D32.1 – Future Internet Platform Federation Specifications and Architecture – M12 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:	01 11 2012
Revision:	1.0

VERSION HISTORY

VERSION	DATE	NOTES AND COMMENTS
0.1.1	31.5.2012	TRANSFERRED TO MSEE TEMPLATE
0.1.2	21.6.2012	ANNOUNCED TO PARTNERS
0.1.3	19.7.2012	REASSIGNMENTS TO PARTNERS
0.1.4	6.9.2012	COMMENTS BY SAP
0.1.5	13.9.2012	IMPROVEMENT OF CHAPTER 2.1 (SAP)
0.1.6	14.9.2012	CONTRIBUTIONS 3.2, 3.3 (SAP)
0.1.7	20.9.2012	CONTRIBUTIONS 2.2 (ENG)
0.1.8	21.9.2012	CONTRIBUTIONS 2.1 (ENG)
0.1.9	25.9.2012	CONTRIBUTIONS 2.3, 3.3 (SAP), IMPROVEMENTS (SINGULAR)
0.1.10	26.9.2012	CONTRIBUTIONS 2.7, 2.8(SINGULAR)
0.1.11	26.9.2012	CONTRIBUTIONS CHAPTER 3 (SINGULAR)
0.8	26.9.2012	INTERNAL REVIEW VERSION
0.93	29.9.2012	REVIEW CYCLE 1 – INITIAL COMMENTS- DRAFT
0.96	23.10.2012	REVIEW CYCLE 1 – READY FOR REVIEW
0.99	29.10.2012	REVIEW CYCLE 2 – READY FOR REVIEW
1.00	01.11.2012	FINAL VERSION

DELIVERABLE PEER REVIEW SUMMARY

ID	**Comments**	**Addressed (✔) Answered (A)**
1	Remove direct content from other deliverables, summarise and reference appropriately	✔
2	Re-examine relation with Service Delivery Platform (WP43)	✔
3	Specify the innovative characteristics of marketplace application in first prototype	✔
4	Consider including Complex Event Processing in IoT application	✔

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

FI: Future Internet
B2B: Business to Business
B2C: Business to Consumer

Executive Summary

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

In this context, deliverable D32.1 "Future Internet Platform Federation Specifications and Architecture" aims to explore ways that Future Internet software assets, delivered through an open platform federation, 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 (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 FI-inspired platforms such as those provided by FI Assembly projects.

In order to identify potential enablers that fit to the MSEE needs, a comprehensive analysis of existing projects in the FI Assembly-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.

Regarding the FI PPP Core Platform, the services selected as relevant to MSEE are the following:

- Applications and Services Ecosystem and Delivery Chapter
  - Marketplace Generic Enabler
  - USDL Repository Generic Enabler
  - Registry Generic Enabler
- Internet of Things Services Enablement Chapter
  - Backend Things Management Generic Enabler
  - Backend Device Management Generic Enabler
  - Gateway Device Management Generic Enabler
  - Gateway Data Handling Generic Enabler
  - Gateway Protocol Adapter Generic Enabler
- Data and Context Management Chapter
  - Publish/Subscribe Broker Generic Enabler
  - Complex Event Processing Generic Enabler
  - Query Broker Generic Enabler
- Interface to Networks and Devices Chapter
  - Cloud Edge Generic Enabler

This means that in the MSEE project we will adopt the open specifications of the GEs above, build our own applications on top of some reference implementations of such GEs and remotely access them via open APIs made available by the FI-WARE Testbed.
Moreover, through the analysis of other FI-inspired Platforms, services and components from the following platforms have been initially selected for further investigation:

- COIN (Enterprise Interoperability and Enterprise Collaboration Services)
- SLA@SOI (Service Level Agreement Management)

- SERENOA (Context aware service mobile user interfaces)
- PERSIST (Personalised and Context-Aware Services)

The above-mentioned selections, in conjunction with a set of criteria based on functionalities necessary for the MSEE Use Cases resulted in the definition of two new service-based applications that will be integrated into the first prototype of this work package (D32.3 “Future Internet Platform Federation – first prototype”). These applications, named “Consumer Marketplace” and “IoT Manager” will be based on the following Enablers from FI-PPP:

- Consumer Marketplace
  - Marketplace Generic Enabler (FI-WARE Applications and Services Ecosystem and Delivery Chapter)
- IoT Manager
  - Backend Device Management Generic Enabler (FI-WARE Internet of Things Services Enablement Chapter)
  - Gateway Device Management Generic Enabler (FI-WARE Internet of Things Services Enablement Chapter)
  - Protocol Adapter Generic Enabler (FI-WARE Internet of Things Services Enablement Chapter)

In the final prototype we will develop additional service-based applications, which will be based on resources from FI-WARE chapters and other FI-inspired platforms. In the case of other FI-inspired platforms, we will first focus research and development efforts on those we have initially selected at this stage of research (COIN, SLA@SOI, SERENOA, PERSIST) without, of course, excluding the possibility of extending or modifying this selection.

As a final prototype is planned, further research is conducted and there are constantly new developments emerging in the FI world, this deliverable will be updated in M21 as D32.2 “FI Platform Federation specifications and architecture – M21” in order to take into account the experience to be gained during the development process of first prototype of the Future Internet Platforms Federation, and of course of all other knowledge coming from the FI PPP and the FI related domains.

1. Introduction

1.1. Context and Purpose of the Deliverable

MSEE WP32 stands to gain from integrating and reusing existing already available or emerging components from other platforms, by assembling such modules or even extending them according to its own needs to cover a broader spectrum of functionality and applicability. 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 FI-inspired platforms, which will altogether generate added value to the MSEE solution and to its use cases. In this context, MSEE project deliverable D31.1 "Functional and Modular Architecture of Future Internet Enterprise Systems" has already presented and explained the concept of federation with Future Internet Platforms.

In order to meet these demands, Deliverable D32.1 titled as "Future Internet Platform Federation Specifications and Architecture" has the following distinct objectives:

- To examine Future Internet-inspired platforms and services that may provide added value to the MSEE project
- To associate Future Internet services and functionalities with the MSEE project, and, more specifically, with its use cases.
- To define the specifications and architecture of the Future Internet Platforms Federation of MSEE and guide the development of its prototype and final implementation

The deliverable at hand provides a set of descriptions of specifications and architectures regarding the Future Internet Platforms Federation that are intended to be used as the blueprints for the development of the prototypes; the first one will be released in M18 and the second in M24. The work plan towards the implementation of the Future Internet Platform Federation follows a two-cycle development (to be checked through milestones M2 and M4) having the first and second prototype versions of the Future Internet Platform Federation as final products. For this reason, Deliverable D32.1 will be updated in M21 (as D32.2), to meet the needs of the second prototype and to include any significant results that should be taken into consideration by MSEE.

1.2. Relation to other Work Packages and Deliverables

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

MSEE Deliverables	Relation to D32.1
D31.1: Functional and Modular Architecture of Future Internet Enterprise Systems	Initial architectural considerations about the FI context in which utility services are placed, including a more refined analysis of their use cases.
D52.1: User Requirements Analysis for Virtual Factories & Enterprises in MSEE	The use cases and the analysis of their elements and requirements is used in D32.1
D32.3: FI Platform Federation first prototype	The prototype will be based on the specification set by D32.1
D32.2: FI Platform Federation specifications and architecture	Updated and final version of D32.1.

D32.4: FI Platform Federation final prototype	The updated prototype will be based on the updated version of D32.1
D33.1: 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
D34.1: 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: D32.1 and its relation to other MSEE deliverables

1.3. Structure of the Document

D32.1 follows a structure which is conceptually organised into three blocks:

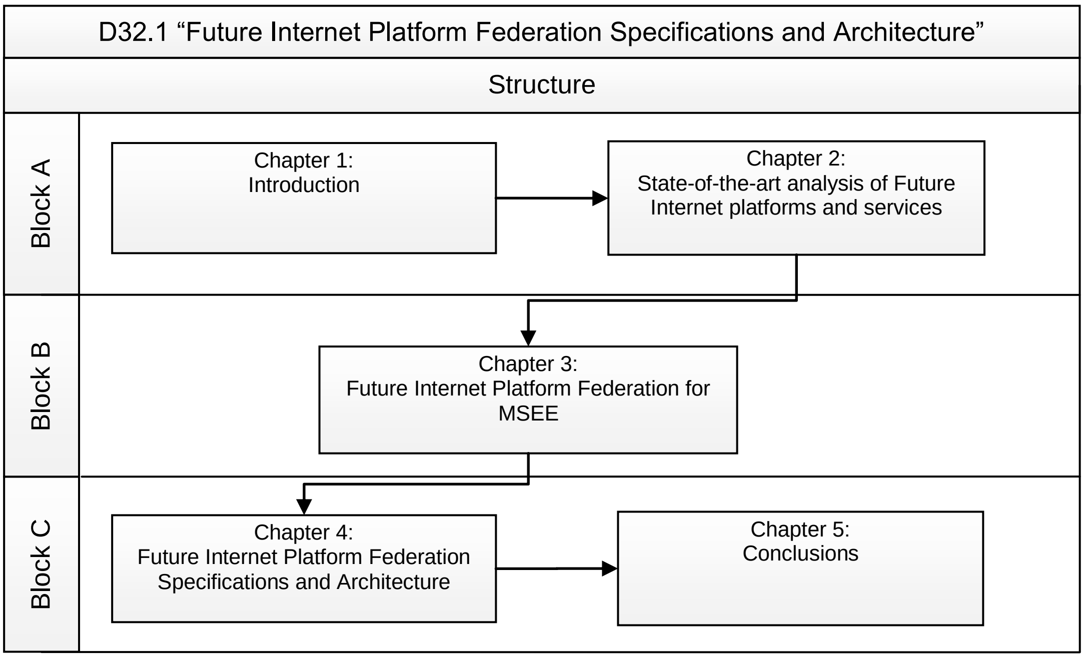

Figure 1: D32.1 Structure

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. Then, 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 a single chapter. Chapter 3 "Future Internet Platform Federation for MSEE" contains an analysis of the MSEE uses cases, in order to determine and prioritise necessary Future Internet functionalities, followed by a selection of appropriate Future Internet assets from FI-WARE and other platforms.

Finally, **Block C** provides the specifications of the Future Internet Platform Federation based on the results and decision of the previous chapters. Chapter 4 “Future Internet Platform Federation Specifications and Architecture” contains an examination of the specifications and architecture of MSEE’s Future Internet Platform Federation. This chapter contains 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, providing recommendations and pointing out the next steps that should be taken.

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. Platforms and projects in the FI Assembly

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	
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]. COIN will be further investigated in coordination with WP33.

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

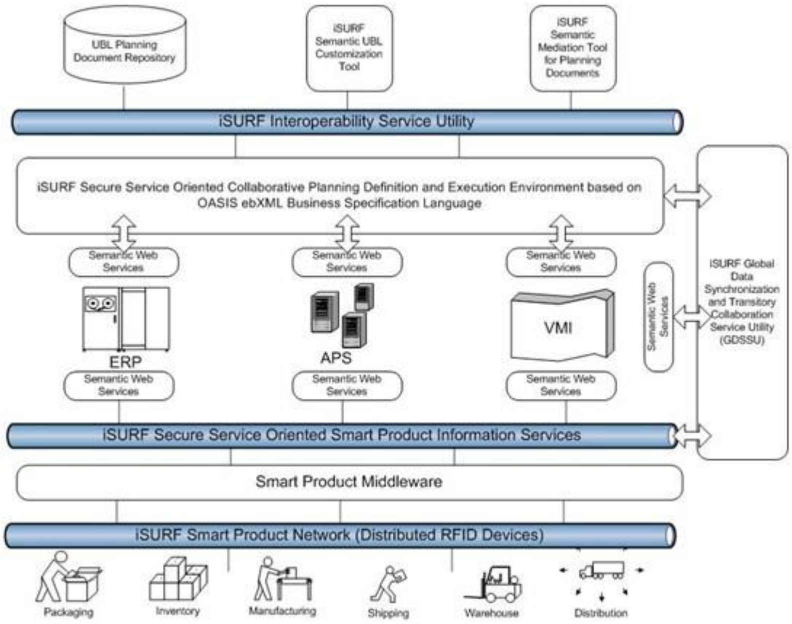

Figure 2: iSurf Architecture[4]

<u>Potential Contribution to MSEE</u>

iSURF can provide tools for implementing IoT services based on RFID-enabled devices. 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]. iSURF will be further investigated in coordination with WP33.

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.

<u>Potential Contribution to MSEE</u>

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 bind together the physical environment with the digital world), and of the social and physical

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

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. 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.

[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

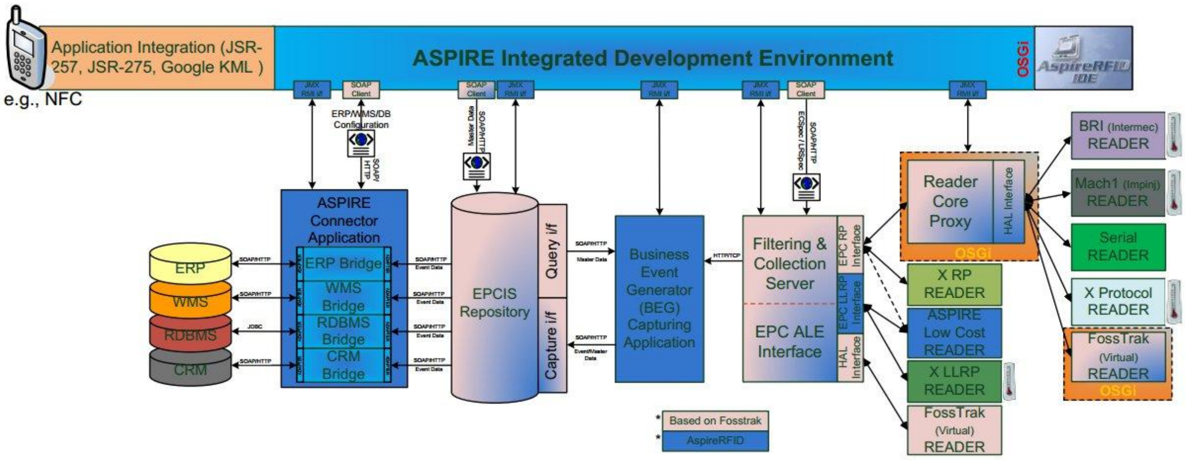

Figure 3: Aspire Middleware Architecture[46]

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 RFID 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.

<u>Potential Contribution to MSEE</u>

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. The middleware components are available as open-source software[10].

2.1.2.2. 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.

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

OpenIoT is leveraging previous research in IoT, including ASPIRE (via AspireRFID). 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.

2.1.2.3. 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.

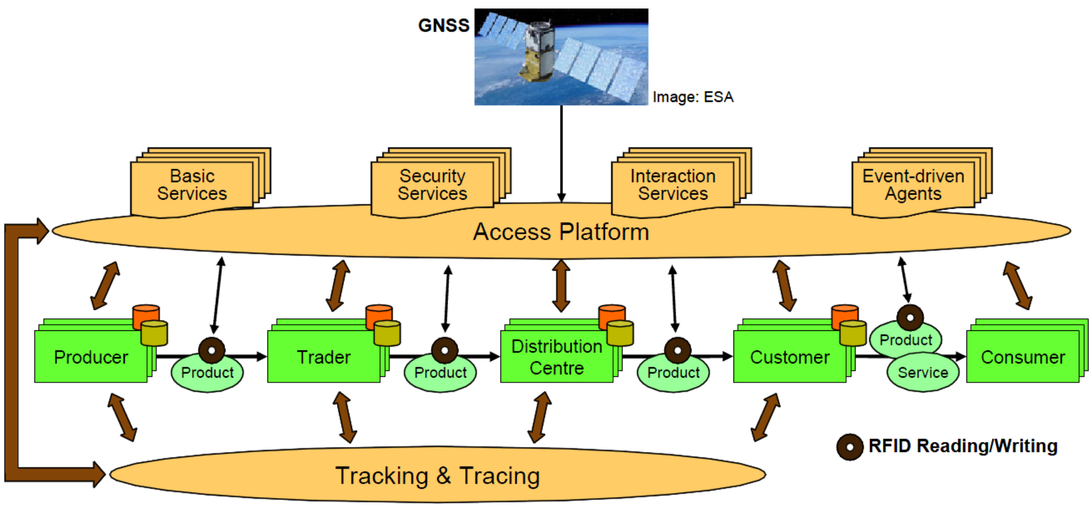

Figure 4: Cuteloop Architecture[47]

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.

<u>Potential Contribution to MSEE</u>

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[11].

2.1.2.4. 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.

<u>Potential Contribution to MSEE</u>

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

The webinos framework is open source and is currently available[12] 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[13].

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.

[12] http://webinos.org/downloads/
[13] 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.

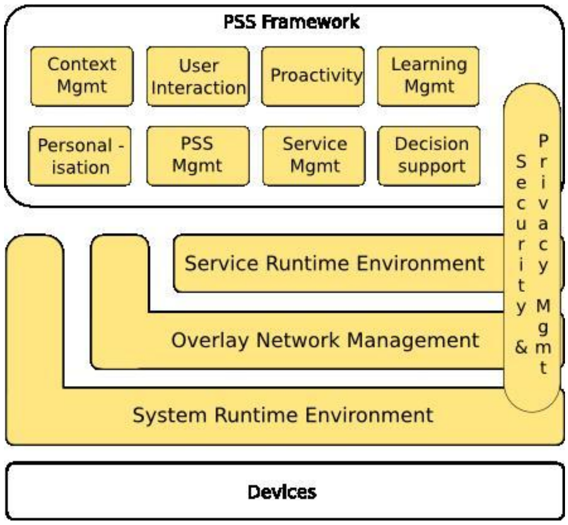

Figure 5: PERSIST Architecture[15]

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[14]

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). Specifically, FI-PPP current projects, organised include [18]:

- Technology Foundation:
    - FI-WARE
- Use Cases:
    - ENVIROFI
    - FINSENY
    - FI-CONTENT
    - FINEST
    - INSTANT MOBILITY
    - OUTSMART
    - SAFE CITY
    - SMARTAGRIFOOD
- Capacity Building and Infrastructure:
    - INFINITY
- Programme Facilitation and Support:
    - CONCORD

The timetable of the FI-PPP programme is illustrated in the following figure [19]:

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

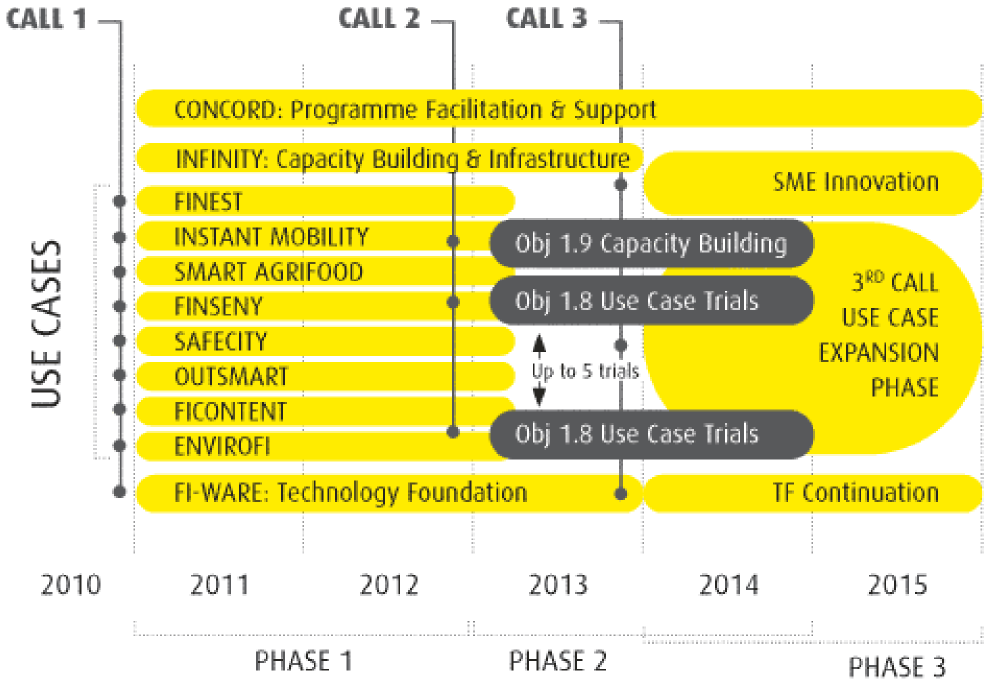

Figure 6: FI-PPP constituent initiatives and timetable[19]

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 September 2012, 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. Methodological Approach

All previous chapters presented briefly or in detail the role and engagement of FI-PPP and Future Internet Projects, performing a SoTA analysis on their actual role in parallel with MSEE project.

This chapter examines the specific MSEE use cases and associates them with various Future Internet functionalities.. These cases have been examined in deliverable D52.1 "User Requirements Analysis for Virtual Factories & Enterprises in MSEE" as an initial reference for case requirements, which have then been refined in D31.1 "Functional and modular architecture of Future Internet Enterprise Systems – M6".). Further updates to this section will also take under consideration results included in the updated D31.1 "Functional and modular architecture of Future Internet Enterprise Systems – M18".

Then, we examine the chapters of FI-WARE and select specific Generic Enablers as suitable for the purposes of MSEE and its use cases. The chapter then includes a preliminary assessment on the usability of other FI-inspired platforms, which will be extended, refined and finalised in the next version of this deliverable. Finally, an initial selection of FI-inspired services for MSEE is presented.

3.2. Use case descriptions and requirements

3.2.1. 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 [40]:

- *"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

3.2.2. 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 [40]:

- *“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

3.2.3. Bivolino

As described in D52.1, Bivolino introduces the following two cases into MSEE[40]:

"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

3.2.4. 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. [40]

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

3.2.5. 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).

<u>Internet of Services</u>

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 [41] 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[40]. 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[40].

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[40].

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.

The results of this analysis are collectively presented in the following table:

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: Use case functional commonalities

3.2.6. FI resources for the MSEE platform

The following section contains an initial evaluation of the suitability of FI resources to the MSEE platform and use cases. In this version of the deliverable, we focus mainly on capabilities derived from FI-WARE, and we provide an initial assessment for capabilities offered by other platforms. In the final version of the deliverable we will update this analysis, and provide a deeper examination of their added value and technical characteristics.

3.2.6.1. 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. These tools are 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 MSEE project in order to determine those that can be potentially leveraged and provided added benefits to the MSEE project as a whole. The initial 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]. These results will be updated in the final version of the deliverable.

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) and for third parties in M27 (May 2013 corresponding to MSEE M23).[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.

3.2.6.1.1 Applications and Services Ecosystem and Delivery Framework

According to Wikipedia[31], *"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]:

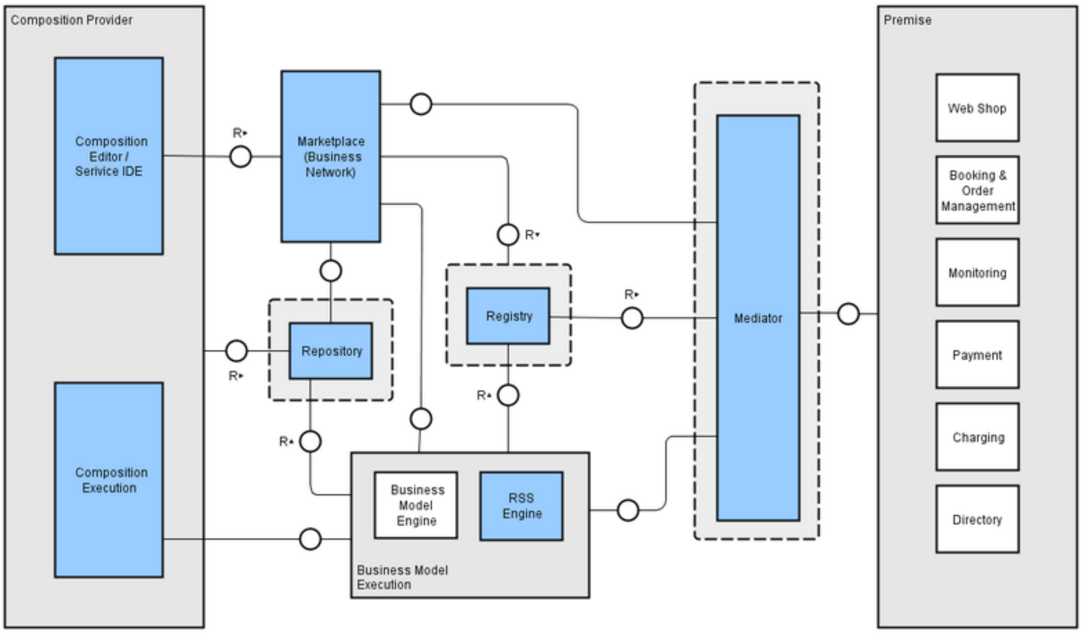

Figure 7: Apps and Services Ecosystems and Delivery Framework overview[32]

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].

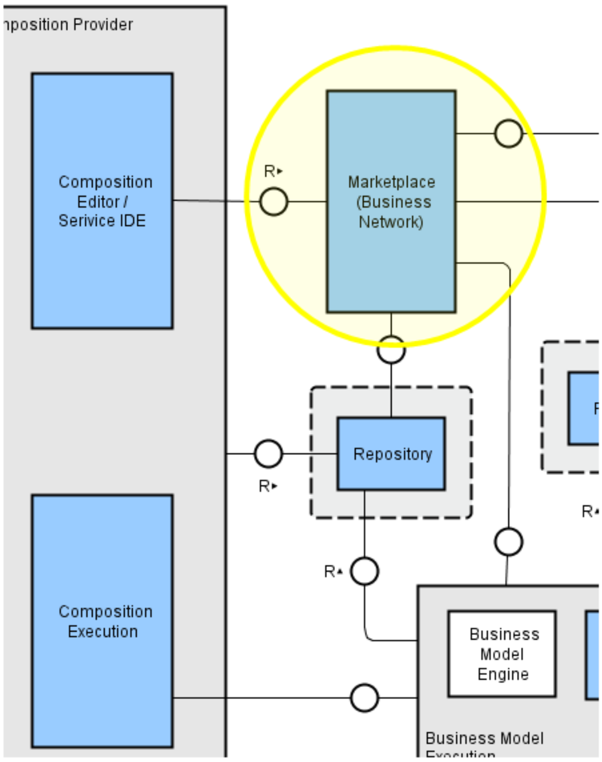

Figure 8: Marketplace in the context of the Business Framework[34]

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 at this moment 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, all 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[15] 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[16] 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.

3.2.6.1.2 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 is illustrated in the following figure [35]:

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

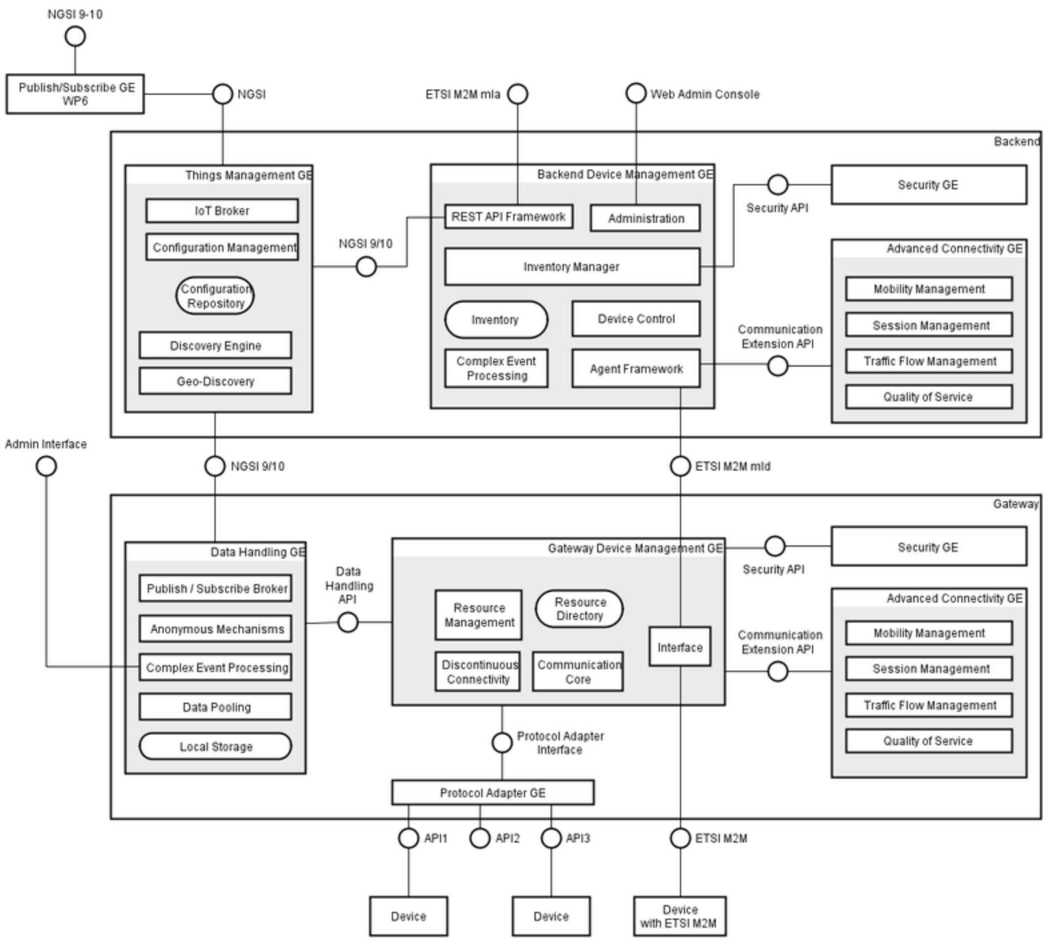

Figure 9: FI-WARE GEs in the "Internet of Things services enablement" Architecture[35]

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.

<u>Evaluation</u>

The IoT Enablers are also a very good match for MSEE. As noted before, three out of four use cases make use of IoT concepts. Indesit and Ibarmia need to communicate with devices away from the companies' premises, both in terms of passively retrieving information, as well as controlling embedded systems. The Bivolino case may include the use of RFID scanners to retrieve real time data on tagged garments.

The best candidate Enablers at this moment are the Backend Things Management GE, the Backend Device Management GE, the Gateway Device Management GE, the Gateway Data Handling GE and the Gateway Protocol Adapter GE. All five enablers correspond to the first release of FI-WARE.

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[17]. 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[18],[19],[20] support registration, update and subscriptions.

The Backend Device Management GE[21] 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.

[17] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.IoT.Backend.ThingsManagement
[18] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/TM_GE_Southbound_Interface_Open_RESTful_API_Specification_(PRELIMINARY)
[19] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/TM_GE_Northbound_Interface_Open_RESTful_API_Specification_(PRELIMINARY)
[20] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/TM_GE_Southbound_Interface_Open_RESTful_API_Specification_(PRELIMINARY)
[21] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.IoT.Backend.DeviceManagement

On the Gateway side, the Gateway Device Management GE[22] contains the core functionality of the IoT Gateway and handles the following interactions:

- Resource management (CRUD operations on the Resource Directory for IoT resources)
- Resource Access (communication with resources also via the Protocol Adapter GE and the Data Handling GE)

A Northbound API to the Backend is specified, as well as a Southbound API to devices (notably, it is an IETF CoRE RESTful API in the first release [45])

The Gateway Data Handling GE[23] 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[24] deals with incoming and outgoing traffic from the Gateway and the registered Devices. It is aimed at IP-enabled devices that do not support ETSI M2M and uses Embedded Web Services. The GE includes some support for so-called "legacy devices", i.e. devices that support protocols other than IP. It supports device discovery and registration, resource and capabilities listing, as well as subscription to device and reception of the corresponding updates.

3.2.6.1.3 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].

[22] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.IoT.Gateway.DeviceManagement
[23] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.IoT.Gateway.DataHandling
[24] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.IoT.Gateway.ProtocolAdapter

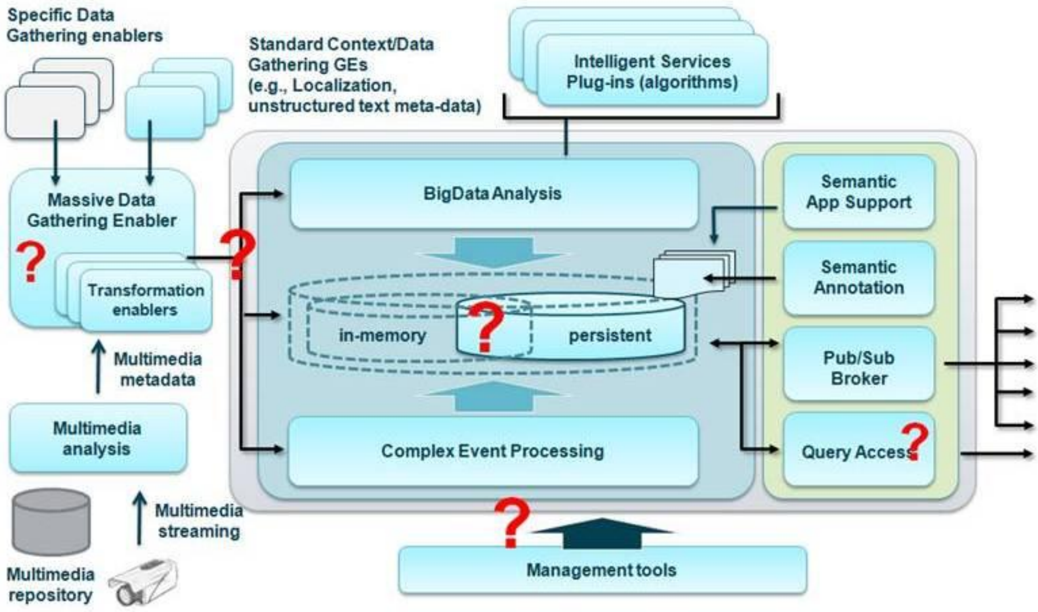

Figure 10: FI-WARE Data/Context Management Enablers[36] [25]

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
- **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. IoT use cases (Indesit, Ibarmia, Bivolino), as well as Philips, may require

[25] Question marks indicate associations and elements that are still under discussion within FI-WARE.

elements of Complex Event Processing. Those cases may also 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[26], 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.

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[27] 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.

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

The Query Broker GE[28] 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.

3.2.6.1.4 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]:

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

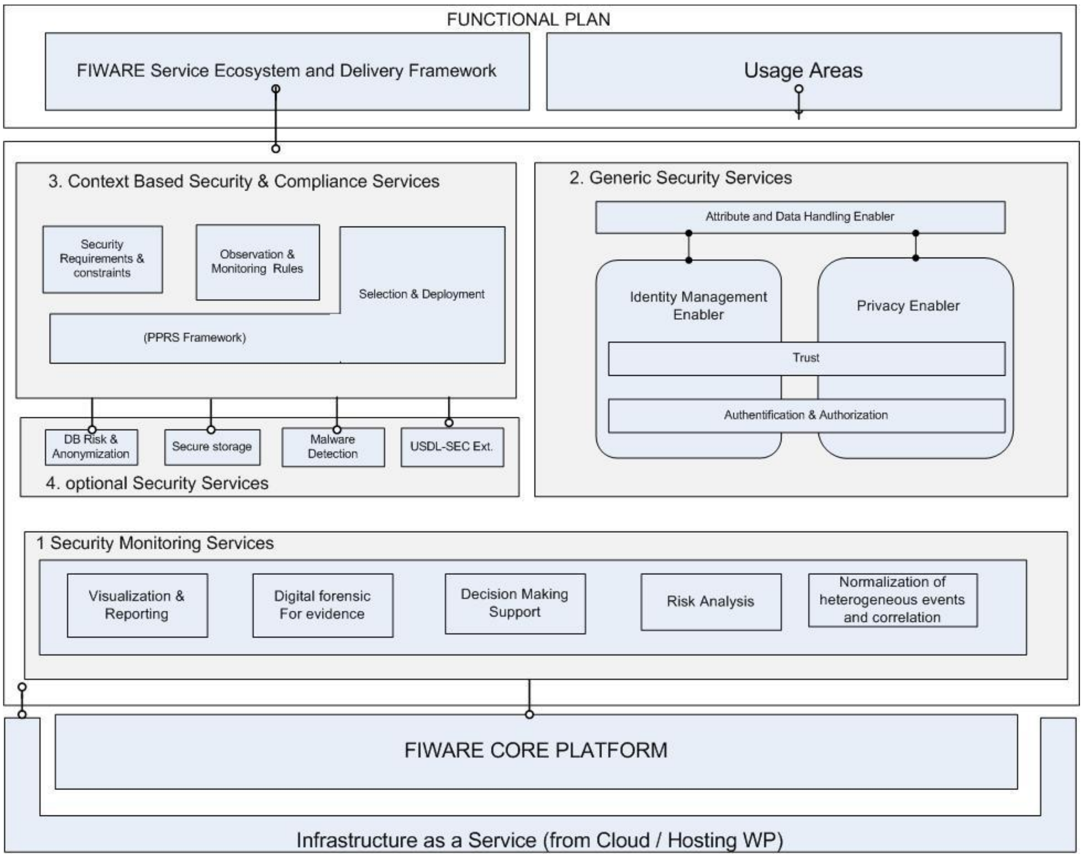

Figure 11: FI-WARE security enablers and associations[37]

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 is expected to be handled via a WP33 Utility Service (Single Sign On). Further security and privacy functionalities appear to be beyond the scope of WP32 at this stage.

3.2.6.1.5 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]:

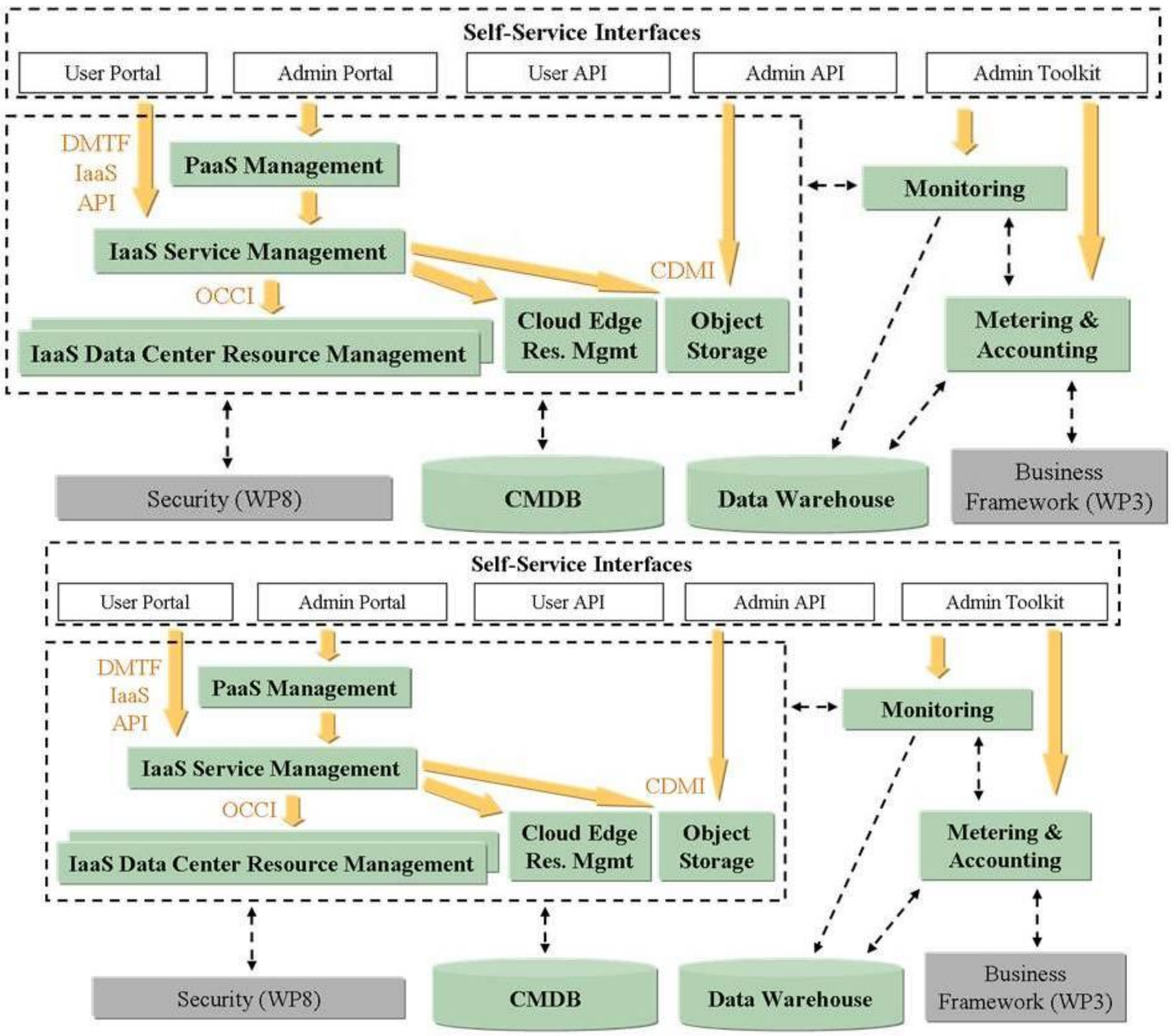

Figure 12: FI-WARE cloud hosting enablers and associations[38]

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. Some interfacing and coordination may occur between the Cloud Proxy GE and IaaS Cloud-edge Resource Management GE in this chapter and the Cloud Edge GE in the Interface to Networks and Devices chapter, which will be further investigated in the final version of this deliverable.

3.2.6.1.6 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]:

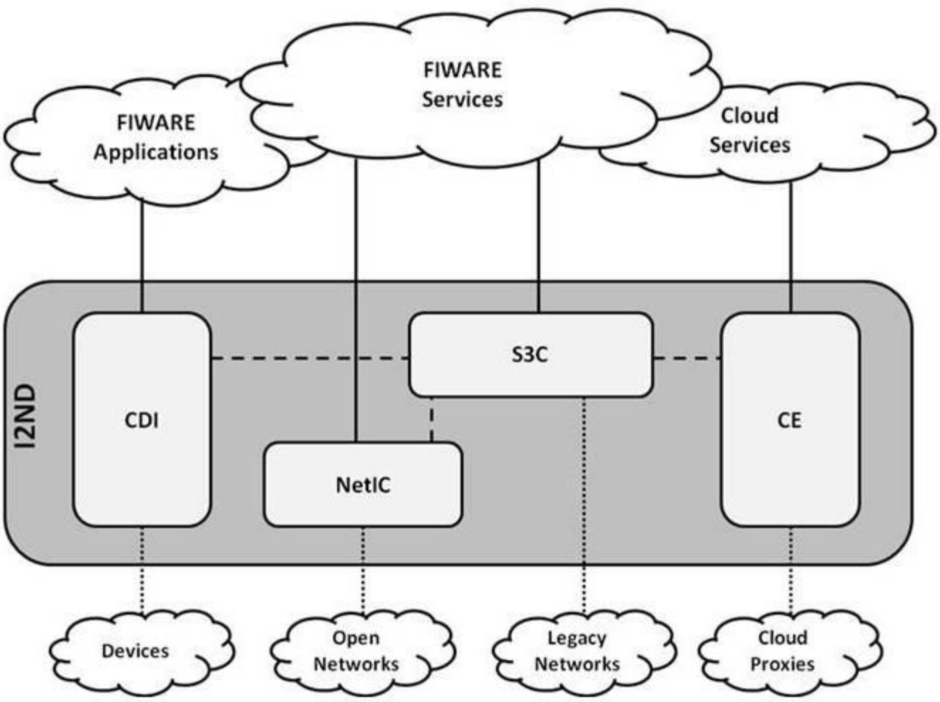

Figure 13: FI-WARE Interface to Networks and Devices Enablers[39]

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[29], 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

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

- 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.

3.2.6.2. Services from other FI-inspired platforms

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. In the current version of this deliverable we attempt an initial evaluation of potentially useful FI-inspired platforms. An in-depth technical investigation of candidate services will be performed and further elaboration will take place in the final version of the deliverable (D32.2).

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, which will be examined in coordination with WP33.

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 may 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, ASPIRE and OpenIOT are good candidates for services reuse. ASPIRE is a mature project which covers RFID, which may prove useful in the Bivolino case, primarily, as well as the Indesit and Ibarmia cases, in a tangential manner. CUTELOOP also provides similar functionalities, and should be considered as an potential alternative to ASPIRE.

OpenIOT builds on the experience of 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 Philips 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.

3.2.6.3. FI Resource Initial Selection

For each of the functionalities identified in the previous sections, there is a need to select one prime candidate service out of those that have been surveyed so far.

The functionalities of End-user service discovery, IoT Device Communications, RFID object tracking, Real Time Data and Event Processing and Context/Location Aware Services can be covered by the different FI-WARE chapters, i.e. Applications and Services Ecosystem and Delivery Framework, Internet of Things Services Enablement, Data and Context Management and Interface to Networks and devices. These correspond to Generic Enablers that are available in the first release of FI-WARE and are preferred over other platforms (EzWeb, ASPIRE, OpenIOT, CUTELOOP, webinos) as they support federation natively.

COIN services may provide added value in the fields of Interoperability and Ecosystem Participant Collaboration. In the last area, COIN appears to be most suited for B2B collaboration. COIN also supports federation natively, which may reduce development and integration overheads.

In the case of SLA management, it may be necessary to use resources from SLA@SOI. The suitability of those resources needs to be further investigated. The same activity must take place in the case of Mobile Access channels, where the best alternative appears to be the re-use of SERENOA components. Service Personalisation can be potentially be supported by PERSIST components.

CONVERGENCE may prove useful for content tracking, however, the necessary resources may not be available soon enough, and therefore more investigation is necessary.

In the case of KPIs, business intelligence, and social media integration, it is necessary to perform further research to determine whether there exist any potentially applicable resources.

The selected services and rationales are 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: FI Resource Selection

The analysis documented in this deliverable will be updated and expanded in the final version of the deliverable in M21, and will be informed by:

- Experience gained from the development of the first FI Platform Federation prototype (M18)
- Documented Suitability of 1st release FI-WARE resources
- Documented Suitability of FI-inspired platform resources
- Evolution of projects in the Future Internet domain
- Evolution of use case requirements
- Coordination with efforts in WP33 and WP34 (Utility Services and Value-added services)

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 may include both FI Core Platform Enablers as well as other FI services.

This initial approach will be updated in M21 with the next version of the deliverable (D32.2), and using the experience from the first prototype (M18) and of the results in related MSEE domains.

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[30] 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 5: 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 infrastructure 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.

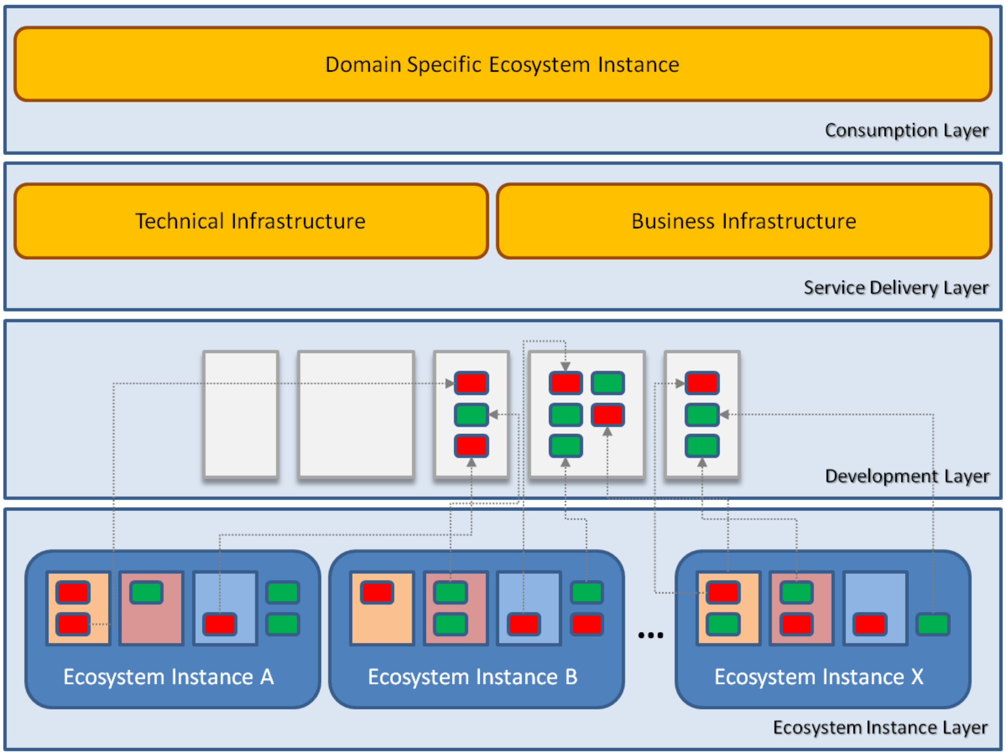

Figure 14: Future Internet Platform Federation conceptual architecture

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

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

Ecosystem Instance Layer
This layer consists of (a) FI-WARE Generic Enablers, and federation with existing services running on FI-inspired Platform instances elsewhere necessary to complete the functionality of the platform, and (b) existing components from available FI platforms based upon integration (which may necessitate assembly and extension of such components).

Development Layer
This layer corresponds to the actual federation, extension and/or customization of components or services derived from the ecosystem layer.

Service Delivery Layer
Deployment of developed components or services derived from the Development Layer, based on the technical infrastructure.

Consumption Layer:
Corresponds to the consumption of the enterprise applications by the end-users; on one hand the ecosystem will allow the user to use the services in a direct way; on the other hand, at a more general level, the services will be registered with the Generic Service Delivery Platform and will be accessible to anyone by its functionalities.

4.3. Future Internet-enabled services

Through the initial exploration of the use cases, an initial set of basic functionalities that can be provided using FI resources has been selected and is described in the following lines.

4.3.1. Service Selection

From the use case analysis in the previous chapter, we have identified specific functionalities for all use cases. In selecting candidate services we have examined a set of criteria for each one:

- Relevance to each use case
  - As a core element
  - As an optional element
- Dependencies on other functionalities
- MSEE synergies
- FI resources required

The results of this analysis are presented 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 6: Service selection decision matrix

In selecting functionalities to be included in the first prototype, preference is given to those that are are included as core elements to most use cases, have no dependencies on other functionalities and can be implemented using resources documented in detail at this stage of WP32.

From the matrix above, we can reach the following conclusions:

- End-user service discovery
    - Is central to all four Use Cases
    - 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
    - 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.

For these reasons, we can reasonably suggest that End-user service discovery, IoT Device Communications and IoT RFID object tracking are good candidates for inclusion in the services of the first prototype (D32.3).

End-user service discovery is a core element to all Use Cases and can be developed immediately, using FI-WARE resources. IoT and RFID services are also central to three out of four cases and can also be developed immediately. They also open an “implementation path” for more services. Specifically, we can use the IoT and RFID functionalities from the first prototype to support the implementation of others, such as Real Time Data and Event Processing, SLA Management and Mobile Access Channels in the final prototype.

4.3.2. Service Specification

For the reasons cited above, the first two FI-enabled services we propose for implementation in D32.4 are

- The **“Consumer Marketplace”** implementing the End-user service discovery service by utilising the FI-WARE Marketplace GE from the Applications and Services Ecosystem and Delivery Framework Chapter
- The **“IoT Manager”,** implementing both IoT Device Communication and elements of RFID object tracking (by treating RFID scanners as IoT-enabled devices) by utilising the FI-WARE Backend Device Management, Gateway Device Management and Protocol Adapter GEs from the IoT Services Enablement Chapter.

4.4. Consumer Marketplace

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

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

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.

4.4.1. 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.

4.4.2. Requirements

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

Interface

- The application will include pages for
  - Provider and consumer profiles
  - Store administration
  - Store browsing
  - Service listing
  - Search
  - Marketplace administration

User Access

- Users will be able to browse Consumer Marketplace content and be redirected to Service Provider portals
- Service Providers will need to register with the application in order to manage stores and service offerings
- Optional: Service consumers may be registered with the Consumer Marketplace
- Optional: Service providers may limit access to service offerings to specific registered service consumers

Service Browsing

- The Consumer Marketplace browsing will support the following features:
  - Store listing by category and provider (ecosystem)
  - Service listings within stores
  - Service listings by category
  - Simple Service/Store search with text
- Users will be able to compare service offerings side by side, based on their attributes

Federated search and recommendations

- Service offerings in the Consumer Marketplace can be associated with specific keywords or phrases describing relevant services.
- When retrieving and displaying a service offering, the application will be able to perform a search across known and trusted marketplaces for service offerings matching the associated keywords or phrases.
- Trusted marketplaces will be defined by providers for specific services according to business and marketing needs

Data (minimum)

- For each service provider, the following information will be maintained and provided
  - Registered services

  - Profile
  - Optional: reviews and ratings
- For each service listing, the following information will be maintained and provided:
  - Title, description, provider
  - Keywords
  - Service Metadata (defined per ecosystem)
  - Access privileges
  - Provider link for purchase
  - Relevant trusted marketplaces for federated search
  - Optional: reviews and ratings
- For each trusted marketplace instance
  - Description
  - Access point
- Optional: For each service consumer, if registered, the following information will be maintained and provided: Profile information
  - Organisation and personal information (for business buyers)
  - Personal Information (for consumers)
  - Optional: reviews and ratings

The application will need to cover the following non-functional requirements:

Scalability
The Consumer Marketplace will need to be expandable with an arbitrary number of stores and services.

Security and privacy
In B2B markets, marketing information is often considered sensitive. The application will provide adequate security and privacy for market activities, by allowing the definition and enforcement of user access privileges.

4.4.3. 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 so far:

- Service Provider Registration
- Store management
- Service offering Registration
- Store listing browsing
- Service listing browsing
- Store and service search
- Service comparison
- User account administration
- Federated search
- Optional: Service consumer registration

4.4.4. Proposed Architecture

The Consumer Marketplace implementation will rely heavily on Generic Enablers from the Applications/Services Delivery and Ecosystem Chapter.

For user authentication purposes, the application may also rely on identity management functions from the proposed Single Sign On utility service developed in WP33[31], at least from the user's perspective within MSEE.

The Consumer Marketplace architecture is illustrated in the following figure:

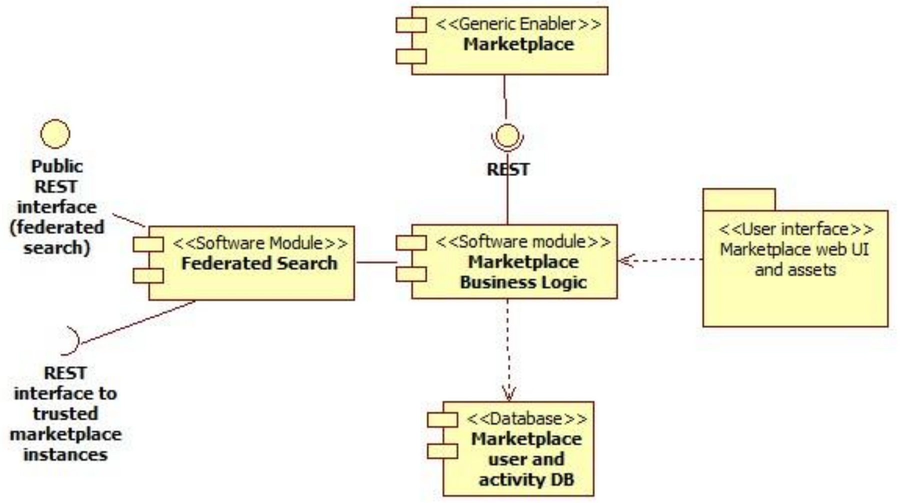

Figure 15: Consumer Marketplace proposed architecture

The Consumer Marketplace relies on a set of external and internal components. These components are the following:

Marketplace Generic Enabler

A Marketplace GE instance is provided either externally or as an internal resource. The Marketplace GE provides access to information about service offerings, stores and marketplace actors. Marketplace participants (Providers, Consumers, Resellers and Guests) can be registered. Stores can be created and populated with service offerings, along with relevant metadata. Furthermore, it allows searches within its listings. The GE is accessed via its REST API by the "Marketplace Business Logic" module.

Marketplace Business Logic

The Marketplace Business Logic module (MBL) manages the operation of the marketplace instance. It accesses the Marketplace GE to construct and interact with the marketplace implemented by the Marketplace GE. Interactions are received by end users via the Marketplace UI, and include store management, service management, service queries, etc.

Marketplace user and activity DB

This database maintains additional information on marketplace entities (stores, participants), as well as activity logs.

Marketplace Federated Search

This module is associated with the MBL and provides the federated search capability. Using keywords associated with a particular service offering, it performs a search across known and trusted marketplaces. The target marketplacesare included in the service offering metadata and specified by the Service Provider according to their business and marketing strategy. The result is aggregated and returned to the users via the Marketplace Business API. This module

[31] MSEE Deliverable D33.1 FI Utility Services specifications and architecture – M12 issue

also mediates access to the Marketplace GE instance for federated searches from other trusted parties.

Service Marketplace web UI
This software package refers to the implementation of a web interface for the Marketplace Business Logic. It allows users to browse the service marketplace, create stores, register services, etc.

4.4.5. MSEE Context

Conceptually, the Consumer Marketplace can be used as a gateway for consumers to search for and access Use Case services. 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 will be used by the following types of users:

Use Case	Service provider	Service Consumer
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
Bivolino: Shirt-as-a-Service	Bivolino personnel, managing the “Shirt-as-a-Service” service group, offering subscriptions, laundry schemes, warranties, etc.	Consumers browsing the “Shirt-as-a-service” offerings
Bivolino: Interoperable Cutfile	Bivolino personnel, managing the “Interoperable Cutfile” service group	Clothing manufacturing decision makers, browsing offerings such as subscription schemes, order packages and digital cutter lease plans
Philips	Philips personnel, managing the NetTV ecosystem	Decision making personnel at Philips partners (or potential partners), browsing participation schemes in the NetTV ecosystem

Table 7: Consumer Marketplace users

4.5. 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 sensors, actuators and embedded devices is a critical aspect of at least three out of four MSEE use cases (Indesit, Ibarmia and Bivolino).

The IoT Manager aims to abstract the complexities of accessing devices in the physical realm into an easy to use interface and its operation will be based on the reuse of Generic Enablers from the Internet of Things Services Enablement Chapter of FI-WARE.

4.5.1. 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.

4.5.2. Requirements

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

- Be accessible via a simple, commonly defined API.
- Focus on IP-enabled devices.
- Provide the following basic operations to applications:
    - Register a new device
    - Send commands to the device
    - Receive information about the status of the device
    - Collect readings from the device
- Access to devices needs to be properly authenticated.
- Access to devices needs to be logged for service and platform administration purposes
- Include an adequate set of error and fault messaging, covering loss connection, device faults, software faults.

The IoT Manager will also follow the non-functional requirements listed below:

- **Interface Simplicity**. The IoT Manager API will support a minimum required set of operations required to successfully perform the operations required in the functional requirements section.
- **Performance**. The IoT Manager will be designed bearing in mind that IoT services place a premium on near-instantaneous operations. An important element of IoT applications is the ability to communicate with remote devices in "real time", requiring, for example, the ability to detect faults or issue commands instantly.
- **Robustness**. Devices existing in the field may have intermittent connectivity, face adverse conditions and may fail unexpectedly. The IoT Manager should be able to withstand loss of connectivity, malformed input, etc. without causing the degradation of the rest of the system it participates in.

4.5.3. Use Cases

The following use cases have been identified for the operation of the IoT Manager:

- Application registration / deregistration
- Device Registration / deregistration
- Receive device status
- Receive device measurements
- Receive device events
- Send commands to the device
- Retrieve operations log

4.5.4. Proposed Architecture

The IoT Manager uses FI-WARE Generic Enablers from the IoT Services Enablement Chapter: The Backend Device Management GE, Gateway Device Management GE and the Protocol Adapter GE. Enablers in this chapter are designed as parts of stack structure, composed of a Gateway Enabler group, providing access to devices and a Backend group, providing access to applications. Note that the Protocol Adapter GE is part of the Gateway group.

Instances of those enablers could either be part of the same provider infrastructure (thus internally interconnected) or provided separately. Some or all enablers could also be implemented internally. In cases where communication between the Enablers are not handled by a third party, we have included a separate interoperability component, the Enabler Relay, to pass messages between the these components.

The proposed architecture for the IoT Manager is illustrated in the following figure:

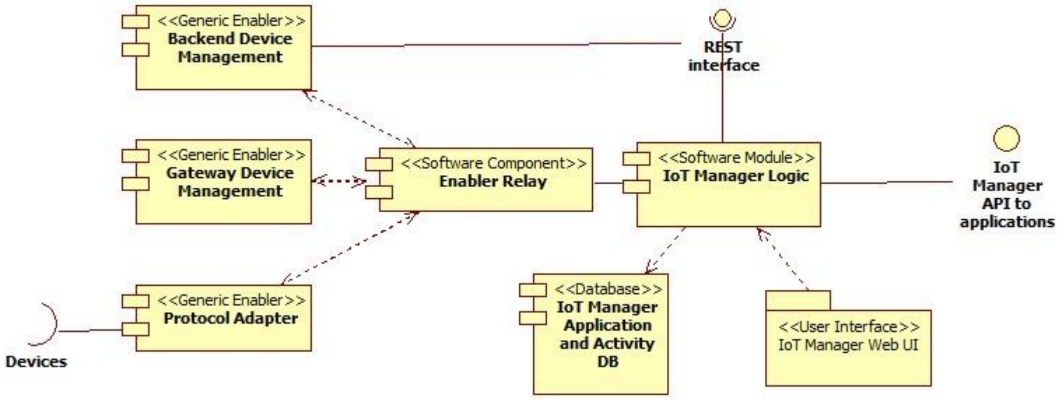

Figure 16: IoT Manager Proposed Architecture

Backend Device Management GE

The Backend Device Management GE provides the main point of access to devices. It allows operations such as the registration of devices, retrieval of device information, status updates and measurements. To provide this functionality, the enabler is expected to interface with the Gateway Device Management GE. This communication may happen internally to the instance provider or may be controlled via the Enabler Relay module by the IoT Manager.

Gateway Device Management GE

This Enabler handles access to the devices. The Enabler is specified to directly interface with IP-enabled, ETSI M2M compliant devices or the Protocol Adapter GE. The Gateway Device Management GE enabler allows access to devices, with particular emphasis to handling situations of reduced connectivity, a critical feature of IoT applications. In this case, the enabler collects messages by backend applications and forwards them whenever the device is online.

Protocol Adapter GE

The Protocol Adapter GE handles communications with devices which use a range of different protocols. It maps communications with these devices to a uniform internal API, ensuring interoperability with the rest of the the FI-WARE IoT stack. The component communicates with devices both using an IP-based protocol or a non-IP based one.

Enabler Relay

Enablers are interconnected as parts of the same “stack”. This component mediates and monitors those transactions, especially in the cases where enabler instances are not part of the same implementation. Even in the case of enablers being provided together, a separate relay may be necessary to control communication and ensure robustness, or even improve interoperability between them.

IoT Manager Logic

The IoT Manager Logic component acts is the central functional module of the IoT Manager. On the device side, it communicates with the Backend Device Management GE directly, passing requests for device registration and access. In the presence of an Enabler Relay, it controls communication between the Backend and Gateway Device Management GEs.

On the application side, it exposes an API to perform the basic functionality described in the requirements, i.e. registration of devices, access, control, and logging. It also implements an access control policy to devices, limiting access to specific services and ecosystems. To perform those tasks, it conducts CRUD operations on the IoT Manager Application and Activity DB. For development and administration purposes, it also exposes a rudimentary UI (indicated in this architecture as a web-based implementation and described separately) providing basic functionality, i.e. the API operations, access control, logs and administration.

IoT Manager Application and Activity DB

This database contains data on applications accessing the IoT manager, registered devices, logs and access control information.

IoT Manager web UI

The IoT Manager includes a rudimentary interface that provides indirect access to its API, and provides access to basic administration operations, respecting access control limitations.

4.5.5. 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 will focus on establishing a reliable channel of communication with external devices. This is not a trivial task, and will require bridging several enablers and communication across a set of different protocols and components. Once this connection is established, the low-level communication layer created by the IoT Manager can be used to support other, high-level, applications. For example, event processing applications (e.g. a “Device Alert” service) could be built on top of the IoT by exploiting the Complex Event Processing Generic Enabler (from the Data and Context Management Chapter of FI-WARE) to provide intelligent monitoring of devices in the field.

5. Conclusions

This deliverable reports on the current state of research in MSEE regarding the Future Internet Platforms Federation. The survey in the field of the Future Internet by examining 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. In this deliverable, emphasis was put on the FI-WARE platform and its services, and an initial assessment of other FI-inspired platforms has been performed. Through this evaluation, an initial a set of candidate services from FI-WARE evolved, which fits the criteria of being both available and relevant to the needs of the use cases and the research scope of MSEE.

Finally, an initial description of two applications for MSEE, based on Future Internet functionalities has been done, alongside with a description of their architecture and specifications. These are the “Consumer Marketplace”, a web-based marketplace, where service consumers can browse and search for service offerings, and the “Internet of Things Manager” which allows MSEE applications and service developers to access IP-enabled devices. Both services have been selected based on the perceived value they bring to the MSEE, as well as the availability and documentation of the necessary Future Internet resources.

The Future Internet resources and services described in this deliverable are an initial approach that will guide the development of D32.3 “FI Platform Federation first prototype”. In this context, the results presented in this deliverable at hand will be further extended and refined in the final version of the deliverable, D32.2. Also, the list of Future Internet-inspired platforms will be refined and a deeper investigation of their specifications and services will be performed, in order to expand the range of Future Internet resources available to MSEE. Additionally, the final version of this deliverable will be informed by the experience of developing the first prototype services and will include specifications for more. This final version will guide the implementation of D32.4 ”FI Platform Federation final prototype”.

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 D52.1: "User Requirements Analysis for Virtual Factories & Enterprises in MSEE"
[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"