Third party innovation enablement in FI-WARE

Please go to:

https://docs.google.com/document/d/1NhjB0ctRxhoz5y9bEwJijAJzIbt_WB3WV1P2yQ1mrSg/edit?pli=1#bookmark=id.rte7xbzgl3cq

Target audience: Reviewers, Architecture Board, FI-WARE steering committee
Contributing Partners: chapter/GE level, SAP/TID providing hints, coordinating editing and dealing with common parts
Dissemination Level: public
Timeline:

- 9. July – introduction provided to partners/chapters
- 9. July – 20. July contributions by partners/chapters
- 30. July – final version ready for review

Review:

- 2. August – Peer reviewer 1
- 5. August – Review by Deliverable Lead
- 10. August – Release to commission

Content:
This deliverable will document the “key choices” made in FI-WARE with respect to 3rd party innovation. According to the description in deliverable in the DoW (to be taken as terms of reference but with freedom to interpret it along the lines of what we believe is meaningful/useful):

> During the design of FI-WARE, the FI-WARE project will make choices that will affect the way FI-WARE can be used by third parties. Some of these choices will allow and some will limit the possibilities that third parties will get to innovate on top of the platform. It is expected that such choices relate to architectural design and/or to the business model of FI-WARE. This deliverable will document the key choices made and will analyse their effect on future third party innovation. As such, it will provide a justification of these choices against the ultimate objective of enabling third party innovation. It is expected

that the deliverable will address topics such as architecture and innovation; neutrality issues; openness; lock-in; data portability; interoperability; patents; standards; specifications; access rights; open source and licensing; and so on.

1 Introduction (TID)

Here, we will elaborate on:

1. the roles that different stakeholders may play with respect to the FI-WARE GEs.
2. Then, we will elaborate on what "third party innovation" means in FI-WARE, i.e., the ability to innovate by a party that intends to play some of the defined roles. Then, the idea is to explain that some of the design decisions in FI-WARE have been made with the intent of achieving the highest freedom and capabilities to innovate by parties playing any of the defined roles.
3. Last but not least, we will elaborate on what are the instruments we will put in place to test innovation capabilities (FI-WARE Testbed, FI-WARE Open Innovation Lab, …)

The FI-WARE project intends to support the development of innovation-driven value chains around Applications and Services. Hereby driving innovation is mainly realized by the efficient and seamless composition of services that bring about new value. 3rd party innovators can take various roles in this innovation process supported by FI-WARE technologies and specifications. The following table describes the different roles envisioned for the FI-WARE-enabled Value Chains.

Role	Description
Application Developer	Future Internet Application Developers are encouraged to develop smart applications targeting either mass markets or individual enterprises and organizations (which in turn may have a limited number of users or, again, the mass market as end users). These Applications should offer flexible means for deployment, provisioning and runtime operation "on the Cloud". Future Internet Applications are intended to be meaningful and stand-alone, implementing a number of functions they export "as a Service" to End Users through a number of User Interfaces but also to third Applications in some cases, through well-defined Service APIs (Application Programming Interfaces). They typically rely on functions provided by a number of Enablers, which can be specific to the Application Domain or Generic (meaning they are general purpose).
Enabler Developer	Enabler Developers are encouraged to develop software components or more complex systems that can be instantiated to provide functions easing the development, provisioning and/or

	runtime operation of Future Internet Applications. Enablers are intended to be universal, that is, they refer to multiple Applications that rely on the functions they implement. Those functions are exported “as a Service” to third Applications through well-defined Service APIs and also to End Users in some cases, through a number of User Interfaces. Enabler Developer may integrate several lower-level Enablers to realize new and more powerful Enablers.. Note that Applications and Enablers resemble each other in their architecture since both implement functions that they export as services. The central differentiator between them is the primary users to be addressed (End Users in the case of Applications, other Applications in the case of Enablers). Note, however, that some products may qualify equally as Applications or Enablers. Enablers can run as stand-alone components and offer services to multiple Applications. Alternatively, they can be packaged and deployed together with the particular Applications they serve. We may distinguish between developers of: ● **Generic Enablers (GEs)**, which are intended to be general purpose ● **Domain-specific Common Enablers**, which are intended to cover needs of applications linked to a concrete Application Domain (e.g., eHealth) FI-WARE Generic Enablers refer to Generic Enablers whose Specifications have been defined or adopted in the FI-WARE project.
(Application / Enabler) Service Provider	Service Providers are in charge of deploying, provisioning and operating either Applications or Enablers. Stakeholders playing the Application/Enabler Developer role may also play this role. However, this is not always the case (e.g., a Public Administrator may be playing the Service Provider role with respect to applications developed by third parties and that the city offers to its citizens)
(Application / Enabler) Service Hosting Provider	Service Hosting Providers provide and operate the hosting infrastructure on top of which Applications or Enablers are deployed. They entwine themselves with the Service Providers to reduce the costs for service provision. Service Hosting Providers may provide Cloud Services for hosting Applications and Enablers. Note that, in that case, they can indeed be considered a concrete case of Enabler Service

	Provider (here, the Enabler is the Cloud providing hosting services) so they may be also referred as “Cloud Hosting Provider”. In many cases, an entity playing the Enabler Service Provider role also hosts the Enablers it provides, therefore also playing the Enabler Service Hosting Provider role. However, note that this is not strictly required (one may think about a Enabler Service Provider that provides a number of enablers, all of them being deployed on Amazon hosting services).
(Application / Enabler) Service Aggregators	Service Aggregators select Services from a broad variety of Service Providers and compose them to build new service offerings that address the specific requirements of niche End Users.
(Application / Enabler) Service Brokers	Service Brokers bring together a multitude of Services from diverse Providers and publish them on marketplaces where end users can compare them, matching their requirements with capabilities of published Services. they should exploit economies of scale and protect investments in the long run. Finally, the ability to combine applications from different sources necessitates innovative revenue sharing models across partners and potentially also customers (e.g. crowd-sourcing) which have to be adapted dynamically as market conditions change.
(Application / Enabler) Service Channel Makers	Often specialized services must be adapted to the specific environment in which they are used. Without such adaptation it can be difficult to make profitable use of some service, e.g., in hospitals. Service Channel Makers are in charge of such adaptation.
Application End Users	We may distinguish between two different types of End Users: End Consumers (individuals in the mass market) and Enterprises and other Organizations: ● **End consumers** want to gain access and easily consume applications that can effectively assist them in daily life situations (e.g., purchasing goods, managing bank accounts, or planning travels). Some of the underlying problems involved are the management of the ever-growing data and information (e.g. from their sensor-enabled environments) and the seamless access anywhere, anytime and from any device. They also ask

	for improved means for communication and collaboration within their social networks, families, neighbourhoods in real-time and while being mobile, meeting security and privacy requirements. Overall, these capabilities would transform communities, homes and cities into safer and better places to live and leverage the Internet as an additional societal utility. The Future Internet that aims at more fine-grained and individualized services has to respond to their requirements as a central issue of innovation. Even though these end consumers do not bring about innovation they nevertheless indirectly become the drivers of innovation because they ultimately decide about the success or failure of innovation. ● **Enterprises and other Organizations** on the other hand, wish to get closer to their customers in order to deliver an even more compelling user experience and better service. For this reason, they would like to exploit contextual user data which may lead to a more personalized interaction experience and service offering, and would like to realize a stronger participation of users in all phases of product and service lifecycles, thereby bringing the lessons of the Web 2.0 phenomena into the services space. In order to develop and operate their services, new methods, technologies and tools are needed to speed up the time to market, to establish value added services which may be better configured in partnership with others and to simplify access to relevant resources and capabilities, e.g., from the Internet of Things. Additional requirements on business services include reduced complexity of ICT provisioning, scaling, global availability and meeting security requirements from customers and legal authorities. An appropriate Future Internet platform would greatly contribute to meeting these demands from business customers. Enterprises and organizations possess the capacity to investigate consumer behaviour and drive development of innovative services and applications based on these investigations. Despite we are referring here to Enterprise and Organizations playing the End User role, they can partake in the Future Internet in different ways so that we will individually consider these roles in the following. They can appear as End Users as well as in

	any of the previously described roles. Relations between enterprises and organizations are often governed by long-term contract, which provides another area of possible innovation.

Summing up, the described roles represent various pathways to innovation that are open for 3rd parties. In these roles they find plenty of ways to design new service offerings, whether by increasing the efficiency of services or by specialisation to particular needs. These roles may also guide potential 3rd party providers to find way towards new service offerings, introducing the business patterns for a future service economy.

The concepts of FI-WARE Instance, FI-WARE Testbed and FI-WARE Open Innovation Lab would come next. Emphasis will be given to description of their relationship to roles described above

Products implementing FI-WARE GEs can be picked and plugged together with products implementing complementary enablers in order to build FI-WARE Instances, operated by so called FI-WARE Instance Providers. FI-WARE Instance Providers can play one or more of the roles described above:

- They can for sure be categorized as *Enabler Service Providers* with respect to the Enablers (both FI-WARE GEs and complementary enablers) they provide and operate.
- They may also host the Enablers that conform the particular FI-WARE Instance they operate, in which case they also play the role of *Enabler Service Hosting Provider*. Note that this, as already mentioned in the description of roles, is not strictly required (one may think about a FI-WARE Instance Provider that provides a number of FI-WARE GEs, all of them conforming a particular FI-WARE Instance, but deployed on some third-party hosting services). When they provide Cloud Hosting capabilities and they are based on FI-WARE Cloud Hosting GEs, they can be also referred as FI-WARE Cloud Hosting Providers.
- They can play the role of Service Aggregators. When playing this role based on FI-WARE GEs defined to support that role, they are referred as FI-WARE Service Aggregators.
- They can play the role of Service Brokers. When playing this role based on FI-WARE GEs defined to support that role, they are referred as FI-WARE Service Brokers.

The FI-WARE project will generate a FI-WARE Instance, hereunder referred to as **FI-WARE Testbed**, which will allow partner Use Case projects (including a number of Use Case projects that are part of the European FI PPP initiative) to run and test Future Internet Applications based on FI-WARE Generic Enablers. Note that the FI-WARE Testbed is different from FI-WARE Instances linked to execution of trials or commercial services (exploitation phase). Those are typically referred to as “FI-WARE Instances in production”.

The FI-WARE Testbed is aimed to be complete, in the sense that it will comprise reference implementations of all Generic Enablers defined in the FI-WARE Architecture. This means that it

will help to experiment all the above mentioned roles. The FI-WARE Testbed will not necessarily be centralised, but will be under central control and be accessible from a dedicated website. The FI-WARE partners will provide support to Use Case projects for the deployment of applications (e.g., the conceptual prototypes) on top of the FI-WARE testbed. Tests run by the partner Use Case projects, coordinated with tests defined by the FI-WARE project, will help to validate Generic Enabler Open Specifications, the reference implementations of FI-WARE Generic Enablers developed within the FI-WARE project, as well as the conceptual prototypes developed by Use Case projects.

In order to pave the way for a successful exploitation and sustainability, the FI-WARE project will work on setting up a FI-WARE Open Innovation Lab around the FI-WARE testbed after the second release of FI-WARE. This Open Innovation Lab will support community involvement beyond the initial partner Use Case projects, offering a space where future innovations on top of the generic enablers provided by FI-WARE can be nurtured. Availability of the FI-WARE testbed per se does not guarantee innovation as such, therefore the FI-WARE Open Innovation Lab will comprise all what is needed to stimulate awareness among target application providers and users so that they feel attracted to participate and build a community. It will also bring tools helping members of the community to share their experiences, needs, etc.

(note from Telefonica: the following text in blue, originally provided by SAP, should be enhanced or dropped because the examples do not cover enough number of roles)

For example, we find possible areas of innovation through all FI-PPP Use Case projects, only mentioning two of them

1. *Increase of transparency and improvement of process tracking*: For instance, in the SmartAgriFood project an obvious problem is to track the origin of food through the processing and transport chain, which is important if retailers and end consumers want to track their products back to the producer, a demand that we can more and more observe. Each party in the chain possesses knowledge of their contributions so that it only requires a *service aggregator* who could combines the required services as possible innovation.
2. *Increase* of *flexibility in business transactions*: In the FINEST project it became clear that it is important for logistics providers to immediately react to interrupts in the transport chain. Here it is important to quickly find alternative ways of transportation. A *service broker* could provide a variety of transport services from which a logistics company could choose suitable services to successfully accomplish the transfer order. This appears to also be an innovation in logistics.

For both cases the FI-WARE infrastructure provides the basis for the innovation. We find that such innovation follows certain patterns that we can often relate to roles that certain parties take in the scenarios. These roles rather refer to *innovative business models* than to *new technologies*. To understand these roles is the core to understand the business opportunities for 3rd party innovators on the Future Internet.

3 Key choices at chapter level enabling third party innovation (each Chapter team)

Expected contribution per chapter: minimum two pages, desirable 3 pages
Aspects to cover per chapter:
Each chapter should elaborate on what decisions they took during the design and architecture process, that will foster innovation by any third party (i.e., non-FI-WARE partners) that is going to play any of the roles described in the introduction. These decisions may have been influenced by requirements coming from the UC or Business Units of the FI-WARE partners (that, in turn, are supposed to be fed by their customers). You should cover as well an analysis on what limits and what allows possibilities of third parties to innovate on top of the Generic Enablers in the chapter..
Questions to bear in mind:

> Architecture and innovation; neutrality issues; openness; lock-in; data portability; interoperability; patents; standards; specifications; access rights; open source and licensing; business models, Specific requirements for your contributions in the open innovation lab/Testbed.

Please try to reference existing wiki pages (e.g. architecture definitions, legal notices, open specifications, usage information…)
Guiding questions:
What roles, played by 3rd parties, are addressed by your contributions? What can be used or implemented by third parties in your chapter?
What is substantial to the design of the Architecture in your chapter that will enable those 3rd parties to innovate ?
What parts of your contributions are IP-protected?
Do you provide already information on what business model you will provide your functionality with?
What can hinder 3rd party innovation? Will this be overcome during the course of the project?

3.1 Applications & Services

The Application and Services Ecosystems delivery framework in FI-WARE will provide the infrastructure that enables the provision, composition, and trading of services in the Future Internet. Both Services and Applications encompass technical, business and operational perspectives. In order to make services tradable, all of the relevant aspects like the description,

the provided interfaces, terms and conditions, pricing, SLA conditions, should be described in an open and very generic format. For this purpose, we use Linked-open-Data USDL, that is flexible enough to fulfill these needs.
The Apps Chapter provides GEs for **Repositories**, **Registries**, **Marketplaces**, **Stores**, **Business Models and Elements** (including revenue sharing and SLA management) as well as **Service Mediation** and **Service Composition**.
The combination of the Generic Enablers of the Application and Services Chapter could be seen as a blueprint of a technology framework that allows completely new, flexible and innovative service ecosystems since it allows composing services on the Marketplace to completely new offerings to which various 3rd parties can contribute.

This open and flexible framework provided by the Application and Services Ecosystems delivery framework supports 3rd Party innovation at 5 different levels:

- The FI-WARE **composition and business framework** facilitates the introduction of simple service compositions by 3rd Party providers without undue technical implementation efforts.
- Implicitly the FI-WARE Apps Chapter provides an **architecture blueprint** that describes how the fundamental Enablers such as Marketplace, Repository, and Composition Environment interact with each other 3rd Party providers can build on this core. They can introduce their own Services and Applications as extension of this core within the platform framework of GEs provided by FI-WARE.
- In collaboration with the FI-PPP Use Case projects, the chapter does not only drive technological innovation but also fosters innovative **business approaches** including novel business models, based on the combination of existing service business models. The business framework supports such combinations of business models respecting the existing revenue sharing and SLA management.
- Finally, the current FI-WARE project partners already form the core of **future service ecosystems** that are open for new 3rd Party players to participate and to further develop this ecosystem with respect to new domains and technologies.

The Unified Service Description Language (USDL) enables the general communication throughout the different Generic Enablers provided by FI-WARE. We expect USDL to realise universal exchangeability of information between services. Based on the use of USDL in FI-WARE we see this as the core of a future community of practice that exploits the full capacity of this language in various domains. In particular the Linked Open Data approach regarding USDL prevents this approach to become a bottleneck and rather invites multiple domains to include their own specific extensions. This approach enables third parties to bring in their own specifics such as their domain knowledge to extend USDL with respect to their concrete needs while retaining the principle interoperability of Services and Applications.

Third Party Roles

The following compilation of examples illustrates how FI-WARE design and architectural decisions in the Apps & Services chapters may lead to innovation introduced by parties, which

play the roles described in the previous section:

- *Application Service Providers:* They provide new domain services that aim at the increase of efficiency of service provision. For example, this increase is realised by a simpler coordination and orchestration of services and is supported by the FI-WARE platform, which provides development tools and the core functionalities like hosting and security services. Specific aspects of the FI-WARE platform that support these service providers are:

    - The Unified Service Description Language (USDL) is a means that allows for the specific description of such services in a general form that makes them searchable on the marketplace or service platform.
    - USDL relies on a Linked-open-Data approach that helps to avoid time-consuming and error-prone data translations.
    - This Linked-open-Data approach allows for the description of all possible domain specific facets of data, this increases the efficiency of service provisioning and the service processing.
    - USDL is designed to be supplemented by aspect and domain specific vocabularies. Application providers can extend the USDL descriptions according to the needs of innovative applications relying on adequate descriptions of arbitrary aspects on top of USDL. For instance, a logistics provider could introduce new USDL descriptions that support the handling of cold chains.
    - As part of the service platform, the FI-WARE Marketplace is available to all 3rd Parties so that it will foster the competition between a large amount of service offerings. Thus, it works in favour of the most suitable and efficient available service, which can be chosen to build better and more innovative applications easily
    - The Generic Enablers developed in this chapter use Web APIs that allow Application Developers to build innovative solutions for their target environments being part of the business framework.
        - API mediation makes it possible for Application Service Providers to query S3C by using the Web prefered protocol: HTTP. Moreover API mediation combines well with the ECE composition engine and Mashup Factory and Mediator.
    - Instruments and tools provided by the Apps Chapter support the development of new Business Elements and Business Models and help application service providers to cover the business aspects of new services more efficiently . An example for this is the GE for revenue sharing.
- *Service Broker:* FI-WARE supports service brokers in innovation through:
    - USDL yields a formalised service description and thus enables a more efficient comparison of services including relevant business and operational aspects leading to increased competition. Hereby innovators with novel and more attractive service offerings are favoured even though they might not yet be well

established.
    - the web-scale distribution of service repositories, which gives higher flexibility for different kinds of instances of the business framework: This, in return, will help realize a wide range of innovative applications and business cases.
    - The marketplace that provides the general overview of existing services and means of expressing feedback from end consumers that drive competition, which can be leveraged for innovative businesses.

- *Service Hoster:* Specific aspects regarding Service Hosters:
    - By including hosting services, FI-WARE, as commercial service platform, provides opportunities for innovative hosters since the uniform description simplifies the exchange of GE-based hosting services, for example, changes to more innovative hosters. In this way the FI-WARE infrastructure avoids “lock-in” effects.
    - The same features support the provision of specialized hosting services and increase the competition between hosters, and fosters innovation.

- *Service Aggregators:*
    - FI-WARE’s GE concept and USDL simplify the aggregation of services since they provide a generic umbrella including a description of various aspects for aggregation and composition. FI-WARE provides several composition environments for this purpose addressing different application scenarios. This shows the potential for the integration of new upcoming innovative composition and aggregation technologies. These environments diminish the effort of service aggregators considerably.
    - The marketplace makes it easier to find specific services, which are required for composition and specialised offerings, enabled by USDL.
    - The support of generic Business Elements & Business Models including RSS and SLA help supplement new service aggregations with a suitable business infrastructure.The Open Specifications hereby specify how to interact with the various GEs..
    - The basic point of Composition Environments is that they stimulate innovation through quick service creation. A list of available services can trigger new combinations or additional steps leading to service improvements.
    - For tailor-made services this approach allows new business opportunities with a clear long-tail perspective.

- *Channel Makers:* FI-WARE Apps supports innovation in this respect in various ways:
    - Multi-channel and multi-device features help to offer services through social and content platforms as well as operating systems and handsets of devices.
    - The I2ND chapter provides a set of open and standardised interfaces to network and devices that supports these features.

- *Consumers:* In a long-term perspective the FI-WARE infrastructure can even support consumers in innovation, enabling them to adapt service to their personal needs.
    - The adaptability of Composition Editors to personal needs further encourages the personalised use of services offered on the FI-WARE platform.
    - Composition environments target to a simplification of processes and communication, customization and bundling of services and tasks and thus change consumers organisational habits. Services are essential for organisational tasks in private and business life. A growing demand for individualized ICT services that are tailored to personal needs are an important innovation to manage individual resources and communications.
    - Composition Environments enable Enterprises and other Organizations (especially SME) to explore new organizational processes by means of web, media and telco services. Allowing non-programmers to compose and evaluate situational technical solutions based on several enabling services drive flexible organizational processes for more efficiency, faster time-to-market and even new business processes with customers.

Generally we can say that the Apps chapter diminishes the following **barriers** for 3rd party innovation:

- lack of guaranteed SLAs for compositions of service
- complexity in handling service compositions
- complexity during the integration of specialized services
- proprietary environments for services that are to be composed
- unacceptable business conditions of services providers that exclude composition..

[kay, Thu, 15:01] I have some bad news for us: Juanjo don't really feel happy about our chapter. I thought, it is ok in this way, but it wasn't. There were some hints but no big complaints (until yesterday) about the structure. But I hope we can fix this issue ASAP.

- Our section is in fact too long for each GE in I2ND. This document should not be about, what each GE is able to provide. We need to shrink all these texts to 1-2 paragraphs for each GE in I2ND and set this to the "introduction" of I2ND. That means, we need to restructure our whole chapter.
- The MAIN FOCUS lies on the possibility of the given roles of section 1 in this document. A structuring reference is the apps & services chapter (please scroll up). This means, that we have one overall introduction and then the bullet point list of the different roles and their opportunities with our I2ND.
- A good approach was given by CE & NetIC in the text. I will restructure and hope for your additional contributions.
- Despite the fact that we are contributing really different GEs in I2ND, we should try to put all of our GE functionalities together to each specific role with the given hint, which GE offers which special opportunity. I am also aware that some roles can be mixed up,

honestly I, have up to now, just two approaches how to solve this. One idea could be to put this into one role and just name the other roles. One other is to summarize them in one bullet point - *this is currently the structure.*

3.5 I2ND ---- NEW

The focus of I2ND GEs inside the FI-WARE platform is to exploit functionalities from the connected devices, the network and the end termination towards the applications and services providers. Therefore it is possible to create an additional set of 3rd party applications incorporating the special functionalities of such Generic Enablers. Via special APIs, the 3rd party applications and/or services are able to retrieve network status information, set up application based QoS parameters, exploit functionalities of an end-terminal or from the network side and put processing power as well as storage capacity towards the network-cloud-edge. Four GEs are defined in the I2ND architecture:

- *CDI (Connected Device Interface):* It is a GE which provides a set of runtime APIs available to application developers. The CDI GE provides two main types of API, on-device API which is available to Developers who create applications which execute on the device (like an installed application), and off-device API which is available to Developers who create cloud hosted (server hosted) applications, which need to interact with connected devices, similar to a network service API. The APIs provide access to device specific features, such as local hardware, and integrate with the network through which the devices are connected to provide advanced QoE (Quality of Experience) by monitoring user interactions and using that data to drive changes in any network dependant QoS (Quality of Service) configuration.
- *CE (Cloud Edge)*: This GE can be seen as a “super gateway”. It is located at the edge of the home, in between the WAN (xDSL, cable ...) and the LAN (Ethernet, WiFi, home-automation networks etc., ...). It has the capacity to locally execute downloadable applications in virtual machines (or in containers), thus giving the 3rd party *application developers* an easy and wide access to all the features of a machine that is located at the user’s premises (a Linux computer in fact).
- *NetIC (Network Information and Control):* The API of this GE is intended to facilitate access to network information and network control features for a wide range of 3rd parties. This will facilitate the operation of a new class of services which require a tailored quality of service for optimum service delivery; on the other hand the network operators retain all network control needed to maintain safe operation of their networks.
- *S3C (Service, Capability, Connectivity and Control):* The API of S3C is intended to facilitate the access to information and control of the services offered by a Future Internet network as well as to the means for managing the connectivity parameters. The new API enables the application platforms and the services to dynamically adapt their delivery parameters through the specific network while the control of the network is maintained in the operator exposing the API. The parameters that can be adapted and exposed through the specific API, address the telecom application adaptation, mediation and exposure, the device connectivity and remote management and the exposure of

connectivity parameters such as location, resource reservations, subscription profiles and charging.

Third Party Roles

The following compilation illustrates how innovation can be introduced by parties which play the roles described in the previous sections, through the exploitation of the FI-WARE design and architectural solutions provided by the I2ND chapter GEs.

- *Application Developers:* The I2ND GEs support 3rd party Application Developers by lowering the cost of application development, aiding innovation, and making it easier for developers to provide applications with excellent user experiences across multiple device types, functioning across multiple network infrastructures, and avoiding 'link bottlenecks' when operating at the boundary of a *cloud universe*. Application Developers can leverage more specifically:
    - The core functionalities provided by the CDI GE such as:
        - Application platform ubiquity: the cross platform development problem is addressed, and it aids developers by providing an single easily understood and familiar development environment based on web standards - first of all HTML and JavaScript - which can be supported by most of the key connected device categories;
        - Application connectivity and device management: new applications need to understand the capabilities of - and reach out the - end user's devices, this will lower the cost for 3rd party Application Developers by providing a framework which allows cloud hosted elements to query connected devices for their capability, and to reach out and connect to these devices via a remote device management functional block;
        - Advanced support for connected application development: API accessing key technologies like QoE and QoS will provide 3rd party Application Developers the means to instrument, measure and control the experiences their distributed applications deliver to end users.
    - The entry point of the home system computing and storage capabilities, which can be directly used and controlled by the cloud actors via the Cloud Edge GE functionality. Application Developers, as well as Cloud Operators, may download executable codes on the Cloud Proxy which will interact with their cloud-based applications.
    - The NetIC API enablement for Network Operators allows Application/Enabler Developers for making their applications/enablers react flexibly on network status information and changes. Furthermore, the new API will also enable Application/Enabler Developers to customize the underlying network to the needs of their applications/enablers (within the limits given by e.g. network operators' security constraints). As the API is intended to provide an abstract view on the network, the Application/Enabler Developers can focus on the required network features and won't need to customize their applications/enablers to specific networks.

    - The S3C layer, between access & core network and service platforms, which allows the Application Developers to have an interaction between the service operators, the network behavior and the end user's system. It helps the Developer to request network parameters for QoS, get information of the current core network and enhance the quality of experience of a specific service via obtaining additional service specific information from the network side. S3C provides different kind and levels of APIs that could be tailored to answer different needs. For example the API mediation aims at managing HTTP based APIs, the OpenEPC is available through a Diameter interface and some S3C APIs comply with the OneAPI specification.

- *Enabler Developers*: The FI-WARE platform provides opportunities for Enabler Developers (4th Party Developers) to participate in the FI-WARE value chain. More specifically, JavaScript Plugin Library Developers can create new plugins to extend the functionality originally provided by the CDI GE, and new services can be built upon the remove device management features, providing additional support for new methods of device management.

- *Service Providers, Service Hosting Providers, Service Aggregators:*
    - Having these enhanced applications/enablers at hand, provision of the new NetIC API by network operators will enable Application/Enabler Service Providers, Application/Enabler Service Aggregators up to Service Hosting Providers to flexibly exploit the features of the underlying networks for the sake of an optimal service provision to the Application End Users. The new API provided by NetIC GE might also *create new business in Future Internet*: Local network owners can cooperate with more global operating connectivity providers or Virtual Network Providers/Operators. In this case, the intended new API will allow control of different underlying local networks on an abstract level, thus reducing significantly the integration effort for the overlaying Network Provider. This might enable also local Network Operators to get their share of the network business as they can offer a common API to an integrating Virtual Network Operator.
    - Cloud Operators, as well as Application Developers, will be able to use and control the capabilities to download code from cloud applications onto the Cloud Proxy, which shall be executed in isolated Linux containers. Such capabilities are offered by Cloud Edge Service Management API and will contribute to enhance security aspects.
    - The S3C layer enables the opportunity to create new service functionality with the inclusion of the current network situation, event notifications and additional core network functionality. The future services are able to use the wide bandwidth of telecommunication technologies towards an end user’s terminal, including established technologies and services (e.g. SMS), seamless mobility via common and new radio and fixed access technologies, service enhancement via location information and network based identity management via SIP/IMS authentication.

    - Service Providers will be able to provide networks functionalities (such as the mentioned location based services and network identity management) *as a feature* for future services; these additional network functionalities can help building an innovative service, but they can hardly be conceived as stand-alone services for market places, hence the possibility to be exploited by stakeholders as Service Aggregators.

- *Service Brokers:*
    - Via the OTT API, which connects to the OpenEPC Core Network Platform, billing end users will be made available by the S3C GE.
    - Application Store providers can offer End Users (and developers) a marketplace for showcasing and downloading CDI based applications.
    - Further Network functionalities such as Location and Identity Management can be exploited by Service Brokers by means of the APIs provided by S3C GE.

- *End Users:* Utilising services which integrate functionalities of the I2ND chapter will be profitable for end users (End Consumers and Enterprises) too:
    - With comprehensive network functionalities for service providers, the end users can get a better QoE/QoS from the application.
    - With the focus of enterprise customers, I2ND's interfaces to core network can manage the network architecture with new and standard radio & fixed access technologies, seamless/uninterrupted connectivity over different access networks and provide QoS levels towards the employees over the connected access network, although it will not provide (physical) infrastructure to the customer.
    - The application downloading capability enabled by the Cloud Edge GE allows a better experience of the offered services to the end user, as some "real time" features (such as the upload of content for example) can be intelligently "cached" by the local application.

In summary, the I2ND GEs address the needs of several 3rd party innovators targeted by the FI-WARE platform, Application Developers and Service Providers being the prevailing ones. New or enhanced services can benefit of the additional functionality from the network side to improve their existing services or even create completely new ones. On the Application Developer side, it is possible connecting via the CDI GE to a homogeneous Interface and obtain and set settings of the end user's device while interacting to a service which is simultaneously able to obtain the network status, request flow/application based QoS parameters, receive additional information from the network side and will therefore be capable to adapt the flow settings and content specifications of its application.

3.5 OLD I2ND

The focus of I2ND chapter is to exploit functionalities from the connected devices, the network and the end termination towards the north - the applications and services providers. Therefore it is possible to create an additional set of 3rd party applications with special functionalities. Via special APIs, the 3rd party applications or services are able to retrieve network status information, set up application based QoS parameters, exploit functionalities of an end-terminal or from the network side and put processing power as well as storage capacity towards the network-cloud-edge.

Four GEs are defined in the I2ND architecture:CDI, CE, NetIC, and S3C. The CDI GE deals with end-devices, whereas the CE GE covers cloud edge nodes. Both GEs will offer a rich suite of functionalities which can be offered to third parties such as service and content providers as well as third party application and software developers. The NetIC GE interfaces directly to the network infrastructure, the S3C GE deals with more abstract features on top of the infrastructure. Both GEs will have only restricted interfaces towards service and content providers, and third party developers. This means in one direction, providers and developers will have a restricted control, which is managed and regulated by the owner of the infrastructure - network service provider. In the other direction, the providers and developers will have the possibility to receive parameters and states of the infrastructure. A special third party interface will be the planned lawful interception access control.

CDI

Connected Device Interface, or CDI is a GE which provides a set of runtime APIs available to application developers. The CDI GE provides two main types of API:

- An on-device API which is available to developers who create applications which execute on the device, like an installed application.
- An off-device API which is available to developers who create cloud hosted (server hosted) applications which need to interact with connected devices, similar to a network service API.

The APIs provide access to device specific features, such as local hardware (cameras, microphone etc), and integrate with the network through which the devices are connected to provide advanced QoE (Quality of Experience) by monitoring user interactions and using that data to drive changes in any network dependant QoS (Quality of Service) configuration.

These APIs are provided across a wide range of devices, which are collectively referred to as “Connected Devices”. The main device types that the CDI GE will attempt to address include:

- Traditional computing devices: PCs / Laptops / Tablet-PC hybrids (Windows 8 Tablets)
- Mobile Phones
- Tablets

- IVI (In Vehicle Infotainment)
- Set-top boxes
- Internet TVs

The CDI GE supports 3rd party application developers by lowering the cost of application development, aiding innovation, and making it easier for developers to provide applications with excellent user experiences across multiple device types, functioning across multiple network infrastructures.

To achieve this the CDI GE provides three key elements:

- Application platform ubiquity
  The range of devices supported by the CDI GE have each traditionally been addressed by their own development platforms, tools, and languages; examples include Java on Android, C++ on Symbian / Desktop Windows, C# on Windows Phone / Desktop Windows, and Objective C on iOS devices. This technology fragmentation has increased the cost of application development, and effectively reduced the range of mobile platforms which can be supported by an application developer. If a customer requires an application to be developed for a range of devices they must pay for an individual application to be created for each platform.

  The CDI GE addresses this issue and helps application developers by providing a single application platform across all devices. Each of the main device categories have evolved and adopted specific technologies making it hard for a developer to create a single application which can function across each of them.

  Even though the devices have evolved their own specific technologies, each device has always maintained a core set of equivalent functionality - a web browser, a necessity for internet access. This common technology is a natural base for a cross platform solution and the addition of HTML has made it increasingly more difficult for users to distinguish between an application and a web site. A number of “web runtimes” exist and each is usually targeted a specific device form factors, e.g. Adobe’s PhoneGap is for phones, and Microsoft’s Metro works on tablets and PCs.

  While each of the available cross platform web technologies share a common core web technology, their specific interfaces and APIs are sufficiently different that it is impossible to create a single application which executes on each web runtime.

  To mitigate the current cost of supporting multiple code bases developers typically expend considerable creative efforts in creating frameworks and coding patterns that can be easily ported between platforms. This is time and effort which developers end up investing in non-functional code (logic in an application which does not deliver functionality to the end user).

CDI addresses the cross platform development problem and aids developers by providing an single easily understood and familiar development environment based on web standards which can be supported by all of the key device categories.

CDI removes the overhead of porting an application between technologies and platforms, maintaining multiple code bases, and prevents the developer from needing to invest considerable effort in non-functional code. This aids innovation by allowing the developer to really focus their creative efforts on the functionality which will make their application stand out in the marketplace.

- Application connectivity and device management ubiquity

  In the past, applications were simply regarded as stand-alone software, installed directly on the device upon which they would primarily be used. The internet has changed this and now applications often consist of a number of networked elements, from user visible interfaces installed on a consumer's / end user's device through to elements running inside the cloud. Running together each of the elements form a modern application.
  Elements of such applications running in the cloud need two key elements of functionality to operate successfully:

  - *They need to understand the capabilities of the end user's device.*

    This is often required so that the cloud elements can provide data in formats which the connected device can understand, for instance provide video in QuickTime format for iOS, and MP4 for Android.

  - *They need to be able to reach out and contact the end user's device.*

    This is often required when the cloud element needs to notify the user or the device of new data or services. This is often referred to as a push service.

  Currently there is no solution which addresses both of these across multiple platforms. The different platform developers have traditionally developed their own solutions, making integration of cloud services with a range of devices difficult and costly. The CDI GE addresses this and lowers the cost for 3rd party application developers by providing a framework which allows cloud hosted elements to query connected devices for their capability, and to reach out and connect to these devices via a remote device management functional block.

- Advanced support for connected application development

  As modern applications are distributed across various compute devices the network and the communication between the compute devices can and does have a direct impact on the experience the end user observers. To address this the CDI component includes two key technologies:

- *QoE (Quality of Experience)*

  The QoE APIs present in the CDI GE allow a 3rd party application developer to instrument their application which allows the QoE system monitoring the behaviour of the user. Monitoring the behaviour of the user allows the QoE system to automatically detect when a poor experience is being delivered. As a simply example, when a video is not streaming well the user may repeatedly press the play/pause button as the content is buffered. This scenario can be detected and the QoE engine can initiate steps for improving the experience. One of these steps could be to make a direct request to the network to increase the available bandwidth for the video stream. This type of request is made to the QoS System.

- *QoS (Quality of Service)*

  The QoS APIs present in the CDI GE allow 3rd party developers to directly request network resources for their application. This this achieved by communicating to the network via OpenEPC. In the past this level of control of the network has not been available to application developers, they were at the mercy of the network provider. When a media stream failed to play due to network issues the only advice the application developer could provide the user was to ask them to change network provider, or network media (switch from 3G to WiFi). With QoS the developer can directly ask the device to switch network media (if possible), or directly ask the network for additional resources.

These two technologies aid 3rd party application developers by giving them the means to instrument, measure and control the experiences their distributed applications deliver to end users.

Enabler Developers, otherwise referred to as 4th party developers, can leverage the architecture provided to add additional services and functionality which 3rd party application developers can leverage. There are two specific areas within the CDI GE which may be of interest to 4th party developers. These include:

- JavaScript Plugin Libraries

  4th party developers have an opportunity to provide JavaScript plugin libraries for 3rd party developers who create on device applications. A similar market currently exists in the traditional web world where 4th party developers offer plugins to the jQuery framework which enables new functionality to be offered to others.

- Advanced remote device services

  The remote device management elements within the CDI GE provide connectivity and device discovery, but additional services could be built on top of this framework by 4th party developers.

The CDI GE supports a number of new business opportunities, these include:

- *Application Stores*: This is the place where developers can download CDI based applications.

- *Application provisioning directly from web sites*: Where websites, rather than linking to a third party application store can directly distribute on device applications, directly from their existing online presence.

- *Cross skilling of existing workforce*: As the CDI GE is built upon web technologies it allows existing professionals from the web ecosystem to move into application development.

- *Device manufacturers / Platform creators*: The CDI provides a thorough specification which can be implemented by anyone. Device manufacturers can implement the specification directly on their devices, enabling them to support CDI applications directly. Platform creators, anyone creating software or hardware platforms can integrate CDI compatible implementations into their platform allowing their application to host CDI applications.

- *Non CDI application developers*: can expose functionality in CDI which enables application integration.

CloudEdge

The Cloud Edge can be seen as a “super gateway”. It is located at the edge of the home, in between the WAN (xDSL, cable ...) and the LAN (ethernet, wifi, home-automation networks etc ...). It has the capacity to locally execute downloadable applications in virtual machines (or in containers), thus giving the 3rd party **application developers** an easy and wide access to all the features of a machine that is located at the user’s premises (a Linux computer in fact).

- Avoid the “link bottleneck”: The idea of having a Cloud Proxy located at the Cloud Edge comes from the assessment that the link between the Cloud and the end consumer still offers a relatively low bandwidth in many scenarios. Actually, typical ADSL or cable bandwidths are in the range of a few Mbit/s while a private LAN like a home network offers at least 100 Mb/s and more and more commonly 1 GB/s. In order to enhance the service to the **end user** despite this relative low bandwidth, the idea of the Cloud Proxy consists in offering at the entrance point of the home system computing and storage capabilities which can be directly used and controlled by the cloud actors.
- Enhance the user’s experience and give more flexibility to the 3rd party application

developer: cloud operators, as well as **application developers** may download executable codes on the Cloud Proxy which will interact with their cloud-based applications. This will allow a better experience of the offered services to the **end user** because some “real time” features (such as the upload of content for example) can be intelligently “cached” by the local application. In order to improve security, the code downloaded from cloud applications onto the cloud proxy shall be executed in isolated linux containers. Cloud operators and **applications developers** will be able to use and control these capabilities offered by the cloud proxy via the Cloud Edge Service Management API for which a first set of method has already been proposed and implemented.

- Allow local minimal applications to provide continuity of service despite possible uplink problems: the Cloud Proxy - based applications can implement a minimal level of local intelligence that can provide the user with a minimal service even when the uplink (WAN) is down. For example, a heating or air conditioning regulation system can continue to be active even if the link with the cloud-based management application is broken. This give the 3rd party **application developers** new opportunities by giving them (and their **application users)** access to both feature-rich applications running in the cloud and safe local uninterruptible minimal applications.

The Cloud Proxy major innovation is to allow cloud-based applications to have their “proxies” running locally at the customer’s premises thus giving application developers access to the “best of two worlds”: the richness and flexibility of cloud-based applications while still keeping the advantages of having locally-executed applications providing continuity of service and giving the user a better experience by hiding possible bandwidth issues.

The Cloud Proxy is also giving new opportunities to the FI-WARE instance providers because some applications are simply not "implementable" without this local intelligent device and the fact it allows avoiding many potential technical problems (bandwidth bottleneck, continuity of service in the event of a datalink breakdown etc ...).

As an example, we can take the same example we used above: a heating or air conditioning system: a very smart cloud-based application can help the user to setup his/her airco system, this cloud application could offer very intuitive and powerful user interfaces, could offer advanced features such as a coupling with the energy provider website and/or with weather forecast etc …The main problem is “continuity of service”: for example, the data link between the home and internet (ADSL …) could be interrupted. If the cloud-based application directly manages the heaters or airco devices, this could let devices switched on or off until the link is activated again, thus creating much trouble for the user and thus rendering the whole application not practical. A locally executed application, hosted in the cloud proxy and interacting with the cloud-based application could solve this issue: if the data link falls down, the local intelligence continues to regulate the devices (although without offering much in term of user interface).

NetIC - Network Information and Control

The NetIC API is intended to facilitate access to network information and network control features for a wide range of 3rd parties. This will facilitate the operation of a new class of services which require a tailored quality of service for optimum service delivery; on the other hand the network operators retain all network control needed to maintain safe operation of their networks.

- **Application/Enabler Developers**

The new API shall easily enable network operators to allow Application/Enabler Developers making their applications/enablers react flexibly on network status information and changes. Furthermore, the new API will also enable Application/Enabler Developers to customize the underlying network to the needs of their applications/enablers (within the limits given by e.g. network operators' security constraints). As the API is intended to provide an abstract view on the network, the Application/Enabler Developers can focus on the required network features and won't need to customize their applications/enablers to specific networks.

- **Application/Enabler Service Providers**, **Application/Enabler Service Aggregators, Application/Enabler Service Aggregators**

Having these enhanced applications/enablers at hand, provision of the new API by network operators will enable Application/Enabler Service Providers, Application/Enabler Service Aggregators or even Application/Enabler Service Aggregators to flexibly exploit the features of the underlying networks for the sake of an optimal service provision to the Application End Users.

The new API provided by NetIC GE might also **create new business in Future Internet**: local network owners can cooperate with more global operating connectivity providers or virtual network providers/operators. In this case, the intended new API will allow control of different underlying local networks on an abstract level, thus reducing significantly the integration effort for the overlaying network provider. This might enable also local network operators to get their share of the network business as they can offer a common API to an integrating virtual network operator.

S3C - Service, Capability, Connectivity and Control

The S3C API is intended to facilitate the access to information and control of the services offered by a Future Internet network as well as to the means for managing the connectivity parameters. The new API enables the application platforms and the services to dynamically adapt their delivery parameters through the specific network while the control of the network is maintained in the operator exposing the API. The parameters that can be adapted and exposed through the specific API address the telecom application adaptation, mediation and exposure, the device connectivity and remote management and the exposure of connectivity parameters such as location, resource reservations, subscription profiles and charging.

The S3C API is based on the exposure of standard interfaces based on standard protocols and

information exchange such as GSMA OneAPI, 3GPP IMS SIP, 3GPP SUPL, 3GPP EPC Rx, Sh, Rf Diameter based interfaces and HTTP. The standard interfaces are available in complete form in the FI-WARE S3C API, thus not requiring any specific adaptation. These standards are publicly available and require no licenses in order to be used.

Four different types of interfaces are defined: (1) Telecom interface, (2) Web interface, (3) machine-to-machine control interface, and (4) an interface to support “Over the top” applications and services. 3rd party developers will have restricted access control (managed by a network service provider) and will be able to request network parameters. A special interface for legal interception can be considered.

On top of the standard interfaces specific features are developed as extensions of the S3C API such as the extension of the 3GPP EPC exposure towards charging and subscription profile information. These specific features follow the standard mechanisms in the new message exchanges and in the information transported. Through the FI-WARE specification documents, these extensions are planned to be publicly available and require no licenses in order to be used.

This Generic Enabler is able to exploit the functionality of the network towards service providers. Over the different APIs, the service providers and 3rd party developers get access to different kinds of network functionalities to enhance their own products and services, extend QoE towards the potential customers and users. In addition, an eco-system is built for new business especially for content and 3rd party providers. The open specification of the sub GEs APIs should provide sufficient information as a blueprint for exploiting the rich functionalities of the S3C Generic Enabler.