D31.2 –
Functional and Modular Architecture of Future Internet Enterprise Systems
–
M18 issue

Document Owner:	Markus Heller (SAP)
Contributors:	Markus Heller (SAP), Alexandru Oprea (SAP), Mauro Isaja (ENG), Ioan Toma (UIBK), Emmanouil Ntanos, (SiLo), Jerry Dimitriou (SiLo)

Dissemination:	Public
Contributing to:	WP 31
Date:	29.03.2013
Revision:	1.0

VERSION HISTORY

1	**DATE**	**NOTES AND COMMENTS**
2	**01.03.2013**	**DRAFT**
3	**28.03.2013**	**FINAL INPUTS AND CONSOLIDATION**
4	**29.03.2013**	**READY FOR SUBMISSION**

DELIVERABLE PEER REVIEW SUMMARY

ID	**Comments**	**Addressed (✓) Answered (A)**
1	No specific note	
2		
3		
4		

TABLE OF CONTENTS

1. Executive Summary

In the sense of a servitization of products, manufacturing enterprises need to be able to design, offer, and operate services around their core products in the future. Enterprise Ecosystems around these products are responsible for delivering the extended service experiences to the end customers (of both products and services in a bundle). The delivery of advanced product-service solutions to customers within such Enterprise Ecosystems poses several architectural challenges, for example the challenge to devise a pragmatic and modern architecture for enterprise systems based on the Internet as is core backbone.

D31.1 contained architectural blueprints for Future Internet Enterprise Ecosystems and Manufacturing Core Platforms. It was highlighted that there the traditional role of monolithic enterprise systems (e.g. CRM, ERP) was modified for better achieving service experiences around products for the end customers (of both product and services). Instead, a layer on top of these traditional enterprises systems was established to serve the requirements of such systems (described in D.31.1). Traditional enterprise systems may more than ever before be in the need to leverage several co-existing Future Internet Enterprise Ecosystems (instead of solely dealing with their own ES-specific ecosystem).

This architecture-focused deliverable updates the first architectural blueprint for Future Internet Enterprise Ecosystems (in the manufacturing domain) introduced in D31.1 and reports changes and evolutions since then.

It is important to note, that this deliverable discusses ecosystems and manufacturing platforms from the **architectural viewpoint only**. Other viewpoints on MSEE ecosystems and platform are also possible and are described in other deliverables (e.g. SP1, SP2, and SP4).

2. Introduction

For this deliverable (as in D31.1), we identified two major trends that need to be taken into account for the exercise of devising an architectural blueprint: (1) the servitisation of products and (2) the Future Internet architectures and platforms with a large set of networks and objects managed by these networks connecting with each other.

First, as described in D31.1, servitisation of products represents a major trend in the manufacturing domain deals with the design and delivery of services around products. Second and more importantly, Future Internet applications will be constructed from services by end-users and developers using assembly instructions provided by vendors. Services will come from a variety of vendors and will be offered on cloud-based platforms – they form an ecosystem.

The *Future Internet* Initiative (e.g. [15], [16]) strives to develop Internet-based future-proof architectures, platforms, applications, systems, and services that can be leveraged for such ecosystems. In this spirit, applications will be constructed from services by end-users. For example, the FI-WARE project in the FI-PPP private-public-partnership initiative (http://www.fi-ware.eu/) is currently developing an ICT platform for FI Internet services for Europe. The FI core platform is a major outcome of this project and it supports the operation of internet-based services on a large scale and in a reliable way. Similar projects like FINES

([22], [17], [18]) Cluster's (e.g. COIN [20], NISB, EBBITS) projects or other interoperability-related programmes (e.g. [21]) develop or utilize comparable ICT service support platforms.

Within SP3, several core results are developed:

- A FINES reference architecture with two architectural blueprints in WP31,
- FI Platforms Federation (Consumer Marketplace & IoT Manager) in WP3.2,
- FI Utility Services (SSO & Feedback management) in WP3.3,
- Value Added Services (Production Planning & Scheduling, Team Building, Product Maintenance) in WP3.4.

The architectural blueprint for GI ecosystems is described in Chapter 3 to establish a basis for the subsequent architecture discussion of a Core Manufacturing Platform in Chapter 4.

3. Future Internet Ecosystems Architectural Blueprint Update

The architectural blueprint for FI ecosystems as described in D31.1 has been evaluated and only *minor modifications* are needed which are reported in this deliverable.

FI ecosystems are built upon a platform which supports all relevant processes that are carried out by the stakeholders of the ecosystem. The architectural blueprint for a general platform and for a domain-specific platform is described in D31.1 – adapted to the manufacturing domain. This domain can profit from the servitization principles and usage of FI technologies.

As shown in Figure 1, with service-product bundles being introduced for enriching manufacturing products, combinations of partners, suppliers, and customers form specific *MSEE ecosystem instances* (i.e. for Indesit, Ibarmia, or others).

In such an ecosystem instance, both the products and the accompanying services for these products are built and managed along their entire lifecycle phases – from idea stage until product / service delivery and consumption of product and services by the customers.

All ecosystem stakeholders use a specific instance of the *Manufacturing Core Platform* in their MSEE ecosystem instance. This platform instance is adapted to the special use case. In the context of Enterprise Ecosystems, and of MSEE in particular, multiple enterprises take part in a single ecosystem, while the same enterprise may take part in multiple ecosystems. Users may be directly registered as ecosystem members, or indirectly as corporate employees.

Core building blocks of such FI platforms can make use of specific elements of an underlying *FI Core Platform,* like the sets of FI generic enables (e.g. FI-WARE reference implementations for FI GEs or other similar implementations), FI utility services see also [19] for a different viewpoint),, FI federation services, FI services, and value-added services. Components from other FI platforms can also be utilized within the Manufacturing Core Platform.

This blueprint of an ecosystem remained valid during the investigation in MSEE so far and it served as blueprint for the different ecosystem implementations in MSEE with respect to the **architectural viewpoint** (other viewpoints are also described in other MSEE work streams / deliverables).

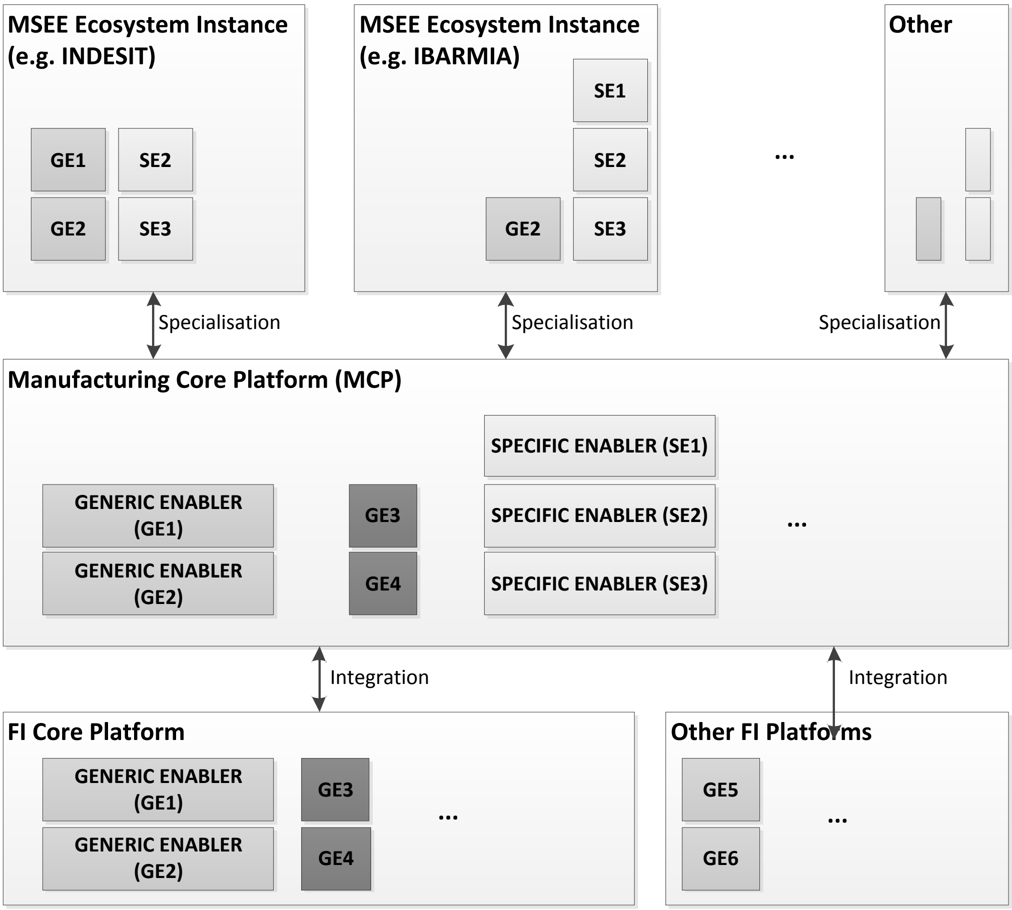

Figure 1 Applying FI Core Platform to Manufacturing Ecosystems (adapted from [5])

In the next subchapters the **core architectural components** of FI ecosystems are discussed:

- FI Federated Services,
- FI Utility Services,
- FI Value-added Services.

All components can be seen as enablers (generic enablers (GE) or specific enablers (SE)) in the sense of the underlying adopted FI-WARE architecture terminology.

3.1. FI Federation Services

3..1.1. The concept of Service Federation in the FI Environment

Service federation is an important feature of modern ICT systems. It refers to a situation where several service providers work closely together to provide a seamless service to their combined group of end-users. Prime examples of federation are international telephony and GSM roaming. In both cases, the end-user only has a formal relationship (the subscription) with its own operator, but is able to access users and resources in the domain of other operators. Technical and business agreements between all involved operators ensure that the end-users experience a seamless service, as if it were offered by one global telecom operator [9].

The following important types of service federation are distinguished:

- Communication service federation,
- Content service,
- e federation,
- Context service federation.

These types of service federation are not mutually exclusive. A single service, offered across domains, may include several types of service federation. The types of federation are named after the group of services where they are most important. [9].

Service Federation aspires to be a paradigm shift for the Future Internet. It adopts practical ideas and tools that already exist and widely used by developers in the fields of Software as a Service (SaaS), Infrastructure as a Service (IaaS) and Software Engineering. At the same time it goes a step beyond the specification mechanisms and architectures of federations: it also considers processes on how one can exploit such mechanisms and architectures in a convenient way during the operating phase.

As far as it concerns Future Internet, a federation framework should allow to automatically instantiate the fixed, mandatory part of the core platform as well as it should provide the necessary freedom to discover, select and compose the optional GEs. Once core platform instances have been composed and instantiated, federation mechanisms come into play. Federating distributed, usage-area-specific testbed resources should allow integrating distributed CPIs into local testbed components and to interconnect distributed elements to create connected, virtual testbeds across administrative domains [11].

Therefore the overall service lifecycle comprises a creation/composition phase, a federation phase and an application programming and deployment phase, based on standardized APIs.

3..1.2. Federation of FI resources for MSEE

In MSEE, the goal of Future Internet Platform Federation is to provide Future Internet collaboration functionalities to the MSEE platform, thus enabling the implementation of FI-inspired applications and services. These functionalities are to be derived from other FI platforms, either by interfacing with instances of services/applications running on other platforms or through the re-use of their components. Candidate platforms include the FI Core Platform developed within FI-WARE[1], as well as other FI-inspired projects.

The conceptual architecture of Future Internet Platform Federation has been initially presented in deliverable D32.1 “Future Internet Platform Federation Specifications and Architecture” (M12)[5]. While maintaining adherence to a “stack” paradigm, this basic architecture has been refined, as a result of the work done towards D32.3 “FI Platform Federation First Prototype”(M18)[6].

The conceptual **architecture** consists of four layers:

- *Future Internet Resource Layer*: corresponds to FI resources from other platforms, either in the form of service instances running in other platforms or reusable components.
- *Importation layer*: corresponds to mechanisms and frameworks for importing FI-functionalities derived from the previous layer into the MSEE. This may include e.g. interfaces to running instances, compatibility adapters for code or executable components, etc. This layer makes FI-inspired functionalities available for development. However, they are still unrefined and unsuitable for consumption by MSEE platform actors.

[1] http://www.fi-ware.eu/

- *Service Delivery Layer*: Based on the products of the previous layer, this layer corresponds to the development of logic / interfaces for the creation and provision of specific FI-inspired services/applications to MSEE platform actors.
- *Consumption Layer*: corresponds to the use of FI-based services by MSEE actors and applications, or with registration with MSEE's Service Delivery infrastructure for re-use.

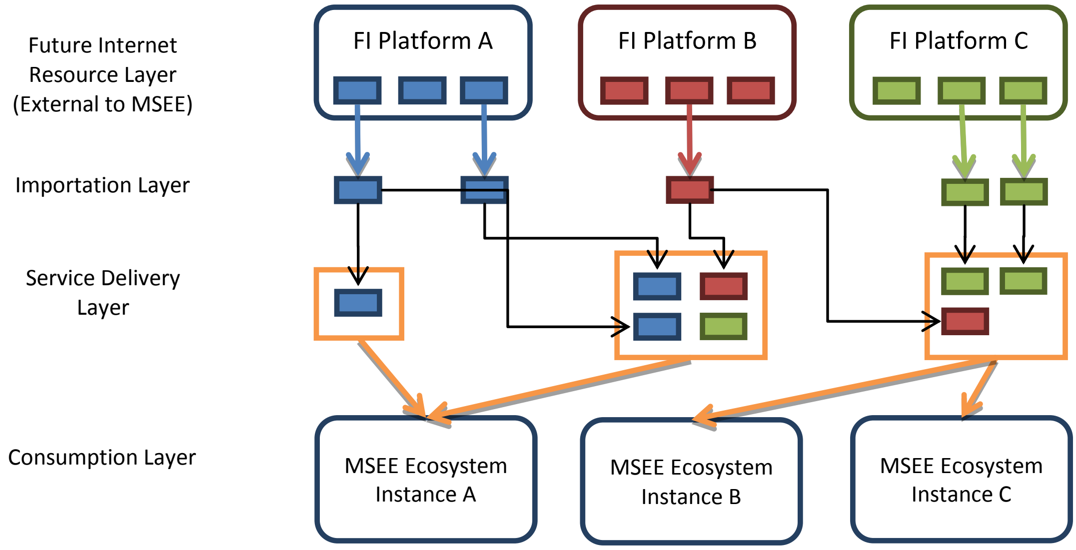

Figure 2: Conceptual Architecture of MSEE FI Platforms Federation

In developing FI-federated services for MSEE, two factors must be taken into consideration: First, the state and availability of resources from the FI domain, and, second, the areas where MSEE could gain the maximum benefit.

In the FI domain, several projects have the potential to provide useful assets. Various platforms in the domains of Internet of Things, Internet of Services, Internet of Contents/Knowledge and Internet for and by the People have been examined and several potential candidates have been identified.

The necessary functionalities have been identified by examining the needs of manufacturing service ecosystems, as reflected by the project use cases. Common features include interactions between service providers and service consumers for the manufacturing and handling of real-world, material items and goods, as well as the bundling of services with products and provision of goods "as-a-service".

The analysis focused on selecting functionalities providing the largest amount of value, combined with relatively favorable conditions regarding federation and re-use. Based on this work, two FI-inspired services have been selected for implementation in the first prototype and are currently being implemented: the Consumer Marketplace and the Internet-of-Things Manager [5].

3..1.3. Consumer marketplace

The Consumer Marketplace will be an application based on the FI-WARE "Marketplace" Generic Enabler, enabling "service consumers" to browse service offerings owned by "service providers". For service providers, the application will allow the creation and administration of stores and service offerings, while, on the consumer side, it will enable searching and browsing.

The Consumer Marketplace will also include a prototype implementation of the "Federation of Platform Services" concept described in [5]. The MSEE Consumer Marketplace application is based on the FI-WARE "Marketplace" Generic Enabler. The "Marketplace" GE implements a "generic" store, meaning that, in theory, it can be re-used to create stores with varied offerings, each running their own "Marketplace" GE instance. Since their API is standardized, it's possible to create functionalities based on the interoperability between "Marketplace" GE instances.

In the Consumer Marketplace application, this will be leveraged by the "Federated Search" feature. It will allow a store, based on the Consumer Marketplace service (which, in turn, is based on the "Marketplace" GE), to dispatch search queries to other "Marketplace" GE-based stores. From a business perspective, a store using "Federated Search" capabilities may complement a particular offering with relevant offerings from ecosystem partners (e.g. purchasing an "intelligent maintenance" service for a CnC machine, while also selecting from a range of branded maintenance consumables).

From an architectural standpoint, the Consumer Marketplace application will be implemented as a standard web application with a **three-tier web architecture**, which will include a frontend (both as a REST API and a web user interface), a request-response processing layer implementing communication and business logic and an information management layer including business and functional data.

A **detailed description** of the application can be found in the documentation included in deliverable D32.3 "FI Platform Federation first prototype" [7].

3..1.4. IoT Manager

The "IoT Manager" will be an application aiming to provide an abstraction layer between MSEE platform actors and devices participating in MSEE ecosystems.

Its goal is to provide a standardised interface through which MSEE applications will be able to discover, query and manage IoT-enabled devices. The initial specification of the IoT manager, as provided in [5], recommended the use of Generic Enablers from the Internet-of-Things Enablement chapter of FI-WARE[2]. These enablers would provide a, more-or-less, complete range of base functions for device management. However, alternative resources are also being considered, e.g. re-usable components from Internet-of-Things FI projects such as OpenIOT[3].

It is expected that several instances of the IoT Manager Service will be running in a given MSEE ecosystem, corresponding to multiple applications interoperating with the ecosystem's IoT-enabled devices. The application will enable the dynamic discovery of devices, as well as providing a unified and consistent addressing scheme.

---

[2] http://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/ Internet_of_Things_(IoT)_Services_Enablement_Architecture

[3] http://openiot.eu/

Its main use cases (depending on the capabilities of the underlying devices) are:

- Polling devices (i.e. retrieving data from devices, e.g. sensors)
- Pushing to devices (i.e. sending commands and data)
- Subscription to devices (i.e. establishing a connection for the reception of dynamically available data, such as events and other ad-hoc messages)

Architecturally, this application will employ a **three-tier architecture**, with a frontend including a web-based administration UI and a REST API, a Domain layer handling Authentication/Authorisation, Auditing and Device Management and a Data Layer.

A **detailed description** of the application can be found in the documentation included in deliverable D32.3 “FI Platform Federation first prototype” [8].

3..1.5. Further Work

Development of Future Internet Platform Federation for MSEE follows a two-stage model, with an initial prototype and a final version. Both of the services described above have been specified for inclusion in the first prototype (D32.3 – M18). This specification was based on an initial assessment of the availability of FI resources and the needs of the MSEE use cases.

For the final version, several issues will need to be investigated, such as the inclusion of new services, the evolution of the existing two prototypes and the examination of usable resources from FI-inspired platforms. The results of this investigation, the availability of FI resources and the needs of the MSEE use cases will determine, to a significant extent, the development roadmap for FI platform federation in MSEE.

3.2. FI Utility Services

The first release of this document (M6 issue) [1] provided a definition of what Utility Services are and how they can be categorized, in the Future Internet context. In this new iteration, and in the light of lessons learned from applying these concepts to the MSEE Use Cases, we can summarize this vision in two brief sentences: *an Utility Service is a commodity service addressing any common functionality that is used in Enterprise Ecosystems; in the MSEE context, Utility Services provide basic and generic services that are exploited by use cases.*

Considering Enterprise Ecosystems as an Usage Area for the Future Internet, we may see Utility Services as Specific Enablers for next generation Enterprise Systems. Figure 3 shows the relationship among Generic Enablers, Specific Enablers and Value-added Services.

The first release of the **FI Utility Services specification and architecture** document [2] (M12 issue) listed and described different categories of Utility Services, starting from necessities and requirements elicited from MSEE Use Cases. It identified two concrete Utility Services: **Single Sign On** and **Feedback Management**. The first prototype implementation of these services was released as deliverable D33.3, M18 issue.

In the following sections we provide an architectural overview of both – see the above mentioned deliverable [3] and related factsheets for further details.

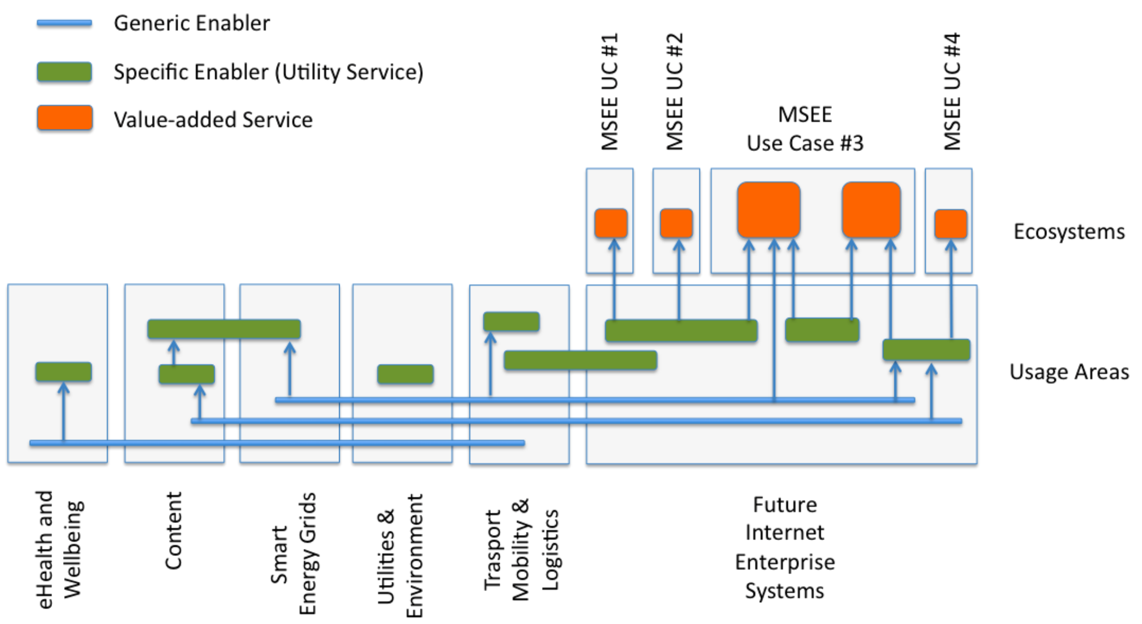

Figure 3: Relationship among Generic Enablers, Specific Enablers and Value-added Services.

Note that the original Single Sign On Utility Service was later renamed to **Federated Single-Sign-On**, to underline the new functionality layer introduced on top of a *traditional* single-sign-on solution.

3..2.1. Federated Single-Sign-On

Single-Sign-On (SSO) is an authentication mechanism which allows users to provide their credentials once, and then have seamless access to a vast array of protected network resources. With SSO in place, authentication can be enforced across components of a complex system with no negative impact on user experience, as users are not required to submit new credentials (*sign-on*) to each application they reach. Furthermore, the global security of the system is enhanced: user credentials travel on secured communications channels, and are never disclosed to applications.

This scenario is not new, and is widely seen in action on corporate networks: any Enterprise may leverage SSO systems to wire together several applications living on its own intranet / extranet. A typical SSO installation is a centralized server which on one side provides a common login page and a security token service, and on the other is connected to a corporate user database or directory. SSO is then used as a system integration tool.

In the context of Enterprise Ecosystems, and of MSEE in particular, the above described SSO usage scenario falls short. Multiple Enterprises take part in a single Ecosystem, while the same Enterprise may take part in multiple Ecosystems. Users may be directly registered as Ecosystem members, or indirectly as corporate employees – and in the latter case, they should be considered as members by any Ecosystem of which their employer is member. To address these issues in the MSEE project, an enhanced version of SSO was devised: **Federated SSO**.

Federated means that while authentication is still centralized, user data is not: each node of the Federation may qualify as a **Local Security Authority (LSA)**, *responsible* for the authentication of a subset of users and *trusted* by all the other nodes. User data is never shared or duplicated outside the boundaries of the LSA it belongs to. Every Ecosystem instance in a Federation has the role of LSA, while Enterprises may choose to run their own

LSA Service (even more than one - e.g., one per business unit) or not. Consequentially, users of an Ecosystem Federation can be registered at the Enterprise or the Ecosystem level – or even globally, if the node which provides the central authentication service also runs a top-level LSA Service. To summarize, Federated SSO is an *enterprise integration* tool.

Figure 4 shows the relationships existing between Federated SSO components (in red) and the other actors of an MSEE Ecosystem Federation.

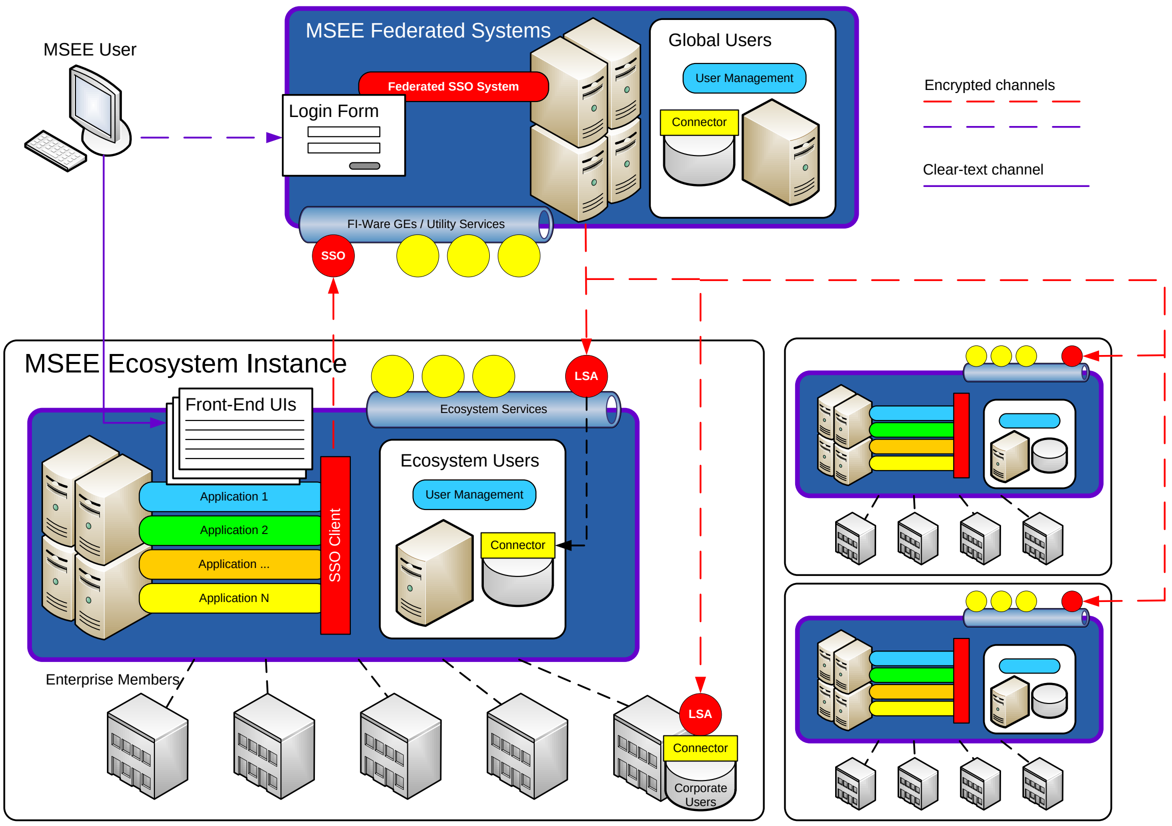

Figure 4 Federated SSO: the Big Picture

The Federated SSO Utility Service is not limited to user authentication, but also supports the *attribute release* feature - i.e., the capability of assigning a set of attributes to an user as the result of a successful login. In other words, each trusted LSA may *make statements* about a specific user, and these statements may then be used at the application level to check security constraints and/or customize the user experience. In the context of MSEE Federated Ecosystems, a standard schema of user attributes is defined, providing common ground on which access control policies may be built – for details on this, see the relevant technical documentation [4].

Finally, the relation of mutual trust between the centralized Federated SSO server and each LSA Service is protected by the use of secure communication channels and by digital certificates as a guarantee of the actual identity of each actor.

3..2.2. Feedback Management

The rapid advance of ICT technologies and their increasing importance in the manufacturing domain brings an exponential growth in on-line communication opportunities. Being able to communicate and engage via a multitude of Internet, Web, Web2.0, and social channels becomes more and more important for single enterprises and virtual manufacturing enterprises. The growing number of communication channels and interaction opportunities generates new challenges in terms of scalability. Single as well as virtual manufacturing enterprises require new skills and more efficient access means to scale and filter the exponentially increased offer. They actually need to actively listen to 'listen the voice' of customers and partners to continuously improve the offer and the overall business process. According to a study realized by the Aberdeen Group in 2011, companies using customer feedback management services have better customer retention's rate.

A scalable solution that enables enterprise to react in a competitively advantageous manner to customers', competitors and partners comments and reviews is needed. Such a solution will help them shape their product and service offerings to reflect the public perception, and manage their reputation online.
The goal of the Feedback Manager Service in MSEE is to provide a solution to the challenges mentioned before. The Feedback Management Service aims at providing a unified publication method in a multichannel environment and the associated automatic aggregation of the feedback provided there, releasing from the burden of manually publish and collect information from these channels separately. Additionally, the Feedback Management Service will offer scalable means to publish on various channels (Facebook, Twitter, LinkedIn, etc.).

In short, the Feedback Management Service is used to distribute information in various channels, as well as collect and analyze feedback from those channels and actively engage in conversations (i.e. reply to comments). The functionality of the Feedback Management Service is provided as REST service and data can be exchanged with the service either in XML or JSON format. The Feedback Management Service is implemented using Ruby on Rails[4], and is conceptually split into two different sub-components. One sub-component is the *Core* part that includes the REST API and the other is the *Adapters* that are responsible for the communication with the different channels. From a functional point of view the Feedback Management Service contains the following modules:

- ***Role Management*** – this module provides role management functionality for those that are using the Feedback Management Service. Two roles are envisioned and supported for the moment i.e. system *user* and *administrator*. A *user* is allowed to use the functionality of the service. An *administrator* has additional rights being able to administrate (add, delete, etc.) user accounts and has an overview about the health of the service.
- ***Publication*** – this module is responsible for the publication of content into channels. The Publication module is also responsible for adaptation of content to be published to the specific requirements of the different channels. The adaption is made on the base of the channel capabilities, e.g. shortened messages are posted to microblog platform, such as twitter. To achieve scalability the Feedback Management Service uses a message-oriented architecture to implement an asynchronous publication process. For this purpose the Advanced Message Queuing Protocol, short AMQP[5] is used.
- ***Feedback Collection*** – this module is responsible for the collection of various feedbacks across multiple channels. Feedback can have different forms. It can either

[4] http://rubyonrails.com/

[5] http://amqp.org/

be textual (comments, replies, etc.), a certain amount of positive (like, +1, thumbs up, etc.) or negative feedback (thumbs down, bury, down vote, etc.), or some other measurement of a user's response to a published item. The Feedback Management Service can collect all these different types of feedback using various polling strategies.

3.3. FI Value-Added Services

Three value-added services are developed within WP3.4: A "Production Planning App" (TXT), a "Team-Building App" (SILO), and a "Product Maintenance App" (SAP). As described in D2.4.1, the first two smart enterprise ecosystem applications support the innovation phase of product-service bundles where a service-bundle is defined and implemented. The third application supports the consumption phase of product-service bundles where the bundle is delivered to consumers.

Looking from the perspective of a stakeholder in the *innovation phase* of product-service bundles, the two following applications are currently serving as examples:

First, the "**Product Planning App**" aims to facilitate the creation of distributed production plans providing to the production manager the possibility to plan its production based on ecosystem tangible assets (see D3.4.2[factsheet]). The ecosystem virtualised tangible assets collaborative and their capacity available published by the producer are used as a bases for the definition of the plan. This allows the manager to rely on the whole ecosystem in the planning involving faster and easily other actors that can easily work collaboratively on that.

Second, the "**Team Building App**" aims to facilitate the creation of personnel teams and partnerships in the unique environment of manufacturing ecosystems supported by the MSEE platform. Traditional Human Resource activities involve a top-down paradigm, where personnel assignment and team building processes were confined within organisational borders. However, the concept of the manufacturing ecosystem requires close coordination with partners, meaning that the traditional modes of defining inter-organisational collaborations and teams may be inadequate.

The Team Building App will support an enhanced method for team creation, better suited to MSEE ecosystems. This will be achieved by introducing a common environment for these operations, including a common knowledge base and tools for assigning individuals to tasks and teams. This common environment will also be adequately scalable to support both intra- and inter-organisational collaboration. When applied to the MSEE use cases, the necessary intangible asset repositories will be populated with HR data relevant to the specific use case and supplied by the pilot users.

For example, the application will be used in the IBARMIA "intelligent maintenance" case for effectively matching skilled technicians to specific maintenance tasks. The detailed requirements and specifications of the Team Building App have been described in D34.1 "Detailed Specifications of Next Generation ESA value-added services" [12]. A notable modification to the original specification is the decision to avoid re-use of existing Singular Logic HR products in favour of working with an independent, custom-built architecture for MSEE. A detailed description of the application specifications can be found in D34.2 [13].

Looking from the perspective of a stakeholder in the *consumption phase* of product-service bundles, the "**Product Maintenance App**" (Figure 5**Error! Reference source not found.**) supports this phase and can be used from mobile devices and desktops.

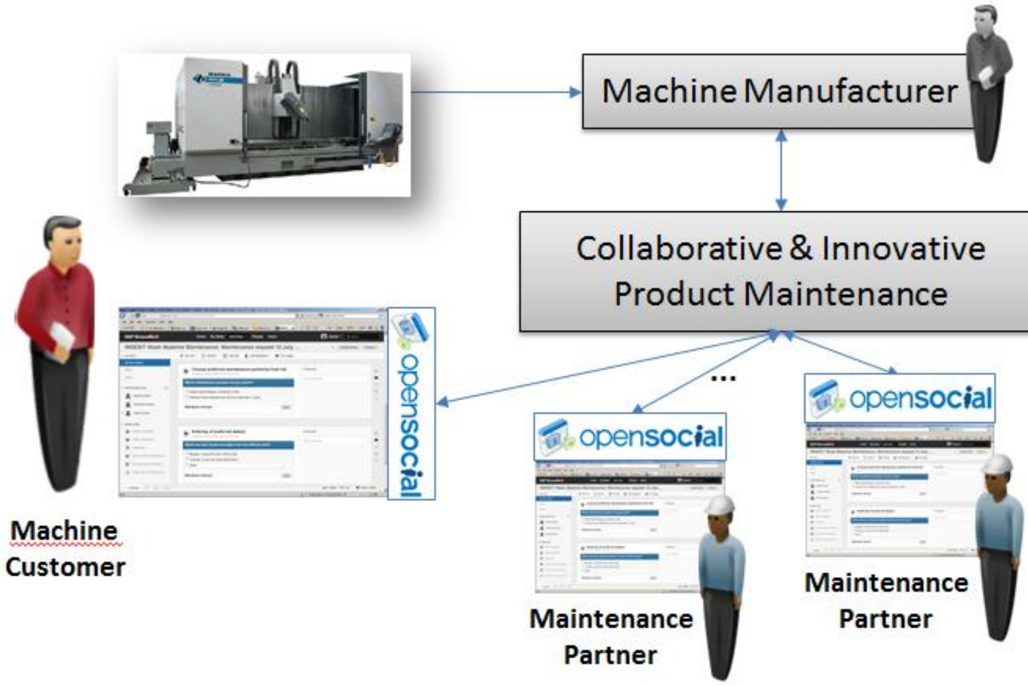

Figure 5: Product Maintenance Application

The Product Maintenance Application connects customers of a product, the product manufacturers, and a set of maintenance service providers in an innovative collaborative maintenance case handling scenario. Therefore, it offers an innovative experience of an added-value business service for the customer of a bundled product (like a machine) and establishes a direct communication channel between him and the product manufacturer and maintenance service providers.

A customer of a product can use the application to check if maintenance is needed for his product or to access helpful product information extracted from specialised enterprise systems. The app guides the user to the problem resolution – either by providing help-yourself information (if wanted) or by ordering maintenance service and scheduling a maintenance visit with a preferred service provider.

As its most important feature, the customer can use this social *collaborative* product maintenance application in a flexible and natural way to talk directly with the product manufacturer and/or maintenance service providers (like a communication using a collaborative channel like Twitter). The Product Maintenance Application can be *adapted* to innovative maintenance service scenarios and is not restricted to a fixed use case scenario of a product + service bundle. The application can access other value-added-services which may be built either on top of the MSEE platform or built into existing enterprise systems, or provided in the Open Internet.

In an example use case, a company *AdProduct Inc*. in Italy owns a set of manufacturing machines sold by the *company IBARMIA* in a plant. Each machine carries a QR code as an additional product identifier which is used for customer service scenarios for *IBARMIA* customers.

The Product Maintenance App offers the following features to its users:

- Identification of a maintenance need for a product: An employee Mr. Miller from *AdProduct Inc*. uses the Product Maintenance Application to scan the QR code of one of the IBARMIA machines in the plant to identify make, model, and serial number. For this machine, he can access relevant product information and - if needed – he can create a maintenance order.
- Finding & selecting optimal maintenance partners: Beyond typical state-of-the-art, the customer can then select from an automatically optimized set of optimal maintenance providers according to his own preferences, for example *BestService Ltd*. and other geographically near maintenance providers.
- Scheduling maintenance actions between customer and maintenance partner: Mr. Miller can schedule a maintenance visit with a maintenance expert from his preferred BestService *Ltd*. and he can communicate directly with him about details of this visit using the application.
- Feedback collection as a basis for continuous maintenance service improvement: The maintenance visit is tracked with the Product Maintenance Application and after successful completion of the maintenance case the feedback from the customers is collected.

The Product Maintenance Application provides *increased business value* beyond the state-of-the-art for its users:

- *Customer*: Benefit from one unified, easy-to-use, and flexible maintenance service for the own products; Communicate directly with product manufacturer and maintenance partners in a social and collaborative way; Schedule maintenance actions with preferred maintenance partners at preferred dates; Access important product information with an easy-to-use application from everywhere and at every time.
- *Product Manufacturer*: Manage the maintenance needs for sold products effectively and reduce costs for individual maintenance case handling; Establish a direct social connection to the product customers; Coordinate a fleet of maintenance service providers time- and cost-effectively.
- *Maintenance Service Providers*: Acquire additional maintenance orders via this business channel with minimal acquisition costs; increase the customer's satisfaction through interactive direct communication with the customer; lower costs of scheduling maintenance actions (e.g. reducing contacts of customer calling a product support call center for this purpose)

All three applications represent prominent examples how FI value-added services can be realized following the architectural blueprints in D31.1 and this deliverables. They represent applications which make use of underlying platform features on the Frontend Layer, the Backend Services Layer, and the Enterprise Layer of the Manufacturing Core Platform.

All three assets are described in more detail in deliverable D3.4.1 and D3.4.2.

3.4. Integration

As described in D31.2, in order to build a *domain-specific* core platform such as, for example, the Manufacturing Core Platform, existing components (implementations of generic enablers) from available FI platforms may be utilized as well implementations of the core components mentioned above. All used components are integrated into a platform instance. Integration comprises assembling and extending such components in this context.

4. Manufacturing Core Platform Architectural Blueprint Update

In the last deliverable we have presented an initial version of an architectural blueprint for a *Manufacturing Core Platform*. This platform is used to create and operate domain-specific manufacturing ecosystems. In this section we report on the second revised version of the platform blueprint.

As a guideline, the architectural guidelines did not need to be modified in the Large, instead some *incremental* modifications have been observed and they are reported here.

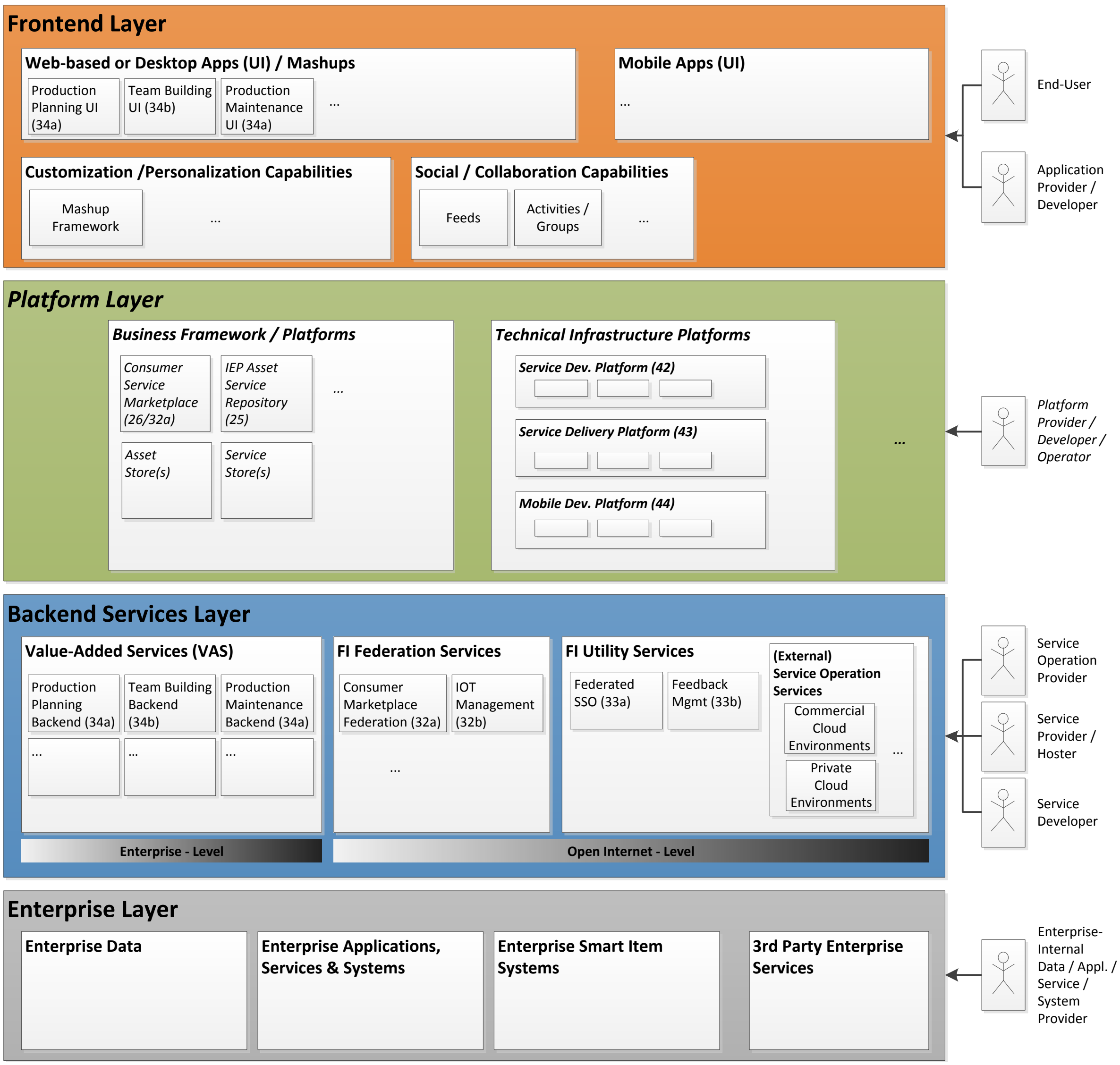

Figure 6 Revision of the architecture blueprint of *Manufacturing Core Platform*

The **Manufacturing Core Platform** (Figure 6) consists of the following layers: Frontend Layer, Platform Layer, backend Services Layer, and Enterprise (-internal) Layer. All four layers are described in the following paragraphs.

The platform is described with the focus on the **architectural viewpoint** (only) in this deliverable. Other viewpoints highlight different dimensions of the platforms. For example, in SP4 deliverables, aspects of the partial architecture for the Technical Infrastructure Platforms are described from not only the architectural viewpoint but also from other viewpoint perspectives as well.

The **Frontend Layer** contains all application frontend / user-interface (UI) capabilities for offering the interaction with the application users:

- *Mobile applications*, *web-based applications*, and *desktop applications* can be accessed to interact with the applications.
- If needed, state-of-the-art *communication capabilities* (like social feeds, groups) are offered to users in a variety of forms and implemented in various tools.
- *Customization capabilities* allow for the tailored customization of working environments to the individual, personal needs of the end-users. As an example, mashup user-interface frameworks can be used to provide personalized and situation-specific user experiences for the end-user.

The **Platform Layer** contains all

- The *Business Framework / Platform* contains all assets conceptualized for commercial operations on services and/or data moved between providers and consumers. For example, marketplaces are logical components of this platform. In particular, asset stores, and service repositories are also part of the business platform. Examples of such assets are shown in the Figure, such as Consumer Marketplace (WP26), IEP's Assets Repository (WP25).
- The *Technical Infrastructure Platform* represents all technical capabilities that are provided by the platform. In this case the entire platform assets from SP4 are part of the Technical Infrastructure Platform, such as service development platform, service delivery platform, mobile development platform, and so on. (all developed in SP4 and described there in more detail). Please note that in the MSEE context there are typically no more components in the Technical Infrastructure Platform of the Platform Layer beyond SP4's assets, but in general this can be the case, of course).

The **Backend Services Layer** addresses the following entities (e.g. developed in SP3):

- *FI Federation Services*: Such services provide capabilities that federate a set of heterogeneous or distributed entities. In MSEE, examples of federation services are the Consumer Marketplace Federation Service (WP32a) and the IOT Manager. They have been described in Chapter 3.1 above and in more detail w.r.t architecture in D32.3 ([3] [7] [8]).
- *FI Utility Services*: Services with a generic nature (depending on the viewpoint) are located in this part of the platform. In MSEE, the federated SSO service (WP33a) and the Feedback Management Service (WP33b) are prominent examples. Utility Services in FI are described in Chapter 3.2 and in D33.3 ([3]).

- *Value-Added Services*: Services with special capability offerings that provide an added values to their consumers (depending on the viewpoint) are located here, as described before. In MSEE, examples are the Production Planning and Scheduling App, the Team Building App, and the Product Maintenance App (all addressed briefly in Chapter 3.3 and in more detail in D34.2 ([13]).

The **Enterprise Layer** is comprised of the following artifacts:

- The existing collection of *enterprise applications, databases, systems, and services* which are used from the enterprises is located on this layer. Examples are in the MSEE context seen in the “Personal Knowledge Database” (Production Planning App) and other similar enterprises.
- In particular, enterprise-level *smart item systems* can be connected in this layer. Such systems provide capabilities to collect, process and forward smart item data into the Manufacturing Core Platform. Consumers of such data can be in themselves services in the overall MSEE framework (for example, the IOT manager service can consume data from such systems in the enterprise layer).

A set of **stakeholder roles** can be typically located for the single platform layers (Figure 6), for example as follows:

- *Frontend Layer*: *End-user* and *Application Provider / Developer* can access UI / frontend capabilities or develop implementations of such capabilities for the user. Likewise, developers can make use of MSEE capabilities for application or service development in particular (e.g. developed in SP1/SP2/Sp4), of course.
- *Platform Layer*: *Platform Provider* role, *Platform Developer*, or *Platform Operator* address platform aspects for the complete lifecycle management for platform assets from initiation, development, testing, operation, and de-commissioning of the platform modules. SP4 develops the needed functionality within MSEE (please see SP4 deliverables for extended details).
- *Backend Services Layer*: the roles *Service Developer*, *Service Provider / Hoster*, *Service Operation Provider* address the aspects of lifecycle management for services located on this layer. Of course, these roles make use of the frameworks on the Platform Modules Layer (above), most prominently the Service Development / Delivery Platforms in SP4.
- *Enterprise Layer*: The artifacts managed on this layer are managed by the roles *Enterprise-Internal Data Provider*, *Application Provider*, *Service Provider*, or *System Operator*. Their responsibility is to create, offer, maintain and operate assets which are outside of the core platform located in the enterprises.

Please note, that this role-layer assignment *example* shows how roles may be applied to layers of the Core Manufacturing Platform with its focus on the **architectural viewpoint** adopted in this deliverable. Other viewpoints are discussed in SP1, SP2, and SP4 also which may lead to different possible assignment structures which may be valid at the same time due to different adopted viewpoint perspectives.

In D31.1, the relationship of the Manufacturing Core Platform and the Service Delivery Framework (SDF) [14] has been described in detail. *No changes* were necessary for the revised version of the architecture blueprint so far. The Manufacturing Core Platform addresses the requirements which have been discussed in D31.1.

The architecture blueprint for the Manufacturing Core Platform is inspired by the architectural principles of the FI Core Platform (FI-WARE), as described in D31.1. A single platform instance – the MSEE platform - comprises a selected subset of the components / modules of the manufacturing core platform in order to address tailored needs for the application domain and application scenario to be supported. In SP4, the functional and modular architecture of MSEE platforms is addressed in detail within WP4.1. The architecture of instantiated MSEE platform is in each case a specific instantiation for a specific usage / application scenario and it is inspired by the architectural principles in WP3.1 described in D31.1 and D31.2.

5. Conclusion

The deliverable contributed an architectural blue-print for Internet-based platforms and utility and value-added services for *Future Internet Enterprise Ecosystems* in its second iteration.

In particular, the deliverable detailed this blue-print for the manufacturing domain. In addition, we contributed an initial set of architectural principles for the existing enterprise applications that prescribe how they can deliver their functionality as value-added services. The architectural aspects of the different main components of both the Ecosystem Architecture as well as the Core Manufacturing Platform have been reported in their most recent state summarizing them from the architectural viewpoint.

While D31.2 serves as the **anchor deliverable** for the SP3 architecture results, more detailed technical and architectural information about the main components of the blueprint can be found in D32.3 (Federation Services), D33.3 (Utility Services), and D34.2 (Value-added Services).

Please note, the M24 issue of this deliverable will further refine the architectural blueprints a third time to arrive at a final stage according to the MSEE project’s progress after M18.

6. References

[1] MSEE D31.1 Functional and Modular Architecture of Future Internet Enterprise Systems
[2] MSEE D33.1 FI Utility Services specifications and architecture
[3] MSEE D33.3 FI Utility Services first prototype
[4] MSEE D33.3 Factsheet#1 Federated SSO Server
[5] MSEE D32.1 Future Internet Platform Federation Specification and Architecture
[6] MSEE D32.3 FI Platforms Federation First Prototype
[7] MSEE D32.3 Factsheet: Consumer Marketplace
[8] MSEE D32.3 Factsheet: IoT Manager

[9] M. Oskar van Deventer, Paul Tilanus, Mike Schenk, Eelco Cramer and Joost Adriaanse: Future Internet: towards context information brokering. In G. Tselentis et al: Towards the Future Internet. IOS Press, 2009, pp. 250-262

[10] Christos Tranoris and Spyros Denazis: Federation Computing: A pragmatic approach for the Future Internet, IEEE Communications Society, International Conference on Network and Service Management (CNSM), 2010, pp. 190-197

[11] Sebastian WAHLE, Thomas MAGEDANZ and Anastasius GAVRAS: Conceptual Design and Use Cases for a FIRE Resource Federation Framework. In G. Tselentis et al: Towards the Future Internet. IOS Press, 2010, pp. 51-62

[12] MSEE D34.1 Detailed Specifications of Next Generation ESA value-added services

[13] MSEE D34.2 First Prototypes of ESA value-added services

[14] 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

[15] H. Couturier, B. Neidecker-Lutz, V.A. Schmidt, D. Woods: *Understanding the Future Internet*, International Research forum 2010, New York, Evolved Technologists Press, 2011.

[16] Domingue, J.; Galis, A.; Gavras, A.; Zahariadis, T.; Lambert, D.; Cleary, F.; Daras, P.; Krco, S.; Müller, H.; Li, M.-S.; Schaffers, H.; Lotz, V.; Alvarez, F.; Stiller, B.; Karnouskos, S.; Avessta, S.; Nilsson, M. *Preface*. In Future Internet Assembly 2011: Achievements and Technological Promises. Springer Lecture Notes in Computer Science, Vol. 6656 , Subseries: Computer Communication Networks and Telecommunications , 1st Edition., 2011

[17] Man-Sze Li, Mehmet Kürümlüoglu, Margaretha Mazura, Roelof van den Berg. *Future Internet Enterprise Systems (FInES) Cluster – Position Paper*. Final Version 3.0, 1st September 2009, European Commission – Information Society and Media.

[18] Angelucci, Daniela; Missikoff, Michele; Taglino, Francesco. *Future Internet Enterprise Systems: A Flexible Architectural Approach for Innovation*. In Domingue, John; Galis, Alex; Gavras, Anastasius; Zahariadis, Theodore; Lambert, Dave; Cleary, Frances; Daras, Petros; Krco, Srdjan; Müller, Henning; Li, Man-Sze; Schaffers, Hans; Lotz, Volkmar; Alvarez, Federico; Stiller, Burkhard; Karnouskos, Stamatis; Avessta, Susanna; Nilsson, Michael (eds): The Future Internet. Springer, Lecture Notes in Computer Science, 2011.

[19] Man-Sze Li: The Economics of Utility Services in the Future Internet. In Georgios Tselentis, Alex Galis, Anastasius Gavras, Srdjan Krco, Volkmar Lotz, Elena Paslaru Bontas Simperl, Burkhard Stiller, Theodore Zahariadis (Eds.): Towards the Future Internet - Emerging Trends from European Research. IOS Press 2010, ISBN 978-1-60750-538-9

[20] Patrick Sitek, Sergio Gusmeroli Marco Conte, Kim Jansson, Iris Karvonen (Editors): The COIN Book Enterprise Collaboration and Interoperability, 1st Edition, 2011, Verlag-Mainz, Germany

[21] Man-Sze Li, Ricardo Cabral, Guy Doumeingts, Keith Popplewell: Enterprise Interoperability Research Roadmap Final Version, 31 July 2006

[22] Cristina Martinez Gonzalez, Man-Sze Li, Nikolas Provatas, Sylvie Woelfflé: Future Internet Enterprise Systems (FInES) Cluster - Cluster Book, ICT2010 EVENT VERSION, June 2010