configuration and deployment of algorithms for collection, and filtering information streams stemming from the internet-connected objects, while at the same time generating and processing important business/applications events. OpenIoT aims to provide an open source solution, at first by providing extensions of existing open cloud infrastructures, and then by supplying a fully customizable toolkit for the Internet of Things into the cloud.

2.1.2.4. CUTELOOP

The CuteLoop[12] project built methods and tools to support the usage of enhanced RFID-based systems and Global Navigation Satellite Systems in an Integrated Enterprise. The project offered a framework to build a system consisting of the following components[47]:

- Access Platform. The Access Platform is the interface between services/agents and existing application. The access platform is implemented according to service-oriented principles and serves as the entry point to the overall system functionality. It follows uses service-oriented-architecture.
- Services. Services are used to communicate within the other parties. They are handling security aspects and to support interactions within the supply chain.
- Agents. Agents are entities inside the workflow (e.g. Producer, Customer etc.). They are actors that communicate in a decentralised and asynchronous manner with each other (i.e. they use an Event-driven-architecture)

The next figure depicts the CuteLoop architecture and its components.

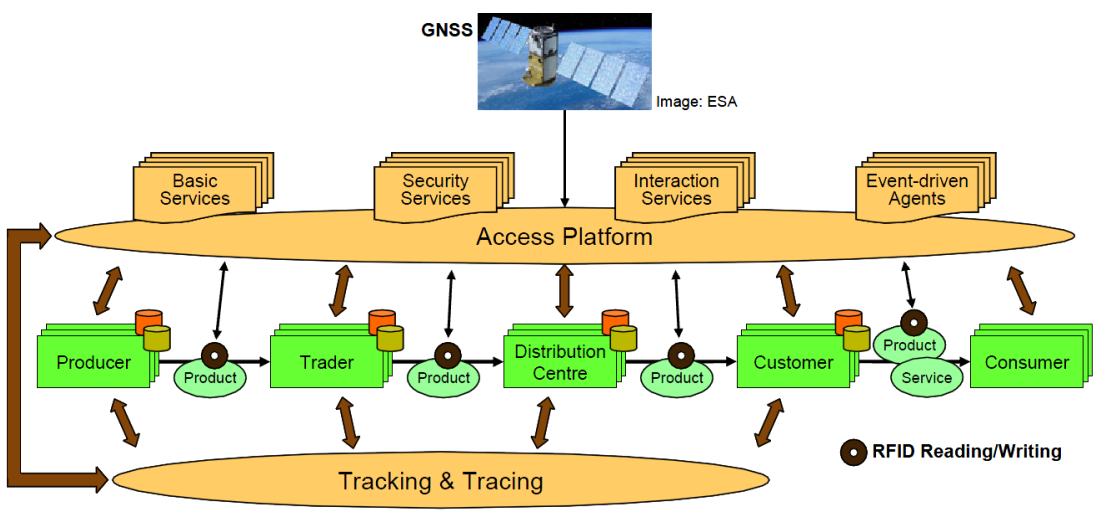

Figure 4: Cuteloop Architecture[47]

To fulfil its goals CuteLoop used technologies such as RFID and GNSS. The use of RFID enables possibilities like uniquely identifying products (transport information along the

supply chain, monitor temperature during transport for fruits etc.). Combined with GNSS it is also possible to track time and location of products.

Potential Contribution to MSEE

The CUTELOOP project offers a set of tools for RFID tracking, enhanced by GNSS services. This toolset may be used to implement an RFID-tracking stack, from scanners to applications. The source code and documentation of the platform components are mature and currently available[14].

2.1.2.5. Webinos

Webinos[13] will define and deliver an Open Source Platform and specific components for the Future Internet, which will enable web applications and services to be used and shared consistently and securely over a broad spectrum of converged and connected devices, including mobile, PC, home media (TV) and in-car units.

Promoting a "single service for every device" vision, webinos will move the existing baseline from installed applications to services, running consistently across a wide range of connected devices, ensuring that the technologies for describing, negotiating, securing, utilizing device functionalities and adapting to context are fit for purpose.

Innovations in contextual description will be broad covering but not limited to device capabilities, network access, user identity and preferences, location, behaviourally induced properties and finally the more complex issue of the users' social network context. webinos will directly address security and privacy issues as part of Quality of Service that users of web services expect.

Webinos aims to provide an inter-operable, standardised, open source technology utilizable across domains with direct commercially exploitable value. The webinos platform is well documented at the moment, and includes the following categories:

- Webinos core interface
- Service discovery and access: Allows applications to discover services/applications on other devices or on network servers and access these remote services.
- HW Resources APIs: APIs allowing applications to access information and functionality relating to device HW resources such as GPS, camera, NFC, SIM and smart card readers, sensors, etc.
- Application Data APIs: APIs allowing applications read and write access to application capabilities such as contact items, calendar information, messages, media files, etc.
- Communication APIs: APIs allowing applications to communicate with other applications in the same or another device.
- Application execution APIs: APIs allowing webinos applications to manage its execution or launch other webinos and native applications.
- User profile and context APIs: APIs allowing applications access to user profile data and user context.
- Security and Privacy APIs: APIs related to the security model for webinos.

Potential Contribution to MSEE

[14] http://www.cuteloop.eu/index.php?aid=12

The webinos framework is open source and is currently available[15] for use by developers. Several of the functions described above seem to align with the MSEE project, as well as the MSEE use cases, warranting further investigation on how they could be leveraged to provide added value in the form of mobile interfaces to IoT-enabled services.

2.1.3. Internet of Contents/Knowledge

2.1.3.1. CONVERGENCE

CONVERGENCE[14] aims at enhancing the Internet with a content-centric, publish-subscribe service model, based on a common container for any kind of digital data, including representations of people and Real World Objects. This common container, named Versatile Digital Item (VDI), is a "package" of digital information with a unique identifier, independent of the machine where the VDI is hosted. VDIs will be designed to handle all possible kinds of digital information, from media to information about services, people and physical objects, independently of the structure or geographical location of the content.
CONVERGENCE targets professional and non-commercial providers and consumers of digital content, allowing them to publish, control, search for, and use content, independently of the structure or geographical location of the content. Users will be able to define their own policies for using, authenticating, protecting and revoking VDIs. The functionality provided by VDIs supports new models of use and new business models, difficult or impossible to implement on the current Internet architecture.

The CONVERGENCE framework will significantly extend opportunities for interactions among different actors along the value chain for digital items. The digital value chain includes media creators, distributors, copyright authorities, etc., finally ending at the consumer. Traditional media value chains are one-way, meaning that once content is delivered to the next stop of the chain (e.g. from the creator to the distributor) no further alterations can take place. On the other hand, VDIs can always be modified by any authorised actor in the value chain, even remotely.

Potential Contribution to MSEE

The project has produced a reference implementation of the CONVERGENCE middleware and of a selected set of Applications, which may be useful to MSEE, especially in content-oriented applications. These include the CONVERGENCE PEER KIT (a sample Application, built on top of the CONVERGENCE middleware, and a release of the current status of the CONVERGENCE middleware). The middleware extends the MPEG-M specification, for an ecosystem of interconnected services supporting distributed applications that create, trade and consume digital objects in so called "media value chains". The CONVERGENCE demonstrator application is openly available[16].

2.1.4. Internet for and by the People

2.1.4.1. PERSIST

The objective of PERSIST[15] was to develop Personal Smart Spaces that provide a minimum set of functionalities which can be extended and enhanced as users encounter other smart spaces during their everyday activities.

[15] http://webinos.org/downloads/
[16] http://www.ict-convergence.eu/demodownloads/

These spaces were to be capable of learning and reasoning about users, their intentions, preferences and context. They were endowed with pro-active behaviours, which enabled them to share context information with neighbouring Personal Smart Spaces, resolve conflicts between the preferences of multiple users, make recommendations and act upon them, prioritise, share and balance limited resources between users, services and devices, reason about trustworthiness to protect privacy and be sufficiently fault-tolerant to guarantee their own robustness and dependability.

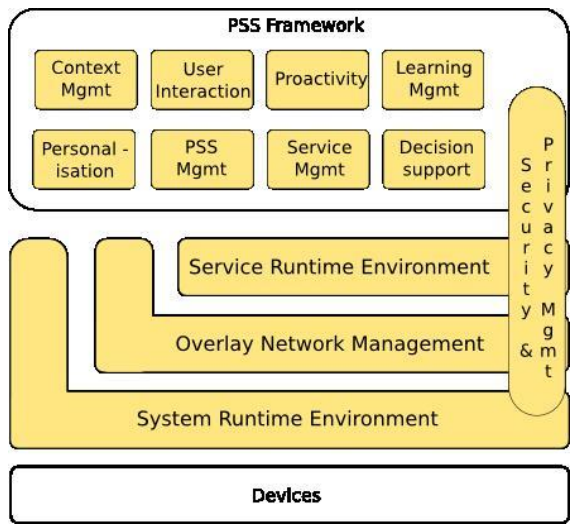

Figure 5: PERSIST Architecture[15]

The architectural layers of PERSIST are:

- Layer 1 – Devices: devices may either implement the PSS stack or part of it, or simply interact with the rest of the PSS framework
- Layer 2 - System Run-Time Environment: an abstraction layer between the underlying device operating system and the PSS software in order to achieve as much platform independence as possible.
- Layer 3 - Overlay Network Management a Peer-to-Peer (P2P) management and communication layer.
- Layer 4 - Service Run-Time Environment: a container for the PSS services. It supports service life cycle management features and provides a service registry, as well as, a device registry. Moreover, it allows for service management in a distributed fashion among multiple devices within the same PSS.
- Layer 5 - PSS Framework: The PSS Framework layer is the core of the PSS architecture. Its functions include discovering and composing PSS and 3rd party services, as well as, managing context data and user preferences. Moreover, the PSS Framework layer supports automatic learning of preferences and inference of user intentions. This information, together with data from recommender systems, enables the proactive behaviour of the PSS platform. Grouping of context data and preferences, as well as, conflict resolution are also provided by the PSS Framework layer.

Potential Contribution to MSEE

The work carried out under PERSIST is of interest to the MSEE, as some of the services provided by the use cases may involve mobile devices and interaction with “smart objects”

(such as appliances and sensors). The project's middleware framework for the provision and operation of a Personal Smart Space (PSS) has been delivered as part of the results dissemination of PERSIST[16] and is currently available both as executable and as source code[17]

2.2. The Future Internet PPP

In 2011, the European Commission launched the Future Internet Public-Private Partnership Programme. The FI-PPP programme aims to support research and development on European, large-scale platforms, business models and technologies that revolve around the Internet of the Future. At the same time, the programme aims to update and harmonise the legal, political and regulatory frameworks in Europe, as well as the various relevant policies.[17]

At the time of writing, the FI-PPP includes a set of research and development initiatives touching on all of the Future Internet domains (Internet of Things, Internet of Services, Internet of Contents/Knowledge, Internet for and by the People, Future Internet Infrastructures). [18]:

The FI-PPP Programme is implemented via three phases:

- Phase 1, 2011-12, budget 90 million euros
    - Laying the technology foundation
    - Defining "use case scenarios" in different industry sectors
    - Making an inventory of available (public) infrastructures via capacity building
    - Programme support

- Phase 2, 2013-14, budget 80 million euros
    - Developing use case pilots and platforms
    - Setting up infrastructures

- Phase 3, 2014-16, budget 130 million euros
    - Expansion of use cases by developing applications and services
    - Extending the technology foundation

The FI-PPP is covering a wide scope of usage areas through a set of use case projects, listed below, that make use of the FI-PPP core platform in setting up trials of advanced Future-Internet-based services and applications. FI PPP phase 1 included 8 use case projects, and the ongoing phase 2 includes 5 use case trials, which will lead to SME-driven use case implementations in Phase 3.

- TRANSPORT, LOGISTICS AND AGRI-FOOD
    - FINEST (Phase 1)
    - SmartAgriFood (Phase 1)
    - FIspace (Phase 2)
- PERSONAL MOBILITY
    - Instant Mobility (Phase 1)
- SOCIAL CONNECTED TV, MOBILE CITY SERVICES, AND VIDEO GAMES
    - FI-CONTENT (Phase 1)
    - FI-CONTENT 2 (Phase 2)
- SMART CITIES AND PUBLIC SECURITY

[17] http://sourceforge.net/apps/trac/psmartspace/wiki

  - SafeCity (Phase 1)
  - OUTSMART (Phase 1)
- SMART ENERGY
  - FINSENY (Phase 1)
  - FINESCE (Phase 2)
- MANUFACTURING
  - FITMAN (Phase 2)
- E-HEALTH
  - FI-STAR (Phase 2)

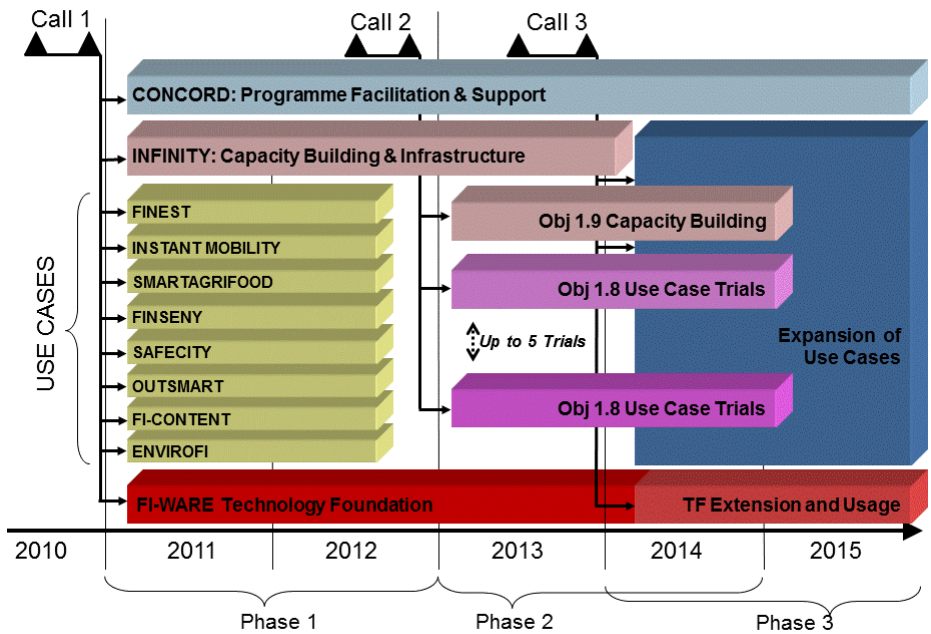

Figure 6: FI-PPP constituent initiatives and timetable[19]

2.2.1. Future Internet Technology Foundation: FI-WARE

The FI-WARE project aims to build an open and harmonised technological framework for Future Internet platforms and services. Experience has shown that applications and services provided across the Internet have gradually defined a set of basic functions, encountered repeatedly over multiple implementations and platforms. For example, user authentication is nearly ubiquitous in today's Internet landscape.

FI-WARE relies on a fundamental assumption: that the Internet of the future will also have ubiquitous functions that will be used in the majority of its ecosystem of applications. For this reason, there will be significant benefits if we were to define what those functions will most likely be, standardise them openly, and then have them provided "as-a-service" to the entities of the Future Internet. These comprise the "Future Internet Core Platform".

The project defines those "atomic building blocks" as "Generic Enablers" covering a range of basic domains (or Chapters as defined in FI-WARE)[49]:

- Cloud Hosting: Functions and tools for providing Infrastructure, Platform and Software-as-a-Service

- Data/Context Management: Data management, multimedia, location, semantic metadata, social networks etc
- Internet of Things (IoT) Services Enablement: Communications, Resources, Data Handling and Automation
- Applications/Services Ecosystem and Delivery Framework: Service delivery, composition, management etc
- Security: Security, Identity management and Privacy
- Interface to Networks and Devices (I2ND): Interoperability across various platforms, devices and networks

As of July 2013, the project has published documentation on preliminary API specifications for its Generic Enablers. [20]

Potential Contribution to MSEE

The FI-WARE project and its Enablers are of particular interest to the MSEE project, for the following reasons: Firstly, the defined Generic Enablers, which implement commonly encountered functionality used across domains and platforms, are good candidates for use in MSEE and MSEE use cases. Secondly, as these Enablers are gradually reaching maturity, they are likely to be ready to be used in a reliable manner within the scope of MSEE, as the composition of the FI-WARE Core Platform is more-or-less agreed at this point and can be relied on to plan on architectures exploiting it. FI-WARE APIs are available, and implementations are already being produced by FI-WARE partners. Lastly, as the Enablers are being defined at a central level (as a basic technological foundation) they have a good chance of remaining available and supported for the lifetime of MSEE.

3. Future Internet Platforms Federation for MSEE

3.1. Introduction

This section presents the methodology used for the selection of appropriate FI resources for MSEE, as well as for the selection of the services for the first and final FI Platform Federation Prototypes (Deliverables D32.3 and D32.4 respectively). Section 3.2 "Methodological Approach" focuses on the selection process and its main results, while section 3.3 "FI Resources for the MSEE Platform" contains technical descriptions and evaluations of FI resources used in the first and final FI Platform Federation prototypes.

The initial selection of FI resources and prototype applications has been described in depth in the first iteration of this deliverable, D32.1 (M12). Since then, however, some portions of this analysis have been superseded by more recent work or correspond to technologies and methodologies not used in the final implementation. Specifically, these are:

- The initial use case analysis performed on the use case descriptions from D52.1 (M4) has been superseded by the results in D52.2 (M15). The D52.2 deliverable provides the most recent updates to the use cases, a much deeper analysis as well as the final requirements from WP32.
- Detailed technical descriptions and evaluations of several FI-WARE chapters as well as FI Resources not used for the FI Platforms Federation prototypes.

Since the relevant sections provide little added value to the updated version of the deliverable, they are not included in the main body of this document. However, they can be found in the Appendix.

3.2. Methodological Approach

Selecting appropriate Future Internet Resources, in order to implement applications supporting the project use cases is a complex task, and depends on the following factors:

- The needs of the individual use cases
- The suitability and availability of FI resources
- The services selected for implementation
- The implementation stage of those services

The first version of this deliverable, D32.1 (M12), provided an analysis of the project use cases in order to determine their particular needs, based on the initial work performed for D52.1 "User Requirements Analysis for Virtual Factories & Enterprises in MSEE – M4"[58].

The analysis was necessary for determining associations between the nature of each use case and Future Internet domains: the Internet of Services, Internet of Things, Internet of Contents/Knowledge and Internet for and by the People, along with the specific areas of "Trust, Security and Privacy" and "Cloud and Network Infrastructures". Implementing applications for functions that are common across use cases would provide maximum benefits and an optimum investment of effort. The result of this work was an enumeration of potential Future Internet functions appearing in each use case. (see Appendix)

The main results of the analysis are summarised in the table below (core functionalities indicate functionalities mentioned in the initial use case descriptions in D52.1, while optional functionalities are not explicitly mentioned but may provide added value):

Functionality	Indesit	Ibarmia	Bivolino	Philips
End-user service discovery	Core functionality	Core functionality	Core functionality	Core functionality
SLA Management	Core functionality	Core functionality	Core functionality	Core functionality
KPIs and business intelligence	Optional	Optional	Optional	Core functionality
IoT device communications (sensors, actuators, etc)	Core functionality	Core functionality	-	-
RFID object tracking	Optional (RFID tracking of parts)	Optional (RFID tracking of parts)	Core functionality (RFID tracking of shirts) – note scannable QR codes may be selected instead	-
Real time data and event processing	Core functionality	Core functionality	Core Functionality (tracking of shirts)	Core functionality (KPI collection)
Interoperability	Core functionality (ecosystem data exchanges e.g. orders, deliveries)	Core functionality (ecosystem data exchanges e.g. orders, deliveries)	Core functionality (Interoperable Cutfile, ecosystem data exchanges)	-
Mobile access channels	Core functionality (device data and control)	Core functionality (device data and control)	Core functionality (service status information for consumers)	-
Context/location aware services	Optional	Optional	Optional	-
Ecosystem participant communication / collaboration	Optional (Ad-hoc maintenance)	Optional (Technical personnel collaboration)	Optional (Collaboration with manufacturers / communication with customers)	Optional (ecosystem partner collaboration)
Personalised services	Core functionality	-	Core functionality	-
Content tracking	-	-	-	Optional
Social media integration	Optional	-	Optional	Core functionality

Table 3: Initial analysis of use case functional commonalities[59]

Following this task, a detailed technical analysis was performed, examining the most promising of those platforms: FI-WARE and its Generic Enablers. FI-WARE was selected for the preliminary analysis because it presents certain important advantages:

- It aims to cover several Future Internet domains under a single unified research effort, while other projects usually focus on a single domain.
- FI-WARE Generic Enablers are designed with the explicit intent to be reusable components for other applications. Most other projects generally provide open source components that need to be adapted programmatically, whereas FI-WARE GEs have documented open interfaces.

- The Generic Enablers in each chapter are designed so as to provide a well-defined functionality for the corresponding domain. This means that individual GEs can be used in various custom combinations, depending on specific applications.
- FI-WARE offers extensive documentation on enabler design and implementation, facilitating the necessary technical analysis.

The preliminary analysis helped identify specific sets of GEs from each FI-WARE chapter suitable for the particular needs of MSEE. A secondary analysis took place identifying potential complementary services from other FI-inspired platforms, while providing a brief evaluation. (see Appendix). The result of this work was a preliminary shortlist of potential candidate FI resources, both from FI-WARE and other platforms, summarised in the following table:

Functionality	Selected Resources (Initial)	Rationale for selection
End-user service discovery	FI-WARE (Applications and Services Ecosystem and Delivery Framework Chapter)	Documented; Supports federation
SLA Management	Potentially SLA@SOI components	Open source Components
KPIs and business intelligence	To be determined	-
IoT device communications (sensors, actuators, etc)	FI-WARE (IoT Chapter, Cloud Edge)	Documented; Supports federation
RFID object tracking	FI-WARE (IoT Chapter)	Documented; Supports federation
Real time data and event processing	FI-WARE (Data/Context Management Chapter)	Documented; Supports federation
Interoperability	Potentially COIN services	Documented; Mature; Supports federation
Mobile access channels	Potentially SERENOA components	Open source Components
Context/location aware services	FI-WARE (Location Platform, Cloud Edge),	Documented; Supports federation
Ecosystem participant communication / collaboration	Potentially COIN services (for B2B applications)	Documented; Mature; Supports federation
Personalised services	Potentially PERSIST components	Documented; Open source Components
Content tracking	To be determined	-
Social media integration	To be determined	-

Table 4: Initial FI Resource Selection[59]

The combined analysis of business needs and technical opportunities helped define two FI-based services that combined maximum impact given the available use case definitions and the potential for implementation through federation with FI-WARE assets. The criteria used included the relevance to each use case (as a core or an optional element), dependencies on other functionalities, MSEE synergies and the required FI resources. The results of this process are summarised in the following table:

Functionality	Number of use cases as a core element	Number of use cases as an optional element	Dependencies on other services	Selected Resources (Initial)
End-user service discovery	4	0	None	FI-WARE (Applications and Services Ecosystem and Delivery Framework Chapter)
SLA Management	4	0	Enhanced by real time data and event processing	Potentially SLA@SOI components

KPIs and business intelligence	1	3	Enhanced by real time data and event processing, Content tracking (Philips)	To be determined
IoT device communications (sensors, actuators, etc)	2	0	None	FI-WARE (IoT Chapter, Cloud Edge)
RFID object tracking	1	2	None	FI-WARE(IoT Chapter),
Real time data and event processing	4	0	Requires IoT device communications for Indesit and Ibarmia, Enhanced by RFID object tracking for Bivolino, and Content tracking for Philips	FI-WARE (Data/Context Management Chapter)
Interoperability	3	0	None	Potentially COIN services
Mobile access channels	3	0	Requires IoT device communications for Indesit,Ibarmia, None for Bivolino	Potentially SERENOA components
Context/location aware services	0	3	Requires Mobile Access Channels, IoT device communications (Indesit, Ibarmia), Enhanced by RFID object tracking (Bivolino)	FI-WARE (Location Platform, Cloud Edge),
Ecosystem participant communication / collaboration	0	4	None	Potentially COIN services (for B2B applications)
Personalised services	2	0	Enhanced by Context/Location Aware services	Potentially PERSIST components
Content tracking	0	1	None	To be determined
Social media integration	1	2	None	To be determined

Table 5: Service selection decision matrix[59]

The decision process gave preference to functionalities that were apparent core features of most use cases, do not depend on other functionalities, and can be implemented using the documented resources. From this process, End-user service discovery, IoT communications and RFID tracking were selected as the most likely candidates. Specifically:

- End-user service discovery
  - Is central to all four Use Cases in
  - Has no prerequisites
  - Can be implemented using a single FI-WARE resources (Applications and Services Ecosystem and Delivery Framework Chapter)
- The IoT device communications service
  - Is central to two Use Cases (Indesit and Ibarmia)
  - Has no prerequisites
  - Is a prerequisite for Real time Data and Event processing, Mobile Access Channels and Context/Location aware services for Indesit and Ibarmia
  - It supports enhanced SLA management, KPIs and Business Intelligence for these two cases
  - Can be implemented using well documented FI-WARE resources, i.e. the IoT Services Enablement Chapter.
- RFID object tracking
  - Is central to the Bivolino Use Case (in D52.1)
  - Has no prerequisites
  - Enhances Real time Data and Event processing for the Bivolino case.
  - Supports SLA management, KPIs and Business Intelligence for the Bivolino case
  - It can be implemented using the FI-WARE IoT Services Enablement Chapter..

- Other functionalities either have prerequisites, do not reach their full potential unless used in conjunction with others or cannot implemented with resources documented in depth at this stage in WP32.

Following this process, two FI-based services were proposed and specified:

- Consumer Marketplace (based on the FI-WARE Applications and Services Ecosystem and Delivery Framework chapter)
- IoT Manager (initially based on the FI-WARE Internet of Things Enablement chapter)

Following the preliminary design and specification work, development started on the prototype applications for deliverable D32.3 "FI Platform Federation First Prototype" (M18). During this period, the development of the Consumer Marketplace prototype proceeded as planned, based on the Reference Implementation of the FI-WARE Marketplace GE.

On the other hand, the development of the IoT Manager application proved problematic. Access to the necessary resources from the relevant FI-WARE chapter could not be granted in time for the development of the first prototype. For this reason, various alternatives were examined from the initial list of IoT-related projects.

During the examination of the OpenIOT platform, it became apparent that one of its base technologies, the Global Sensor Networks Middleware, could provide a significant part of the required functionality. The GSN Middleware is licensed as open-source and it is mature, therefore it can be reused and adapted to the needs of the project.

In the meantime, the IoT-related use case requirements (IBARMIA, INDESIT) were refined and became focused on remote sensing [40]. A detailed technical analysis of GSN confirmed that it is an ideal candidate for the IoT Manager application, and was put into use for the first prototype.

For the second prototype in M24 (D32.4), the individual services will remain the same, but they will be updated with further functionality and other improvements. This decision was based on the following factors:

- The final version of the user requirements (D52.2 – M15) confirmed the assumptions of the initial analysis in D32.1. The Consumer Marketplace and IoT Manager will support key functionalities in three out of four use cases. In particular, the IoTManager will provide IoT connectivity for the implementation of the Ibarmia and Indesit use cases. Additionally, while the Consumer Marketplace will be aimed towards the Bivolino use case, it is generic enough to be applicable to all use cases if required [40].
- There is a trade-off between the quantity and quality of developed services. The apparent usefulness of the selected services indicates that the best investment of design and development effort would be towards enriching their features and improving their reliability.
- The final use case requirements relevant to WP32, as described in deliverable D52.2 [40] are focused on the Consumer Marketplace and the IoT Manager. These requirements are currently being used in the use case pilot design. Any new services would fall outside this framework, causing deviations from the original pilot specifications and would increase coordination overheads.

Detailed specifications on the specifications of the final FI Platform Federation prototype can be found in Chapter 4 of this deliverable.

3.3. FI resources for the MSEE platform

The following section contains the updated technical evaluation of the suitability of selected FI resources to the MSEE platform and use cases.

3.3.1. FI-WARE

The core concept behind FI-WARE is to offer a toolbox to build and instantiate so called Future Internet platforms. FI-WARE offers a set of specifications of components offered in the toolbox, called Generic Enablers. Generic Enablers represent basic, "atomic" Future Internet functions, which are already present "en masse" in the Internet domain, such as Consumer Marketplaces, identity management, etc. FI-WARE Generic Enablers are classified into a number of areas considered to be parts of the Future Internet: IoT, Cloud, Security, Interface to the Network, Data Context Management, Service Ecosystems and Delivery, and Developer Tools.

Generic Enablers are designed to be accessed via an openly specified RESTful interface, and be provided "as-a-service" by external providers.

FI-WARE will offer reference implementations for many of the Generic Enablers and a "test bed" platform instance for test purposes where all Generic Enablers implemented within FI-WARE will be instantiated.

An analysis of FI-WARE Generic Enablers has been performed by SP3 partners in the MSEE project in order to determine those that can be potentially leveraged and provided added benefits to the MSEE project as a whole. The results of this analysis are summarised in the following section. The analysis has been performed using information from the specifications of FI-WARE Enablers[42] and their published APIs[43].

Note that FI-WARE Generic Enablers will be released along three major milestones:

- 1st Release: Q3 2012 (Already Released) – corresponds to MSEE M12
- 2nd Release: Q3 2013 – corresponds to MSEE M24
- 3rd Release: Q2 2014 – corresponds to MSEE M33

A complete release of the FI-WARE Testbed will be available to FI use cases in FI-WARE's M15 (which is August 2012 corresponding to MSEE M11).[30]

Note that FI Platform Federation prototypes are expected in M18 and M24. Therefore, the usefulness of each Enabler is limited by its release date. Enablers released at multiple releases (e.g. 1,2, etc) have various functionality improvements and additions with each subsequent release.

The major criteria for evaluating Enablers are availability and alignment with the purposes of MSEE and its use cases. Availability is influenced by the release cycle of a particular enabler, as well as the presence of instances, downloads and API specifications. With FI Platform Federation prototypes expected in M18 and M24, there is a strong preference for Enablers included in the first FI-WARE release.

Alignment to the purposes of MSEE, means that the Enabler may be a useful component for an MSEE applications, either related to a use case, to the MSEE platform, or to the

manufacturing domain in general. The selection of useful functionalities is also dependent on functionalities developed within the rest of SP3 as Utility Services (WP33) and Value Added Service (WP34). WP32 will focus on developing new services, which are not included in the aforementioned service groups.

3.3.1.1. Applications and Services Ecosystem and Delivery Framework

According to Wikipedia[31], (the) *"Internet of Services is extending today's Internet to become service-enabled. The Internet of Services is not an overhaul of the Internet but, for the most part, an application of it, where classical barriers and inefficiencies are removed from service access."*

The Internet of Services concept has been developed in the TEXO project and supported by a number of further projects[33]. The IoS notion has been taken up by the FI-WARE project to become a part of the Future Internet Core Platform.

The overall architecture illustrating the GEs and their relationships offered by Apps and Services Ecosystems and Delivery Framework is shown in the following figure[32]:

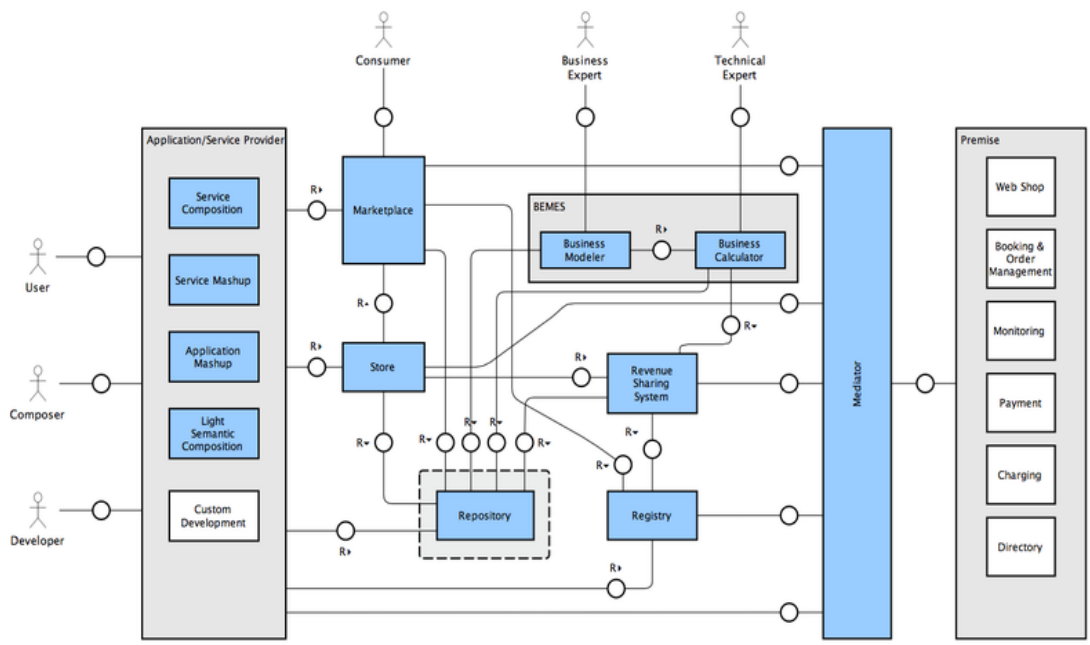

Figure 7: Apps and Services Ecosystems and Delivery Framework overview[32]

FI-WARE chapter "Apps and Services Ecosystems and Delivery Framework" specifies a number of Generic Enablers (GE) intended to support creation of IoS platforms for applications and services ecosystems. The GEs comprise a Broker (a business framework consisting of a Marketplace, Registry, Repository, Business Models Definition and Execution, and SLA management) Aggregators (Composition and Mashup), Mediators and Channel Makers. For some of the Generic Enablers reference implementations will be made available.

A marketplace brings together sellers and buyers or offer and demand. In the context of IoS a marketplace brings together service offers published by service providers and potential service consumers. The core business task of the Marketplace is to provide functionality to discover and match service offerings from providers and/or other intermediaries (e.g. published by different shops, stores, web pages) with the demand of consumers. This

functionality is made accessible through a uniform service interface and provides a basis for extended services depending on the domain and nature of the target markets[34].

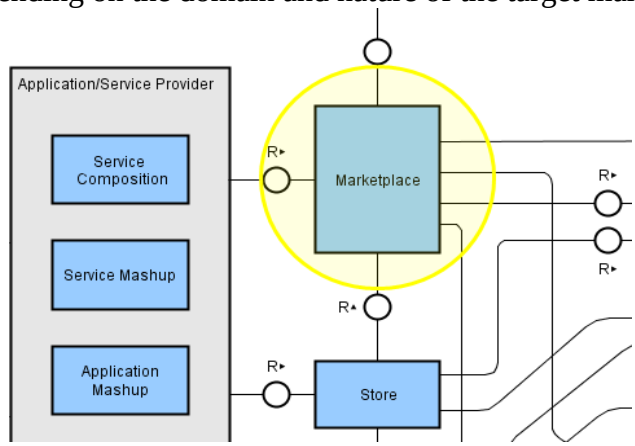

Figure 8: Marketplace in the context of the Business Framework[34]

Furthermore, FI-WARE specifies two further core GEs in the context of IoS which are the Registry and Repository enablers.

The Repository Enabler is used to store complete service descriptions which are more or less static and change only rarely. The repository provides a consistent uniform API to USDL service descriptions and associated media files for applications of the business framework. A service provider can use the Repository to publish the description of various aspects of the service according to a unified description language. The repository relies on the "Linked USDL" version of USDL.

The Registry Enabler is used to store information on service instances necessary for run-time execution. The registry serves as a kind of directory and for example can store detailed settings for concrete infrastructure components as well as information about human or computing agents. The information can range from stable to extremely volatile and is needed to make specific settings for and adjustments to other components in the platform. For example, the Registry can be used by the Marketplace in order to register stores, providers, persons, infrastructure components, etc.

Regarding the other GEs in the IoS context (Revenue Sharing, Business Models Definition, and SLA monitoring), which are of value for productive systems working in the context of prosumer-based scenarios, they are considered not to be closely related to the concepts behind the current use cases of MSEE and therefore they are not considered for analysis at this stage. Furthermore, they will be released by FI-WARE project too late in order to be consumed by MSEE.

Evaluation

The most promising Enablers are the Marketplace Enabler, the USDL Repository Enabler and the Registry Enabler. All are available at the first release and show an immediate correlation with the needs of MSEE. As noted already, use cases involve the concept of an accessible service store, while a "Marketplace/Store" combination appears to be a basic component of a Manufacturing Core Platform.

The Marketplace GE[18] provides a service interface to "discover and match application and service offerings from providers and sources, with demand from customers". Within FI-WARE, it uses services from the Identity Management GE and the USDL Repository/Registry. Its functionality is composed of the following blocks:

- Registration and directory
  - Registration of marketplace entities (such as stores, participants and business roles), updates and deletion.
- Offering and demand
  - Allows the creation, update and deletion of service offerings (i.e. information about services) and their listing/retrieval
- Discovery and matching
  - Supports service consumers in the search for specific service offerings / stores matching their particular needs. It allows search for offerings and stores, as well as comparisons
- Review and rating
  - Supports the creation, update, retrieval and deletion of textual feedback for rateable entities (e.g. stores and services)
- Recommendation
  - Supports recommendations for services dependent on specific user attributes

At this moment, there exist preliminary Restful APIs for

- Marketplace offering (offering and demand)
- Marketplace registration (registration and directory)
- Marketplace search (discovery and matching)

The USDL Repository and the Registry Enabler, while technically two separate entities, are considered together, since their functionalities are intertwined. The USDL Repository is used to store service descriptions and other models. Its main operations are the management of resources (CRUD operations on models, descriptions, etc.), collections, content listing, service listing (a list of services provided by the repository), searches and queries. Its current API specification includes operations for Managing Collections, Managing Resources, Listing Services, and Search. The Registry GE[19] acts as a directory of information for the maintenance, administration, deployment and retrieval of services. It is generally assumed that it will be used by all other GEs of this chapter, as well as certain GEs from other chapters. It enables registration, retrieval and de-registration of entities/attributes, and at the time of writing, its API specification covers all this functionality.

3.3.2. Global Sensor Networks (GSN) Technical Analysis

The Global Sensor Networks (GSN) open source platform aims at providing a flexible middleware to accomplish the dynamic integration and management of sensor networks and their produced data streams [54]. Notice that concerning the GSN project, outcome of which is the GSN platform, essential information has been presented in paragraph 2.1.2.1.

According to [54], the GSN platform is distributed under GPL for the general public. Until release 1.1.2, GSN is distributed, for the general public, under the GNU General Public

[18] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.Apps.Marketplace
[19] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.Apps.Registry

License (GPL). From release 1.1.3, a new branch x-gsn (eXtended Global Sensor Networks) is created and released under the GNU Lesser General Public License v3 license (LGPLv3).

Detailed documentation concerning GSN is included in the GSN Book produced by the GSN Team of NCCR-MICS [55]. The platform assumes the model shown in the following figure. A network of sensors is supposed which internally may use arbitrary multi-hop, ad-hoc routing algorithms to deliver sensor readings to one or more sink node(s). A sink node is a node which is connected to a more powerful base computer which in turn runs the GSN middleware and may participate in a (large-scale) network of base computers, each running GSN and servicing one or more sensor networks.

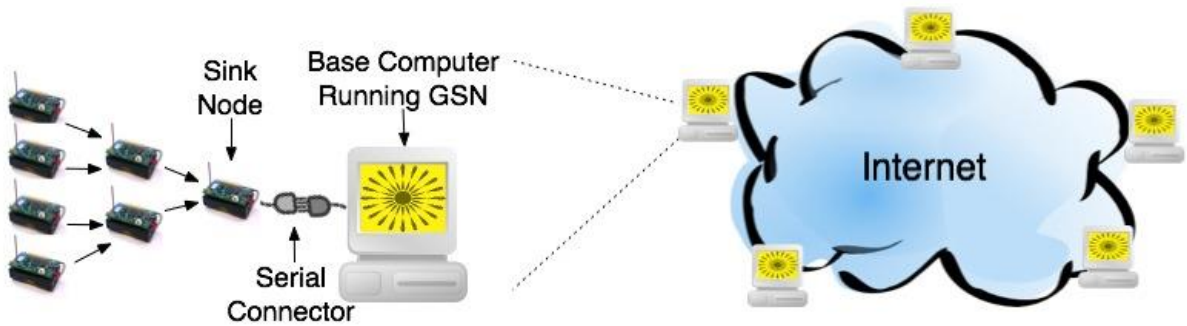

Figure 9: GNS overview [54]

GSN can be configured to acquire data from various data sources. The high number of data sources in GSN allows for sophisticated data processing scenarios. In the unlikely event that data sources are not supported wrappers can be developed in order to make almost any hardware work with GSN [55].

The GSN implementation consists of the GSN-CORE, implemented in Java, and the platform-specific GSN-WRAPPERS, implemented in Java, C, and C++, depending on the available toolkits for accessing specific types of sensors or sensor networks[57].

The implementation currently has approximately 20,000 lines of code and is available from SourceForge (http://sourceforge.net/apps/trac/gsn/).

The high capacity and the capability of GSN to handle large amount of data coming from real sensors is being proved every day in practise as the GSN platform serves as the primary streaming data database for the ETH Center for Competence Environment and Sustainability (CCES). According to the GSN web page in sourceforge, through this project, the usage of GSN is currently approaching somewhere in the region of 80 - 100 million data points, while GSN is used as a repository for all types of data, from national networks to dense, short term local deployments and integrates sensors of a variety of types, from groundwater monitoring, to plant ecology, cryospherics and meteorology.

GSN is deployable on several platforms, from servers and PC workstations to mobile smart devices. Finally it has to be noted that GSN also includes visualization systems for plotting data and visualizing the network structure, which is an extremely useful feature in cases of complex networks of sensors.

Evaluation

GSN has been thoroughly examined as an alternative to the FI-WARE IoT Generic Enablers since there have been several problems with the accessibility and availability of the IoT GEs.

The analysis performed has shown that the open source platform of GSN can fully cover the needs of the use cases of MSEE in terms of sensor data acquisition and in general in the development of IoT-enabled services.

GSN middleware instances support device management and the possible device-related operations described in the use cases of Ibarmia and Indesit, involving registration of devices and reading of sensor values from them.

The GSN Middleware provides the following, required for the aforementioned cases, integration points (among others):

- A SOAP service that provides methods for
  - Management of Virtual Sensor Descriptions (VSDs)
  - Data retrieval, with advanced querying and data stream processing
- A callback-based HTTP streaming method
- A pure push, connection-oriented data push interface.

Based on the fact that GSN is a platform that has been tested and successfully integrated in large IoT solutions in the framework of OpenIoT and other sensor-related projects and given that the middleware is available and mature, it is an ideal solution for covering MSEE needs in IoT services as they derive from the use cases. Given the limited availability of the FI-WARE IoT GEs, the GSN solution can be characterised as the most appropriate option.

4. Future Internet Platform Federation Specifications and Architecture

4.1. Methodological approach

The goal of the Future Internet Platform Federation is to act as a provider of Future Internet functionality to MSEE. This functionality is derived from external to MSEE FI-inspired platforms, and includes both FI Core Platform Enablers as well as other services and frameworks.

4.1.1. Federation Enablers

As discussed in the previous chapters, the main purpose of FI-WARE Generic Enablers is to provide interface specifications and reference implementations used as a toolbox for instantiation of FI-inspired platforms.

Hence, first at all, it is possible to instantiate platforms based on different combinations of subsets of FI-WARE GEs so that a rich palette of platforms with a different scope and capabilities becomes available. However, it is also clear that a concrete FI platform is created with a domain-specific business goal in mind, so that FI-WARE can contribute just some core capabilities to a platform. Further specialisation/adaptation of the GEs might become necessary but also specification and implementation of Specific Enablers (SEs) or integration with additional Utility Services may become necessary to complete the capabilities of the platform. This combination of GEs, SEs, and Utility Services can be considered as ***federation of capabilities*** leading to the creation of a platform with required features.

Second, assuming that FI-WARE concepts and technologies get certain footprint in the real business, there will be a wide range of instantiated domain-specific platforms supporting FI-WARE GE interfaces (for instance, the FI-PPP program is piloting 8 use cases from different domains that will use FI-WARE as the core platform). This opens the perspective for consumption of services directly from such platforms. To make this possible ***federation of platform services*** is necessary which can be typically achieved through composition of GEs and SEs. A composition of GEs and SEs leading to a service federated across two or more platforms is called a **Federation Enabler**.

4.1.2. FI Platform Perspective

The MSEE platform can be considered conceptually as a domain-specific FI platform providing support for service ecosystems in the manufacturing domain. In this platform, specific GEs can be combined with SEs and MSEE Utility Services, as well as custom code and operations, to achieve a set of "Federated" capabilities.

Furthermore, it can be assumed that many competing ecosystems will be created on different instantiations of such a platform offering the consumers and providers a choice to join one or more ecosystems with their service offerings.

Also, it is likely that there will be many ecosystems complementing each other, e.g., logistic services offered at a transport and logistics ecosystem would complement a manufacturing ecosystem by providing services for the transport and delivery of physical goods.

The ecosystems may decide to collaborate to increase the reach of their offered services for the potential customers. This would require a number of federated services like, federated identity and federated search. For instance, to realize a "pull-based" federated query on an

MSEE platform the following FI-WARE Marketplace "Search for Offerings" API[20] can be used to query a number of FI-WARE compliant marketplaces:

Verb	URI	Additional Path Parameters	Description
GET	/search/offerings/fulltext/{searchString}	filter, index, limit, sortBy, order, minScore	Search for offerings where the services description matches the specified search string

Table 6: Pull-Based federated search on MSEE platform using FI-Ware Marketplace

These query results have to be aggregated to present a federated search result to the requestor.

4.2. Conceptual Architecture

The Future Internet Platform Federation provides FI-based services to various internal actors through appropriate interfaces.

In relation to MSEE, the Future Internet Platform Federation is not part of the MSEE ecosystem platform. As described in D41.1, it is part of an external "Open Internet" platform, which involves utility services and value-added services, which provide reusable services across MSEE.

The conceptual architecture of MSEE's Future Internet Platform Federation presented in deliverable D32.1 has been further updated and refined in deliverable D31.2 "Functional and Modular Architecture of Future Internet Enterprise Systems"(M18), as illustrated in the following figure:

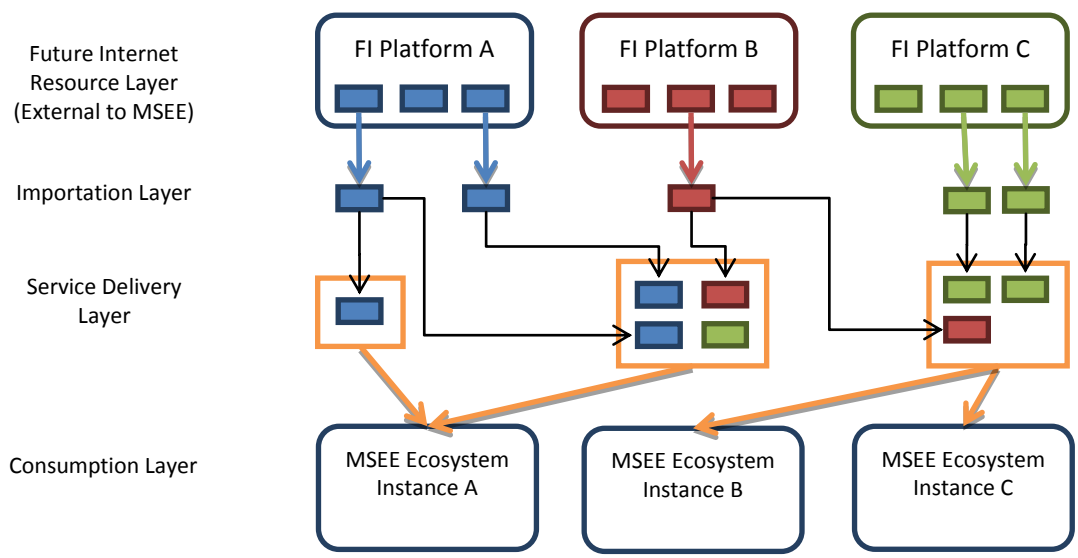

Figure 10: Future Internet Platform Federation conceptual architecture[56]

The figure above illustrates the basic conceptual structure of Future Internet Platform Federation for MSEE. From top to bottom, the layers involved are:

[20]http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Marketplace_Search_Open_RESTful_API_Specification_(PRELIMINARY)

- **Future Internet Resource Layer:** This layer contains FI resources from other platforms. These may correspond to service instances running in other platforms or reusable components.
- **Importation layer:** corresponds to mechanisms and frameworks for importing FI-functionalities derived from the previous layer into the MSEE. This may include e.g. interfaces to running instances, compatibility adapters for code or executable components, etc. This layer makes FI-inspired functionalities available for development. However, they are still unrefined and unsuitable for consumption by MSEE platform actors.
- **Service Delivery Layer:** Based on the products of the previous layer, this layer corresponds to the development of logic / interfaces for the creation and provision of specific FI-inspired services/applications to MSEE platform actors.
- **Consumption Layer:** corresponds to the use of FI-based services by MSEE actors and applications, or with registration with MSEE's Service Delivery infrastructure for re-use.[56]

In the final prototype, two FI-inspired services will be included: the Consumer Marketplace (based on the Marketplace Generic Enabler) and the IoT Manager, (based on the GSN Middleware). Given this particular selection, the conceptual schema of Future Internet Platform federation contains the following elements, per layer:

Conceptual Layer	**Realisation**	
Future Internet Resource Layer	FI-WARE Marketplace GE	GSN Middleware
Importation Layer	Federation with Reference Implementation of GE	Re-use of open-source component
Service Delivery Layer	Consumer Marketplace	IoT Manager
Consumption Layer	• Bivolino ecosystem • Other use case ecosystems (if applicable)	• Ibarmia ecosystem • Indesit ecosystem

Table 7: Correspondence of Conceptual Layers to the MSEE project

4.3. 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 in federated FI-WARE Marketplace instances, as well as displaying service offerings in them, bringing together Service Providers and Consumers. 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.

Operating in the manufacturing domain means that this service needs to be flexible enough to handle both Business-to-Consumer as well as Business-to-Business market relations. These

types of markets operate differently and the application specifications will need to be adjusted accordingly.

Furthermore, the Consumer Marketplace can provide an opportunity to demonstrate the innovative aspects of the "federation of platform services" made possible by the open and common interfaces and capabilities of enablers. In this case, the Consumer Marketplace will be able to use the search API of other Marketplaces and recommend combinations of service offerings.

The Consumer Marketplace application shows significant relevance to all MSEE use cases, and as such, it is considered a prime candidate for implementation.

4.3.1. Users

The Consumer Marketplace has the following classes of users:

- 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
- Administration: host platform governance and oversight personnel

In most cases Service Provider personnel mainly include people from the sales or marketing departments. In the B2C use cases, Service Consumers are individual consumers, while in the B2B use cases, they represent procurement decision makers, including technical and administrative personnel.

4.3.2. Use cases

From the user analysis above, potential use case users can be rather divergent (they range from industrial decision makers to consumers and clothing to industrial equipment procurers), but they follow the simple motif of "service provider" and "service consumer". As such, use cases will be defined using these roles. The following use cases have been identified:

- Service Provider Registration
- Store management, including multiple stores
- Service offering Registration
- Store listing browsing
- Service listing browsing
- Store and service search
- Service comparison
- User account administration
- Federated Search

4.3.3. Requirements

The Consumer Marketplace application will need to conform to the following functional requirements:

The application will support the following functionalities:

- Service Consumer browsing:
    - Browsing of stores
    - Browsing of offerings per store
- Service Provider Store Administration

  - Administration of stores (registration & update)
  - Management of offerings per store (CRUD)
- Search and comparison
  - Search stores with text
  - Search offerings
  - Store listing by category and provider
  - Service listing per category and/or store
  - Service attributes comparison side-by-side
  - Keyword/phrase annotations on Services and Offerings
  - Related offerings
  - *Optional:* Reviews and Ratings
- Access control
  - Service Providers will need to register
  - Service Providers will need to login before accessing stores
  - *Optional:* Registration of Service Consumers
  - *Optional:* Access control at service level: a Service Provider may specify which Service Consumers view a service

Additionally, the application has to conform to the following supporting requirements:

- Browse consumers & providers profiles
- Consumer Marketplace has to be expandable with an arbitrary number of stores and services

For the development of the final prototype, apart from completing, finalising and validating the required functionalities described above, a set of additional enhancements will be investigated:

- Federated SSO integration for all Consumer Marketplace modules (CM frontend, CM API, FI-WARE Marketplaces):
  - This feature will support authentication & authorization functions across all layers of the component architecture, including the FI-WARE Marketplace RI, using the Federated SSO Utility Service developed in WP33.
- Update of FI-WARE Marketplace Reference Implementation to enhance RDF parsing and indexing for MSEE:
  - By Customising the RDF model parser and indexer in the FI-WARE Marketplace Reference Implementation, the application will be able to better support MSEE USDL service descriptions.
- Enhanced search functionality:
  - Based on above feature, the introduction of targeted search according to specific attributes (e.g. service price) will be examined and introduced.
- Introduce USDL repository:
  - Usage in the first prototype demonstrated that the lack of a USDL repository to manage service descriptions effectively. A dedicated USDL repository will be included in the system architecture of the final version.
- Automate USDL generation via templates:
  - Provide a set of templates relevant to MSEE use cases allowing users to parameterise and publish them.

4.3.4. Consumer Marketplace Architecture

In architectural terms, the Consumer Marketplace includes:

- A frontend layer that provides Consumer Marketplace functionality via a REST API and a web application UI.

- A synchronous request-response processing layer that handles API- and UI-originated requests, implementing the core business logic of Consumer Marketplace.
- Information management layer (database, logging)

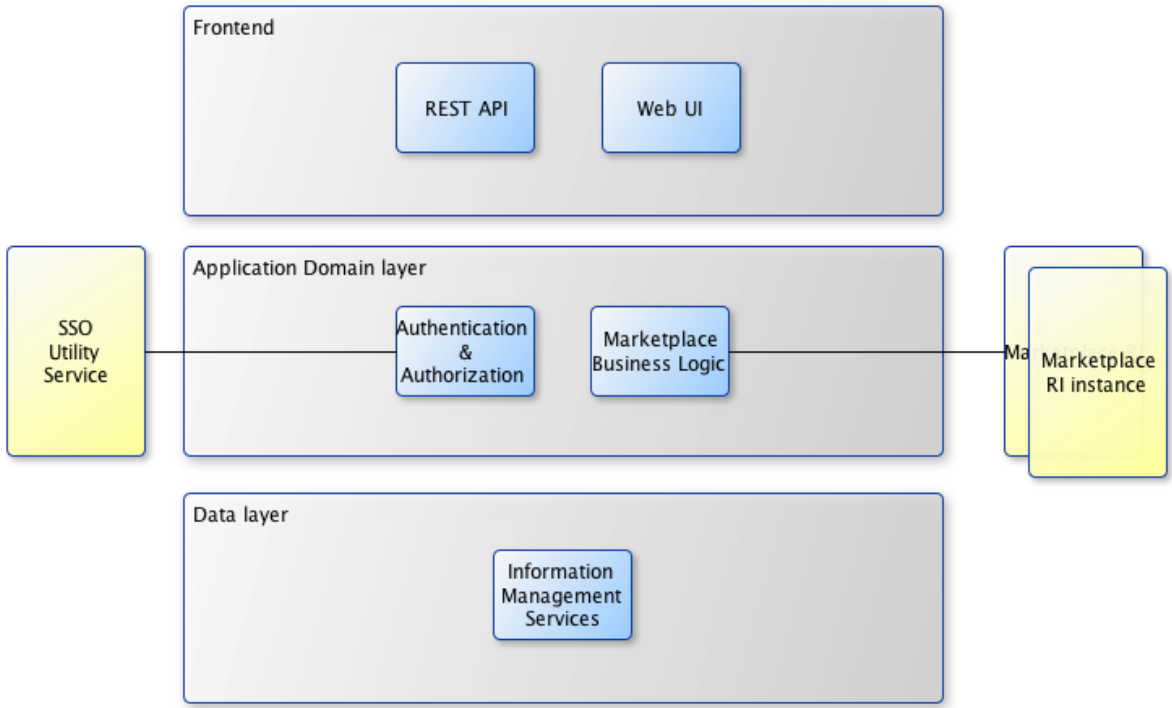

Figure 11 Consumer Marketplace Architecture

The Consumer Marketplace depends on external FI-WARE Marketplace RI instances[21] to which it provides federated access. In the final prototype, another security layer is to be implemented, integrating the MSEE SSO utility service to the FI-WARE marketplaces.

Front end layer

- Consumer Marketplace Web Front-end

The web front-end provides a Graphical User Interface for the functionality of the Consumer Marketplace service as a web application.

- Consumer Marketplace REST API

This module provides an API, allowing programmatic access to the functionality of the Marketplace Business Logic Module.

Application Domain layer

- Marketplace Business Logic

This module implements the core business logic of the Consumer Marketplace service.
The module also mediates access to any federated FI-WARE Marketplace Generic Enablers (Reference Implementation instances).

- Authentication and Authorization

The Authentication and Authorisation module handles access control to devices and actions based on specific user privileges. This module interoperates with the Federated Single Sign-On Utility Service developed in WP33 to provide advanced access control services.

[21] https://github.com/service-business-framework/Marketplace-RI

Data Layer

- Information Management

This module contains all data sources for the operation of the services (e.g. DataBases, etc)

4.3.5. MSEE Context

Conceptually, the Consumer Marketplace can be used as a gateway for consumers to search for and access Use Case service offerings. Access to these services to end-users (such as consumers and customers) would be mediated by the Consumer Marketplace application.

For each MSEE use case, the Consumer Marketplace could be used by several types of users. An indicative list is provided in the following table:

Use Case	Service provider	Service Consumers
Main application (as defined by user requirements[40])		
Bivolino	Bivolino personnel managing the service groups	Customers browsing Bivolino offerings and clothing Manufacturing decision makers browsing order packages, digital cutter lease plans etc
Further potential applications		
Indesit	Indesit personnel managing the "Carefree Washing" service group	Washing Machine Buyers, browsing offerings from the "Carefree Washing" service group
Ibarmia	Ibarmia personnel providing machine monitoring and enhanced maintenance services via the "Intelligent Management" service group	Industrial decision makers, browsing offerings related to machine monitoring and enhanced maintenance in the "Intelligent Management" service group
TPVision	TPVision personnel managing the SmartTV ecosystem	Decision making personnel at TPVision partners (or potential partners), browsing participation schemes in the SmartTV ecosystem

Table 8: Consumer Marketplace users

4.4. IoT Manager

The Internet of Things Manager (IoT Manager) will operate as an easy to use gateway for MSEE applications to reach devices in the MSEE ecosystems, as access to embedded devices is a critical aspect of at least two out of four MSEE use cases (Indesit and Ibarmia).

As described earlier, although the initial IoT Manager architecture was designed with the intent to reuse Generic Enablers from the Internet of Things Services Enablement Chapter of FI-WARE project, it has been decided that – without amending the specified functionality – a new architecture shall be implemented, based on the Global Sensor Network (GSN) open source framework.

The updated 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 the IoT Manager's API will remain the same and clients will operate normally.

4.4.1. Users

The human users of the IoT Manager are MSEE service developers building applications that require access to devices in the field. These users will be in a position to take advantage of the IoT Manager's API to retrieve data, send commands, and retrieve device information.

Non-human users are MSEE applications accessing devices in the field, using the IoT Manager's API.

4.4.2. Requirements

The IoT Manager will need to conform to the following functional requirements:

- Management of devices (registration, discovery):
  - The application will support registration/ de-registration of devices
- Collect readings from device:
  - The application will be able to collect device data from devices in the field
  - *Recommended:* The application will provide a real-time data sink API The application will be able to capture high-frequency sampled data.
  - *Recommended:* The application will be able to perform basic processing before storage of sensor data, in order to minimize storage requirements, as the multitude of deployed sensors may produce enormous amounts of data.
- Receive status information from device:
  - The applications will be able to receive information regarding the status of any connected device
- Authenticated and authorized access to device:
  - The application will provide functionality for access control to the managed devices
  - Access control features apply to both human users and other applications accessing its API interface
  - *Recommended:* The application will provide fine-grained access control and security features by interoperating with the MSEE Federated Single Sign-On Utility Service developed in WP33
- Logging device access for service and platform administration purposes:
  - This may include actions, states, data and message exchanges (and other relevant information)
  - Actions will be logged for human users and applications
- Web interface for administration:
  - The application will provide a web-based administration panel
  - *Recommended:* Provide enhanced sensors management with options to edit virtual sensors in depth via the frontend admin user interface and manage federated GSN instances
  - *Recommended:* Provide visualization and monitoring features including device data, status and alerts.

Additionally, the IoT Manager shall be developed bearing in mind that IoT services place a premium on near-instantaneous operations. An important element of such applications in order to meet expected **performance** requirements is the ability to communicate with remote devices in "real time", requiring, for example, the ability to detect faults or issue commands instantly.

Moreover, it has to be noted that devices existing in the field sometimes face intermittent connectivity, adverse conditions and may fail unexpectedly. The IoT Manager, in order to meet expected **robustness and fault tolerance** requirements should be able to withstand loss

of connectivity, malformed input, etc. without causing the degradation of the rest of the system it participates in.

4.4.3. Use Cases

The IoT Manager will track devices registered with the federated GSN middleware instances and provide browsing and searching facilities, so that MSEE Ecosystem Services can dynamically discover registered devices for further interaction. Moreover, the IoT Manager will provide a unified device addressing scheme, so that MSEE Ecosystem Services may locate and communicate with devices across several federated GSN middleware instances.

Service- device communication may happen according to the following use cases:

- Service requests data (the service polls a remote device)
- Service subscribes to updates from devices in order to receive data at the time they are available (A publisher / subscriber model is applied, with the MSEE service "subscribing" to the device)

The availability of the above modes of communication may vary with different device capabilities, however, the IoT Manager will abstract these details from MSEE Ecosystem Services by providing a unified communication layer on top of the federated GSN middleware instances.

4.4.4. IoT Manager Architecture

As described earlier, the updated design of IoT Manager uses components from the GSN Middleware open source project to provide backend implementation of sensors data acquisition. The architecture of the IoT Manager is illustrated in the following figure:

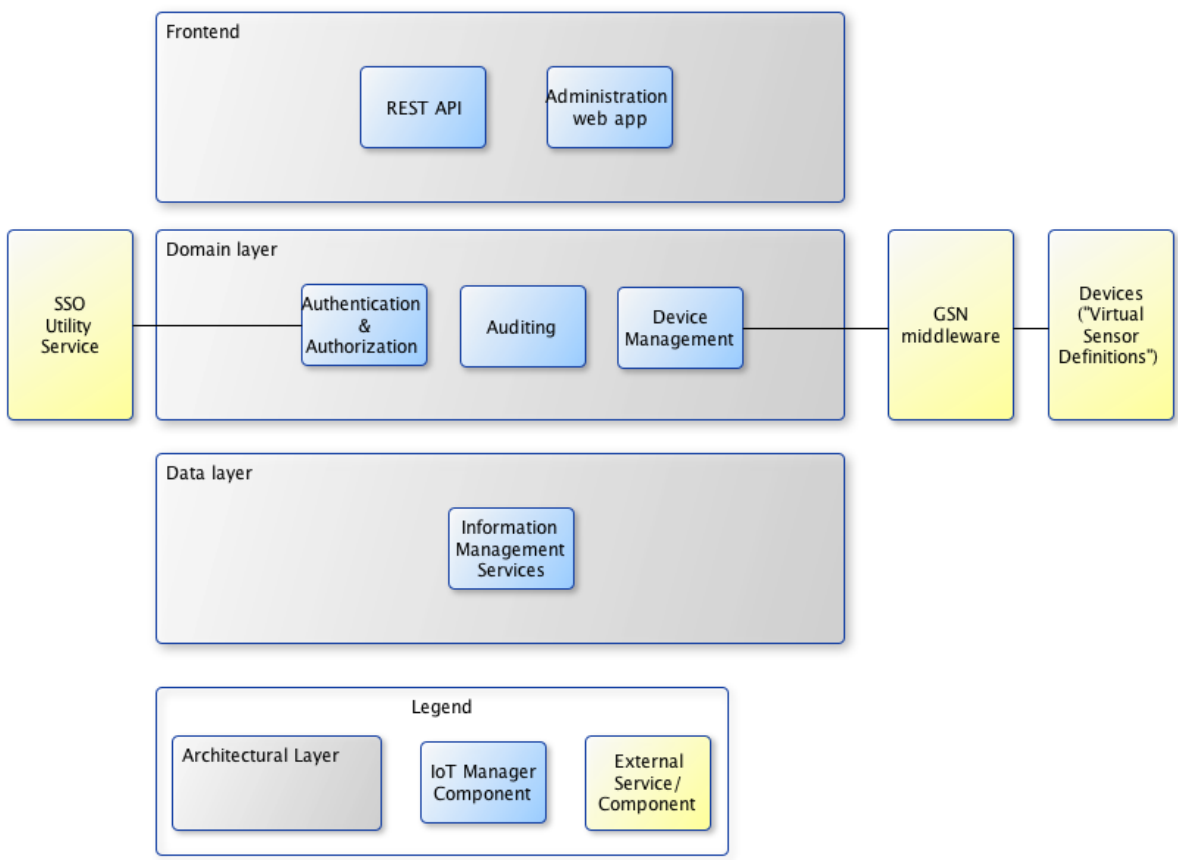

Figure 12: IoT Manager Architecture

The IoT Manager relies on interfacing with instances of the GSN Middleware, which in turn, are interfacing with IoT devices in the field. The architectural components of the IoT Manager are described below:

Frontend:

- Administration web application:

The administration web application is a Java web application with a simple & clean interface for the IoT Manager administrator to:

- Register and unregister GSN instances available for federation
- Manage (create, read, update, delete) Virtual Sensor Descriptions per each GSN instance.
- View logs and audit trails for each GSN Middleware Instance

- IoT Manager REST API

The IoT Manager's REST API provides a programmatic frontend for other MSEE services to take advantage of federated access to sensor data and management. The API provides functionalities for the management of federated sensor networks, registration of sensors and data acquisition from remote sensors for MSEE services and applications.

Domain Layer:

- Device management

The Device Management module interfaces with instances of the GSN middleware via a SOAP API. The GSN mediates communication with devices in the field by providing the following features:

- A SOAP service that provides methods for
  - Management of Virtual Sensor Descriptions (VSDs)
  - Data retrieval, with advanced querying and data stream processing
- A callback-based HTTP streaming method
- A pure push, connection-oriented data push interface.

The Device manager also provides an internal API that allows other components of IoT Manager to communicate with particular GSN middleware instances and obtain information about Virtual Sensors and data streams.

- Auditing

The Auditing module monitors all events and activities in an IoT Manager instance and maintains logs of events, actions and related actors. Logged information includes:

- Administration events such as changes to virtual sensors descriptions
- Device register/unregister
- Subscription updates
- Queries for data
- Commands sent to devices

- Authentication and authorization

The Authentication and Authorisation module handles access control to devices and actions based on specific user privileges. This module interoperates with the Federated Single Sign-On Utility Service developed in WP33 to provide advanced access control services.

Data Layer

- Information Management

This module contains all data sources for the operation of the services (e.g. DataBases, etc)

4.4.5. MSEE Context

In the context of MSEE, the IoT Manager provides a central interface that MSEE applications can use to access external devices. It is used by services developed using the Service Development infrastructure of MSEE and could also registered to the Service Delivery infrastructure. In this respect, the IoT Manager is expected to be frequently invoked, as part of the normal execution of MSEE ecosystem services involving IoT elements.

In the first prototype, development work has focused on establishing a reliable channel of communication with external devices. In the final version, the application will be enhanced to support real-time data acquisition, improved sensor management features and advanced visualisation and alerts.

5. Conclusions

This deliverable updates the current state of research in MSEE regarding the Future Internet Platforms Federation. The deliverable updates D32.1 based on the work performed in the project and on the updates and outcomes of other work packages of the project.

In this context, D32.2 updates the survey in the field of the Future Internet including projects and platforms in the domains of the Internet of Things, Internet of Services, Internet of Contents/Knowledge and Internet for and by the People resulted in the compilation of an initial "short list" of projects and platforms which show strong correlations to the objectives of MSEE and its use cases, and may provide significant added value.

By closely examining the identified MSEE use cases, the usefulness of services offered by external platforms has been determined. Emphasis was put on the FI-WARE platform and its "Applications and Services Ecosystem and Delivery Framework" chapter. Apart from FI-WARE, specific attention was given to the Global Sensor Networks (GSN) platform, constituting the most appropriate alternative to FI-WARE GEs for implementing IoT services.

The two applications selected for MSEE are the "Consumer Marketplace" and the "Internet of Things (IoT) Manager". The Consumer Marketplace is a web-based marketplace, where service consumers can browse and search for service offerings, and is based on the FI-WARE Marketplace Generic Enabler. The "IoT Manager" is a service which allows MSEE applications and service developers to access IP-enabled devices, and is based on the Global Sensor Network Middleware.

Both services were selected on the basis of the perceived value they bring to the MSEE, as well as of the availability of the necessary resources. The selection of the two services, which was initially documented in D32.1, was confirmed and validated in the present deliverable, especially given the results of the updated user requirements analysis in deliverable D52.2 (M15), and the development of the first prototype D32.3 (M18). In the second prototype, these services will be enhanced with additional features and improvements.

Specifically, the technical roadmap for the development of these two services involves the following stages:

Consumer Marketplace
Activities to date
M18: First prototype of the Consumer Marketplace including:

a. Integration with instances of the Marketplace GE Reference Implementation
b. REST API for programming access
c. Web User Interface
d. Registration of service providers (i.e. store owners), administration of stores and service offerings
e. Browsing Stores and Store Offerings
f. Store and service listing by criteria
g. Comparison of service attributes
h. Keyword and phrase annotations of Services and Offerings
i. Basic store federation

M19-M21: Evaluation of the first prototype, design updates and further development towards the final prototype

M21: Updated Specification released (this document)

Future activities

M21-M23: Development of the final prototype:

a. Inclusion of USDL repository for service descriptions
b. Search upgrades
c. Performance improvements when handling Federated Marketplace instances
d. Modification of Marketplace GE Reference Implementation to enhance RDF parsing and indexing for MSEE
e. Development of use case-specific templates for easier USDL generation
f. Integration with Federated SSO Utility Service (WP33)
g. Other enhancements as necessary

M23-M24: General testing and evaluation by Bivolino end users

M24: Final prototype of the Consumer Marketplace including additional features:

a. Access control
b. Complex search via complex criteria and text
c. Assisted service description/publishing via templates
d. "Related offerings" are displayed when viewing a particular service offering
e. Improved performance for Marketplace federation functions
f. Various improvements in existing features as suggested during final evaluation

IoT Manager

Activities to date

M18: First prototype of the IoT Manager including:

a. Integration with instances of the GSN Middleware
b. REST API for programming access
c. Web User Interface
d. Administration of Federated Sensor Networks
e. Administration of individual Virtual Sensors
f. Connection with registered Virtual Sensors
g. Activity logging

M19-M21: Evaluation of the first prototype, design updates and further development towards the final prototype

M21: Updated Specification released (this document)

Future activities

M21-M23: Development of the final prototype including:

a. Addition of a real-time data sink API for high-frequency data
b. Visualisation and alerting features, relying (in part) on real-time capabilities
c. Development of deeper control panel for Sensors and sensor networks
d. Integration with Federated SSO Utility Service (WP33)
e. Other enhancements as necessary

M23-M24: General testing and evaluation by Ibarmia end users

M24- Final prototype of the IoT Manager including the following additional features:

a. Improved Federated Sensor Network and individual sensor management
b. Real time/ High frequency data acquisition
c. Enhanced User Interface with improved visualization features and alerts
d. Access control
e. Various improvements in existing features as suggested during final evaluation

As shown in the technical roadmaps for both applications, the present deliverable provides updated specifications for the services under development, including both functional and non-functional requirements, defining the links to the use cases and describing in what way the selected services could be applied to the updated use case scenarios.

The specifications and the requirements analysis provided in the present deliverable will guide the development of D32.4 “FI Platform Federation final prototype” which will be the main and final outcome of the Project Work Package 32.

6. References

[1] MSEE Deliverable D31.1 ”Functional and Modular Architecture of Future Internet Enterprise Systems”
[2] Future Internet Assembly Meeting Report, Madrid, Spain, 9th – 10th December 2008 retrieved from http://ec.europa.eu/information_society/activities/foi/library/docs/madrid-conference-report-v1-1.pdf
[3] http://www.coin-ip.eu
[4] http://www.srdc.com.tr/projects/isurf/
[5] http://www.sla-at-soi.eu
[6] Software & Services FP7 Project Portfolio, ”Internet of Services, Software and Virtualisation”, Call 5, FP7-ICT-2009-5 – Objective 1.2, JUNE, 2010.
[7] http://www.serenoa-fp7.eu/
[8] Nikos Tsouroulas, “SFE OA Presentation at the R2S Conferene Valencia”, 2010
[9] http://ezweb.morfeo-project.org/
[10] http://www.fp7-aspire.eu/
[11] http://openiot.eu
[12] http://www.cuteloop.eu/
[13] http://webinos.org/
[14] http://www.ict-convergence.eu
[15] http://www.ict-persist.eu/
[16] http://sourceforge.net/apps/trac/psmartspace/wiki
[17] FI-PPP official website (http://www.fi-ppp.eu/about-us/)
[18] European Future Internet Portal: http://www.future-internet.eu/home/future-internet-ppp.html
[19] http://www.fi-ppp.eu/about-us/
[20] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Summary_of_FI-WARE_API_Open_Specifications
[21] http://www.future-internet.eu/home/future-internet-ppp/envirofi.html
[22] http://www.fi-ppp-finseny.eu/
[23] http://www.fi-content.eu/?page_id=63
[24] http://www.finest-ppp.eu/aim-a-approach
[25] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Unclassified_Enablers#FINEST
[26] http://www.fi-ppp.eu/projects/instant-mobility/ Retrieved on
[27] http://instant-mobility.com/index.php/public-documents/public-deliverables.html ,
[28] http://www.fi-ppp-outsmart.eu
[29] http://www.safecity-project.eu
[30] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Testbed_Design
[31] http://en.wikipedia.org/wiki/Internet_of_Services
[32] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Architecture_of_Applications_and_Services_Ecosystem_and_Delivery_Framework
[33] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Architecture_of_Applications_and_Services_Ecosystem_and_Delivery_Framework
[34] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.Apps.Marketplace

[35] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Internet_of_Things_(IoT)_Services_Enablement_Architecture
[36] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FI-WARE_Data/Context_Management
[37] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Security
[38] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Cloud_Hosting
[39] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Interface_to_Networks_and_Devices_(I2ND)
[40] MSEE Deliverable D52.2: "User Requirements Analysis for Virtual Factories & Enterprises in MSEE – M15"
[41] Barros, A. & Kylau, U. : Service Delivery Framework --- An Architectural Strategy for Next-Generation Service Delivery in Business Network. In Kellenberger, P. (Ed.): Proceedings 2011 Annual SRII Global Conference SRII 2011, 30 March - 2 April 2011, San Jose, California, USA, IEEE Computer Society Conference Publishing Services (CPS), 2011, pages 47-58
[42] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Summary_of_FI-WARE_Open_Specifications
[43] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Summary_of_FI-WARE_API_Open_Specifications
[44] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/NGSI-9/NGSI-10_information_model
[45] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/IETF_CoRE
[46] ASPIRE deliverable WP3/D3.4b "Core ASPIRE Middleware Infrastructure (Final Version)", 2010, retrieved from http://www.fp7-aspire.eu/fileadmin/aspire/docs/deliverables/ASPIRE_D3.4b_Final.pdf
[47] Cuteloop Concept, 2008, retrieved from http://www.cuteloop.eu/index.php?aid=12
[48] Furdik, K. (2009) Secure Process Oriented Infrastructure for Networked Enterprises, Workshop on Data Analysis WDA'2009, 2009. Retrieved from http://web.tuke.sk/fei-cit/furdik/publik/wda09_spike.pdf
[49] http://www.fi-ware.eu/our-vision/
[50] http://www.smartagrifood.eu/
[51] http://www.future-internet.eu/home/future-internet-ppp/infinity.html
[52] http://www.future-internet.eu/home/future-internet-ppp/concord.html
[53] MSEE deliverable D26.1 "Innovation Ecosystem Platform"
[54] http://sourceforge.net/apps/trac/gsn/wiki
[55] GSN Team (2009) The Global Sensors Networks. Retrieved from http://gsn.svn.sourceforge.net/viewvc/gsn/branches/documentations/book-of-gsn/main.pdf
[56] MSEE Deliverable 31.2 "Functional and Modular Architecture of Future Internet Enterprise Systems"
[57] Karl Aberer,Manfred Hauswirth, Ali Salehi (2006) The Global Sensor Networks middleware for efficient and flexible deployment and interconnection of sensor networks, Technical report LSIR-REPORT-2006-006, Ecole Polytechnique Federale de Lausanne (EPFL)
[58] MSEE Deliverable D52.1: "User Requirements Analysis for Virtual Factories & Enterprises in MSEE – M4"
[59] MSEE Deliverable D32.1 "FI Platforms Federation Specifications and Architecture – M12"

Appendix

Initial use case analysis and Future Internet requirements (D32.1 - M12)

Indesit

The Indesit use case is the "Carefree Washing" concept. Essentially, it is a servitisation upgrade of the basic washing machine product.

"Carefree washing" will provide a set of services to customers using Indesit washing machines providing personalised technical assistance and allowing remote control of the washing operations. Key elements as described in D52.1 are[58]:

- *"Washing machine monitoring and control"* which will enable remote user control e.g. through mobile apps, data and statistics collection as well as fault monitoring and alerts.
- *"User needs forecast"* which will provide recommendations for new services according to perceived user needs. The suggestions are based on usage data.
- *"Flexible assistance service"* which will provide personalised service, maintenance and warranty schemes, as well as scheduling and requesting maintenance work.
- *"Disposal & recycling service"* which will allow users to schedule the retirement and recycling of the washing machine.
- *"Automatic soap/cleaner recharge"* which will involve monitoring of soap/cleaner levels and the provision of appropriate notifications, automatic pre-orders of refills and can be used to connect customers with marketing activities by cleaning agent manufacturers and distributors.
- *"Health & safety service"* which will monitor the washing machine for unexpected and potentially dangerous faults, such as leakages, machine displacement and vibrations, electrical, mechanical and software faults, etc and will produce alarms. This service is also associated with the Flexible Assistance Service

As indicated in D52.1, the initial requirements analysis includes the following notable functionalities and requirements from SP3:

- Potential use of IoT service enablement FI GEs, including a communication channel to the washing machine for remote monitoring and control
- Service offer descriptions, Service ecosystem support and Service Delivery framework / Utility Services for the service ecosystem
- B2C services: The main components of the "Carefree Washing" service group
- B2B services: support of an ecosystem to provide the B2C services, including remote monitoring and control
- Ecosystem support, mobile channels (for monitor and control), Utility Services and Value-Added Services

Ibarmia

The Ibarmia case involves providing machine tools to customers with a more automated, transparent and less problematic support and maintenance service in order to enhance the machine's availability and thus, the customer's productivity and satisfaction.

The concrete services to be offered in the "Intelligent maintenance services" use case as described in D52.1 will be [58]:

- *"Smart Data Management Service"*: acquisition, transmission and logging of machine behaviour and usage data via sensors.
- *"Smart Data Analysis Service"* (Artificial Intelligence and Data Mining based) of acquired data.
    - Machine-level Online Analysis: the embedded software in the machine analyses sensor data and issues alerts on possible faults
    - Offline Analysis: historical data analysis of collected sensor data to discover usage, behaviour and fault patterns
- *"Alert Service":* whenever a problem or an incidence is detected, this service issues alerts to maintenance personnel and subcontractors
    - Send notifications (by email or SMS) to maintenance actors
    - Connect to an Enterprise Information System to perform the actions such as ordering spares, keeping logs, etc. Some interoperability between the customer's EIS and Ibarmia's systems may be present.

As indicated in D52.1, the initial requirements analysis for this case, in relation to SP3, includes the following notable points:

- Support for the Provision and control new services using IoT and IoS capabilities, including a communications channel for remote monitoring and control
- Service offer description
- Access to Enterprise Services
- Event/process aggregation, service composition
- Channel makers
- Support for a Service delivery platform, Utility Services, Instantiation of Service Delivery Framework and Interoperability.
- B2C services: Support for the "Intelligent maintenance services" service group
- B2B services: Support for a B2B ecosystem

Bivolino

As described in D52.1, Bivolino introduces the following two cases into MSEE[58]

"Shirt-as-a-Service" (B2C service)
Customers who have designed and bought a shirt through Bivolino.com will be able to subscribe to a "laundering service". The shirt will be washed by a nearby laundry partner, for example every week during 1 year at a specified maximum frequency over a time period or for a number of washings. The shirt may be tagged through RFID or scannable QR codes and a pick-up/drop-off service may also be included. At the end of the shirt's lifetime, the customer may be prompted to purchase a new subscription with a new shirt.

"Interoperable cut files for third Manufacturer" (B2B Service)
Bivolino shirts may be configured and ordered through e-retailers that use the BivolinoServices.com online configurator. The retailer will usually place an order for a configured shirt with a manufacturer already in Bivolino's manufacturing ecosystem.

The use case involves providing the possibility of placing an order with a third party manufacturer, as long as the manufacturer has digital fabric cutters. In this case, BivolinoServices will create a compatible "cutfile" and relevant materials such as production instructions, auxiliary component lists (buttons, threads, packaging, etc.), 2D pattern pieces, 3D visual references, customer invoice, contact details and tracking information for delivery. The cutfile is imported, the shirt is then constructed and delivered. In some cases,

BivolinoServices may act as a broker enabling third party manufacturers to lease a compatible cutter from another source for two years, before gaining ownership.

As indicated in D52.1, the initial requirements analysis for this case, in relation to SP3, includes the following notable points:

- Service offer descriptions and channels for services, support for service ecosystem
- Service delivery framework and utility services for service ecosystems
- Extension of ecosystem with new partners such as laundries, e-retailers
- B2C services: Possible IoT services for the laundry service (corresponding to the Shirt as a Service scenario)
- B2B services: Interoperability between Bivolino and e-retailers

Philips

As described in D52.1, the Net TV product is, essentially, an internet-enabled television set that offers a set of applications/ services via the Net TV portal. These services includes social media , news services, catch up TV services, video services and video on demand, games and services for the elderly. At this moment, the service portal is accessible in 36 countries, and includes various location-specific applications (e.g. applications for local news).

The Net TV service ecosystem is composed of several entities, and includes the actual Net TV devices, the afore-mentioned service portal, individual applications, a device portal (enabling Net TV device certification and access to the service portal), an advertising management system. A new addition to the ecosystem is the inclusion of a "Partner Portal", where ecosystem partners can develop and register new Net TV applications.

Philips is already using the Net TV service as a way to differentiate the Net TV product, and is not planning any further transition to higher levels of servitisations. For this reason, the contribution of MSEE is to enable Philips to monitor strategic Key Performance Indicators as the Net TV ecosystem expands, by adding several new applications and partners. [58]

As indicated in D52.1, the initial requirements analysis for this case, in relation to SP3, includes the following notable points:

- Next-generation NetTV devices may include novel capabilities such as new types of Digital Rights Management and new interfaces
- Services to support the lifecycle process of services and apps of the NetTV ecosystem
- A Service Delivery Framework and Utility Services for collaboration and feedback
- B2C services: Go Marketplaces
- B2B services: Outsourcing the application certification process and new channel for app developers such as social networks and app stores

Use case analysis and commonalities

In the following section, an initial synthesis of the use case requirements is presented in an attempt to identify common elements encountered in the MSEE use cases. These elements will support determining the specific areas where the provision of applications will achieve synergies thus offering the maximum benefits. In this analysis, both "core" elements of the use cases, i.e. elements that are either explicitly required by the servitisation process or the use case descriptions above, and "optional" elements, which are either implicit or may extend

the added value of the use cases are considered. It has to be noted that this analysis will be finalised in the final version of this deliverable (D32.2).

Internet of Services

In D31.1, the use cases (except Philips which is pending the selection of candidate services) were deconstructed using product-service bundles and collaboration diagrams. The D31.1 deliverable points out that there exist common actor roles for all or most analysed use cases, which correspond to the Service Delivery Framework described in D43.1 and D31.1.

After the analysis in D31.1, the use cases contain instances of the following roles:

- Customer. The final consumer of the service
- Provider/producer. The provider of the selected services, namely, the use case owner (for example see the cases from Indesit, Ibarmia, etc.)
- Broker. A Marketplace/Store where customers discover, compare and purchase services from providers, and where services are monetised
- Gateway. An interconnection to externally provided services for functional tasks such as payment or logistics support, corresponding to external systems or platforms (e.g. such as utility services or FI platforms)

The Channel Maker role, while not explicitly different within all use cases, corresponds to the expected ability of the use cases to utilise multiple channels regarding the provision of services, such as mobile devices, SmartTVs, etc.

The analysis performed in D31.1[1] concludes that the Marketplace and the Store can be considered to be separate, with the Marketplace being a service provided by the ecosystem itself, facilitating service discovery and comparison, while the store is provided by the service provider and supports publishing and transactions.

Given the ubiquity of the Marketplace and Store elements, they appear to be a prime candidate for implementation as a service using federation with Future Internet Platforms. The Indesit, Ibarmia and Bivolino requirements indicate that “service offer description” is necessary. The Philips case also requires this element from SP3, specifically mentioning “new channels for app developers”. For this reason, marketplaces, i.e. mechanisms for End-user service discovery are considered to be a core element of all use cases.

Also, Service Level Agreement management and Key Performance Indicators are important facet of the servitisation process[58]. SLA management and KPI collection (business intelligence) should be present in all cases. SLAs are intertwined with quality service provision; therefore, we consider SLA management to be a core element of all cases. KPI collection is only explicitly required by the Philips case, and is optional for all others as it is not a central element of their business cases.

Internet of Things and Access to Devices

Three out of the four MSEE use cases (Indesit, Ibarmia, Bivolino) revolve around the IoT concept. The first two include sensors and embedded devices (Indesit includes both sensing and control of washing machines), as core elements, while the third may use RFID to track items. This functionality is considered as a core element for these use cases. Note, however, that for Bivolino, the use case requirements indicate that an alternative method may involve scannable QR codes[58].

In the Indesit and Ibarmia cases, RFID functionality may be used to track spare parts and other supplies, as a part of the maintenance services provided, although this is not explicitly required, thus it is considered to be optional.

Interfaces to multiple devices are relevant both as part of IoT implementations, as well as the provision of services via various types of platforms, such as smartphones and tablets, in conjunction with context management functionalities. The Indesit and Ibarmia cases specifically require the use of a mobile channel communications for device access and control. The Bivolino case will also provide a mobile channel for the 'Shirt as a service' scenario for consumers[58].

Data and Context Management

All cases require functionalities that involve the analysis of large amounts of real-time information, such as measurements and events (Indesit and Ibarmia for sensor data, Bivolino, optionally for shirt locations, Philips for KPI management).

As expected in the case of ecosystems, interoperability is also an element of all cases. The Bivolino "Interoperable cutfile" service requires metadata handling and transformation as a core element of its business process. In the Indesit and Ibarmia cooperation and data exchanges between ecosystem partners (e.g. local maintenance providers). In the Philips case, interoperability may be involved in the application lifecycle process, although this is, at the moment, unclear. For this reason, consider interoperability to be central to the Bivolino, Ibarmia, and Indesit cases.

The Philips use case is focused on the provision of Key Perfomance Indicators for the NetTV ecosystem. Since the NetTV ecosystem provides several content-based services to end users, content-tracking applications may provide some business intelligence to ecosystem managers and this is considered to be an optional element of the Philips use case at this stage.

Context management seems relevant to location tracking and semantic annotation of context data. This may useful in the provision of services via mobile terminals to end users, either in a B2C or B2B context. For example, the Ibarmia case mentions the use of mobile terminals (smartphones, tablets, etc) as a means to monitor and control devices. A context-enabled service may allow technicians' tablets to retrieve machine data from the system as they approach Ibarmia-supplied equipment. Such applications may also be possible in the Indesit case for consumers (e.g. as a consumer returns home, washing machine data and alerts are immediately uploaded to their smartphone). In the Bivolino case, location awareness may, for example, allow for product delivery to a user's location as reported by their smartphone. These functionalities are not explicitly required by the use cases and may be considered optional.

Internet for and by the People

Personalisation may prove important in business-to-consumer use cases, especially Indesit and Bivolino. Both the "Carefree washing" and "Shirt as a Service" scenarios require the differentiation of service offerings according to the customer's personal attributes and lifestyle. In the Indesit case for example, washing machine maintenance will have to be scheduled according to washing loads and the customer's schedule. In the Bivolino case, customers' measurements and style preferences may be used in combination with manufacturing ecosystem attributes (e.g. fabric availability, stocks, etc) to provide personalised suggestions for shirts "as-a-service". These are core elements of both use cases.

The Bivolino (shirt as a service), Indesit and and Philips cases, have B2C elements and may involve interoperation with social media platforms, especially since those platforms are

becoming the main communication providers for consumers. For example, notifications regarding laundered shirts could be sent directly from the Bivolino ecosystem to the customer's social media account. In the Philips case, such interoperation is mentioned as a requirement for SP3 as a way to provide alternative channels for app developers. In the Bivolino and Indesit case, integration with social media is mentioned in [40] in the context of SP2. For this reason, this functionality is optional in this analysis.

The Ibarmia, Bivolino and Philips use cases implicitly require the active collaboration of ecosystem participants. In this case, the Ibarmia case will involve communication between technicians from equipment procurers and customers. Collaboration between Bivolino personnel and clothing manufacturers may be involved in the Bivolino case, especially as the former instruct the latter in the creation of Bivolino-designed garments. Also, Bivolino personnel will coordinate with other service providers such as laundries for shirts in the shirt-as-a-service scenario. In the Philips case study, we expect collaboration between Philips ecosystem managers and partners (e.g. developers for app certification as mentioned in [40]). The Indesit case may involve collaboration between ecosystem professionals (e.g. third party maintenance service professionals, parts suppliers). On the other hand, from a business-to-consumer perspective, Indesit customers may request ad-hoc technical support from technicians, e.g. via instant messaging, and Bivolino customers may contact service representatives (from Bivolino or other providers e.g. laundries) with queries and requests. As they are not explicitly required, support for these elements is considered optional at this stage.

Finally, it is important to note that service feedback collection is potentially present in all four cases. This functionality has been proposed in WP33 as a Utility Service and will not be examined in this deliverable.

Security, Trust and Privacy

All services require user authentication, security and privacy functionalities. Access to personalised services requires the use of secure credentials. Also, especially in the case of "servitised" consumer products, there is significant need for data usage control and privacy assurance. However, a Utility Service has been proposed in WP33 that may cover such needs.

Cloud and Network Infrastructures

Finally, cloud enablement may be useful, however it must be noted that various cloud-related functionalities are mostly relevant to MSEE's infrastructure layer, which may be beyond the scope of this study.

Functionality	Indesit	Ibarmia	Bivolino	Philips
End-user service discovery	Core functionality	Core functionality	Core functionality	Core functionality
SLA Management	Core functionality	Core functionality	Core functionality	Core functionality
KPIs and business intelligence	Optional	Optional	Optional	Core functionality
IoT device communications (sensors, actuators, etc)	Core functionality	Core functionality	-	-
RFID object tracking	Optional (RFID tracking of parts)	Optional (RFID tracking of parts)	Core functionality (RFID tracking of shirts) – note scannable QR codes may be selected instead	-
Real time data and event processing	Core functionality	Core functionality	Core Functionality (tracking of shirts)	Core functionality

				(KPI collection)
Interoperability	Core functionality (ecosystem data exchanges e.g. orders, deliveries)	Core functionality (ecosystem data exchanges e.g. orders, deliveries)	Core functionality (Interoperable Cutfile, ecosystem data exchanges)	-
Mobile access channels	Core functionality (device data and control)	Core functionality (device data and control)	Core functionality (service status information for consumers)	-
Context/location aware services	Optional	Optional	Optional	-
Ecosystem participant communication / collaboration	Optional (Ad-hoc maintenance)	Optional (Technical personnel collaboration)	Optional (Collaboration with manufacturers / communication with customers)	Optional (ecosystem partner collaboration)
Personalised services	Core functionality	-	Core functionality	-
Content tracking	-	-	-	Optional
Social media integration	Optional	-	Optional	Core functionality

Table: Use case commonalities

Detailed technical descriptions and evaluations of FI-WARE chapters considered for use (D32.1 – M12)

Internet of Things Services Enablement

FI-WARE, in recognition of the future expansion of the IoT concept, has a chapter dedicated to providing Generic Enablers for IoT applications. These enablers and their dependence on other FI-WARE chapters are illustrated in the following figure [35]:

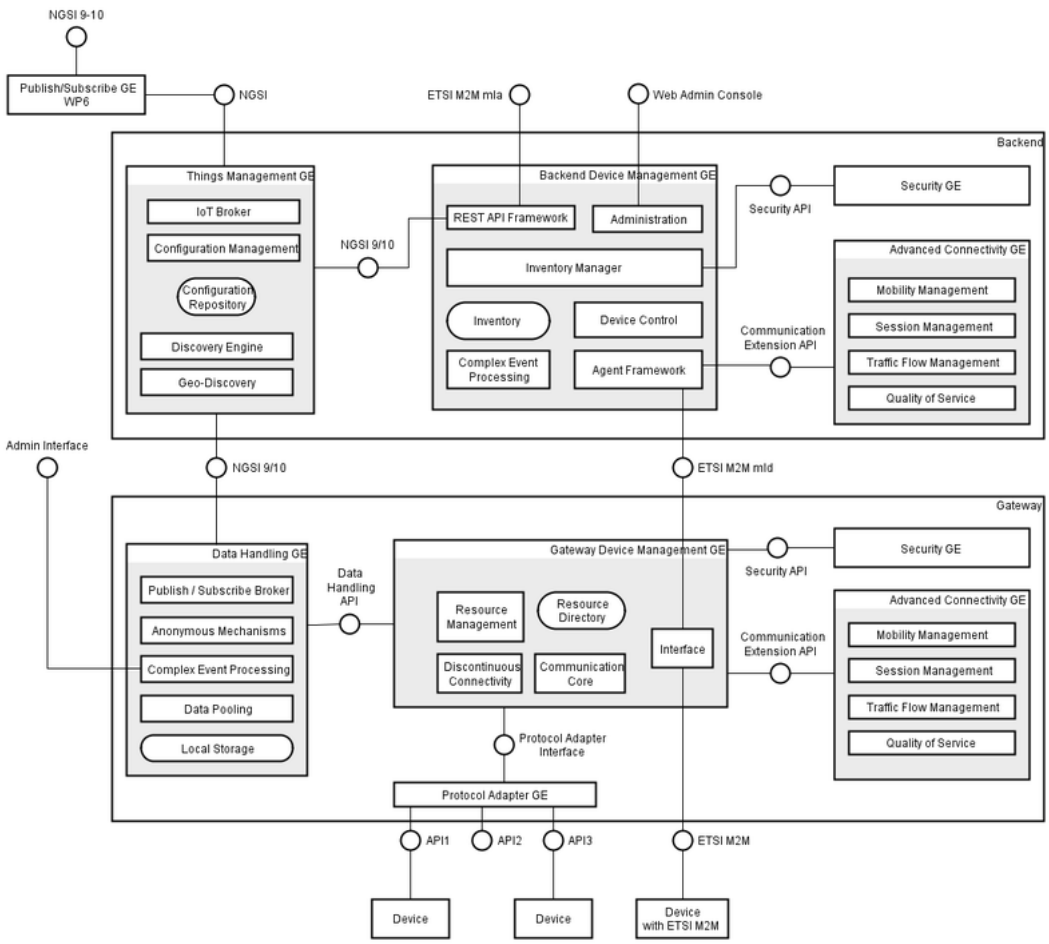

Figure 13: FI-WARE GEs in the Internet of Things services enablement Architecture[35]

This chapter uses three abstraction levels: a Device level (corresponding to hardware units capable of activities such as measurement or actuation), a Resource level (computational elements providing access to Devices) and a Thing level (any object, person or place in the physical world).

IoT Generic Enablers are organised into two main functional areas:[35]

- **Backend:** provides management functionalities for the devices and IoT support for applications. It contains the Things Management GE, the Backend Device Management GE, the Advanced Connectivity GE and an IoT Backend Security GE. A backend Template Handler GE (for the definition and execution of IoT-aware business process is included but not pictured)
- **Gateway:** provides inter-networking and protocol conversion functionalities between devices and the IoT backend. It is usually located at proximity of the devices to be connected. and contains the Gateway Device Management GE, the Data Handling GE,

the Protocol Adapter GE, the Advanced Connectivity GE and an IoT Gateway Security GE.

Evaluation

The IoT Enablers are a good match for MSEE as far as it concerns their specifications. As noted before, two out of four use cases make use of IoT concepts. Indesit and Ibarmia need to communicate with devices away from the companies' premises.

As mentioned before, the Internet of Things chapter uses the Device, Resource and Thing abstractions. Devices correspond to physical devices, resources are computational elements providing access to devices and things are objects, people, contexts, etc. It also uses ETSI M2M APIs to communicate externally, as well as an NGSI - 9/10 interface[44] to cooperate with the Publication/Subscription GE.

On the Backend side, the Backend Things Management acts as a point to access information about entities and their attributes[22]. It provides interfaces (using an FI-WARE specified NGSI-9/10 interface) using the PubSub broker GE as a front-end for:

- Registration of IoT agents
- Query handling (i.e. queries for information from IoT agents)
- Subscription handling (i.e. "subscriptions" to IoT agents)
- Notifications (i.e. notifications from IoT agents after subscription)

Current API specifications[23],[24],[25] support registration, update and subscriptions.

The Backend Device Management GE[26] corresponds to a "lower" abstraction level, providing access on a resource-level of remote assets (devices with sensors and actuators) and includes communication capabilities, such as basic IP connectivity and handling disconnected devices. Its main interactions include:

- Retrieve Device Information
- Sending Control Operations to Device
- Device Push Update (e.g. new firmware)
- Device Registration Southbound (the device self-registers with the GE via internal programming)
- Measurement collection

Both "Northbound" (towards higher abstraction levels) and "Southbound" (lower abstraction levels) APIs have been defined, as well as communication with the Device Management GE.

On the Gateway side, the Gateway Device Management GE[27] contains the core functionality of the IoT Gateway and handles both Resource management (CRUD operations on the

[22]http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.IoT.Backend.ThingsManagement
[23]http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/TM_GE_Southbound_Interface_Open_RESTful_API_Specification_(PRELIMINARY)
[24]http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/TM_GE_Northbound_Interface_Open_RESTful_API_Specification_(PRELIMINARY)
[25]http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/TM_GE_Southbound_Interface_Open_RESTful_API_Specification_(PRELIMINARY)
[26]http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.IoT.Backend.DeviceManagement
[27]http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.IoT.Gateway.DeviceManagement

Resource Directory for IoT resources) and Resource Access (communication with resources also via the Protocol Adapter GE and the Data Handling GE)

The Gateway Data Handling GE[28] filters, aggregates and merges data from different sources, using a subscription model (and an NGSI-9/10 interface) and including complex event handling, local storage of data, and data pooling. Current API is at a draft stage.

The Gateway Protocol Adapter GE[29] deals with incoming and outgoing traffic from the Gateway and the registered Devices.

Data and Context Management

The Future Internet is expected to contain large amounts of data and applications that need to capture and process this data. The FI-WARE Data/Context Management Chapter aims to provide Generic Enablers to cover this functionality.

FI-WARE handles data in the form of data elements, having an assigned type, attributes and associated meta-data. Context elements extend the concept of the data element, by adding information on the specific "entity" of the FI-WARE that the data are referring to, thus adding "context" to data. Finally, the chapter supports "events" and "event objects" i.e. handling information about occurrences within a system.[36]

The logical structure and interrelations of those enablers is illustrated in the following figure [36].

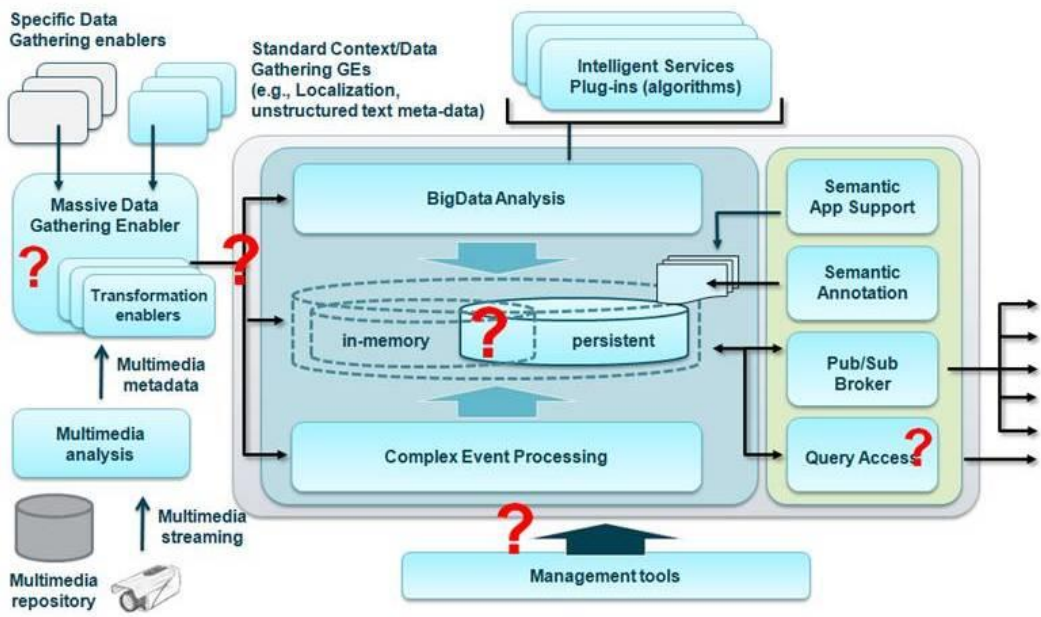

Figure 14: FI-WARE Data/Context Management Enablers[36]

The main Generic Enablers included in this chapter are according to[36]:

- **Publish/Subscribe Broker GE:** Events exchange from "publishing" entities to "subscriber" entities
- **Complex Event Processing GE:** Real-time events stream processing
- **Big Data Analysis GE:** analysis of large data sets, either in real time or off-line

[28] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.IoT.Gateway.DataHandling
[29] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.IoT.Gateway.ProtocolAdapter

- **Multimedia Analysis Generation GE:** Extraction of meta-information from multimedia
- **Unstructured data analysis GE:** Extraction of metadata from unstructured data
- **Meta-data pre-processing GE:** Supports conversion of metadata formats and the creation of objects carrying metadata
- **Location GE:** Device geo-location information as context information.
- **Query Broker GE:** Provides a uniform query mechanism for data retrieval
- **Semantic Annotation GE:** Annotation of multimedia and other data with semantic information
- **Semantic Application Support GE:** Supports functionalities for applications operating with semantic data

In addition to these, the Chapter will include additional enablers, based on the specific demands of individual users of FI-WARE assets. These will cover the areas of:

- Social Network Analysis
- Mobility and Behaviour Analysis
- Real-time recommendations
- Behavioural and Web profiling
- Opinion mining

Evaluation

Data and context management seems promising for utilisation under various MSEE applications. All cases may benefit from location services, as well as the implementation of a Publish-Subscribe mechanism for context data and a Query Broker for access to structured data.

The Publish/Subscribe Broker GE[30], also known as the "Context Broker" GE supports the publication of event information by entities, referred as Context Producers, making it available to Context Consumers, interested in this information. This Enabler supports the following basic functionalities:

- Exchange of Context Elements between Context Producers and Context Consumers
  - Context Elements are generic data structures following the OMA NGSI-9/10 model. Context Elements contain information about an entity's identity, a ste of one or more attributes, as well as metadata about the attributes themselves.
- Context caching
  - Caching context elements to ensure availability to context consumers
- Context Validity
  - Checking against preset expiration periods and timestamps
- Context history
  - Logging of context elements

The GE is based on a NGSI-9/10-style API, as well as an ContextML/CQL over HTTP Open RESTlike API.

The Complex Event Processing GE operates on event data in real-time, and enables instant responses from applications, and supports event-based routing, observation, monitoring and event correlation. It also enables defining and maintaining event processing logic on behalf of other applications.

[30]http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.Data.PubSub

The GE and acts as a mediator between Event Producers and Event Consumers. Also, it may forward events to the Publish/Subscribe Broker GE, which will make them accessible to entities using a Publish/Subscribe model.

The CEP GE operates based on Event Processing Networks, composed of interconnected Event Processing Agents. The GE supports Pattern Detection, rules regarding specific conditions and relevant actions. At this moment, its API supports reception and retrieval of events.

The Location Platform GE[31] provides location services to:

- Third party location clients, using the Mobile Location Protocol or a RESTful Network API for Terminal Location, standardised by the OMA.
- Mobile end-users, which try to determine their location by using applications querying the GE. The GE computes the position based on the data provided. This usage scenario includes the possibility of sharing location information via the Secure User Plane (SUPL) OMA standard.

It comes into play in difficult environments (in terms of location tracking) by using GPS and alternative techniques. The GE also supports privacy and security protection for those services. In its current release, it has a RESTful API supporting location queries, subscription to periodic notifications, and subscriptions to periodic notifications of device location within an area.

The Query Broker GE[32] provides a mechanism for retrieval of data from the FI-WARE data management layer, in addition to the Publish/Subscribe Broker GE. It provides support for integration of query functions into applications by abstracting the access to databases and search engines available in the FI-WARE data management platform as well as access outside data sources. It supports highly structured data such as results of database queries (e.g. SQL), and less structured, like XML. The data involved may range from text to multimedia.

This GE will be capable of federating multiple retrieval services, paradigms and APIs. Its API specification supports receiving queries (based on MPQF), registration and de-registration of a data source and the creation of semantic links.

Security

Security, trust and privacy are critical elements of the Internet now, and are expected to be fundamental parts of any Future Internet applications. A core feature of Internet services is personalisation, and they rely on identity management functions to deliver customised value to users. Trust cross platforms enables the seamless interoperability of platforms, safeguarding both tangible and intangible assets, such as data, processes and systems. Finally privacy considerations are necessary to reach a safe compromise between the disseminative nature of ICT technologies and the real-world need to control and manage the spread of information.

FI-WARE specifies a set of Generic Enablers dealing explicitly with those issues. These enablers and their interrelations with other parts of FI-WARE and external actors is illustrated in the following figure [37]:

[31] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.Data.Location
[32] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.Data.QueryBroker#Introduction_to_the_Query_Broker_GE

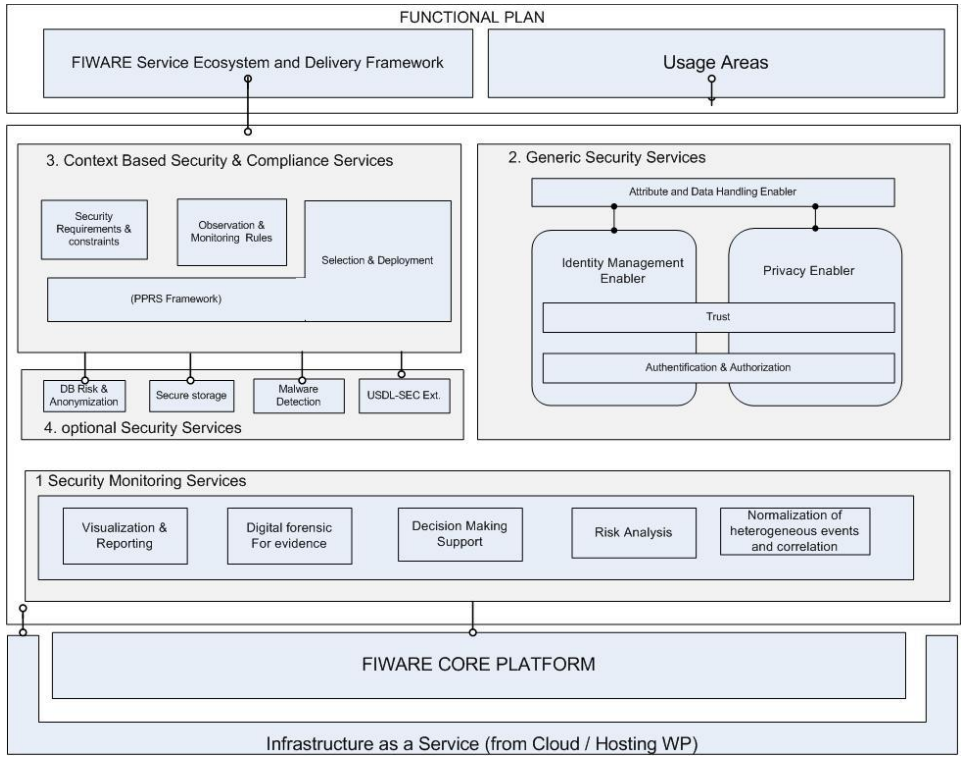

Figure 15: FI-WARE security enablers and associations[37]

The Security Chapter includes the following Generic Enablers according to [37]:

- **Security Monitoring:** An enabler to act as the security framework for all FI-WARE instances, applications and services. It operates as a monitor and coordinator for all other security enablers
- **Identity Management:** Supports user management, authentication, access control, administration and external directory services
- **Privacy:** Supports enhancing Privacy aspects of Identity Management services (authentication, user management, access control, etc) with special credentials disclosed on a "need-to-know" basis to other entities
- **Data Handling:** Supports controlling the usage of data by associating data usage policies to specific data items or sets, as well as providing cryptography functionality
- **Optional Security Services:** Domain or usage specific security services. They are also activated by the Context Based Security and Compliance Enabler
- **Context based security and compliance:** This enabler supports the dynamic configuration of access and security policies depending on the specific context of use i.e. a customised Optional Security Service for particular usage and application domain

Evaluation

The security chapter contains several enablers that relate to security, privacy and identity management. Although necessary for all use cases and applications, identity management will be handled in the MSEE platform via the WP33 Federated Single Sign-On Utility Service.

Cloud

The emergence of cheap and highly reliable communications technologies, as well as massive computing and data infrastructures has facilitated the emergence of Cloud Computing.

Applications and data need no longer to be hosted at or near the point of consumption; they can be located in infrastructures away from the user but offered transparently, without having to be aware of the location or details of their physical hosting, by being "in the cloud".

The same advances have resulted in the practice of providing assets and products "as-a-service". In this paradigm, users no longer own tools, applications and infrastructures as these are, in a sense, outsourced to service providers who supply the use of those tools, applications and infrastructures via subscription models. Such business models are the, now ubiquitous, "Software-As-A-Service", "Platform-as-a service" and "Infrastructure-as-a service" provision schemes. With further technological advances, cloud services are expected to become one of the dominant modes of service provision in the Future Internet.

FI-WARE supports cloud hosting by providing a set Generic Enablers for this purpose. These Enablers and their relationships are illustrated in the following figure [38]:

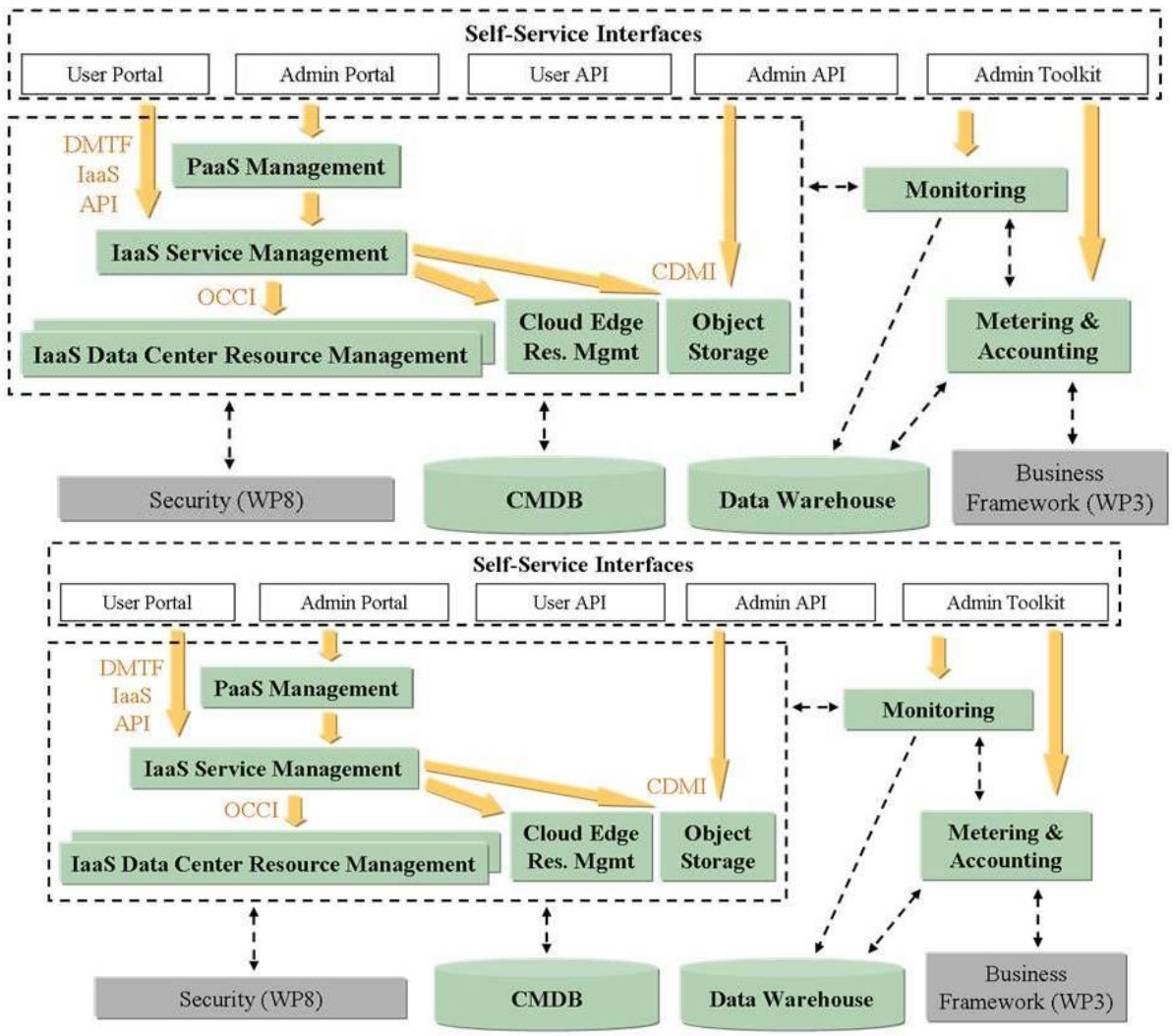

Figure 16: FI-WARE cloud hosting enablers and associations

The Cloud Hosting Generic Enablers include the following, according to [38]:

- **IaaS Data Center Resource Management:** Provides VM hosting capabilities to end users, and manages VM resources
- **IaaS Cloud-Edge Resource Management:** Allows application developers to use resources close to the end users i.e. at the "Cloud Edge"
- **IaaS Service Management:** hosting and management of compound VM-based services and uses the IaaS Resource Management GE to handle individual VMs

- **PaaS Management:** hosting application containers
- **Object Storage:** storage and retrieval of objects and metadata.
- **Monitoring:** collecting metrics and usage data of the various resources in the cloud.
- **CMDB:** storing the operational configuration of the Cloud environment, used GEs.
- **Data Warehouse:** Storage of usage data (collected by the Monitoring GEs)
- **Metering & Accounting:** Collection and processing the data on usage and monetisation of cloud services (via an external Billing system).

Evaluation

In general, Cloud Hosting GEs are intended for use by providers offering Infrastructures-as-a-Service, Platforms-as-a-service, etc. and operate at an infrastructural level. Their usage scenarios do not correspond directly to MSEE use cases. For this reason, Cloud Hosting GEs are beyond the research scope of service development in WP32.

Interface to Networks and Devices

Nowadays, Internet services are consumed over a large variety of devices, platforms and networks. This leads to significant difficulty in creating applications and services that are at the same time interoperable and accessible via the full range of potential software and hardware frameworks. Interoperability is already a major concern and Future Internet applications will need to handle it successfully.

FI-WARE introduces the concept of "Intelligent Connectivity" as a central feature of the Future Internet. It involves the intelligent management of network connections of applications and devices, while, at the same time, exploiting the individual characteristics of each platform. This is achieved through the introduction of specialised Generic Enablers.[39]

The Enablers of the Interfaces to Networks and Devices chapter and associations with external entities are illustrated in the figure below [39]:

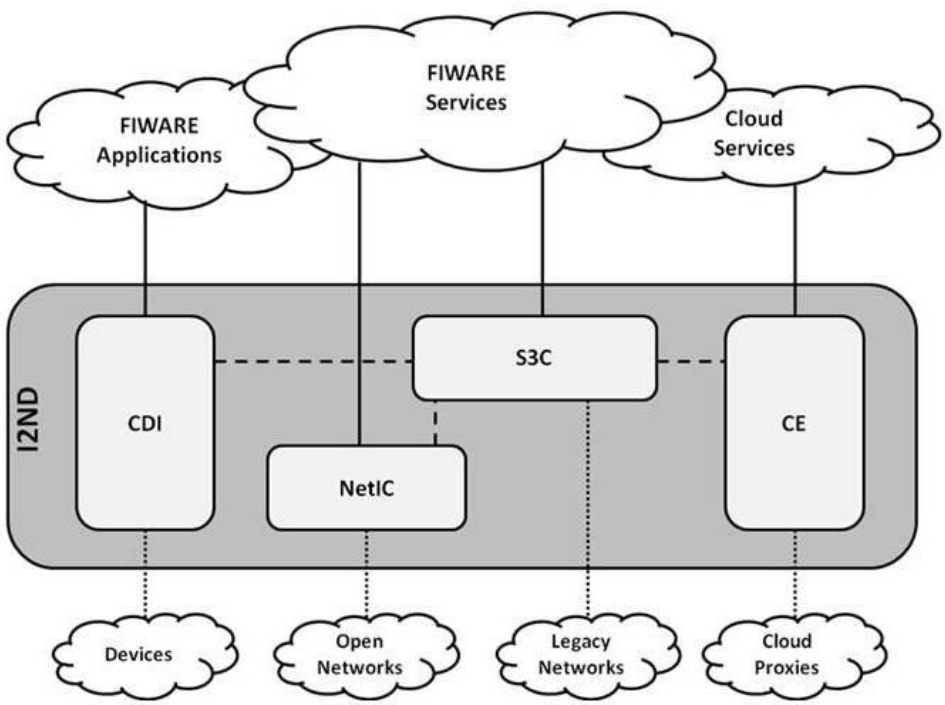

Figure 17: FI-WARE Interface to Networks and Devices Enablers[39]

This chapter includes four Generic Enablers according to[39]:

- **Connected Device Interfacing (CDI):** Interfaces and APIs for connected device, including metadata such as locations, status, etc
- **Cloud Edge (CE):** Interfaces Cloud Proxies with the FI-WARE and legacy cloud services, including interfaces for end-device communication, home system management, virtual machine management, protocol adaptation, etc.
- **Network Information & Control (NetIC):** Homogeneous access to heterogeneous open networking devices, it exposes network status information and enables limited network programmability
- **Service, Capability, Connectivity and Control (S3C):** Access to legacy network devices, capabilities and services.

Evaluation

The Interface to Networks and Devices chapter aims to provide interoperability functionality across devices and platforms. The 1st release of FI-WARE includes the Cloud Edge GE[33], which shows significant promise, and may fit well with the Indesit and Ibarmia use cases.

The "Cloud Edge" concept refers to the border between the "Cloud" and user networks. A "cloud proxy" device is located in the user network and handles links between end user devices and the cloud itself. It stores data within in the user network, and organises access to user devices. The cloud proxy alleviates the problem of harmonising the low bandwidth capabilities of user devices with a permanent, high bandwidth, connection to the cloud.

The Cloud Edge GE introduces the following concepts:

- Virtual Appliances, which are types of Service that the cloud proxy can host: an OS and applications run on top of the virtualised system supported by the Cloud Proxy.
- Images, which are files composing a virtual appliance and associated metadata
- Instances which are virtual machines that run the Service. They are instantiations of a Virtual Appliance.

Main operations of the Cloud Edge GE include:

- Platform features operation: provides information about the platform itself and the resources that can be shared or offered to virtual applications.
- Images Features operation: management of images that are available on the cloud proxy.
- Instances Features operation: Management of Instances that runs on cloud proxy.
- Users Features operation: Management of the user authentication and authorisation.
- Monitoring Feature operation: Provision of information about the state and the behaviour of any Instance.

[33] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.I2ND.CE

Brief evaluation of services from other FI-inspired platforms considered for use (D32.1 – M12)

As noted in Chapter 2, there exists a significant wealth of other Future Internet-inspired platforms that cover areas relevant to the project and may also provide with added value in the form of resources and tools.

On the side of Internet of Services, COIN and iSURF were already examined in WP33, mostly focusing on their implementation approach for the Interoperability Service Utility (ISU) concept. COIN in particular supports federation, and provides a significant set of potentially useful interoperability and collaboration services.

SLA@SOI aims to support a critical aspect of service delivery: Service Level Agreements. The open source tools provided may provide support for this concept, especially considering that the SLA Management GE from FI-WARE will be available in its second release, and will be unavailable for MSEE prototypes.

SERENOA combines a service front end with elements of context management, potentially enabling the creation of context-aware UIs for services through mobile channels. EzWeb, on the other hand, could potentially support a user interface for service discovery, as it explicitly supports the concept of a Consumer Marketplace.

In the Internet of Things domain, good candidates for services reuse are OpenIOT and ASPIRE

OpenIOT builds on the experience of GSN and ASPIRE, along with other projects, and introduces the "Sensing as a service" concept. The OpenIOT middleware may provide an opportunity to construct sensing applications, and, more importantly, services, which can then be provided to the MSEE platform and use cases, especially Indesit and Ibarmia, and Bivolino, should RFID tracking be selected.

Webinos also provides a framework for running common applications on IoT Enabled devices, in conjunction with mobile and static terminals (e.g. smartphones, tablets, desktop computers).

In the area of Internet of Content/Knowledge, the CONVERGENCE project is especially attractive, as it will provide tools for digital content management. The TPVision use case involves the consumption of digital content over the NetTV ecosystem.

The PERSIST project provides tools for the creation of personalised services, especially tailored to each end user. This has the potential to provide added value in use cases that involve business-to-consumer transactions, esp. Indesit and Bivolino.