D2.1: FI-WARE Requirements Backlog

Project acronym: FI-WARE
Project full title: Future Internet Core Platform
Contract No.: 285248
Strategic Objective: FI.ICT-2011.1.7 Technology foundation: Future Internet Core Platform
Project Document Number: ICT-2011-FI-285248-WP2-D2.1a
Project Document Date: 7 Dec. 2011
Deliverable Type and Security: Public
Author: FI-WARE Consortium
Contributors: FI-WARE Consortium.

Abstract: This deliverable provides the backlog of the sprints and releases of FI-WARE.

Keyword list: FI-WARE, PPP, Future Internet Core Platform, Sprint, Agile, Epic, User Story
Following the Agile methodology, items in the FI-WARE backlog(s) correspond to "work to be done" in the project, including "functionality to be supported, therefore developed" in the FI-WARE Platform. Items associated to functionality to be developed are categorized as Themes, Epics, Features or User Stories, depending on the level of granularity of the description of the functionality. The rest of work to be done in the project is captured through additional entities in the backlog, referred to as Work Items.

Description of the functionality to be supported and developed in FI-WARE comes from different sources: the Use Case projects of the FI PPP Programme, the internal FI-WARE business and market analysis activities and global exploitation plan, the individual exploitation needs of the partners of the consortium and the input from third parties not involved in PPP projects. Mechanisms to capture input from Use Case projects and third parties have been put in place. See [1] for further details.

Activities in the FI-WARE project are planned through subsequent Sprints and Releases. Sprints are one month long while Releases comprise three consecutive Sprints, that is three months. User Stories or Work Items with the highest priority are selected at the beginning of each Sprint, then developed and tested until the Sprint is finished. The User Stories come as a result of refining Features that have been selected for the Release that the current Sprint is part of. The Release for which a given Feature is planned, as well as the Sprint and Release within a given User-Story or Work Item is being executed, are captured in the Release Id and Sprint Id fields in the backlog.

Contents of the FI-WARE backlog are dynamically changed, so that this deliverable simply constitutes a snapshot of the FI-WARE Backlog taken at a concrete point in time. You can follow the trackers through which FI-WARE backlog entries corresponding to each FI-WARE chapter are being managed at [3].

See [2] for further details on how the Agile methodology is being applied in FI-WARE.

[3] - Chapter backlogs:
  Cloud Hosting: https://forge.fi-ware.eu/tracker/?atid=204&group_id=14&func=browse
  Data/Context Management: https://forge.fi-ware.eu/tracker/?atid=187&group_id=9&func=browse
  IoT Services Enablement: https://forge.fi-ware.eu/tracker/?atid=193&group_id=11&func=browse
  Interface to Networks and Devices (I2ND): https://forge.fi-ware.eu/tracker/?atid=192&group_id=12&func=browse
  Security: https://forge.fi-ware.eu/tracker/?atid=195&group_id=19&func=browse
Provider information will also be described using a Linked Data vocabulary, which have to be defined.
As a cloud user/administrator I would like that the system implements a Single Sign On mechanism associated to
the resource allocation guarantees and QoS to the user, as well as to enforce them at runtime. Baseline assets
additional capabilities developed as part of this project -- such as resource reservation and capping for CPU,
évaluation of different workloads in the cloud, it is often required to decide where certain VM must be hosts. Due to the cost
mechanism is required. It will be based on existing capabilities of the hypervisors like KVM, enhanced by
order to be able to control the amount of resources allocated for different VMs and workloads over time, as well
This will identify key metrics of compute resources and provide the means to extract those metrics, ready for
using one or more streams of metrics from the distribution channel). Baseline assets such as Esper or Drools can
Using analysis engines, performance guarantees and KPIs can be calculated using rules and logic facilities. The
where a system supports this following a successful authentication, authenticate transparently with other
As a cloud admin, I would like that the PaaS GE could get information about monitoring in order to take decision
Due to the cost
As a developer before entering into the design of the User Portal I would need to get sure that the required
The administrator should be able to access the cloud resources and services by directly using command line
A user will request the details of a virtual machine against an API (OCCI).
As a cloud user I would like to move local infrastructures to remote Clouds (public or private) to cope with peaks
This will allow users manage and manipulate the three main core resources available from a provider, namely
compute, storage and network.
A user will present a request against an API (OCCI) to add an object and related metadata.
A user will present a request against an API (CDMI) to create a virtual machine. The creation of the virtual machine
As a user I would like to be able to define and perform lifecycle management.
As a Cloud User, I would like to move local infrastructures to remote Clouds (public or private) to cope with peaks
Where a system supports this following a successful authentication, authenticate transparently with other
As a cloud user/administrator I would like that the system implements a Single Sign On mechanism associated to
the resource allocation guarantees and QoS to the user, as well as to enforce them at runtime. Baseline assets
additional capabilities developed as part of this project -- such as resource reservation and capping for CPU,
évaluation of different workloads in the cloud, it is often required to decide where certain VM must be hosts. Due to the cost
mechanism is required. It will be based on existing capabilities of the hypervisors like KVM, enhanced by
order to be able to control the amount of resources allocated for different VMs and workloads over time, as well
This will identify key metrics of compute resources and provide the means to extract those metrics, ready for
using one or more streams of metrics from the distribution channel). Baseline assets such as Esper or Drools can
Using analysis engines, performance guarantees and KPIs can be calculated using rules and logic facilities. The
As a IaaS SM admin, I would like that the system deploy a Virtual Data Center.

As a PaaS Management admin, I would like that the IaaS SM recover a previously saved VM clone.

As a IaaS SM admin, I would like the way in which we get the metrics from the probes.

As a IaaS SM admin, I would like to convert from a VM to a template in order to move one VM from one region to another.

As a IaaS SM admin, I would like to delete a previous store VM clone when we do not need to use anymore.

As a Cloud user, I would know the way in which define the Elasticity Rules.

As a IaaS SM admin, I would like that the system could install a probe into a deployed VM.
|------------|-------------------|---------|------|---------|----------------|------|--------|-----------------|-----------------|--------------|-----------------|
A web developer/FIWARE Cloud GE developer receives a HTTP request with an additional header. The additional header contains a description of the device's capabilities. This allows the web developer to make a decision about how to navigate the device, or to provide content tailored to the device's capabilities. The results of these API calls can be used client-side directly. If a browser or a web application needs to access the device's capabilities, it can make a series of API calls to the device. These API calls can be made using a variety of technologies, such as RESTful HTTP APIs or SOAP. The results of these API calls can be used to render content accordingly. The north interface uses SOAP (order from service logic, service notifications to logic) and the south interface uses RESTful HTTP APIs (data driven service logic). The FIWARE developer calls a series of Javascript APIs to detect the device's capabilities and then renders content accordingly. The results of the API calls can be used client-side directly. If a browser or a web application needs to access the device's capabilities, it can make a series of API calls to the device. These API calls can be made using a variety of technologies, such as RESTful HTTP APIs or SOAP. The results of these API calls can be used to render content accordingly.

Using such industry-standard APIs has the following benefits for developers and publishers:

- **Reduced proprietary integrations to multiple operators**: Using such industry-standard APIs, service providers and publishers can reduce the number of proprietary integrations required to support multiple operators. This can simplify the development process and reduce the cost of development.
- **Mashup network APIs into desktop and mobile Web applications**: Developers can use these APIs to create mashups that integrate content from multiple sources, such as social media and web applications, into desktop and mobile web applications.
- **Network level security and AAA**: The GSMA OneAPI v2 defines a set of RESTful HTTP APIs for various telecom services and among them network level security and AAA. These APIs provide a secure way to authenticate and authorize users and to control access to network resources.
- **Provisioning, self-service and operator product and customer administration service**: The GSMA OneAPI v2 also defines a set of RESTful HTTP APIs for provisioning, self-service, and operator product and customer administration. These APIs allow operators to provision services and manage customer accounts in a flexible and scalable way.
- **Network Element Virtualizer (the ALU-I version of the NetIC) that represents the driver for network paths and network paths of the managed physical infrastructure. This functional block is defined inside the network space.** The north interface uses the Network Element Virtualizer (the ALU-I version of the NetIC) that represents the driver for network paths and network paths of the managed physical infrastructure. The south interface uses the Network Element Virtualizer (the ALU-I version of the NetIC) that represents the driver for network paths and network paths of the managed physical infrastructure. This functional block is defined inside the network space.

The FIWARE developer calls a series of Javascript APIs to detect the device's capabilities and then renders content accordingly. The results of these API calls can be used client-side directly. If a browser or a web application needs to access the device's capabilities, it can make a series of API calls to the device. These API calls can be made using a variety of technologies, such as RESTful HTTP APIs or SOAP. The results of these API calls can be used to render content accordingly. The north interface uses SOAP (order from service logic, service notifications to logic) and the south interface uses RESTful HTTP APIs (data driven service logic). The FIWARE developer calls a series of Javascript APIs to detect the device's capabilities and then renders content accordingly. The results of these API calls can be used client-side directly. If a browser or a web application needs to access the device's capabilities, it can make a series of API calls to the device. These API calls can be made using a variety of technologies, such as RESTful HTTP APIs or SOAP. The results of these API calls can be used to render content accordingly.

A method by which each physical resource of a network device can be activated, go beyond a simple User Agent String, to contain concrete device metrics. The north interface uses SOAP (order from service logic, service notifications to logic) and the south interface uses RESTful HTTP APIs (data driven service logic). The FIWARE developer calls a series of Javascript APIs to detect the device's capabilities and then renders content accordingly. The results of these API calls can be used client-side directly. If a browser or a web application needs to access the device's capabilities, it can make a series of API calls to the device. These API calls can be made using a variety of technologies, such as RESTful HTTP APIs or SOAP. The results of these API calls can be used to render content accordingly.

- **Reduced proprietary integrations to multiple operators**: Using such industry-standard APIs, service providers and publishers can reduce the number of proprietary integrations required to support multiple operators. This can simplify the development process and reduce the cost of development.
- **Mashup network APIs into desktop and mobile Web applications**: Developers can use these APIs to create mashups that integrate content from multiple sources, such as social media and web applications, into desktop and mobile web applications.
- **Network level security and AAA**: The GSMA OneAPI v2 defines a set of RESTful HTTP APIs for various telecom services and among them network level security and AAA. These APIs provide a secure way to authenticate and authorize users and to control access to network resources.
- **Provisioning, self-service and operator product and customer administration service**: The GSMA OneAPI v2 also defines a set of RESTful HTTP APIs for provisioning, self-service, and operator product and customer administration. These APIs allow operators to provision services and manage customer accounts in a flexible and scalable way.
- **Network Element Virtualizer (the ALU-I version of the NetIC) that represents the driver for network paths and network paths of the managed physical infrastructure. This functional block is defined inside the network space.** The north interface uses the Network Element Virtualizer (the ALU-I version of the NetIC) that represents the driver for network paths and network paths of the managed physical infrastructure. The south interface uses the Network Element Virtualizer (the ALU-I version of the NetIC) that represents the driver for network paths and network paths of the managed physical infrastructure. This functional block is defined inside the network space.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>FIWARE.Sprint.1.1.1</td>
<td>Open 09/11/2011 16:38</td>
<td>09/11/2011 17:00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FIWARE.Release.1.1</td>
<td>Close date</td>
<td>Context-based Security &amp; Compliance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Optional security services</td>
<td>Close date</td>
<td>Context-based Security &amp; Compliance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FIWARE.Sprint.1.1.3</td>
<td>Close date</td>
<td>Context-based Security &amp; Compliance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FIWARE.Release.1.2</td>
<td>Close date</td>
<td>Context-based Security &amp; Compliance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(As attack trace engine) customize vulnerability impact scores (IOS oriented).</td>
<td>Close date</td>
<td>Context-based Security &amp; Compliance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Feature Security Monitoring</td>
<td>Close date</td>
<td>Context-based Security &amp; Compliance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Security Monitoring</td>
<td>Close date</td>
<td>Context-based Security &amp; Compliance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Context-based Security &amp; Compliance</td>
<td>Close date</td>
<td>Context-based Security &amp; Compliance</td>
</tr>
</tbody>
</table>

**Goal:** Deploy the Secure Storage Service Asset as services (Web Services)

**As a PRRS framework I'd like to use the to be defined USDL-SEC language to perform my queries:**
- A system that would be able to describe security features and requirements.
- A system that would be able to collect all events concerning sensitive services and data.

**Requirement:**
- Developement of the Protocols and Interfaces between the PPL engine and the Credential Based Security Enabler.
- Design of the Protocols and Interfaces between the PPL engine and the Credential Based Security Enabler.

**Description:**
- The current status of the T8.4 Morphus asset (related to the optional security enabler).
- Get the list of SPs who had access to the profile and when, and what usage they did of the data
- Sufficient security to master the storage and the exchange of sensitive data.
- Using this service the DB owner can evaluate the risk of his shared resource and hide data.
- The service that can provide this asset is a remote estimation of the risk for an anonymized source.
- The relevant binary code in a malware, the system extract signatures from it and puts it in a malware database.
- To the IDMix technology the credentials can be anonymous in order to preserve the privacy of the user.

**Context:**
- IBM: Michael Osborne (osb@zurich.ibm.com)
- There are actually two assignees for this Epic:
  - IBM: Michael Osborne (osb@zurich.ibm.com)
  - IBM: Michael Osborne (osb@zurich.ibm.com)

**Entity:**
- Use the correlation of normalized heterogeneous events to detect a variety of attacks and use the correlation of normalized heterogeneous events to detect a variety of attacks.
- Propose the most appropriate mitigation responses based on results of the risk analysis.

**Functionality:**
- Deploy the Secure Storage Service Asset as services (Web Services).
- One of the objective of this task8.4 is to provide optional security services...