Private Public Partnership Project (PPP)

Large-scale Integrated Project (IP)

D.8.1.1b: FI-WARE GE Open Specifications (Security Chaper)

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-WP8-D.8.1.1b

Project Document Date: 2012-10-15

Deliverable Type and Security: Public

Author: FI-WARE Consortium

Contributors: FI-WARE Consortium

Table of Content

1 Introduction

1.1 Executive Summary

This document describes the Generic Enablers in the Security chapter, their basic functionality and their interaction. These Generic Enablers are part of the core framework of the FI-WARE platform by ensuring that the security needs from both the FI-WARE project (development and research activities, market analysis and consortium-specific exploitation requirements) and the FI PPP Use-Case (UC) projects are properly met.

The functionalities of the framework are illustrated with several abstract use-case diagrams, which show how the individual GE can be used to provide key security functionalities such as identity management or security monitoring, in order for them to be delivered as so-called generic security enablers that will be integrated with the design and implementation of the FI-WARE. Each GE Open Specification is first described on a generic level, describing the functional and non-functional properties, and is supplemented by a number of specifications according to the interface protocols, API and data formats.

1.2 About This Document

FI-WARE GE Open Specifications describe the open specifications linked to Generic Enablers GEs of the FI-WARE project (and their corresponding components) being developed in one particular chapter.

GE Open Specifications contain relevant information for users of FI-WARE to consume related GE implementations and/or to build compliant products which can work as alternative implementations of GEs developed in FI-WARE. The later may even replace a GE implementation developed in FI-WARE within a particular FI-WARE instance. GE Open Specifications typically include, but not necessarily are limited to, information such as:

- Description of the scope, behavior and intended use of the GE
- Terminology, definitions and abbreviations to clarify the meanings of the specification
- Signature and behavior of operations linked to APIs (Application Programming Interfaces) that the GE should export. Signature may be specified in a particular language binding or through a RESTful interface.
- Description of protocols that support interoperability with other GE or third party products
- Description of non-functional features

1.3 Intended Audience

The document targets interested parties in architecture and API design, implementation and usage of FI-WARE Generic Enablers from the FI-WARE project.

1.4 Chapter Context

The overall ambition of the Security Architecture of FI-WARE is to demonstrate that the Vision of an Internet that is "secure by design" is becoming reality. Based on achievements to date and/or to come in the short-term (both from a technological but also a standardization perspective) we will show that "secure by design" is possible for the most important core (basic) and shared (generic) security functionalities as anticipated by the FI-WARE project and in accordance with the requirements of external stakeholders and users such as the FI PPP Use Case projects. The “secure by design” concept will, therefore, address both the security properties of the FI-WARE platform itself and the applications that will be built on top of it.

In this section the foreseen high-level functional architecture is described, introducing the main modules and their expected relationships, then depicting the most important modules in detail along with their main functionalities.

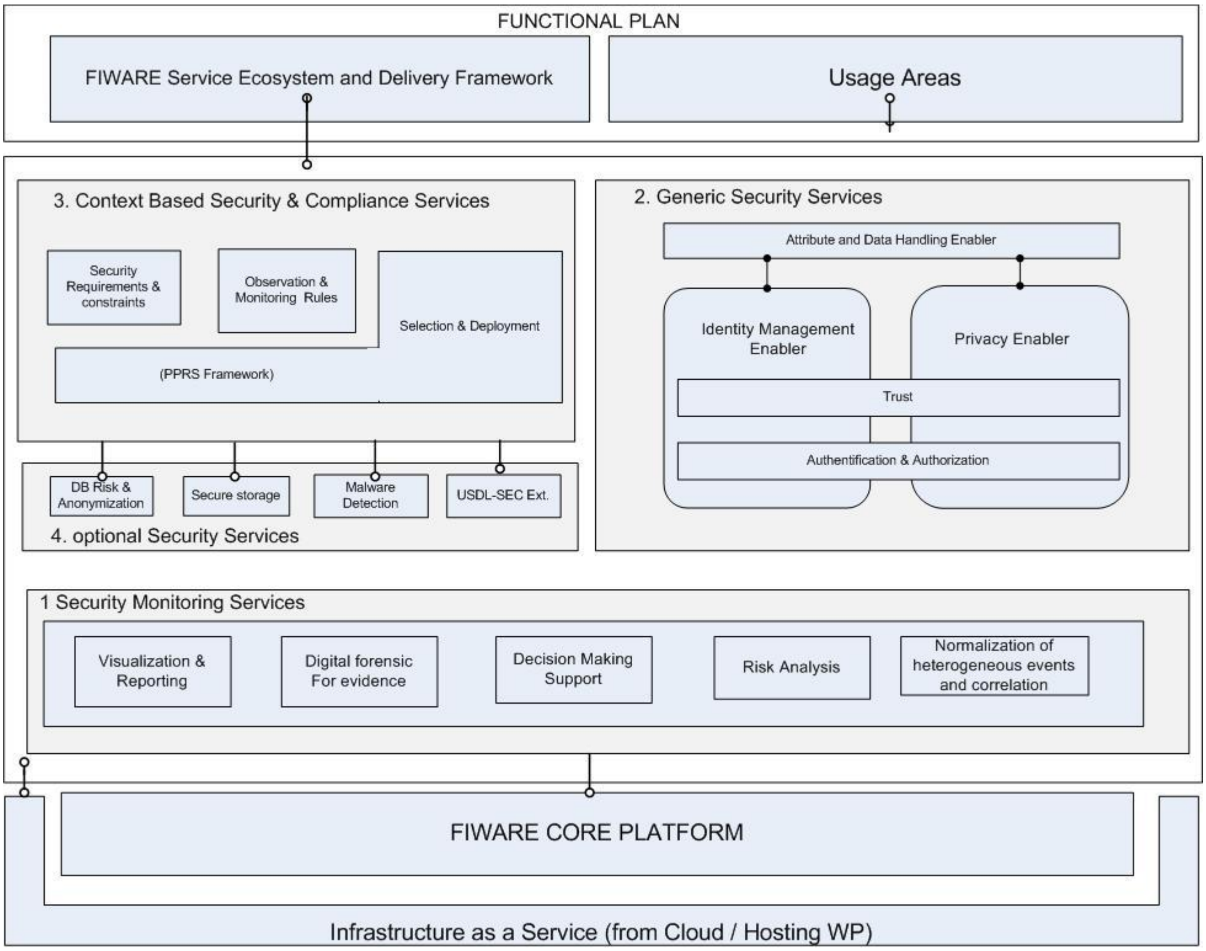
The high level architecture is formed by four main modules: Security monitoring mechanisms (M1), a set of General Core Security Mechanisms (e.g. Identity Management and Privacy solutions) (M2), Context-Based Security and Compliance (M3) where an enhanced version of USDL for security will support the matching of security goals with available security services while addressing compliance management, and a set of universally discoverable Optional Generic Security Services (M4) that will be instantiated at runtime and can be dynamically reconfigured (triggered by M3) based on the needs of specific scenarios.

The overall security plane of the FI-WARE architecture will interlink with practically all its functional modules. In order to simplify the description of these links subsequently the main components as well as their technical relationships with only the Application and Service Ecosystem and Delivery Framework and FI PPP Use Case projects are depicted:

The core general security mechanisms for the FI-WARE project will be provided by M2, including support for Identity Management, Authentication Authorization and Access, and Privacy. M3 will provide the required language and tools for describing services in the FI and their security needs. Where specific scenarios will require optional generic security services these can be consumed on a basis of what is provided by M4. A key architectural assumption is that security services may fail. Security monitoring mechanisms as provided by M1 may detect deviations with respect to the expected behaviour and signal this to M3 to take action (e.g. invoke alternative security services or trigger countermeasures if under attack).

FI-WARE GEs to be developed and/or integrated as part of the Security chapter will materialize the (Security) Reference Architecture sketched in Figure below. This Reference Architecture comprises:

- A component able to dynamically invoke and compose security services to answer related security needs while dealing with constraints which may apply (e.g. regulatory).
- A set of GEs for a number of shared security concerns (i.e. identity and access management as well as privacy and auditing) that are considered core and therefore present in any FI-WARE Instance.
- A set of optional Security GEs to address current and future requests from concrete Usage Areas.
- An advanced security monitoring system that covers the whole spectrum from acquisition of events up to display, going through analysis but also going beyond thanks to a digital forensic tool and assisted decision support in case of cyber attacks.


FUNCTIONAL PLAN
FIWARE Service Ecosystem and Delivery Framework
Usage Areas
3. Context Based Security & Compliance Services
Security Requirements & constraints
Observation & Monitoring Rules
Selection & Deployment
(PPRS Framework)
2. Generic Security Services
Attribute and Data Handling Enabler
Identity Management Enabler
Privacy Enabler
Trust
Authentification & Authorization
DB Risk & Anonymization
Secure storage
Malware Detection
USDL-SEC Ext.
4. optional Security Services
1 Security Monitoring Services
Visualization & Reporting
Digital forensic For evidence
Decision Making Support
Risk Analysis
Normalization of heterogeneous events and correlation
FIWARE CORE PLATFORM
Infrastructure as a Service (from Cloud / Hosting WP)

FI-WARE High Level Security Architecture

More information on the Security Chapter and FI-WARE in general can be found within the following pages:

- http://wiki.fi-ware.eu
- The Architecture of Security in FI-WARE
- Materializing Security in FI-WARE

1.5 Structure of this Document

The document is generated out of a set of documents provided in the public FI-WARE wiki. For the current version of the documents, please visit the public wiki at http://wiki.fi-ware.eu/

The following resources were used to generate this document:

- FIWARE.OpenSpecification.Security.SecurityMonitoring
- Security-Monitoring Mulval Attack Path Engine API Specification (PRELIMINARY)
- Security.Security-Monitoring OSSIM SIEM API Specification (PRELIMINARY)
- FIWARE.OpenSpecification.Security.Context-based security & compliance
- FIWARE.OpenSpecification.Details.Security.Context-based security & compliance_API

FIWARE.OpenSpecification.Security.Data Handling Generic Enabler

FIWARE.OpenSpecification.Security.DataHandlingGE.Open RESTful API Specification

FIWARE.OpenSpecification.Security.Optional Security Enablers.DBAnonymizer

FIWARE.OpenSpecification.Security.DBAnonymizer.Open RESTful API Specification

FIWARE.OpenSpecification.Security.IdentityManagement

Identity Management Generic Enabler API Specification

FIWARE.OpenSpecification.Security.Optional Security Enablers.SecureStorageService

FIWARE.OpenSpecification.Security.Privacy Generic Enabler

FI-WARE Open Specifications Legal Notice

Open Specifications Interim Legal Notice

1.6 Typographical Conventions

Starting with October 2012 the FI-WARE project improved the quality and streamlined the submission process for deliverables, generated out of the public and private FI-WARE wiki. The project is currently working on the migration of as many deliverables as possible towards the new system.

This document is rendered with semi-automatic scripts out of a MediaWiki system operated by the FI-WARE consortium.

1.6.1 Links within this document

The links within this document point towards the wiki where the content was rendered from. You can browse these links in order to find the "current" status of the particular content.

Due to technical reasons not all pages that are part of this document can be linked document-local within the final document. For example, if an open specification references and "links" an API specification within the page text, you will find this link firstly pointing to the wiki, although the same content is usually integrated within the same submission as well.

1.6.2 Figures

Figures are mainly inserted within the wiki as the following one:

```

[[Image:....|size|alignment|Caption]]
```

Only if the wiki-page uses this format, the related caption is applied on the printed document. As currently this format is not used consistently within the wiki, please understand that the rendered pages have different caption layouts and different caption formats in general. Due to technical reasons the caption can't be numbered automatically.

1.6.3 Sample software code

Sample API-calls may be inserted like the following one.

```
http://[SERVER_URL]?filter=name:Simth*&index=20&limit=10
```

1.7 Acknowledgements

The current document has been elaborated using a number of collaborative tools, with the participation of Working Package Leaders and Architects as well as those partners in their teams they have decided to involve.

1.8 Keyword list

FI-WARE, PPP, Architecture Board, Steering Board, Roadmap, Reference Architecture, Generic Enabler, Open Specifications, I2ND, Cloud, IoT, Data/Context Management, Applications/Services Ecosystem, Delivery Framework , Security, Developers Community and Tools , ICT, es.Internet, Latin American Platforms, Cloud Edge, Cloud Proxy.

1.9 Changes History

Release	Major changes description	Date	Editor
v0	First draft of deliverable submission	2012-10	SAP
v1	Second draft of deliverable submission	2012-11-03	THALES
v2	Draft for deliverable submission	2012-11-08	THALES, SAP
v3	Second draft for deliverable submission	2012-11-08	THALES, SAP
v4	Final version for deliverable submission	2012-11-12	THALES, SAP

2 FIWARE OpenSpecification Security SecurityMonitoring

You can find the content of this chapter as well in the wiki of fi-ware.

Name	FIWARE.OpenSpecification.Security.Security_Monitoring
Chapter	Security,
Catalogue-Link to Implementation	Security Monitoring
Owner	THALES, Daniel Gidoin

2.1 Preface

Within this document you find a self-contained open specification of a FI-WARE generic enabler, please consult as well the FI-WARE_Product_Vision, the website on http://www.fi-ware.eu and similar pages in order to understand the complete context of the FI-WARE project.

2.2 Copyright

2.3 Legal Notice

Please check the following Legal Notice to understand the rights to use these specifications

- FI-WARE Open Specifications Legal Notice

2.4 Overview

The Security Monitoring GE is part of the overall Security Management System in FI-WARE and as such is part of each and every FI-WARE instance. The target users are: FI-WARE Instance Providers and FI-WARE Application/Service Providers.

Security monitoring is the first step towards understanding the real security state of a future internet environment and, hence, towards realizing the execution of services with desired security behaviour and detection of potential attacks or non-authorized usage.

This generic enabler deals with the security monitoring and beyond, up to pro-active cyber-security i.e. protection of “assets” at large. It allows to assess the real security state of a future internet environment and, hence, towards realizing the execution of services with desired security behaviour and detection of potential attacks or non-authorized usage.

The main concerns of Security Monitoring are:

1. Detect vulnerabilities and identify risks
2. Score vulnerabilities impact and assess risks
3. Analyze events to correlate and detect threats and attacks
4. Treat risks and propose counter-measures
5. Visualize result alarms and residual risks in order to allow efficient monitoring from the security perspective

2.5 Basic Concepts

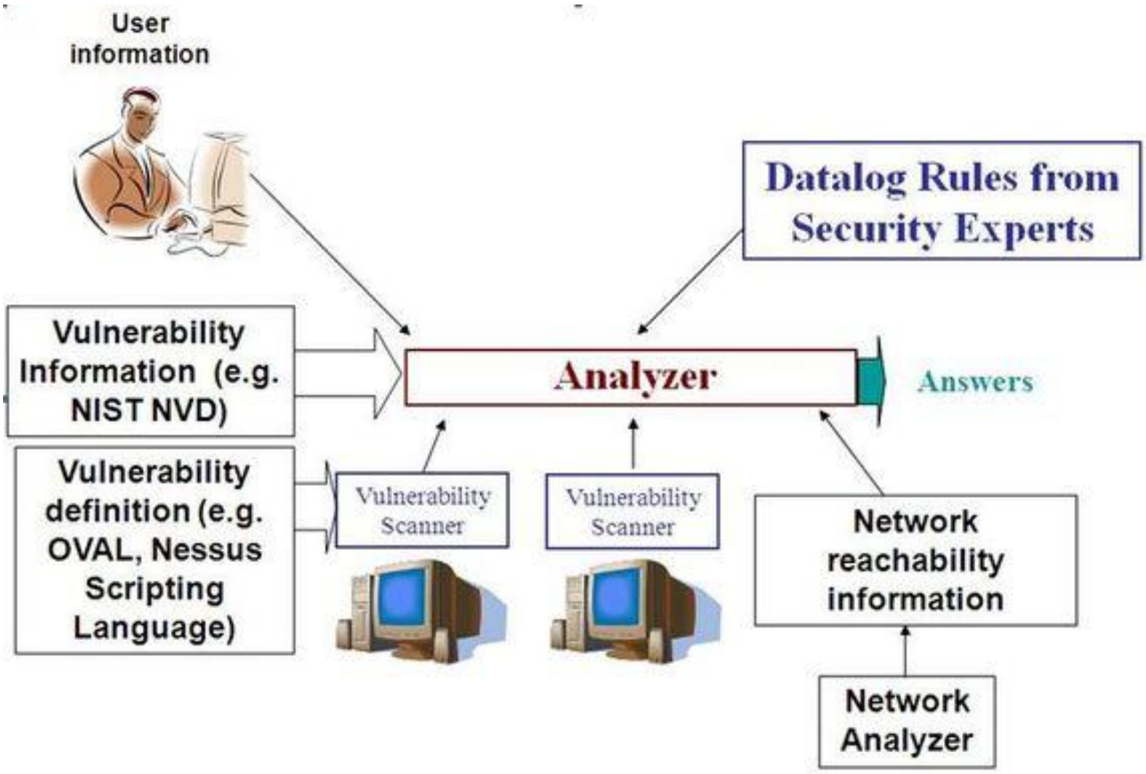
2.5.1 MulVAL Attack Paths Engine

To determine the security impact that software vulnerabilities have on a particular network, one must consider interactions among multiple network elements. For a vulnerability analysis tool to be useful in practice, the model used in the analysis must be able to automatically integrate formal vulnerability specifications from heterogeneous vulnerability sources.

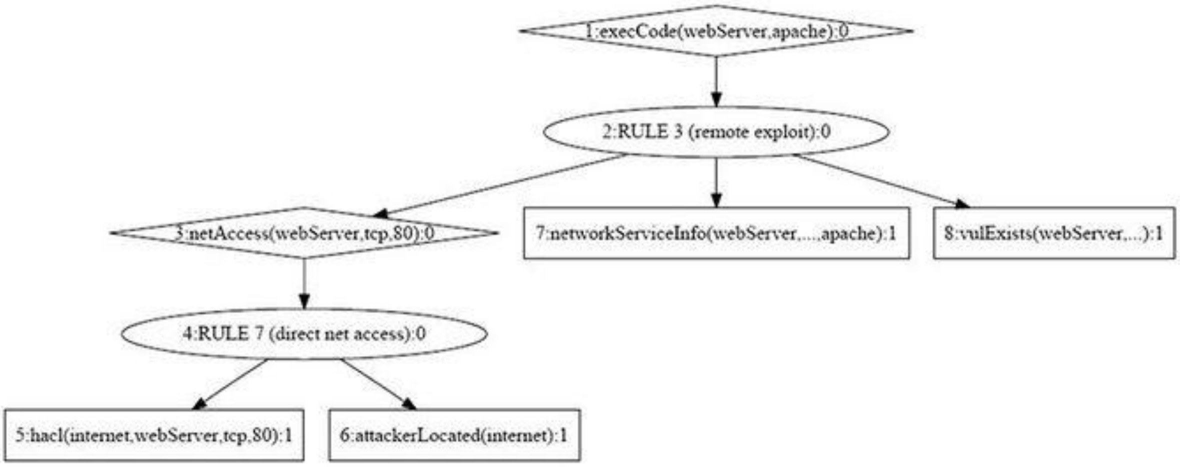
The MulVAL Attack Paths Engine is an end-to-end framework and reasoning system that conducts multihost, multistage vulnerability analysis on a network. The MulVAL Attack Paths Engine adopts Datalog (a query and rule language for deductive databases) as the modeling language for the elements in the analysis (bug specification, configuration description, reasoning rules, operating-system permission and privilege model, etc.). It has leveraged existing vulnerability-database and scanning tools by expressing their output in Datalog to feed the MulVAL Attack Paths Engine.

The inputs to the MulVAL Attack Paths Engine’s analysis are:

- Advisories: What vulnerabilities have been reported and do they exist on my machines?
- Host configuration: What software and services are running on my hosts, and how are they configured?
- Network configuration: How are my network routers and firewalls configured?
- Principals: Who are the users of my network?
- Interaction: What is the model of how all these components interact?
- Policy: What accesses do I want to allow?


User information
Datalog Rules from Security Experts
Vulnerability Information (e.g. NIST NVD)
Analyzer
Answers
Vulnerability definition (e.g. OVAL, Nessus Scripting Language)
Vulnerability Scanner
Vulnerability Scanner
Network reachability information
Network Analyzer

The current MulVAL Attack Paths Engine data model relies on the exploit range (local or remote) and the privilege escalation consequence data that are stored in NIST NVD. The figure below shows a logical attack graph computed by the Attack Path Engine.


1:execCode(webServer,apache):0
2:RULE 3 (remote exploit):0
3:netAccess(webServer,tcp,80):0
7:networkServiceInfo(webServer,...,apache):1
8:vulExists(webServer,...):1
4:RULE 7 (direct net access):0
5:hacl(internet,webServer,tcp,80):1
6:attackerLocated(internet):1

The MulVAL Attack Paths Engine uses Datalog (a subset of Prolog) to produce logical attack graphs. It takes as input a set of first-order logical configuration predicates and produces the corresponding attack graph. These configuration predicates include network specific security policies, binding information and vulnerability data gathered from vulnerability databases. The MulVAL Attack Paths Engine identifies possible policy violations through logical inference.

Attack graph presents a qualitative view of security discrepancies:

- It shows what attacks are possible, but does not tell you how bad the problem is.
- It captures the interactions among all attack possibilities in your system.

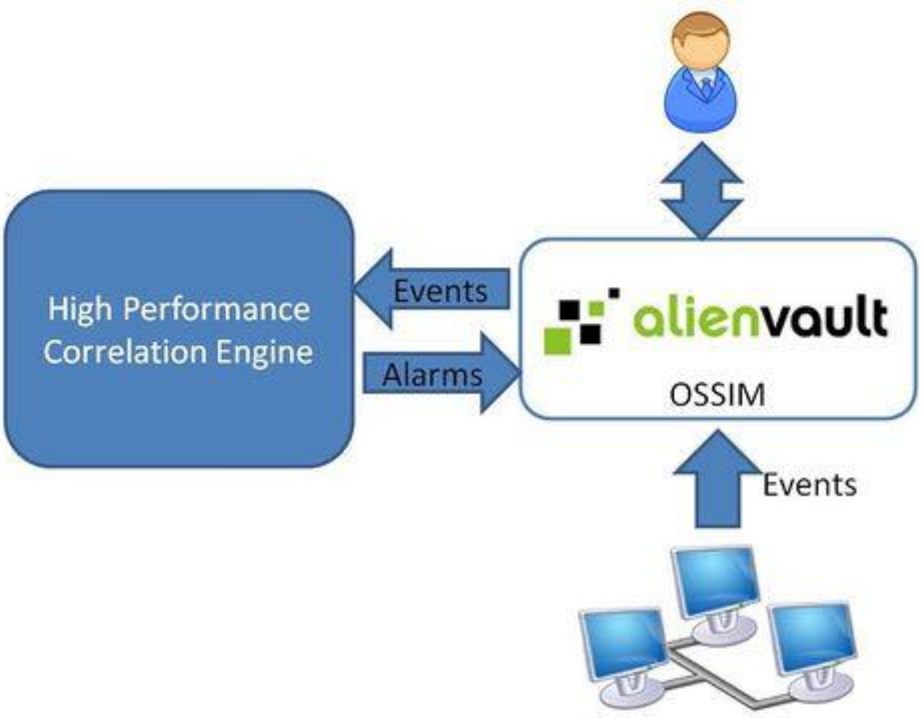
CVSS provides a quantitative property of individual vulnerabilities:

- It tells you how bad an individual vulnerability could be.
- But it does not tell you how bad it may be in your system.

The idea is to use CVSS to produce a component metric, i.e. a numeric measure on the conditional probability of success of an attack step. The MulVAL Attack Paths Engine aggregates the probabilities over the attack-graph structure to provide a cumulative metric, i.e. the probability of attacker success in your system. Suppose there is a “dedicated attacker” who will try all possible ways to attack your system. If one path fails, he will try another. The cumulative metric is the probability that he can succeed in at least one path.

2.5.2 Service Level SIEM

Limitations of current SIEM (Security Information and Event Management) systems are mainly in line with performance and scalability leading to the inability to process vast amounts of diverse data in a short amount of time. Next generation of SIEM solutions should overcome these performance limitations of its predecessors allowing in this way to monitor more systems, to process more complex rules or even to correlate events at different layers. To achieve the above commented goals, the SIEM to be included in FI-WARE will incorporate a high performance parallel correlation engine that will improve drastically the correlation capabilities of the current SIEM solutions available in the market. In the context of FI WARE this high performance correlation engine will be built on top of the OSSIM (Open Source Security Information Management - http://www.ossim.net) however integration with other tools, such as Prelude or Sentinel, could be analysed



High performance correlation engine

2.5.3 Botnet Tracking System

The NXDOMAIN-based Analysis focuses on the detection of «domain flux botnets», where the C&C domain names are frequently changed in order to escape from classical block-lists like the ones provided by DNSBL (Domain Name System Blacklists). This analysis relies on the observation of the behaviour of such botnets, and the way the bots try to locate their C&C servers. In order to find the domain name attached to the C&C server, the bots will request several domain names first, determined by more or less complex Domain Generating Algorithms (DGA): time-based, pseudorandom characters, dictionary based generation, etc. At a given time, such algorithms will generate a list of possible domain names to request, amongst them only one or few will be effectively reserved by the botmaster.

Because only few domain names are really associated to an IP address, bots will generate several DNS requests - answered by DNS errors - until finding an active domain. The target of the proposed solution is to detect abnormal error rates in order to identify and track the underlying botnet. Advantages of such approach is that:

- The DNS error traffic represents only a small portion of the whole DNS flow, thus ensuring a better scalability of our approach, a faster detection and a “less intrusive” analysis for the end-users.
- The DNS errors present a very limited meaning by themselves. Such analysis would not allow users’ profiling, and limit in that way the privacy impact.
- The DNS error traffic presents a very high dispersion compared to the successful traffic. As for example, there will be a huge number of users doing requests to www.orange.com, while the probability for a user to request the non-existing domain whzejdqmvnt.dynserv.com is very low. Such characteristic makes DNS errors traffic easier to analyse in order to detect abnormal behaviours.

2.5.4 Fuzzer

To allow IoT developers to assess the security of their applications, we developed a fuzzing tool for the 6LoWPAN protocol.

Fuzzing is a software testing technique, that involves providing valid, invalid, unexpected or random information as input of an application. Then, the program itself will react to these inputs, reporting exceptions or crashes, failing in the normal behavior or keeping the normal flow. The technique goal is to take advantage of the low-cost computation power to find out unexpected scenarios that lead to situations that escape from the normal flow scenario and produce an unexpected behavior.

In our current case, the target application is either the protocol itself or the application that resides on a remote device; so, in this case, we will be sending messages through the network to test the target device's behavior.

6LoWPAN is the acronym of IPv6 over Low power Wireless Personal Area Networks, also it's the name of the IETF working group in charge of the protocol.

The 6LoWPAN group has defined an encapsulation and header compression mechanism that defines a set of compression/decompression rules taking advantage of the most common messages sent through a typical wireless sensor network, and exploiting some features of the underlaying layer, IEEE 802.15.4, and the upper layer, IPv6. Furthermore, duplication of information is avoided, allowing the protocol to send really short messages, and in this way, helps the devices saving energy. The protocol also defines some special rules to fragment long IPv6 messages, being able in this way, to send minimal length IPv6 message.

2.5.4.1 ***Note on interaction between the Fuzzer and the IoT Work Package***

The Protocol Adapter GE provides an adaptation layer between the Gateway and any IoT device, for devices that include an IP stack and suppoort the CoAP protocol (from the IETF "CoRE" group), and the IoT Work Package concentrates on the application layer, and relies on existing standards for the lower network layers.

6LoWPAN & RPL are simply one of these standards, that allow to communicate with IoT devices using IPv6, and they are also being defined by the IETF (by the "6lowpan" and "roll" groups).

So, there is actually no conflict between this GE and the IoT Work Package, as they don't target the same layers.

The Fuzzer can be used as-is by Use Cases that decide to deploy devices that use the 6LoWPAN stack, and it can support any protocol for which a scapy module exists.

And in the event Use Cases decide to adopt other standards, and have an interest in the Fuzzer, we can also discuss the possibility for us to implement the necessary modules.

2.5.5 Countermeasures

The decision making support will look for the possible topological solutions and the software updates that could reduce / cut a given attack path. A remediation DB is needed; it will be built on publicly-available Security Advisories (for example, coming from CERT-EU); it will be external to the GE, as the Vulnerability DB. Only clear input will be considered, i.e. the advisories which are making a non ambiguous link between one or several CVE-ID and a software update. Remediation will be correlated to the vulnerabilities mentioned in the attack path. For each node, a software or a topological update will be proposed. As a result, a defence path will be produced, mirroring the submitted attack path.

2.5.6 Visualization framework

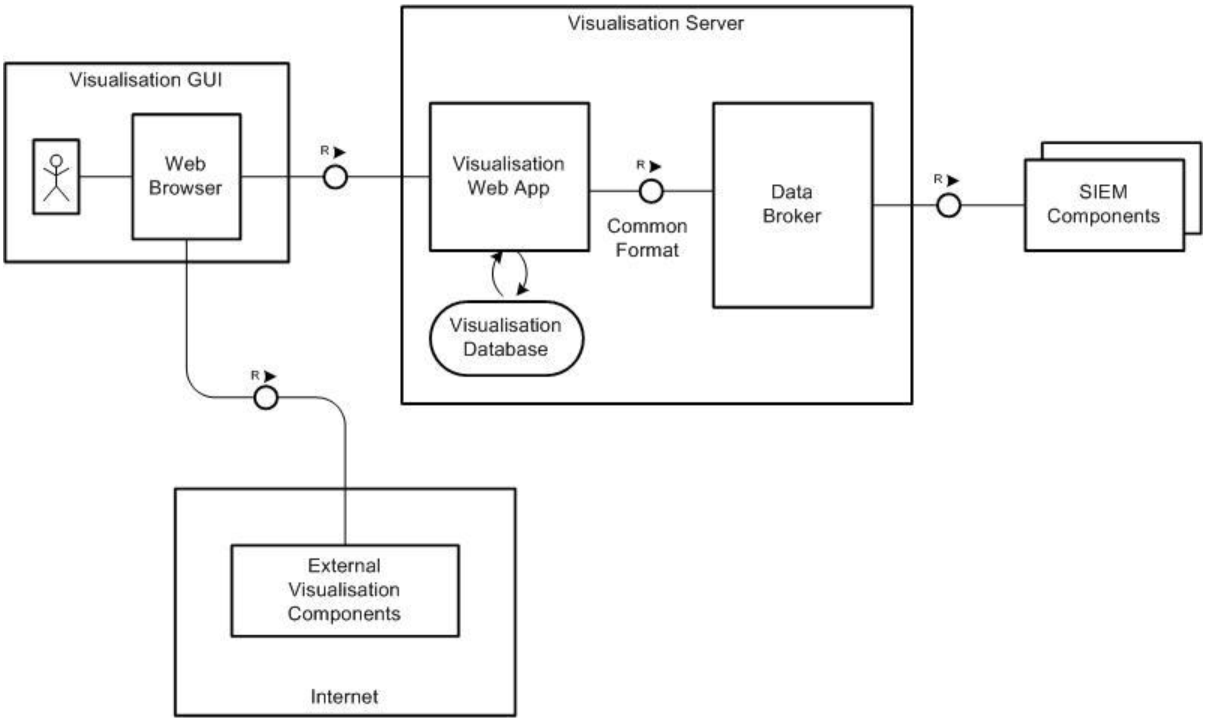
Systems that monitor the security of a network, such as network probes as part of an Intrusion Detection Systems (IDS), can generate a large amount of data. It is generally agreed that one of the most effective ways in which large volumes of data are presented to a human is by the use of visual analytics techniques. The INTERSECTION Visualisation Framework aims to enable large quantities of data to be presented to users in ways that aid their understanding of it. It is a flexible framework that allows the easy combination of multiple sources of data and enables the easy combination of third party and bespoke visualisations. The Visualisation Framework, business-oriented, allows the user to choose which data is visualised and which visualisation techniques are used.

In addition, it is extensible, allowing the addition of new data sources, data processing and visualisations. The design of the system is built around the concept of web-based mash-ups, which combine content from multiple sources into an integrated experience, and rich Internet applications (RIA) have a similar set of features and functionality to desktop applications.

The functionality of the Visualisation Framework can loosely be considered in two parts: the Data Broker, which collates and manages data; and the Visualisation Web Application, which provides and controls the visualisation. The Visualisation Framework’s Data Broker interfaces with the various data sources to collect data. This can be achieved via a message queue, through accessing an external database or via some other means, depending on the source of this data. Data routes are created, as required, between the various data sources and end points. In all cases, the end points are adapters that transform incoming data into a common form, used throughout the visualisation component. Data is also stored in the visualisation database to allow a user to review historical as well as current data.

The Visualisation Web Application provides a number of key functions:

- Serves up pages to a user allowing them to set-up and interact with visualisations.
- Provides access to locally stored visualisations and facilitate the use of third party visualisations through the Internet.
- A conduit between data held on the server and the user who is accessing the visual analytics system from a web-browser.
- Allows the user to choose, configure and map data to visualisation axes as required.



2.6 Security Monitoring Architecture

We detail in the following the interactions between the components of the Security Monitoring Architecture, as well as their respective connections to the FI-WARE framework. We start by the three blocks composing the input for the Heterogeneous Event Normalization Service. The aim of this service is to normalize heterogeneous events so that they can be processed by the Service oriented SIEM. In order to be correlated by the SIEM, the events must be pertinent for the risk analysis.

Events into the front of this service are:

- Context-Based Security & Compliance violation events, from GE provided by ATOS and SAP in WP8
- Secure Storage Service events from GE provided by TCS in WP8.
- Cloud, Internet of Things and Interface-to-networks events from GE provided in WP4, WP5 and WP7.

As for the Heterogeneous Event Normalization Service itself, it is part of WP8 and provides inputs for Service-level SIEM, Forensics Framework, and eventually Complex-Event Processing in Data/Context Management of WP6.

The Service-level SIEM provides its results directly to the Visualization Framework. Complex-Event Processing on the other hand, serves as input for both the Forensics and Visualization Frameworks. It can be deduces that the CEP can potentially be bypassed. The Security Monitoring enabler is intended to be used to assess compliance to the security requirements of Business Framework for the Applications and Services Ecosystem and Delivery (WP3). Security Monitoring employs the Complex Event Processing from Data/Context Management (WP6). Attack Paths Engine in Security Monitoring utilizes the Vulnerability Collections from Cloud/IOT/I2ND, the Vulnerabilities Database (NVD), and the Configuration Management

Database (CMDB) in WP4, the latter being also involved. From the internal viewpoint of Security Monitoring, the Attack Paths Engine includes in its entries the Vulnerability Scanners operating on the network, and the Fuzzer block for assessing the applications' security. Business-oriented Vulnerability requires as input the Configuration Management Database (CMDB) in WP4, and the vulnerability scoring it provides is employed by the Attack Path Engine, along with the other inputs of the latter. By combining the input from the Botnet Tracking System with the one from the Attack Paths Engine, the Counter-Measures App yields the proposed output to the Visualization Framework for further monitoring and decision-making purposes.

The decision making support will aim to provide some help to the security operator by proposing several possible countermeasures / remediation that could be deployed in the monitored system / services. To facilitate the decision making processes, assets contribute to early warning of harmful events, for the detection of suspicious behaviour, for correlation of heterogeneous security events and for the computation of critical attack paths. In addition, the man-machine interfaces ensure that solutions are effectively designed for end-users, providing them with increased efficiency. This would include advanced visualisation techniques to provide a more complete picture to handle complex situations efficiently.

Finally, a digital forensics for evidence consist to develop capabilities to trace illegal activity in cyberspace back to its origin. Correlating events provides the means to support the search for evidence process. Timeframe analysis will can be useful in determining when events occurred. For this, we will can review the time and data contained in the file system metadata, linking error logs, connections logs, security events, alarms and files of interest to the timeframes relevant to the investigation.

The Security monitoring GE Chapter meets the requirements of ISO 27001 (to see 4.2 Establishing and managing the ISMS).

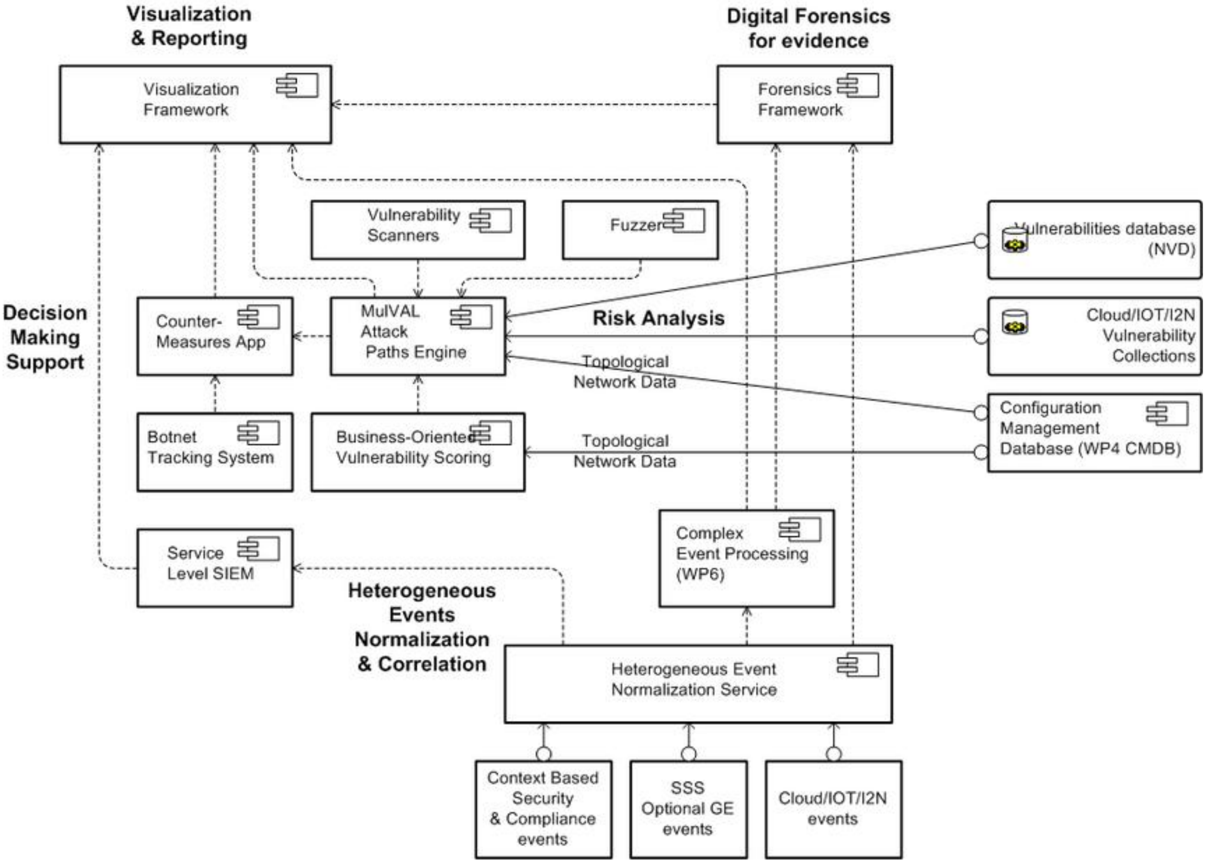
Among others things, it provides an answer to “c” paragraph (...Identify a risk assessment methodology that is suited to the ISMS), to “d” paragraph (identify the risks); “e” paragraph (Analyse and evaluate the risks); to “f” paragraph (Identify and evaluate options for the treatment of risks) and to “g” paragraph (Select control objectives and controls for the treatment of risks.)

In conclusion, the security monitoring enabler is composed of the following functionalities:

- **Normalization of heterogeneous events and correlation**. This functionality covers the normalization and correlation of massive and heterogeneous security events.
- **Risk analysis**. Considering the threat profiles and the related system vulnerabilities, a risk profile is built for each threat, containing qualitative values which measure the impact of the outcome of threats to the organization.
- **Decision making support**. Countermeasures can be selected in order to mitigate the risks, for instance implementing new security practices within the organization, or taking the actions necessary to maintain the existing security practices or fixing the identified vulnerabilities.
- **Digital forensics for evidence**. it deals with the acquisition of data from a source, the analysis of the data and extraction of evidence, and the preservation and presentation of the evidence. The digital evidence is intended to facilitate the reconstruction of events found to be malevolent or helping to anticipate unauthorized actions.

- **Visualization and reporting**. It will provide a dynamic, intuitive and role-based User System Interface for the various stakeholders to use in order to understand the current security situation, to make decisions, and to take appropriate actions.

The GE as envisaged will address security monitoring and beyond, up to pro-active cyber-security i.e. protection of "assets" at large. The figure below provides a high-level initial architectural sketch of the Security Monitoring GE as envisaged in FI-WARE.


Visualization & Reporting
Digital Forensics for evidence
Visualization Framework
Forensics Framework
Vulnerability Scanners
Fuzzer
Vulnerabilities database (NVD)
Decision Making Support
Counter-Measures App
MulVAL Attack Paths Engine
Risk Analysis
Cloud/IOT/I2N Vulnerability Collections
Topological Network Data
Configuration Management Database (WP4 CMDB)
Botnet Tracking System
Business-Oriented Vulnerability Scoring
Topological Network Data
Complex Event Processing (WP6)
Service Level SIEM
Heterogeneous Events Normalization & Correlation
Heterogeneous Event Normalization Service
Context Based Security & Compliance events
SSS Optional GE events
Cloud/IOT/I2N events

2.7 Basic Design Principles

2.7.1.1 *MulVAL Attack Paths Engine*

To determine the security impact software vulnerabilities have on a FIWARE architecture instantiation, one must consider interactions among multiple network components. The model used in the vulnerability analysis is able to automatically integrate formal vulnerability specifications from the bug-reporting community. But also from various vulnerability databases, specific to the cloud hosting, the internet of things, I2N.. Also, the analysis is able to scale to networks with thousands of machines.

To achieve these two goals, the MulVAL Attack paths Engine, composed of an end-to-end framework and a reasoning system, conducts multihost, multistage vulnerability analysis on a FIWARE architecture. The MulVAL Attack Paths Engine adopts Datalog as the modeling language for the elements in the analysis (bug specification, configuration description, reasoning rules, operating-system permission and privilege model, etc.). It easily leverage existing vulnerability-database and scanning tools by expressing their output in Datalog and feeding it to the Attack Path reasoning engine.

The reasoning engine consists of a collection of Datalog rules that captures the operating system behavior and the interaction of various components in the network. Thus integrating information from the bug-reporting community and off-the-shelf scanning tools in the reasoning model is straightforward. Reasoning rules specify semantics of different kinds of exploits, compromise propagation, and multihop network access. The rules are carefully designed so that information about specific vulnerabilities are factored out into the data generated from OVAL (Open Vulnerability and Assessment Language-MITRE)and ICAT (Categorization of Attacks Toolkit-NIST). The interaction rules characterize general attack methodologies (such as “Trojan Horse client program”), not specific vulnerabilities. Thus the rules do not need to be changed frequently, even if new vulnerabilities are reported frequently.

The MulVAL Attack paths Engine uses an exploit dependency graph to represent the pre and post conditions for exploits. Then a graph search algorithm can “string” individual exploits and find attack paths involves multiple vulnerabilities. This algorithm is adopted in Topological Vulnerability Analysis (TVA), a framework that combines an exploit knowledge base with a remote network vulnerability scanner to analyze exploit sequences leading to attack goals. Compared with a graph data structure, Datalog provides a declarative specification for the reasoning logic, making it easier to review and augment the reasoning engine when necessary.

The reasoning engine scales well with the size of the network. Once all the information is collected, the analysis can be performed in seconds for networks with thousands of machines.

2.7.1.2 *Service Level SIEM*

The OSSIM agent will receives normalized event.

The fields of which the standardised event consists are:

- type: Type of event, Detector or Monitor.
- date: date on which the event is received from the device.
- sensor: IP address of the sensor generating the event
- plugin_id: Identifier of the type of event generated
- plugin_sid: Class of event within the type specified in plugin_id
- priority: Possible deprecated (agent can't decide priority, just server)
- protocol: Three types of protocol are permitted in these events. Should a different one reach the server, the event will be rejected:: TCP, UDP or ICMP
- src_ip: IP which the device generating the original event identifies as the source of this event
- src_port: Source port
- dst_ip: IP which the device generating the original event identifies as the destination of this event
- dst_port: Destination port

- log: Event data that the specific plugin considers as part of the log and which is not accommodated in the other fields. Due to the Userdata* fields, it is used increasingly less.
- data: Normally stores the event payload, although the plugin may use this field for anything else
- username: User who has generated the event or user with whom it is identifying, mainly used in HIDS events
- password: Password used in an event
- filename: File used in an event, mainly used in HIDS events
- userdata1: These fields can be defined by the user from the plugin. They can contain any alphanumeric information, and on choosing one or another, the type of display they have in the event viewer will change. Up to 9 fields can be defined for each plugin.
- userdata2
- …
- userdata9

2.7.1.3 *Fuzzer*

The fuzzer engine itself is built around the Scapy packet manipulation framework, to which we have added a module that supports the assembly, disassembly and fuzzing of the 6LoWPAN protocol.

To be able to inject crafted 6LowPAN packets onto the network, we use an Atmel RZUSBstick, a 802.15.4 USB dongle, running a modified version of the Contiki OS, that simply relays packets without altering them.

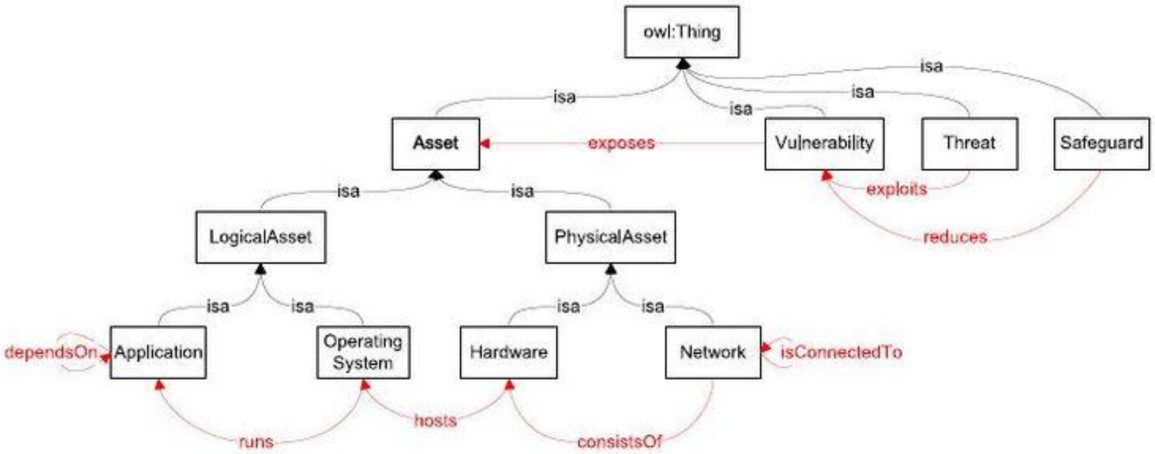
The fuzzer engine is then driven by a scenario written in XML that defines a "normal" sequence of sent and received packets, as well as callback functions written in python that define the fuzzing process — which messages and message fields to alter, and how.

2.7.1.4 *Countermeasures*

The design of the decision making support module will be based on the Ontology Handler asset which has already the capacity to handle remediation (called ‘Safeguard’ in the currently defined ontology). It will coupled with the following capacities:

- a remediation extractor that will find the remediation information in the Security Advisories and will populate the ontology with it
- a defence path engine that will analyze the submitted attack path and will build the solution / countermeasure

The Ontology Handler is in charge of providing the Decision making support with knowledge relative to information systems infrastructure and security information, which are retrieved from the vulnerabilities, by means of an ontology. These two components can be replaced by other components that respect the interfaces of the Ontology Handler. In addition, the instantiated ontologies provided by the Ontology Handler can be directly used by the numerous inference engines to infer knowledge and security concerns.



The ontology repository exposes interfaces to create, access, manipulate and store instances of the Security Ontology. A. Security Ontology in computer science and information science, an ontology is a formal representation of a set of hierarchical concepts within a domain and the relationships between those concepts. It provides a vocabulary of classes and relationships to describe a domain. Ontologies are used to reason about the properties of their domain and to infer knowledge applying rules.

The proposed Security Ontology has been designed to describe the interdependencies between information systems' assets and their identified security issues according to the definitions of the ISO/IEC 13335-1:2004 standard. In this standard, vulnerabilities are considered as a property of a network security system. Assets and components have weak points named vulnerabilities. These vulnerabilities can be exploited by threats, that is, potential causes of unwanted incident which may result in harm to a system or organization, leading to attacks.

The proposed Security Ontology has been partially conceived with these concepts. We used OWL, the Web Ontology Language, endorsed by the W3C. Well-known in Semantic Web, this knowledge representation language utilizes a semantic model intended to provide compatibility with RDF (Resource Description Framework) Schema. OWL ontologies are most commonly serialized using RDF/XML (Extensible Markup Language) syntax. In addition, we integrated the Semantic Web Rule Language (SWRL), which extends OWL with inference rule extensions, within the OWL INSPIRE Security Ontology. An example of an inference rule would be to classify the assets according to their threats: AttackPatternClassification :

- Asset(?x) ^ exposes(?y; ?x) ^ exploits(?z; ?y) □! related to(?x; ?z)

Such a rule can be applied on every instance of the Security Ontology available in the Ontology repository.

The Ontology repository is the key component of the Ontology Handler. It serves as a knowledge base to store information relative to systems infrastructure and their associated security data using the Security Ontology. Each ontology instance corresponds to a version of an information system and its associated vulnerabilities, threats, and safeguards. These security data are retrieved from the vulnerability DB. Finally, the Ontology repository embeds mechanisms to create, save and retrieve instances of the Security Ontology and mechanisms to add, delete, and rename individuals and to set links between the individuals with predefined relationships. These functionalities are exposed in an interface in order for other components to remotely manipulate ontologies. The Ontology repository has been implemented in Java with the Jena library. The APIs of this component were defined in accordance with the principles of the OSS/J specification: they are available remotely by means of a Web Service.

2.7.1.5 *Visualization framework*

- Decoupling of data from visualisations. A fundamental principle is that the data is stored in common formats to allow any visualisation to work with the data and to allow each visualisation to work with multiple different data sets. This requires input data to be formatted in a suitable way to allow this (e.g. identification and location of common fields with other data). New data types can be accommodated, perhaps requiring some minor modifications to the Service. However, highly structured data, and particularly XML formats are preferred.
- Real-time input. The Service is designed to receive data in real time, and is particularly suited for publish-subscribe architectures.

2.8 Re-utilised Technologies/Specifications

2.8.1 Attack Path Engine

The Attack Path Engine uses the report of vulnerability scanners. Scanner can run asynchronously on each host and which adapts existing tools such as OVAL to a great extent—and an analyzer, run on one host whenever new information arrives from the scanners.

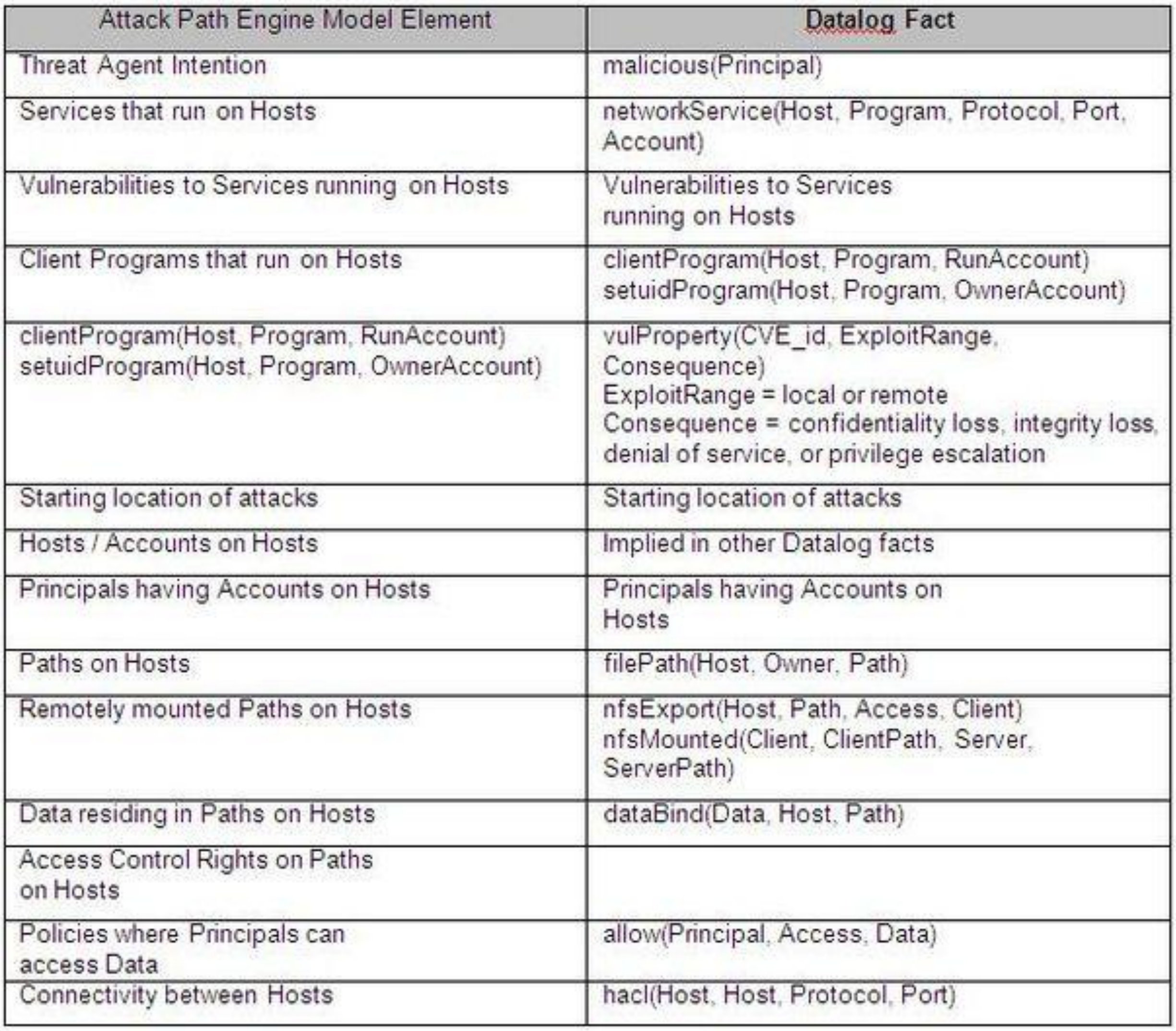
An OVAL scanner takes such formalized vulnerability definitions and tests a machine for vulnerable software. The result is converted into Datalog clauses like the following:

vulExists(webServer, 'CAN-2002-0392', httpd).

Namely, the scanner identified a vulnerability with CVE id CAN-2002-0392 on machine webServer. The vulnerability involved the server program httpd. However, the effect of the vulnerability—how it can be exploited and what is the consequence — is not formalized in OVAL. NVD the vulnerability database developed by the National Institute of Standards and Technology (NIST), provides the information about a vulnerability's effect through CVSS Impact metrics. The relevant information is converted from CVSS into Datalog clauses such as:

vulProperty('CAN-2002-0392', remoteExploit,privilegeEscalation.

The Attack Path Engine models elements in Datalog. The model elements are recorded as Datalog facts. The Attack Path Engine requires all Datalog facts to be defined prior to performing any analysis. Missing or incorrect facts will result in a misleading analysis of the system being modelled. The following table shows the elements modelled by the Attack Path Engine and their Datalog fact statements sorted by the DAP layer in which they belong.



Attack Path Engine Model Element	Datalog Fact
Threat Agent Intention	malicious(Principal)
Services that run on Hosts	networkService(Host, Program, Protocol, Port, Account)
Vulnerabilities to Services running on Hosts	Vulnerabilities to Services running on Hosts

Client Programs that run on Hosts	clientProgram(Host, Program, RunAccount)
	setuidProgram(Host, Program, OwnerAccount)

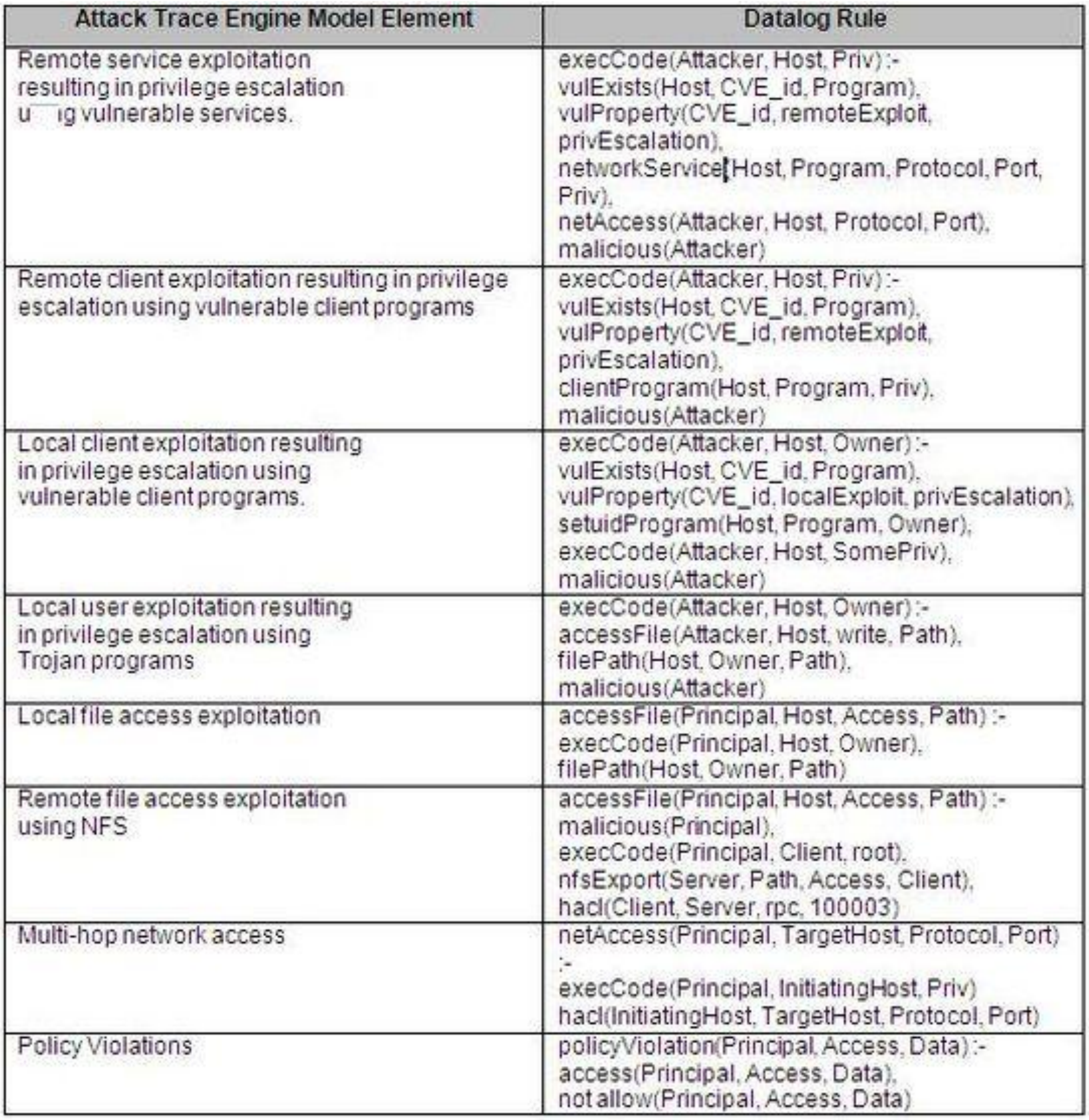
clientProgram(Host, Program, RunAccount)
setuidProgram(Host, Program, OwnerAccount)
	vulProperty(CVE_id, ExploitRange, Consequence)
	ExploitRange = local or remote
	Consequence = confidentiality loss, integrity loss, denial of service, or privilege escalation

Starting location of attacks	Starting location of attacks
Hosts / Accounts on Hosts	Implied in other Datalog facts
Principals having Accounts on Hosts	Principals having Accounts on Hosts
Paths on Hosts	filePath(Host, Owner, Path)

Remotely mounted Paths on Hosts	nfsExport(Host, Path, Access, Client)
	nfsMounted(Client, ClientPath, Server, ServerPath)

Data residing in Paths on Hosts	dataBind(Data, Host, Path)
Access Control Rights on Paths on Hosts	
Policies where Principals can access Data	allow(Principal, Access, Data)
Connectivity between Hosts	hacl(Host, Host, Protocol, Port)

How the Attack Path Engine Datalog facts interrelate is recorded as Datalog reasoning rules that are shown in the following table.



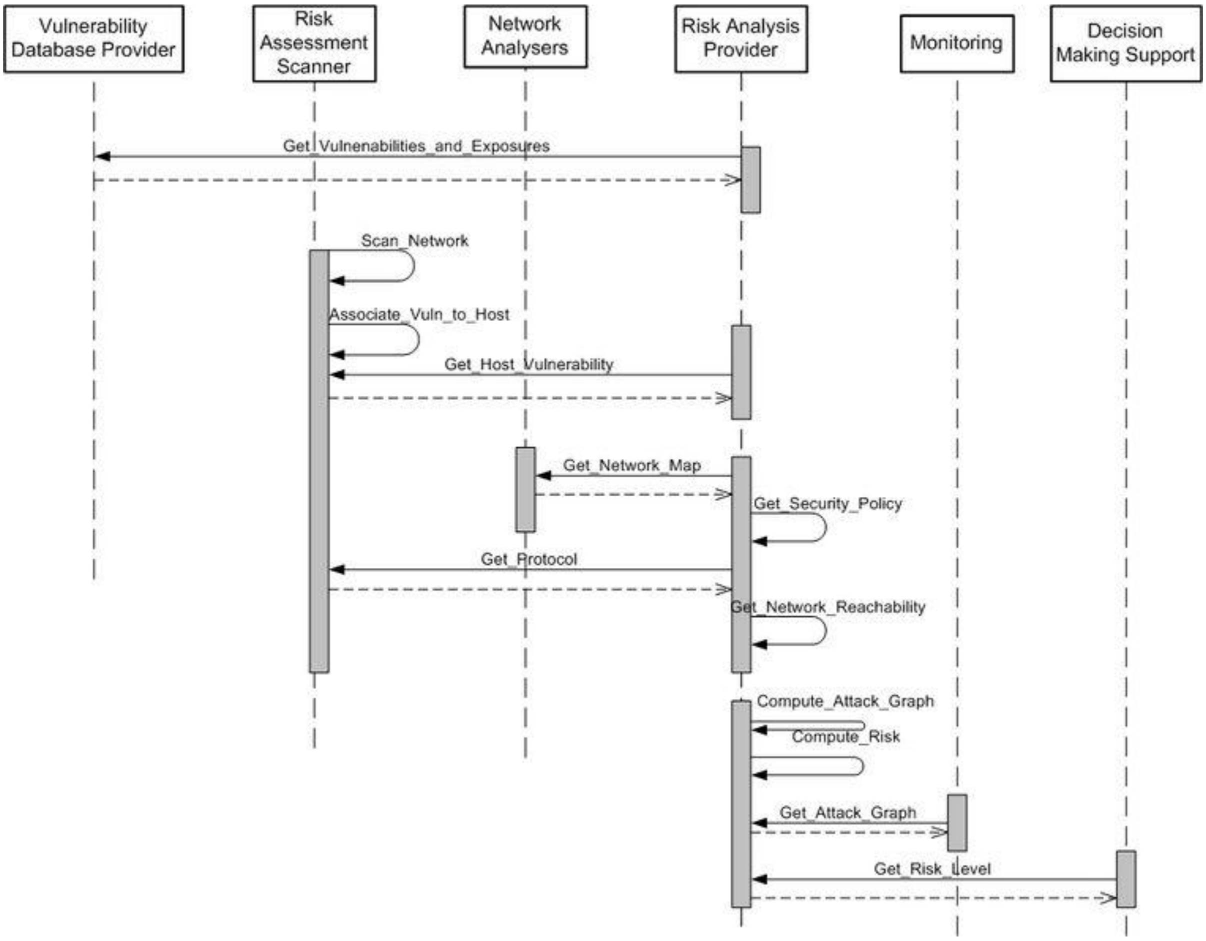
Attack Trace Engine Model Element	Datalog Rule
Remote service exploitation resulting in privilege escalation using vulnerable services.	execCode(Attacker, Host, Priv) :- vulExists(Host, CVE_id, Program), vulProperty(CVE_id, remoteExploit, privEscalation), networkService(Host, Program, Protocol, Port, Priv), netAccess(Attacker, Host, Protocol, Port), malicious(Attacker)
Remote client exploitation resulting in privilege escalation using vulnerable client programs	execCode(Attacker, Host, Priv) :- vulExists(Host, CVE_id, Program), vulProperty(CVE_id, remoteExploit, privEscalation), clientProgram(Host, Program, Priv), malicious(Attacker)
Local client exploitation resulting in privilege escalation using vulnerable client programs.	execCode(Attacker, Host, Owner) :- vulExists(Host, CVE_id, Program), vulProperty(CVE_id, localExploit, privEscalation), setuidProgram(Host, Program, Owner), execCode(Attacker, Host, SomePriv), malicious(Attacker)
Local user exploitation resulting in privilege escalation using Trojan programs	execCode(Attacker, Host, Owner) :- accessFile(Attacker, Host, write, Path), filePath(Host, Owner, Path), malicious(Attacker)
Local file access exploitation	accessFile(Principal, Host, Access, Path) :- execCode(Principal, Host, Owner), filePath(Host, Owner, Path)
Remote file access exploitation using NFS	accessFile(Principal, Host, Access, Path) :- malicious(Principal), execCode(Principal, Client, root), nfsExport(Server, Path, Access, Client), hacl(Client, Server, rpc, 100003)

Multi-hop network access	netAccess(Principal, TargetHost, Protocol, Port) :-
	execCode(Principal, InitiatingHost, Priv)
	hacl(InitiatingHost, TargetHost, Protocol, Port)

Policy Violations	policyViolation(Principal, Access, Data) :- access(Principal, Access, Data), not allow(Principal, Access, Data)

With the occurrence of new vulnerabilities, assessment of their security impact on the network is important in choosing the right countermeasures: patch and reboot, reconfigure a firewall, dismount a file-server partition, and so on.

The next figure show the sequence diagram with the interaction beetwen thhe others components:



Vulnerability Data Interface Description

-<definition class="vulnerability" id="oval:org.mitre.oval:def:99" version="4">

```
 -<metadata>
   -<title>
     IE v6.0 Content Disposition/Type Arbitrary Code Execution
    </title>
```

-<affected family="windows">

```
  <platform>Microsoft Windows 2000</platform>
  <product>Microsoft Internet Explorer</product>
```

</affected> <reference ref_id="CVE-2002-0193" ref_url="http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0193" source="CVE"/> -<description>

```
  Microsoft Internet Explorer 5.01 and 6.0 allow remote attackers to
execute   arbitrary   code   via   malformed   Content-Disposition   and
Content-Type  header  fields  that  cause  the  application  for  the
spoofed file type to pass the file back to the operating system for
handling rather than raise an error message, aka the first variant
of the "Content Disposition" vulnerability.
```

```
</description><
```

-oval_repository>

```
 -<dates>
   -<submitted date="2004-01-27T05:00:00.000-04:00">
      <contributor organization="The MITRE Corporation">Andrew
Buttner</contributor>
    </submitted>
   -<modified comment="modified wrt-222 - changed pattern match"
date="2005-03-07T05:00:00.000-04:00">
     <contributor organization="The MITRE Corporation">Christine
Walzer</contributor>
    </modified>
    <status_change date="2005-03-09T05:00:00.000-
04:00">INTERIM</status_change>
    <status_change date="2005-03-29T05:00:00.000-
04:00">ACCEPTED</status_change>
   -<modified comment="Changed IE registry test to wrt-18"
date="2005-09-20T04:00:00.000-04:00">
       <contributor organization="The MITRE Corporation">Christine
Walzer</contributor>
    </modified>
    <status_change date="2005-09-21T01:27:00.000-
04:00">INTERIM</status_change>
    <status_change date="2005-10-12T05:49:00.000-
04:00">ACCEPTED</status_change>
   -<modified comment="Added negate=true attribute to criteria sub-
block to fix conversion error from OVAL 4.2 to OVAL 5.0" date="2006-
07-03T12:56:00.000-04:00">
     <contributor organization="The MITRE Corporation">Matthew
Wojcik</contributor>
   </modified>
   <status_change date="2006-07-03T12:56:00.000-
04:00">INTERIM</status_change>
   <status_change date="2006-09-27T12:29:41.221-
04:00">ACCEPTED</status_change>
  -<modified comment="Multiple corrections and update to POSIX
compatibility for ste:2878" date="2010-11-29T16:13:00.904-05:00">
     <contributor organization="G2, Inc.">Shane
Shaffer</contributor>
   </modified>
   <status_change date="2010-11-29T16:14:04.414-
05:00">INTERIM</status_change>

    </dates><status>INTERIM</status></oval_repository>
</metadata><criteria comment="Software section"
operator="AND"><criterion comment="the version of mshtml.dll is less
than 6.0.2716.2200" negate="false"
test_ref="oval:org.mitre.oval:tst:3086"/><criterion comment="the
patch q321232 is installed (Installed Components key)" negate="true"
test_ref="oval:org.mitre.oval:tst:3119"/><criterion comment="the
patch q323759 is installed (Installed Components key)" negate="true"
test_ref="oval:org.mitre.oval:tst:3118"/><criterion comment="the
patch q328970 is installed (Installed Components key)" negate="true"
test_ref="oval:org.mitre.oval:tst:3117"/><criterion comment="the
patch q324929 is installed (Installed Components key)" negate="true"
test_ref="oval:org.mitre.oval:tst:3116"/><criterion comment="the
patch q810847 is installed (Installed Components key)" negate="true"
test_ref="oval:org.mitre.oval:tst:3115"/><criterion comment="the
patch q813489 is installed (Installed Components key)" negate="true"
test_ref="oval:org.mitre.oval:tst:3114"/><criterion comment="the
patch q818529 is installed (Installed Components key)" negate="true"
test_ref="oval:org.mitre.oval:tst:3113"/><criterion comment="the
patch q822925 is installed (Installed Components key)" negate="true"
test_ref="oval:org.mitre.oval:tst:3112"/><criterion comment="the
patch q828750 is installed (Installed Components key)" negate="true"
test_ref="oval:org.mitre.oval:tst:3111"/><criterion comment="the
patch q824145 is installed (Installed Components key)" negate="true"
test_ref="oval:org.mitre.oval:tst:3110"/><criteria comment="Windows
2000 Service Pack 4 (or later) is installed" negate="true"
operator="AND"><criterion comment="Windows 2000 is installed"
negate="false" test_ref="oval:org.mitre.oval:tst:3085"/><criterion
comment="SP4 or later Installed" negate="false"
test_ref="oval:org.mitre.oval:tst:3073"/></criteria><criterion
comment="Internet Explorer 6 is installed" negate="false"
test_ref="oval:org.mitre.oval:tst:3090"/></criteria>

>
```

Topological Data Network Interface Description

```
<

 The reachability input is like "hacl(HOST1, HOST2, Protocol,
Port)", where "hacl" means "host access control list".

>
```

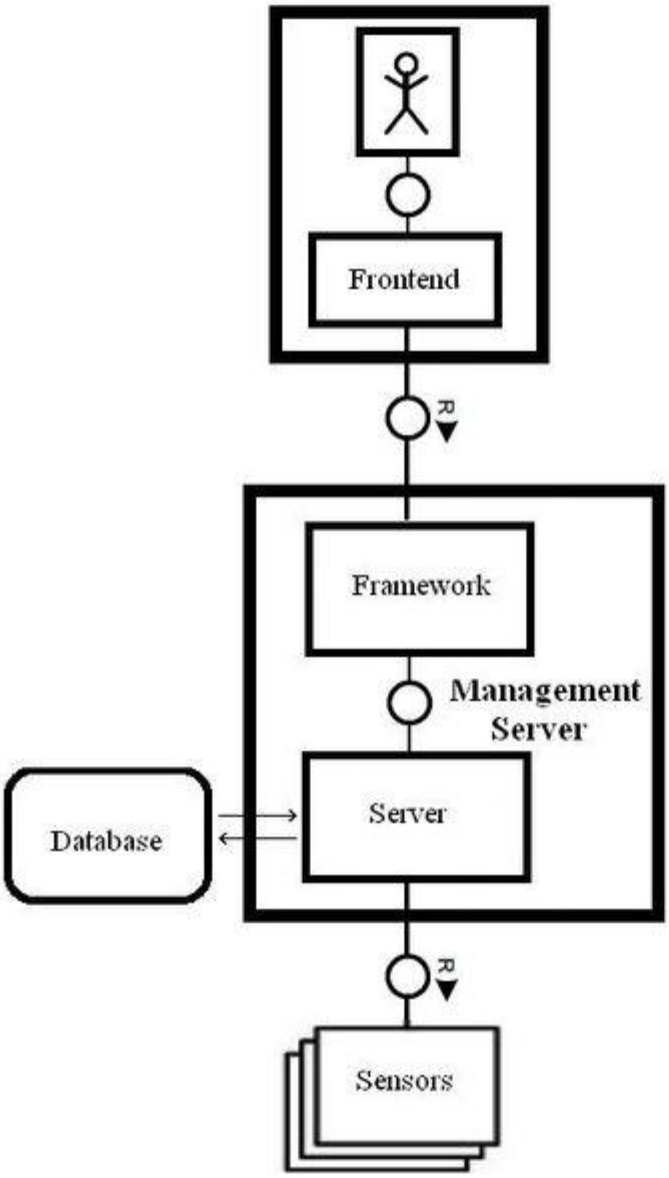
2.8.2 Service Level SIEM

A conventional SIEM deployment is mainly composed of four elements:

1. Sensors: deployed in the networks to monitor network activity. They usually include the low level detectors and monitors that passively collect data looking for patterns

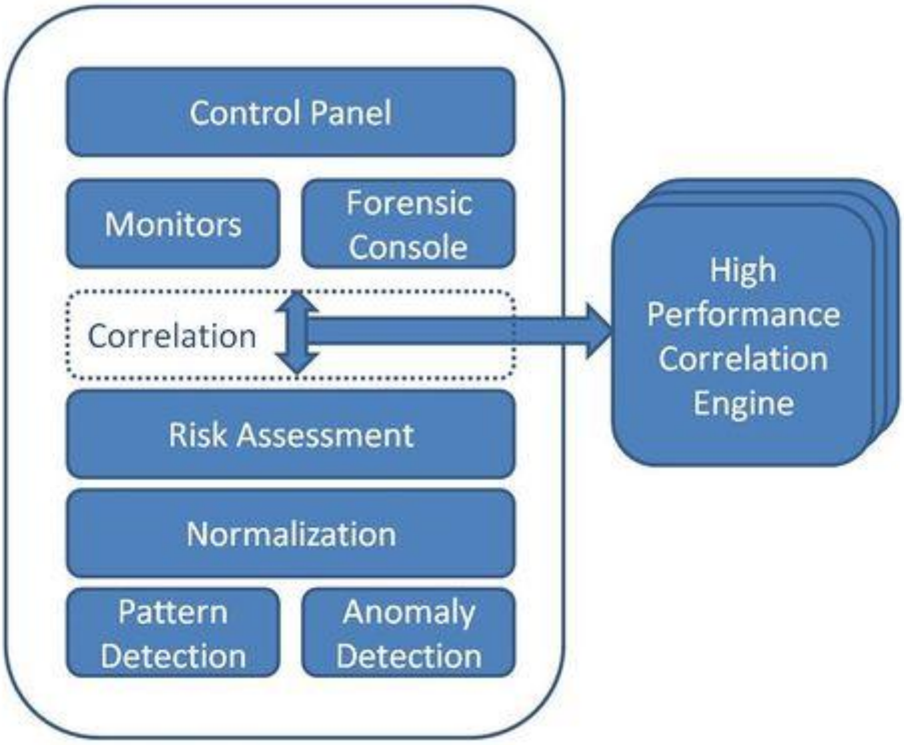
but also, they can include active scanners that try to compile information about node vulnerabilities or agents which could receive data from other hosts of this network.

2. Management Server: this component is in charge of the main processing activities such as normalizing, prioritizing, collecting, risk assessment and correlating engines.
3. Database: where all events and information configuration for the management of the system is stored.
4. Fronted: where the operator can visualize the status of the system and configure the SIEM.



SIEM main elements

From a functional point of view, the OSSIM SIEM stack could be illustrated as showed in the next figure, where also the bypass of the OSSIM correlation engine is depicted.



OSSIM SIEM functional view

2.8.3 Botnet Tracking System

Before the analysis we need to anonymize the IP addresses of the clients in order to preserve their privacy using a reversible hash function. Because some errors can be directly associated to miscon figured softwares, a first step is to filter the error traffic using the following criteria:

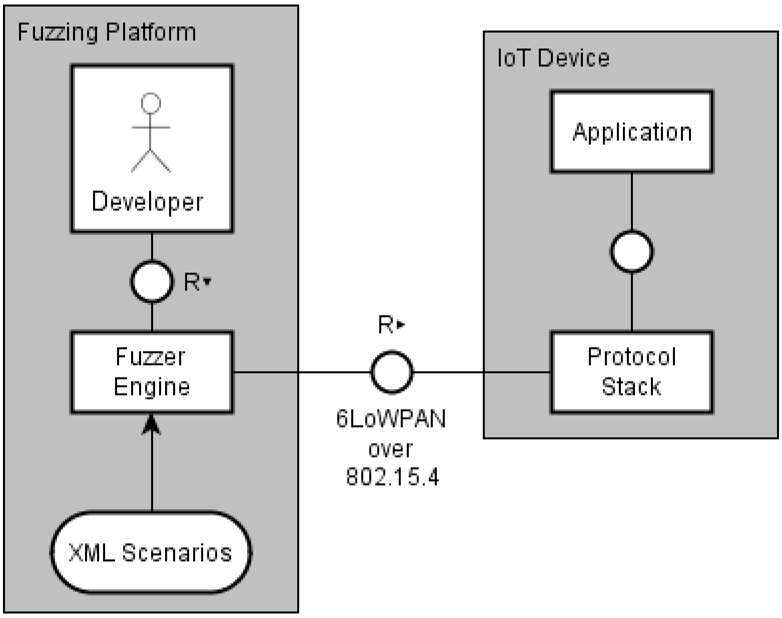
- Only the DNS domain names longer than 6 characters are proceeded, as short domain names have been exhausted since a while by generic web sites and cannot therefore be used for domain flux;
- All the requests made on non-existing Top Level Domain (TLD) like '.home' and '.local' (mostly linked to Apple Bonjour protocol) and '.arpa' (reverse lookup which is rarely implemented) are discarded ; representing the 3rd most popular TLD on the L root server. Such filters are therefore useful more for performance reasons than for algorithm issues. Once the NX error traffic is expurgated from those generic errors, we build up a bipartite graph establishing the relations between failed queries of non existing domains and clients. Such graph allows us to identify communities of users with strong connectivity, i.e. doing similar errors in a short time frame. A cyclic analysis (every 60 seconds) is then made on the identified sub-graphs in order to compute a Malware Probability Factor (MPF) for each erroneous domain.

2.8.4 Fuzzer

The fuzzer communicates with IoT devices, through a 802.15.4 network interface on the fuzzing platform which must be in range of the devices, and must be capable of relaying raw Link Layer frames.

The fuzzer is driven by XML scenarios that define the sequence of packets to be sent, and how the fuzzed system should reply to these packets.

The developer interacting with the fuzzer can then use the provided fuzzing policies, or create new ones, and define how to apply them to the selected scenario, and start fuzzing the device.



2.8.5 Countermeasures

The decision making support will provide the following different interfaces:

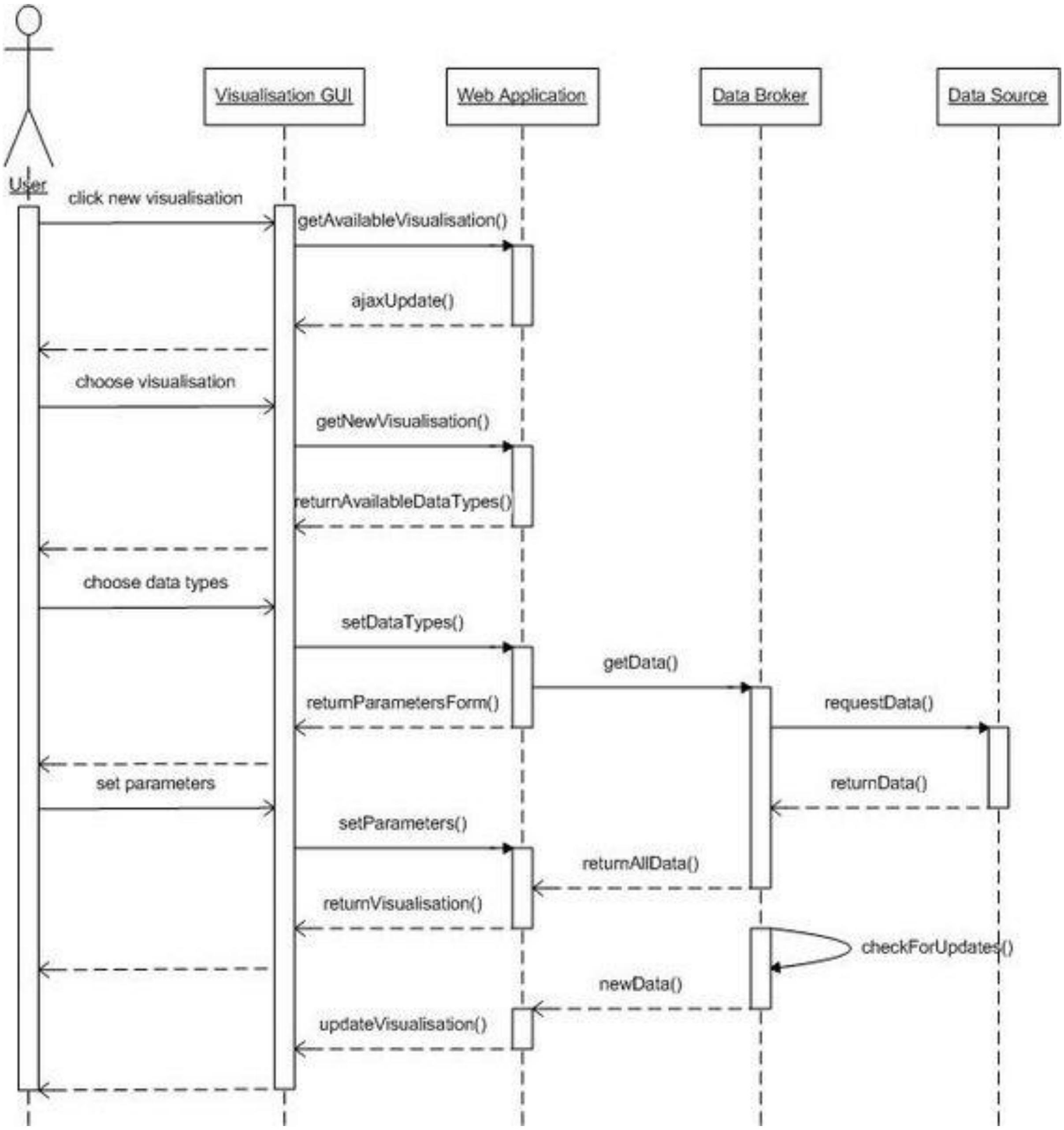
- An internal interface with the ‘Attack Path Engine’ to receive the attack path to be reduced.
- An external interface to send the countermeasure selected by the security operator to the monitored system / services.
- An external interface to collect the Security Advisories
- A GUI for the previously mentioned selection.

2.8.6 Visualization framework

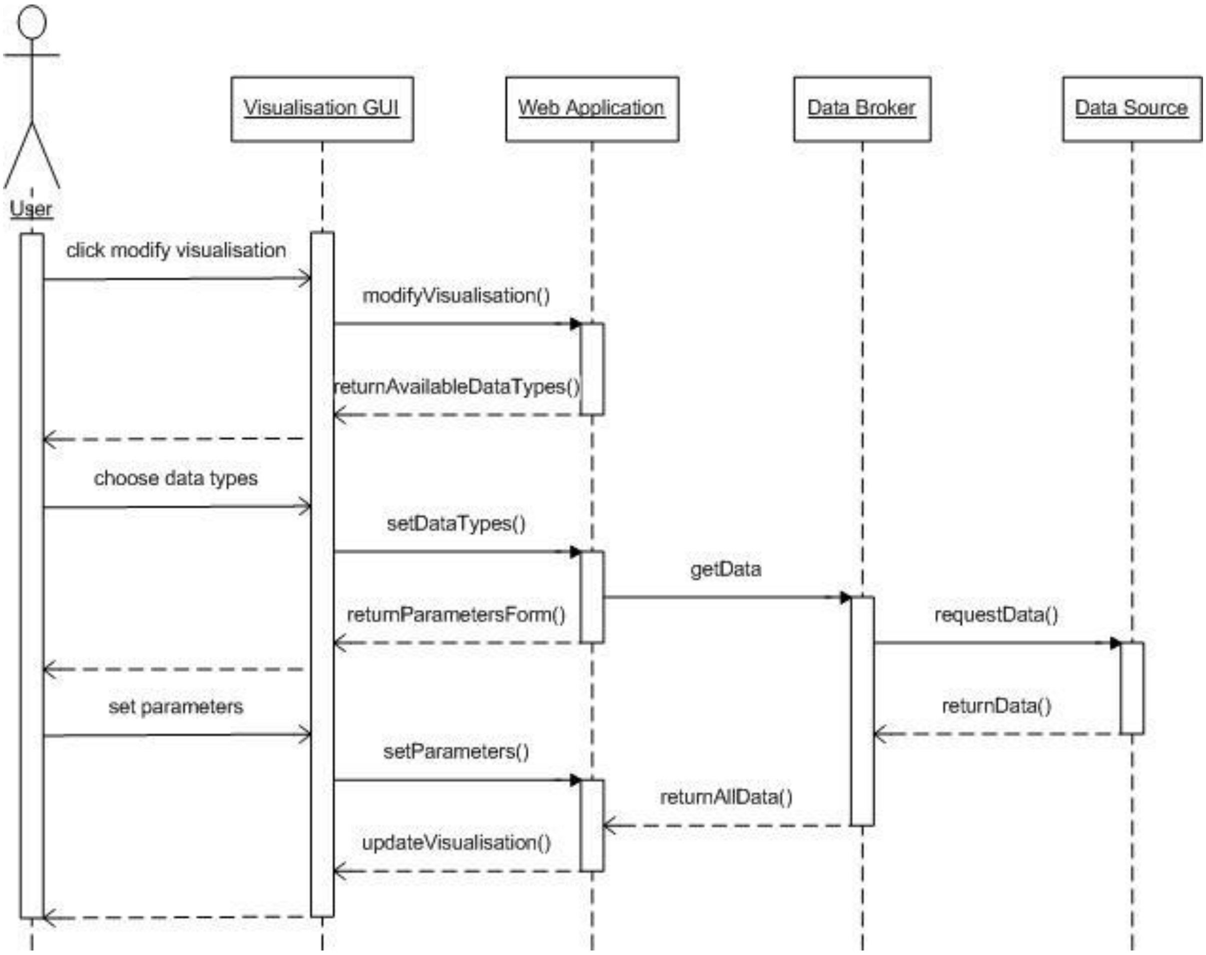
The Visualisation Framework offers a visualisation service that allows users to visualise data from multiple network components. The user accesses the visualisation service through a standard web-browser connected to the web-application server using some network connection (such as the Internet). The user will experience a single integrated application showing multiple visualisations. Behind the scenes, the browser will compliment the information from the visualisation server with data and functionality directly from the Internet.

Users of the framework will follow a similar pattern of creating, interacting with, modifying and eventually removing visualisations. There are therefore the three main interactions between users and the Visualisation Framework: adding a new visualisation; modifying an existing visualisation; removing a visualisation.

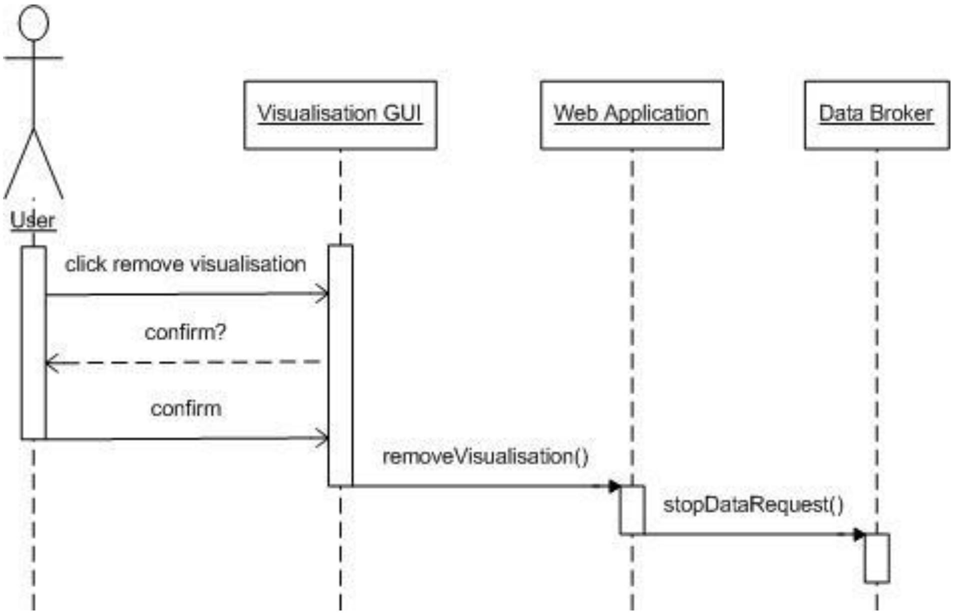
Add new visualisation enables a user to view a new visualisation. The user selects the visualisation and data type from a list of available options. External visualisations that support the existing data formats can also be added. The user can customise the visualisation, e.g. by choosing the size of the window. A sequence diagram for the interaction is shown below.



Modify visualisation enables a user to modify an existing visualisation. The user can change the type of data displayed, the size of the window, how often the visualisation is updated. The interactions for modifying a visualisation are shown in the sequence diagram below.



Remove visualisation enables a user to remove a window containing a visualisation from the display. A sequence diagram for the interaction shown in the following figure.



2.9 Detailed Specifications

Following is a list of Open Specifications linked to this Generic Enabler. Specifications labeled as "PRELIMINARY" are considered stable but subject to minor changes derived from lessons learned during last interactions of the development of a first reference implementation planned for the current Major Release of FI-WARE. Specifications labeled as "DRAFT" are planned for future Major Releases of FI-WARE but they are provided for the sake of future users.

2.9.1.1 *Open API Specifications*

Security Monitoring Generic Enabler APIs are under construction. The following initial functionalities will be available in September 2012:

- Attack graph engine conducting multi-host, multistage vulnerability analysis on a network and showing what attacks are possible. This functionality is included in the Risk Analysis Service
  - Security-Monitoring Mulval Attack Path Engine API Specification (PRELIMINARY)
- Security Information and Event Management tool based on a set of defined assets such as hosts, networks, groups and services.This functionality is included in the Event Correlation Service.
  - Security.Security-Monitoring Service Level SIEM API Specification (PRELIMINARY)

2.10 Terms and definitions

This section comprises a summary of terms and definitions introduced during the previous sections. It intends to establish a vocabulary that will be help to carry out discussions internally and with third parties (e.g., Use Case projects in the EU FP7 Future Internet PPP).

For a summary of terms and definitions managed at overall FI-WARE level, please refer to FIWARE Global Terms and Definitions

- **CVE.** Common Vulnerabilities and Exposures is a dictionary of publicly known information about security vulnerabilities and exposures.
- **Event.** A software message indicating an observable or an extraordinary occurrence.
- **IDS.** Intrusion Detection System.
- **Sensor.** Devices deployed to monitor network activity. They usually include the low level detectors and monitors that passively collect data but they can also include active scanners.
- **SIEM.** Security Information and Event Management is a technology that provides real-time analysis of security alerts. It aggregates data from many sources, providing the ability to consolidate monitored data to notify immediates issues.

3 Security-Monitoring Mulval Attack Path Engine API Specification (PRELIMINARY)

You can find the content of this chapter as well in the wiki of fi-ware.

3.1 Introduction to the Mulval Attack Path Engine API

3.1.1 Mulval Attack Path Engine API Core

This document provides a description of the available interface and presents adapters used by attack path engine to import data file. The adapter transforms the data file to internal data in order to make reporting and decision support in the context of security monitoring G.E.

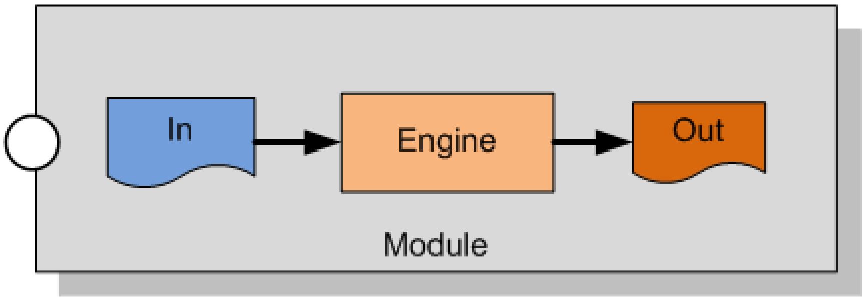


Figure 1: principle of Mulval API The Mulval API can be seen such as a module. This module needs input in order to be processed and compute the results with certain available options. We can sum as below Input: file required by the engine Engine: offers certain flexibility (options) of attack path computation Output: data file which can be consumed by the reporting, visualization and decision support components.

3.1.2 Intended Audience

This document is addressed both software developers and the consumers of attack path engine.

3.1.3 API Change History

This version of the Mulval Attack Path API Guide replaces and obsoletes all previous versions. The most recent changes are described in the table below:

Revision Date	Changes Summary
August, 2012	• V1.0, first release
Janauary, 2012	• V1.1 release • Nessus scanner supported

	• Attack path generated from the file exported by the Nessus scanner.
...	• ...

3.1.4 How to Read This Document

Along the document, some special notations are applied to differentiate some special words or concepts. The following list summarizes these special notations:

- A bold, mono-spaced font is used to represent a module.
- An italic font is used to represent an example

3.1.5 Additional Resources

The attack path engine is identified such as an innovative way to assess the security risk. The API is still provided in summary version. In the case of providing this component to the FI-WARE community, we explicit the API for the first time. For the moment, any reference of the API cannot be found. Fortunately, many publications on the attack path can be quoted by these following links:

(all references can found here)

- MulVAL: A logic-based network security analyzer. Xinming Ou, Sudhakar Govindavajhala, and Andrew W. Appel. In 14th USENIX Security Symposium, Baltimore, Maryland, U.S.A., August 2005.
- A logic-programming approach to network security analysis. Xinming Ou. PhD dissertation, Princeton University, 2005.
- A scalable approach to attack graph generation. Xinming Ou, Wayne F. Boyer, and Miles A. McQueen. In 13th ACM Conference on Computer and Communications Security (CCS 2006), Alexandria, VA, U.S.A., October 2006.
- Googling attack graphs. Reginald Sawilla and Xinming Ou. Technical report, Defence R & D Canada -- Ottawa. TM 2007-205, September 2007.
- From attack graphs to automated configuration management - an iterative approach. John Homer, Xinming Ou, and Miles A. McQueen. Technical report, Kansas State University, Computing and Information Sciences Department. January 2008.
- Improving attack graph visualization through data reduction and attack grouping. John Homer, Ashok Varikuti, Xinming Ou, and Miles A. McQueen. In 5th International Workshop on Visualization for Cyber Security (VizSEC 2008), Cambridge, MA, U.S.A., September 2008.
- Identifying critical attack assets in dependency attack graphs. Reginald Sawilla and Xinming Ou. In 13th European Symposium on Research in Computer Security (ESORICS 2008), Malaga, Spain, October 2008. The extended version.
- SAT-solving approaches to context-aware enterprise network security management. John Homer and Xinming Ou, In IEEE JSAC Special Issue on Network Infrastructure Configuration, Vol. 27, No. 3, April 2009. Preprint
-

  Techniques for enterprise network security metrics. Anoop Singhal and Xinming Ou. Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies (CSIIRW) , Extended Abstract, April, 2009.

- A host-based security assessment architecture for industrial control systems. Abhishek Rakshit and Xinming Ou. 2nd International Symposium on Resilient Control Systems (ISRCS), Idaho Falls, ID, USA, August 2009.
- A sound and practical approach to quantifying security risk in enterprise networks. John Homer, Xinming Ou, and David Schmidt. Technical report, Kansas State University, Computing and Information Sciences Department. August 2009.
- Uncertainty and risk management in cyber situational awareness. Jason Li, Xinming Ou, and Raj Rajagopalan. In Sushil Jajodia et al., editor, Cyber Situational Awareness: Issues and Research , chapter 4. Springer, Nov. 2009.
- An empirical approach to modeling uncertainty in intrusion analysis. Xinming Ou, S. Raj Rajagopalan, and Sakthiyuvaraja Sakthivelmurugan. Annual Computer Security Applications Conference (ACSAC), Honolulu, Hawaii, USA, Dec 2009.

3.2 General Mulval Attack Path API Information

The Mulval Attack Path engine is an orchestration of chained modules. A module can be an adapter, core attack graph computation, attack path visualization or metrics analysis.

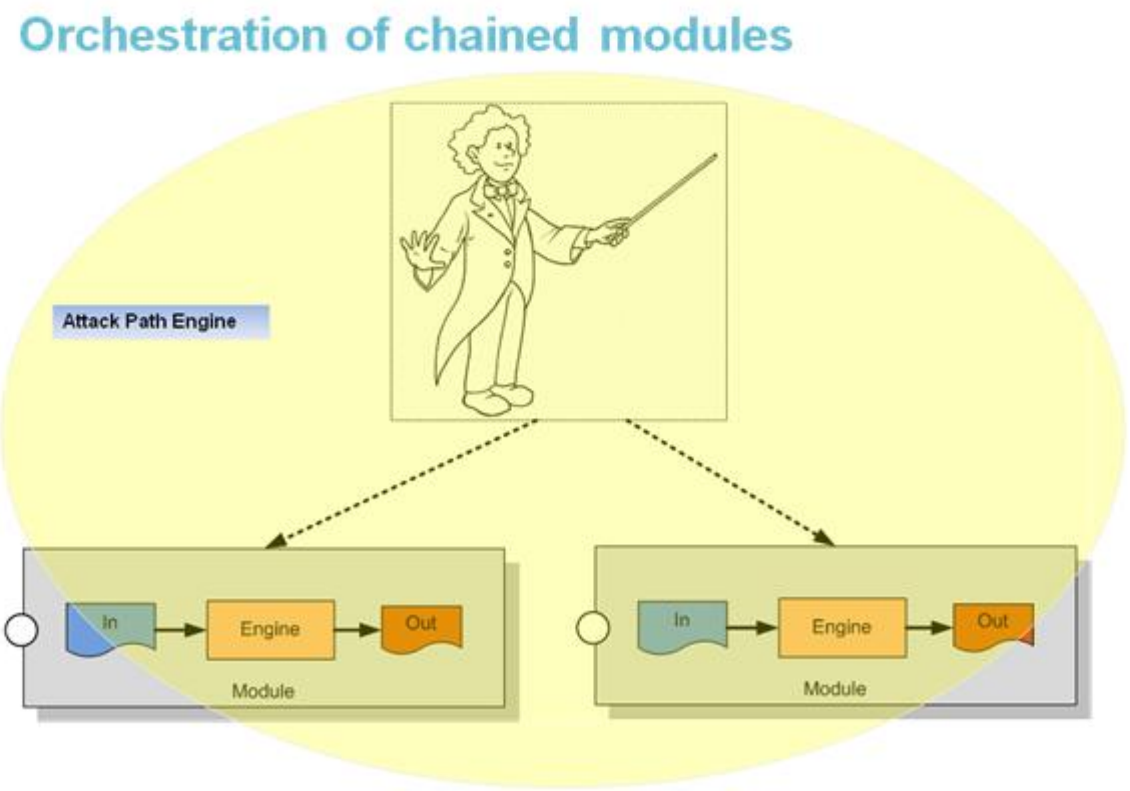


Figure: Orchestration of chained modules.

The attack path engine is composed of four modules:

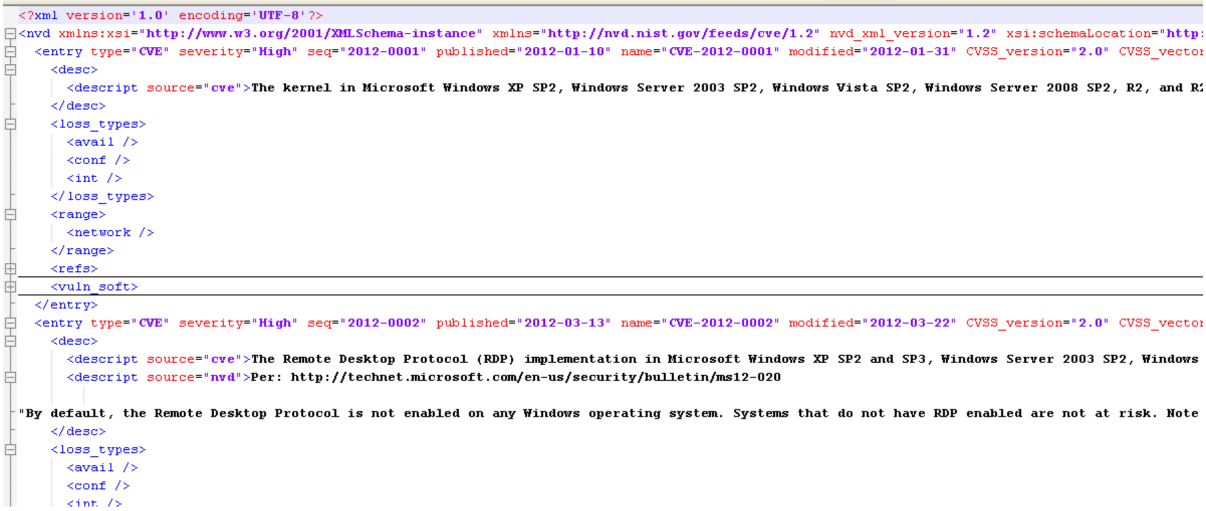
- **1. Adapters**
- **2. Core Attack Graph Computation**
- **3. Metrics analysis**
- **4. Attack Path visualization**

3.2.1 Adapters

What are adapters? The adapters convert / transform the input data files to the internal information which is required by the engine. Regarding of the interface attack path engine in input, the input data files are: NVD database Vulnerability scanners (OVAL and NESSUS)

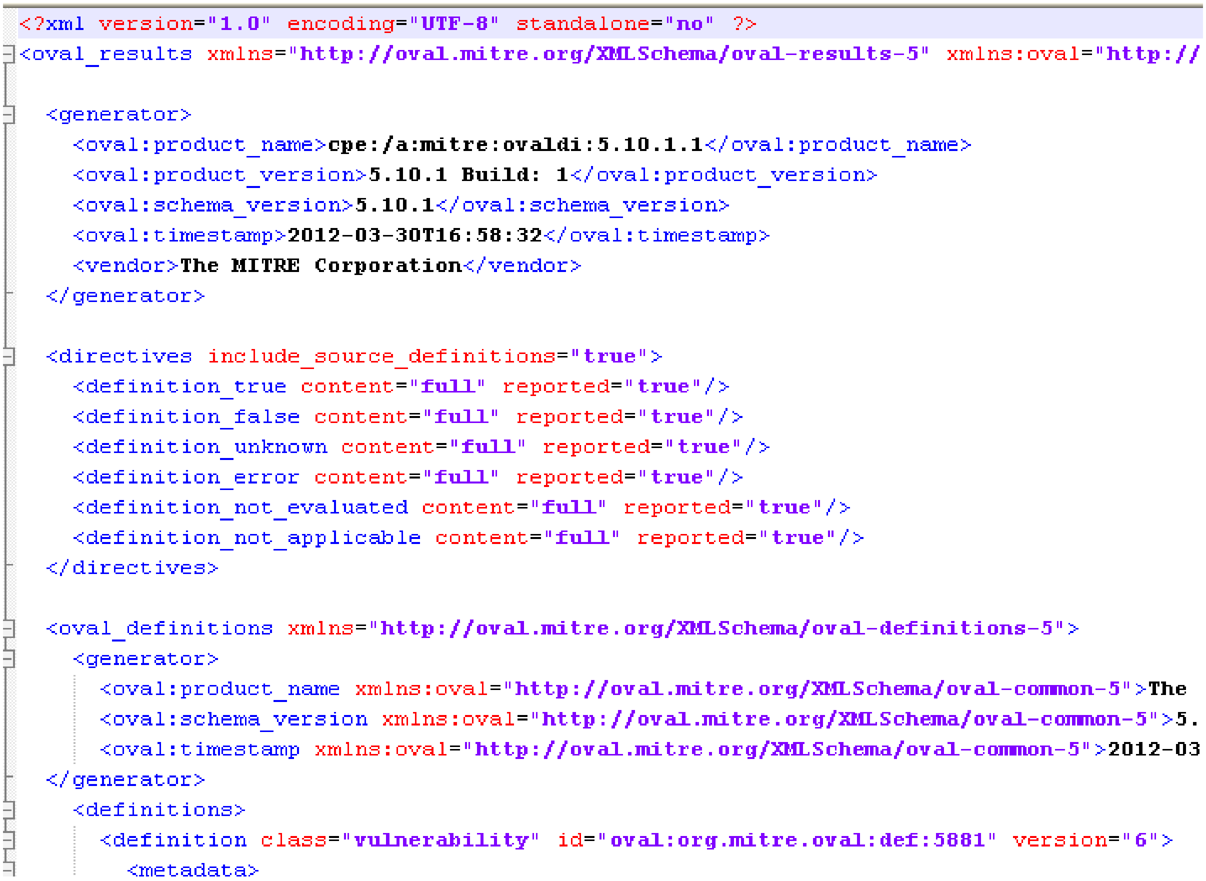
NVD database can be getting directly from the NIST. After getting these XML files. The adapter parsers these files and stored these to the local MySQL database.

NVD Example



```
<?xml version='1.0' encoding='UTF-8'?>
<nvd xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://nvd.nist.gov/feeds/cve/1.2" nvd_xml_version="1.2" xsi:schemaLocation="http:
  <entry type="CVE" severity="High" seq="2012-0001" published="2012-01-10" name="CVE-2012-0001" modified="2012-01-31" CVSS_version="2.0" CVSS_vecto
    <desc>
      <descript source="cve">The kernel in Microsoft Windows XP SP2, Windows Server 2003 SP2, Windows Vista SP2, Windows Server 2008 SP2, R2, and R
    </desc>
    <loss_types>
      <avail />
      <conf />
      <int />
    </loss_types>
    <range>
      <network />
    </range>
    <refs>
    <vuln_soft>
  </entry>
  <entry type="CVE" severity="High" seq="2012-0002" published="2012-03-13" name="CVE-2012-0002" modified="2012-03-22" CVSS_version="2.0" CVSS_vecto
    <desc>
      <descript source="cve">The Remote Desktop Protocol (RDP) implementation in Microsoft Windows XP SP2 and SP3, Windows Server 2003 SP2, Windows
      <descript source="nvd">Per: http://technet.microsoft.com/en-us/security/bulletin/ms12-020

"By default, the Remote Desktop Protocol is not enabled on any Windows operating system. Systems that do not have RDP enabled are not at risk. Note
    </desc>
    <loss_types>
      <avail />
      <conf />
      <int />
```

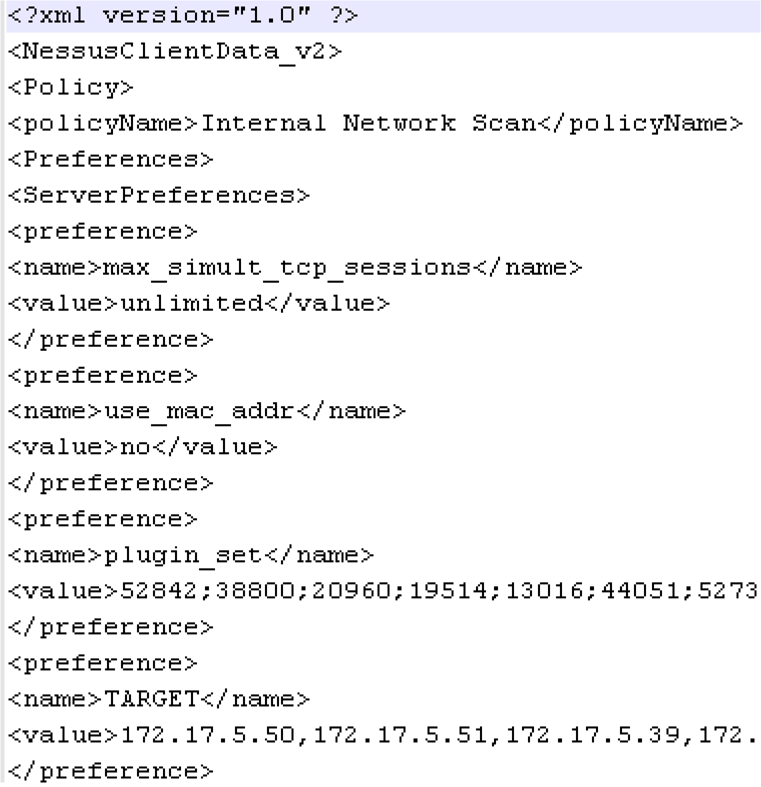
OVAL result is obtained by using an OVAL scanner. In our case, we use the OVAL interpreter which can be downloaded at “oval interpreter lien”. The OVAL Interpreter scan the “vulnerable host” and provides a result on xml file. Example of OVAL result:



```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<oval_results xmlns="http://oval.mitre.org/XMLSchema/oval-results-5" xmlns:oval="http://

  <generator>
    <oval:product_name>cpe:/a:mitre:ovaldi:5.10.1.1</oval:product_name>
    <oval:product_version>5.10.1 Build: 1</oval:product_version>
    <oval:schema_version>5.10.1</oval:schema_version>
    <oval:timestamp>2012-03-30T16:58:32</oval:timestamp>
    <vendor>The MITRE Corporation</vendor>
  </generator>

  <directives include_source_definitions="true">
    <definition_true content="full" reported="true"/>
    <definition_false content="full" reported="true"/>
    <definition_unknown content="full" reported="true"/>
    <definition_error content="full" reported="true"/>
    <definition_not_evaluated content="full" reported="true"/>
    <definition_not_applicable content="full" reported="true"/>
  </directives>

  <oval_definitions xmlns="http://oval.mitre.org/XMLSchema/oval-definitions-5">
    <generator>
      <oval:product_name xmlns:oval="http://oval.mitre.org/XMLSchema/oval-common-5">The
      <oval:schema_version xmlns:oval="http://oval.mitre.org/XMLSchema/oval-common-5">5.
      <oval:timestamp xmlns:oval="http://oval.mitre.org/XMLSchema/oval-common-5">2012-03
  </generator>
    <definitions>
      <definition class="vulnerability" id="oval:org.mitre.oval:def:5881" version="6">
        <metadata>
```

NESSUS result is obtained by using NESSUS scanner. The NESSUS scanner scan the set of IP addresses and offers an export option in order to export the result to xml format.

Example of NESSUS result



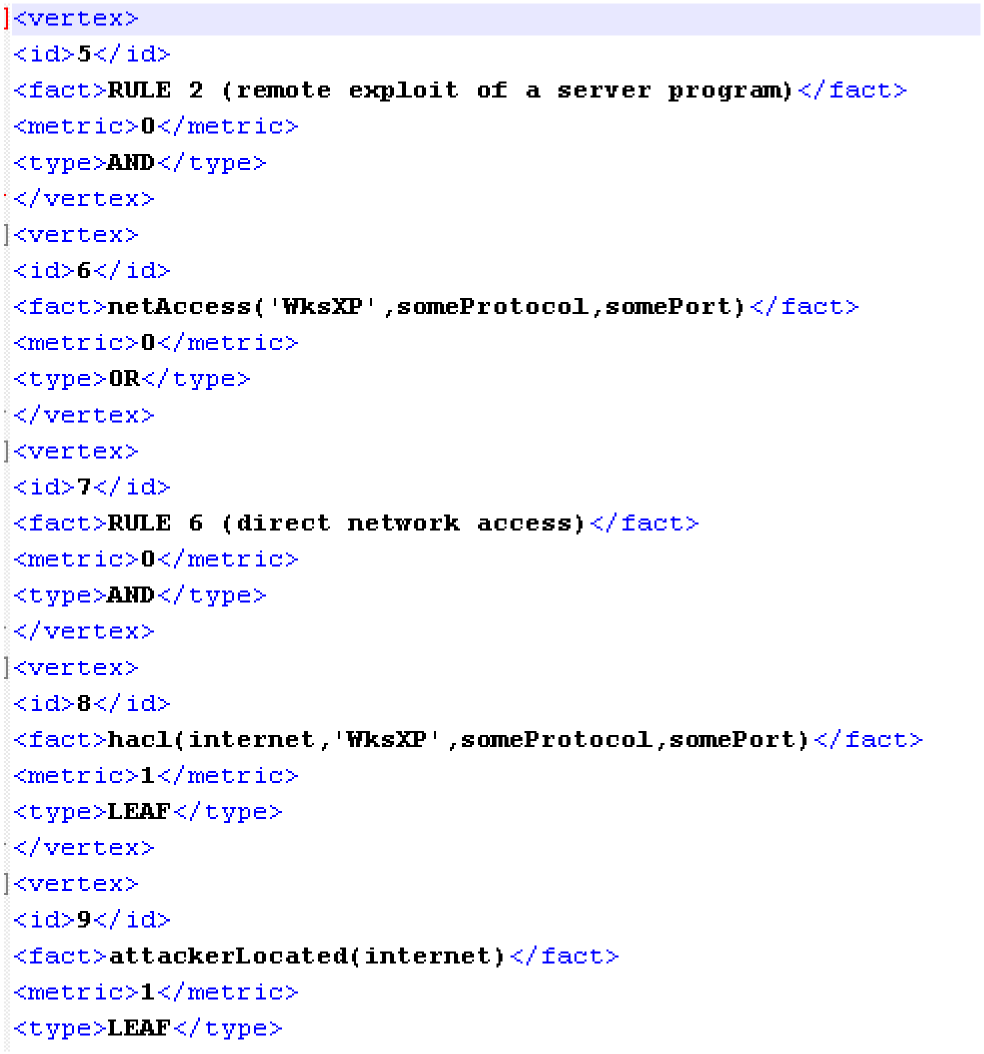
```
<?xml version="1.0" ?>
<NessusClientData_v2>
<Policy>
<policyName>Internal Network Scan</policyName>
<Preferences>
<ServerPreferences>
<preference>
<name>max_simult_tcp_sessions</name>
<value>unlimited</value>
</preference>
<preference>
<name>use_mac_addr</name>
<value>no</value>
</preference>
<preference>
<name>plugin_set</name>
<value>52842;38800;20960;19514;13016;44051;5273
</preference>
<preference>
<name>TARGET</name>
<value>172.17.5.50,172.17.5.51,172.17.5.39,172.
</preference>
```

3.2.2 Core Attack Graph Computation

This module is the core computation. It uses the input files (OVAL or NESSUS scanner) transformed previously by the adapters and combines these input files with the local MySQL database in order to get more information about the vulnerability. The core computation is handled with the ProLog rules. This is rules engine which define conditions of generation of attack graph.

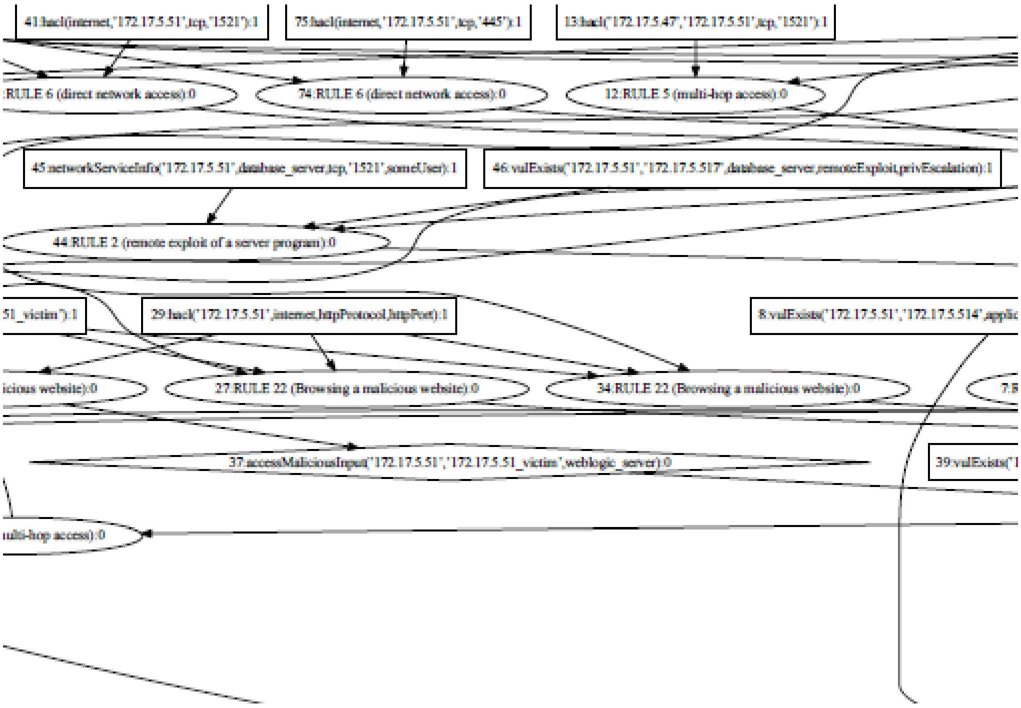
3.2.3 Attack Path Visualization

The attack path visualization is the result of the core computation which can be rendered under different formats: XML, PDF, text file *Example of XML Format*

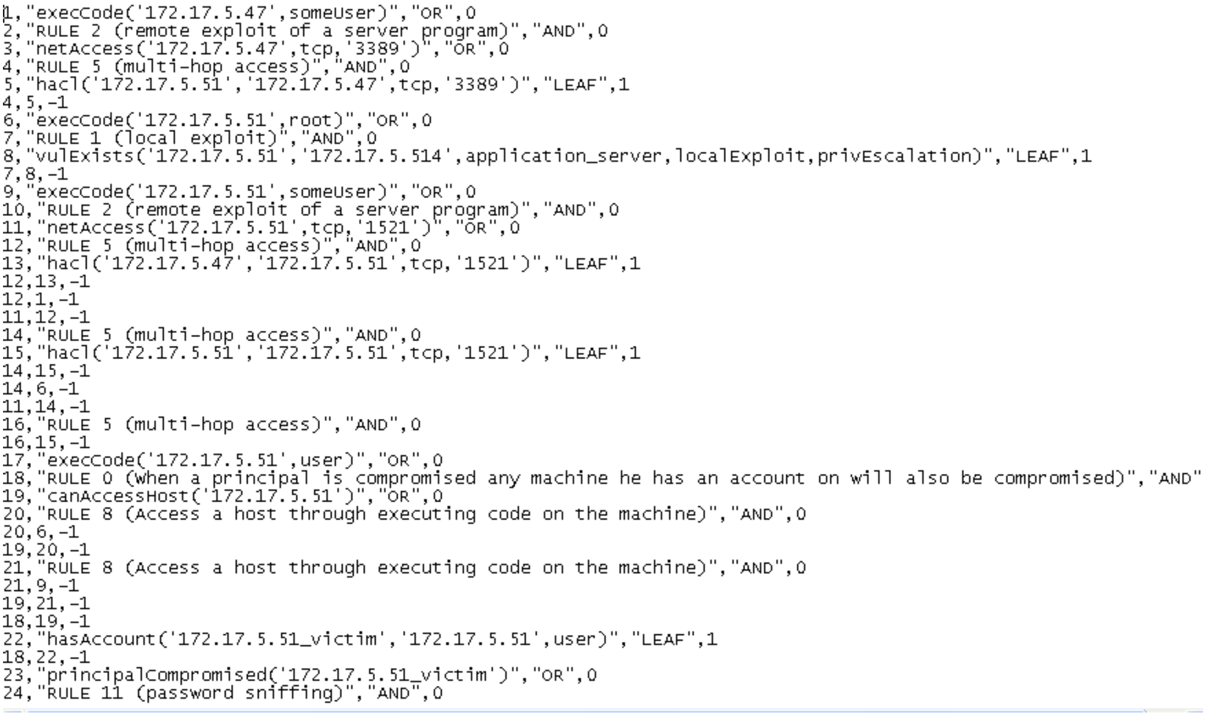


```
<vertex>
<id>5</id>
<fact>RULE 2 (remote exploit of a server program)</fact>
<metric>0</metric>
<type>AND</type>
</vertex>
<vertex>
<id>6</id>
<fact>netAccess('WksXP',someProtocol,somePort)</fact>
<metric>0</metric>
<type>OR</type>
</vertex>
<vertex>
<id>7</id>
<fact>RULE 6 (direct network access)</fact>
<metric>0</metric>
<type>AND</type>
</vertex>
<vertex>
<id>8</id>
<fact>hacl(internet,'WksXP',someProtocol,somePort)</fact>
<metric>1</metric>
<type>LEAF</type>
</vertex>
<vertex>
<id>9</id>
<fact>attackerLocated(internet)</fact>
<metric>1</metric>
<type>LEAF</type>
```

Example of PDF format


41:hacl(internet,'172.17.5.51',tcp,'1521'):1
75:hacl(internet,'172.17.5.51',tcp,'445'):1
13:hacl('172.17.5.47','172.17.5.51',tcp,'1521'):1
:RULE 6 (direct network access):0
74:RULE 6 (direct network access):0
12:RULE 5 (multi-hop access):0
45:networkServiceInfo('172.17.5.51',database_server,tcp,'1521',someUser):1
46:vulExists('172.17.5.51','172.17.5.517',database_server,remoteExploit,privEscalation):1
44:RULE 2 (remote exploit of a server program):0
51_victim'):1
29:hacl('172.17.5.51',internet,httpProtocol,httpPort):1
8:vulExists('172.17.5.51','172.17.5.514',applic
icious website):0
27:RULE 22 (Browsing a malicious website):0
34:RULE 22 (Browsing a malicious website):0
7:R
37:accessMaliciousInput('172.17.5.51','172.17.5.51_victim',weblogic_server):0
39:vulExists('1
ulti-hop access):0

Example of Text format



```
1,"execCode('172.17.5.47',someUser)","OR",0
2,"RULE 2 (remote exploit of a server program)","AND",0
3,"netAccess('172.17.5.47',tcp,'3389')","OR",0
4,"RULE 5 (multi-hop access)","AND",0
5,"hacl('172.17.5.51','172.17.5.47',tcp,'3389')","LEAF",1
4,5,-1
6,"execCode('172.17.5.51',root)","OR",0
7,"RULE 1 (local exploit)","AND",0
8,"vulExists('172.17.5.51','172.17.5.514',application_server,localExploit,privEscalation)","LEAF",1
7,8,-1
9,"execCode('172.17.5.51',someUser)","OR",0
10,"RULE 2 (remote exploit of a server program)","AND",0
11,"netAccess('172.17.5.51',tcp,'1521')","OR",0
12,"RULE 5 (multi-hop access)","AND",0
13,"hacl('172.17.5.47','172.17.5.51',tcp,'1521')","LEAF",1
12,13,-1
12,1,-1
11,12,-1
14,"RULE 5 (multi-hop access)","AND",0
15,"hacl('172.17.5.51','172.17.5.51',tcp,'1521')","LEAF",1
14,15,-1
14,6,-1
11,14,-1
16,"RULE 5 (multi-hop access)","AND",0
16,15,-1
17,"execCode('172.17.5.51',user)","OR",0
18,"RULE 0 (when a principal is compromised any machine he has an account on will also be compromised)","AND"
19,"canAccessHost('172.17.5.51')","OR",0
20,"RULE 8 (Access a host through executing code on the machine)","AND",0
20,6,-1
19,20,-1
21,"RULE 8 (Access a host through executing code on the machine)","AND",0
21,9,-1
19,21,-1
18,19,-1
22,"hasAccount('172.17.5.51_victim','172.17.5.51',user)","LEAF",1
18,22,-1
23,"principalCompromised('172.17.5.51_victim')","OR",0
24,"RULE 11 (password sniffing)","AND",0
```

3.2.4 Metrics Analysis

The metrics analysis uses the CVSS scoring. This score is contained in each vulnerability definition. We have included a quantitative risk assessment algorithm. It combines the CVSS metrics and the attack graph to compute a probabilistic risk metrics for the enterprise network.

3.2.5 Resources Summary

The API is only available here.

3.2.6 Authentication

The usage of Attack Path Engine don't require the inclusion of specific authentication credentials.

The credentials are needed when accessing to the VMWare CentOS where the attack path engine is installed.

3.2.7 Representation Format

The Attack Path Engine supports the XML serialisation. The request and the response format are specified using the XML Content-Type header.

3.2.8 Representation Transport

We can imagine file exchange in order to put the input in a folder.

But presently, it only needs to put directly the files in the correct foler and execute the CLI (Commande Line Interface).

3.2.9 Resource Identification

N.A

3.2.10 Links and References

Report to "Additional Resources" for references.

3.2.11 Paginated Collections (Optional)

N.A

3.2.12 Efficient Polling with the Changes-Since Parameter (Optional)

In this case we can specify the parameter changes-since in a GET method in order that the response will give us only the changed information from the previous request specified through a dateTime format ISO 8601 (2011-01-24T17:08Z).

3.2.13 Limits

N.A

3.2.13.1 *Attack Graph Engine Limits*

Under test.

3.2.13.2 *Absolute Limits*

Under test.

3.2.13.3 *Determining Limits Programmatically*

The Attack Graph Engine is provided such as a black box. This kind of limits depends of the maturity of the tool.

We expect the answer from the developper.

The limits in our possession are to use the CLI to execute the Attack Graph Engine.

3.2.14 Versions

We are V1.1 release.

3.2.15 Extensions

No extension is forecasted for the moment.

3.2.16 Faults

The faults are saved in log files and also indicating in line when executing.

3.3 API Operations

In this section we go in depth for each operation. In order to provide good comprehensive of the API operations, we would suggest to group them into similar functionalities within subsections. e.g. operations related to attack path engine.

3.3.1 <name of subsections | group of similar functionalities>

3.3.1.1 *<Operation>*

Please, position the shell in the right folder. /opt/mulval_v1.1/bin

And execute the CLI >oval_translate.sh XML_REPORT_FROM_IN_OVAL (version NESSUS V2 exported from NEUSS scanner)

And execute the CLI for the attack paph engine

>graph_gen.sh input.P -v -p

4 Security Security-Monitoring OSSIM SIEM API Specification (PRELIMINARY)

You can find the content of this chapter as well in the wiki of fi-ware.

4.1 Introduction to the OSSIM SIEM Open Specifications

Please check the FI-WARE Open Specifications Legal Notice to understand the rights to use FI-WARE Open Specifications.

4.1.1 OSSIM SIEM API Core

A Security Information and Event Management (SIEM) solution is a technology that provides real-time analysis of security events, aggregating data from many sources and providing the ability to consolidate and correlated monitored data to generate reports and alerts. OSSIM (Open Source Security Information Management - http://www.ossim.net), developed and maintained by AlientVault (http://www.alientvault.com), is one of the most widely used Open Source SIEM.

The OSSIM SIEM Component provided for FI-WARE first release is an Atos preconfigured version of the open source OSSIM SIEM v4.0. These modules will be the core of the advanced **Service Level SIEM Component** (currently under development by Atos) that is going to be delivered on future releases of the FI-WARE **Security Monitoring GE**. Consequently, the Open Specification described here for the OSSIM-SIEM Component included in this FI-WARE first release is the same specification offered by OSSIM.

This page provides a description of the available OSSIM SIEM specifications about its collection functionality (collection methods and plugins) and the event types that will be necessary to gather the information to be processed by the future Service-Level SIEM component in the context of the FI-WARE architecture. The normalized events will be sent to the OSSIM SIEM by the **Heterogeneous event normalization service** component also included in the Security Monitoring GE.

4.1.2 Intended Audience

This specification is addressed for both software developers and service providers that will need advanced monitoring features in their environments.

The Security Monitoring GE will include a Service Level SIEM component based on the open source OSSIM SIEM (Security Information and Event management) that will overcome its limitations with a high performance correlation engine. In this first release only the specifications of the OSSIM core that will be required to receive events in the future Service Level SIEM, are included.

4.1.3 API Change History

The most recent changes are described in the table below:

Revision Date	Changes Summary

April 2012	• Initial version
September 2012	• Reviewed initial version

4.1.4 How to Read This Document

The following list summarizes these special notations.

- A bold, mono-spaced font is used to represent code or logical entities, e.g., HTTP method (GET, PUT, POST, DELETE).
- An italic font is used to represent document titles or some other kind of special text, e.g., URI.
- The variables are represented between brackets, e.g. {id} and in italic font. When the reader find it, can change it by any value.

For a description of some terms used along this document, see the Architecture Description document.

4.1.5 Additional Resources

Additional information about OSSIM SIEM open source solution can be found on the official OSSIM - AlienVault Technical Documentation Web Page:

http://communities.alienvault.com/community/documentation.html

Additional information about how to create your own complex rules and plugins to detect attacks can be found in the Service Level SIEM User and Programmer Guide:

https://forge.fi-ware.eu/plugins/mediawiki/wiki/fiware/index.php/Service_Level_SIEM_-_User_and_Programmer_Guide

4.2 General *OSSIM SIEM* Specification Information

Collection is one of the first things required in a SIEM. Data collection is firstly done as a result of the Agents installed in Sensors. Each server can receive data from different sources, but data collection is only possible from:

- Agents: These are the main sources for incoming events.
- Other Server: This is only possible in multi-level architecture.

In the FI-WARE context, the OSSIM SIEM core will receive events coming from the **Heterogeneous event normalization service** component included in the Security Monitoring GE. Consequently. Our main goal here is to describe the specification of the OSSIM agents which collect the incoming events to be proccessed by the SIEM.

4.2.1 Agents

The OSSIM agents are the components responsible for collecting all the data sent by the various devices existing on the network, in order to subsequently send it to the OSSIM server in a standardized way.

The agents are installed on the sensor machines, normally one per machine although it is possible to install more than one if necessary. This will normally only occur in multi-level environments, where one machine with several agents can be sending information to various different servers, each from different devices.

The way in which the agent receives the data (which will then be converted into events for OSSIM) that it is going to send to the server is by means of reading a log file in most cases. The ports to which the agent is connected are:

Port number	Use
40001	Normally the port of the OSSIM server to which they are connected
3306	DB port to which it is connected for monitor requests

Each of the events received by the OSSIM server has always been processed beforehand by an agent in order to standardize them. The point of standardizing events prior to sending them to the server is so that the latter can deal with these events equally and so that storage and processing is simpler and more coherent.

For any device from which one wishes to collect data a **plugin** has to have been created beforehand so that OSSIM is capable of processing it. This is achieved thanks to the creation of a plugin which basically consists of a series of regular expressions and a list that allows the event type being produced to be unambiguously identified, including Reliability evaluations.

4.2.2 Plugins

Plugins are each of the elements defined in the Agent to analyze and standardize the information from a device. Once this has been standardized it is passed to the remaining functionalities of the Agent in order to be sent to the OSSIM server in the form of an event.

In OSSIM there are two types of plugins:

- **Detectors**: Their job is to read from the logs that store the devices and to standardize them so that the Agent can send them to the OSSIM server. Detector plugins passively read a file, socket or process and send events upon pattern-matching lines.
- **Monitors**: These plugins will receive a question from the OSSIM server and send it to the corresponding tool; then as they obtain the reply let the server know whether it agrees or not with what it has asked. Sample monitor plugins would be:
    - Nmap: It receives a monitor request, launches an nmap scan against a specific host:port pair and returns a message stating the open/filtered/closed status of the requested pair.
    - OSSIM C/A: After receiving a Compromise or Attack status request the agent watches for those values inside the OSSIM database, returning and event after having reached it or after the timeout expires.
    - Tcptrack: OSSIM Server asks these agents for specific TCP session information such as duration and bytes sent/received.

In general, each of the plugins can read and send data from a specific device identified by its *plugin_id* and each event type belonging to that plugin is identified as its *plugin_sid*.

The plugins consist of two basic files, one with its configuration, and another with the information that the OSSIM server needs in order to correlate the events subsequently. In order to create a new plugin, it will only be necessary to create these two files as specified in the documentation. One of the most important parts is to create the regular expression which must correspond to the one in the log file of the device for which we are creating the plugin. Both the server and the plugin have to agree on what each *plugin_id* and *plugin_sid* of each event means; both files are therefore inseparable and it is essential to have both in order for the plugin to work.

4.3 OSSIM SIEM Collection Methods

There are several ways of collecting information in OSSIM and it is important to know which ones will be used in order to configure the agents and the plugins required to proccess the incoming data. The most common ways are:

- **Syslog**

  The device from which the logs wish to be extracted can inject information directly into the syslog of an OSSIM sensor. An agent will be active in this sensor to read from this syslog, and will standardize the events so that they can be sent to the server on which it depends.

- **SNMP**

  An agent can receive events in SNMP format. Anyway, in order to receive them from any device, it will be necessary to install in the sensor which is going to receive the data, additional software to establish the connection and make the sensor be able to understand this protocol. This software is available on SNMP Sourceforge web site.

- **Log Files**

  In the same way as with Syslog and SNMP, an agent can be configured in order to read from any log file once a dedicated plugin has been configured for this purpose.

- **Osiris: Unix HIDS**

  Osiris is a Host Integrity Monitoring System that periodically monitors one or more UNIX hosts for change. It maintains detailed logs of changes to the file system, user and group lists, resident kernel modules, etc.

  It is possible to define both an agent and a plugin to extract information from UNIX machines by accessing Osiris stored information.

- **Snare: Collecting from Windows**

  Snare is the method OSSIM uses to extract information from a windows box. Each Windows host with snare agent installed must be able to send UDP port 514 data towards an OSSIM sensor. Then Windows events are normalized into the OSSIM nomenclature and sent to the ossim-server.

- **FW1LogGrabber: Collecting from Checkpoint FW-1**

  It is also possible, by installing in the sensor machine some additional software (available as part of Checkpoints OPSEC API) to download the logs from the

Checkpoint FW-1. These logs, once downloaded and stored in the sensor hard drive, will be read from a plugin, exactly equal as the other plugins in the Agent.

4.4 OSSIM SIEM Event Description

OSSIM defines four types of events that are recognized by the server. The events received will be treated in a different way depending on the type of data. Plugins should parse events from different sources to these standardized ones, typically to the first of them as the other three are dedicated for special situations. Consequently, any developer who wants to implement a new plugin in compliance with the SIEM provided by the FI-WARE Security Monitoring GE should take into consideration this event description.

These event types are:

4.4.1 Normalized event

This is any event that OSSIM server receives from different plugins or devices.

The fields of which the standardized event consists are detailed in the table below:

Field name	**Description**
Type	Type of event: detector or monitor
Date	Date on which the event is received from the device
Sensor	IP address of the sensor generating the event
Interface	Deprecated
Plugin_id	Identifier of the type of event generated
Plugin_sid	Class of event within the type specified in plugin_id
Priority	Deprecated
Protocol	Three types of protocol are permitted: TCP, UDP, ICMP
Src_ip	IP which the device generating the original event identifies as the source of this event
Src_port	Source port
Dst_ip	Ip which the device generating the original event identifies as the destination of this event
Dst_port	Destination port
Log	Event data that the specific plugin considers as part of the log and which is not accommodated in the other fields. Due to the Userdata fields, it is used

	increasingly less
Data	Normally stores the event payload, although the plugin may use this field for anything else
Username	User who has generated the event or user with whom it is identifying mainly used in HIDS events
Password	Password used in an event (HIDS events)
Filename	File used in an event, mainly used in HIDS
Userdata 1 to 9	These fields can be defined by the user from the plugin. They can contain any alphanumeric information, and on choosing one or another, the type of display they have in the event viewer will change. Up to 9 fields can be defined for each plugin

4.4.2 Mac Event

Events which inform of changes in MAC address for specific IPs. This could be used in order to create directives sensitive to ARP Spoofing, for example. These events are generated by arpwatch software, integrated into OSSIM. An event of this type can also be generated in the server thanks to Service event messages coming from the agent. These internal messages can be of the MAC change type; when it reaches the OSSIM server, the latter redoes it and converts it into a MAC type event, treating it in the same way as these.

The fields are detailed in the table below:

Field name	**Description**
Host	Host IP which has changed MAC
Mac	Mac in hexadecimal
Vendor	Card manufacturer
Sensor	IP address of the sensor generating the event
Interface	Deprecated
Date	Date on which the event is received from the device
Plugin_id	Identifier in this case it will always be 1512
Plugin_sid	Without importance, the OSSIM server will assign it
Log	Event data that the specific plugin considers as part of the log and which is not accommodated in the other fields
Userdata 1	Copy of the MAC address

Userdata 2	Copy of vendor
Userdata 3 to 9	These fields can be defined by the user from the plugin. They can contain any alphanumeric information, and on choosing one or another, the type of display they have in the event viewer will change. Up to 9 fields can be defined for each plugin

4.4.3 OS Event

These events will inform of a change in the operating system of a machine.

The fields are detailed in the table below:

Field name	**Description**
Host	IP to which the O.S. has been shown
OS	Operating System displayed for the host indicated
Sensor	IP address of the sensor generating the event
Interface	Deprecated
Date	Date on which the event is received from the device
Plugin_id	Identifier in this case it will always be 1511
Plugin_sid	Without importance, the OSSIM server will assign it
Log	Event data that the specific plugin considers as part of the log and which is not accommodated in the other fields
Userdata 1	In this case it is used to maintain the O.S. for correlation
Userdata 2 to 9	These fields can be defined by the user from the plugin. They can contain any alphanumeric information, and on choosing one or another, the type of display they have in the event viewer will change. Up to 9 fields can be defined for each plugin

4.4.4 Service Event

These events are used in order to keep an inventory of the systems existing on the network, new active applications and open ports being detected. They are also used in cross-correlation, together with the OSVDB DB.

The fields are detailed in the table below:

Field name	**Description**
Host	Host IP
Sensor	IP address of the sensor generating the event
Interface	Deprecated
Port	Open port displayed in the Host machine
Protocol	Type of protocol
Service	Type of service existing in the specified port (www, ssh, ftp…)
Application	Specific application which executes the displayed service
Date	Date on which the event is received from the device
Plugin_id	Identifier in this case it will always be 1516
Plugin_sid	Without importance, the OSSIM server will assign it
Log	Event data that the specific plugin considers as part of the log and which is not accommodated in the other fields
Userdata 1	Copy of the application field
Userdata 2	Copy of the service field
Userdata 2 to 9	These fields can be defined by the user from the plugin. They can contain any alphanumeric information, and on choosing one or another, the type of display they have in the event viewer will change. Up to 9 fields can be defined for each plugin

5 FIWARE OpenSpecification Security Context-based security & compliance

You can find the content of this chapter as well in the wiki of fi-ware.

***[DISCLAIMER]:** The first version of the generic enabler described below is planned to be delivered on the second FI-WARE release. That is the reason why some of the specifications are under discussion and have not been fixed yet. Anyway, most of the sections are expected to have either minor or no changes at all. Those ones which are still not mature enough are clearly identified and they will be more detailed on the second release.*

Name	FIWARE.OpenSpecification.Security.Context-based security & compliance
Chapter	Security,
Catalogue-Link to Implementation	Context-based Security&Compliance - PRRS
Owner	ATOS, Antonio García-Vázquez

5.1 Preface

Within this document you find a self-contained open specification of a FI-WARE generic enabler, please consult as well the FI-WARE_Product_Vision, the website on http://www.fi-ware.eu and similar pages in order to understand the complete context of the FI-WARE project.

5.2 Copyright

- Copyright © 2012 by ATOS

5.3 Legal Notice

Please check the following Legal Notice to understand the rights to use these specifications.

5.4 Overview

The role of the Context-based Security & Compliance Generic Enabler is to support additional security requirements requested by a specific subset of applications as a result of the application of very specific regulatory constraints.

The GE will accept security requests from a client application and will select the best Optional Security Enabler to fulfill it.

The deployed security enabler will implement the compliance between the client security request and any applicable regulation from Private and/or Public sources.

The framework has also monitoring capabilities to overseen the system performance. As a result of this monitoring, if a non-conformance is detected, the framework is capable of performing run-time and context-based reconfiguration of deployed security enablers, such that the client application will be provided with a new configuration for the security enabler it is utilizing, or it can receive instructions to stop using that security enabler and use a newly provided one.

This feature provides the security layer of FI-WARE with context-aware capabilities that allow any of its instances to deal with dynamic and unpredictable context changes. End User applications will send to the GE their requirements and the context description to obtain the security solution that best matches their needs.

Once the GE has got the user request and the applicable rules, it will query FI-WARE security marketplace to get the Optional Generic Enabler that best fulfill the security requirements defined and will deploy it in the End-User context.

At the same time that the security solution is deployed into the end-user environment, the GE also instantiates a runtime monitor with the responsibility of detecting anomalous behavior or non-conformance. In case of a non-conformance detected, the framework will take compensation actions for the automated adaptation of the deployed security mechanism to the changing context conditions.

The security requirements sent by the end-user applications to the GE could be expressed both by single security specifications and by references to a predefined rule, law or agreement stored in the rule repository offered by the GE.

Security services must be registered in FI-WARE marketplace and must describe their features by using USDL-SEC language in order to be successfully discovered by this GE.

5.4.1 Example Scenario

A firm has implemented some communication links between its subsidiaries through the Future Internet by using FI-WARE capabilities.

Each of the subsidiaries is located in a different country and private data from the employees are shared. The communications links must be compliant with different data protection regulations.

The Context-based security & compliance Generic Enabler is requested to deploy an optional security enabler that will guarantee the privacy and make the communication link compliance with the Data Protection Law of each country. In the case of one of the affected countries would change its applicable regulations, the associated metrics will be updated in the GE rules repository by the firm. Then the GE monitors will verify if the communication links with that specific country are still compliant with the new regulations and a trigger will be sent to the GE framework manager if needed.

Finally the optional security enabler will be either reconfigured or redeployed by the Context-based security & compliance Generic Enabler if the monitor reports a non-compliance event.

5.5 Basic Concepts

5.5.1 USDL-SEC

The USDL-SEC language is being developed as a security extension to the latest version of USDL language (Linked-USDL), currently maintained by the Applications/Services Ecosystem and Delivery Framework Work Package. More information on USDL can be found in its Open Specification page.

The available security service features as well as the security specifications to be fulfilled are described by using USDL-SEC language.

Furthermore, the combination USDL/USDL-SEC describes a service along with functional and non-functional properties in a single and complete description file. This feature provides a means to compare and select services according to the consumer needs.

The communication between end-user applications, the deployed security enabler and the framework will be is supported by this service oriented language.

The security extension allows:

- any application to express both high level description of the service and detail functionalities & implementations in a single and complete specification file;
- consumers and providers to agree on a security protocol, through expressions of concrete mechanisms and links to existing standard such as WS-SecurityPolicy, XACML, P3P, etc. with a security model fully defined by its associated properties file.

5.5.2 Security Specifications & Rules

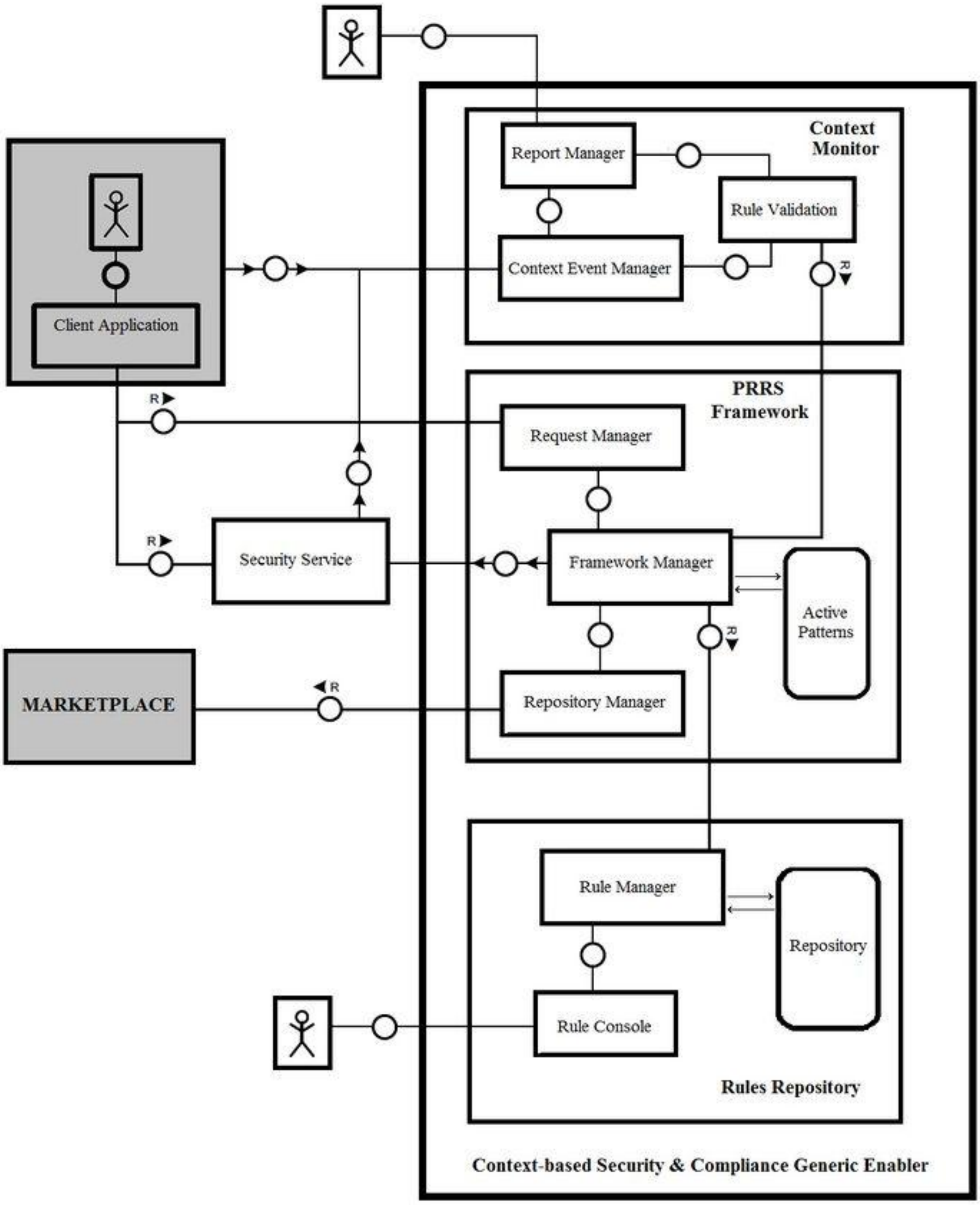
In the context of this Generic Enabler we define these two concepts as follow:

- **Security specification**: Any single security requirement that can be supported by a security service. Some examples could be encryption, authentication and accountability. Each security specification will be expressed as a security service feature in USDL-SEC Language.
- **Rule**: A set of security specifications that describes a complex security constraint that must be fulfilled commonly by an optional security service. Some examples could be a *Data Protection Law* for a specific country or a *Security Service Level Agreement* between two different companies.

5.6 Context-based security & compliance architecture

This section describes software Context-based security & compliance GE modules. The overall architecture will be highlighted as well as the description of its main components.

The architecture of the proposed Generic Enabler is detailed in the figure below. It shows the main components of the GE and the interfaces to be implemented between them and to be offered to external applications.



Context-based Security and Compliance Generic Enabler Architecture

5.6.1 PRRS Framework

This component provides run-time support to applications performing dynamic selection & deployment of security enablers.

The PRRS Framework is the core of the Generic Enabler. It is in charge of controlling the rest of the components of the GE, processing requests from end-user applications and orchestrating the instantiation and monitoring of the Security Enabler selected.

End-user applications send requests in order to fulfill their security requirements to the PRRS Request Manager sub-module either as a specification of security requirements or as a reference to already existing security rules.

The PRRS Request Manager is also going to be in charge of sending any notification from the GE to the End-User applications.

The PRRS Framework Manager is going to deal with the communication between PRRS Framework component and Rule Repository component on one site and the FI-WARE Marketplace on the other.

From Rule Repository, it retrieves the set of security rules to be applied on each Security Request based on the context description provided by the end-user application and it will be triggered by the repository in case of a rule change situation.

From FI-WARE Marketplace, the PRRS discovers the optional security enabler (from the services offered there by the providers) that better matches the end-user security request according the published usdl-sec service descriptions. Examples of optional security enablers developed in FI-WARE context and available in the FI-WARE Marketplace are the DBAnonymizer GE or the Secure Storage Service GE.

Active Patterns database stores useful information related to the Applications request, monitoring systems overseen them, the rules that are being used and the additional security enablers deployed for PRRS Framework Manager decision making support.

Finally, the PRRS Framework Manager is the decision making engine. It compares rules to be applied, user requirement and security enabler features to select the most suitable solution to fulfill End-User requirements.

PRRS Framework Manager also provides external monitoring components with the rules to be checked against the already deployed security enabler and takes the necessary reaction mechanisms in case of non compliance detection by the reactivation, reconfiguration, deactivation and/or substitution of the deployed enabler if required.

5.6.2 Rules Repository

This component will allow the generic enabler to store and manage the binding of security requirements and additional security constraints that arise from relevant user context information at various abstractions levels defined during design-time (for example specific regulatory constraints for Future Internet domains, such as Healthcare or Telecommunications) and also manage the rules to check end-to-end business processes for compliance against the set of applicable constraints during run-time.

The rules to be stored could come from various sources, including laws and regulations, public and internal policies, standards, customer preferences, partner agreements and jurisdictional provisions. A Rule Console will be provided with this purpose.

It could be required the interaction with the FI-WARE Context Broker GE to retrieve context information from different scopes and to publish new security rules in order to put them available for end-user applications or other entities. More details about the relationship between the Context-based security and compliance GE and the Context Broker GE will be available on release two.

In order to manage compliance throughout all phases of business process lifecycle, it must be possible to define and subsequently integrate compliance specifics into business processes and enterprise applications, and assure compliance starting from the process analysis and design phase. Furthermore, the Rule Manager component will be able to reuse fragments of already stored formal rules specifications to build a new formal specification form a new law or rule to be stored. Reuse of rules specification fragments will make the task of compile new laws or rules into formal language easier.

Each high-level rule or specification will be compiled into a formal pattern following USDL-SEC specifications that can be applied and referenced in many scenarios either by end-user applications as a security requirement or any security enabler to describe its characteristics.

Besides, each rule must include a monitoring control that will be used at run-time by the Context Monitors to perform the rule validation and determine when a deployed security enabler has violated some constraint and a recovery action is required in the PRRS Framework to maintain the compliance.

Finally the Rules Repository will be able to trigger PRRS Framework when some rule is modified so that the PRRS Framework can take the necessary actions in case of the modification must be taken into account on compliance measurements.

5.6.3 Context Monitor

Runtime context monitors are the components in charge of monitoring anomalous behavior or changes in the end-user application context and in the deployed security enabler in order to detect non-conformances with the validation rules.

Each context monitor will get context and status events from the end user application and the security enablers it is overseeing. There can be multiple context monitors (including the own FI-WARE Security Monitoring GE) but in any case the Context Monitor component in the Context-based security & compliance GE must include a Rule Validation module with a common interface to receive from the PRRS Framework the rules to be checked. This Rule Validation must understand how to deal with the rules received and add them to the specific monitoring engine. It also must implement the way to retrieve alerts and events from the monitoring engine to trigger the reaction mechanism in the PRRS Framework.

The Context Event Manager module is the monitoring engine itself that will compare the information obtained with the rules provided by PRRS Framework.

In case of non-compliance detection, the assigned event (as well as the identification of the monitored service) will be sent to PRRS Framework by the appointed monitor so that the framework could take the necessary recovering actions.

Additionally, the Context Monitor will provide a Report Manager module with a dashboard that gives the system reporting capabilities. Reports of system performance will be generated once the information from data context has been compared with rules received from PRRS.

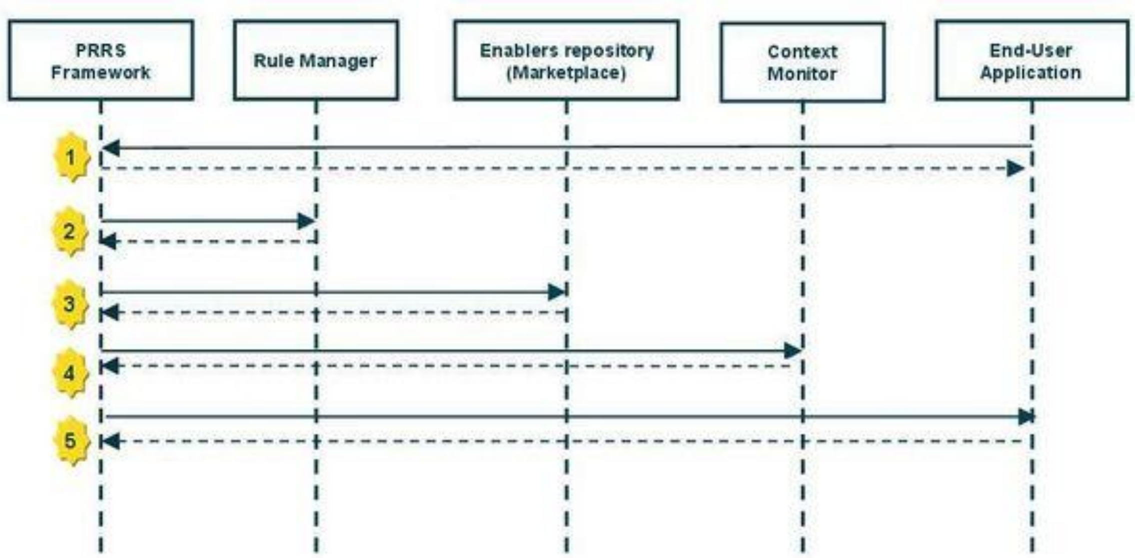
These visual reports will provide useful information about the levels of compliance and performance of the optional security enablers dynamically deployed by PRRS, making the task of identification of root causes for non-compliant situations easy.

5.7 Main Operations

We describe in this section the first approach of the interface that is going to be offered by the GE and will be implemented on the Second FI-WARE Main Release. The interactions between the PRRS Framework and other FI-WARE Generic Enablers (such us the Data/Context Management GE or the Marketplace GE) are still under discussion and the interfaces could suffer some changes.

5.7.1 User Request

This section describes the steps to be followed in a communication between an End-User application and the PRRS Framework, from the moment an user request for an additional security enabler is sent to its deployment by the GE.



User Request Sequence Diagram

1: An End-User application sends a security request to the PRRS Framework providing information about its security requirements and user context. The PRRS Framework stores the information into its internal database and replies the applicant with an assigned requestID for the security service that will be selected by the PRRS Framework.

The user security request can include the following elements:

End-User serviceRequest

Elements	Description
requestServiceName	Allows users to send a direct request to PRRS by service name if known.
requestServiceType	User will specify the type of security goal required for the service.
requestSecuritySpecs	User will specify required security specifications expressed in USDL-SEC vocabulary.
requestContextDesc	User will specify additional security rules and useful information about the user context that can result in the application of specific security constraints or determine the service selection. This context information must have been published in the FI-WARE Context Broker GE and the user will include here a list of context data with its entity scope and id.

End-User serviceRequestResponse

Elements	Description

responseStatus	It will indicate the error level of the response, where 0 is equals to OK and any other value means an error
responseDetails	It will contain the error description if the previous parameter indicates an error or the assigned request ID otherwise

2: The PRRS Framework gets from the Rules Repository the list of additional security specifications associated to the rules and user-context information detailed by the End-User application in the security request.

ruleRequest

Elements	Description
requestRuleName	Rule identificator name if known
requestContextData	Context information provided by the end-user application in the security request that could derive in an additional security constraint or rule.

ruleRequestResponse

Elements	Description
responseStatus	It will indicate the error level of the response, where 0 equals to OK and any other value means an error.
responseRuleName	Rule identificator name.
responseSecuritySpec	Security specification expressed in USDL-SEC vocabulary for the rule to be added to the security requirements provided by the End-User application.

3: The PRRS Framework uses the FI-WARE Marketplace GE to discover the best optional security enabler to be deployed by searching among the available offering services those ones whose usdl-sec (published by the Service Providers) better matches the user security requirements (including the ones derived from the provided context information by the Rules Repository in the previous step).

The communication to be implemented with the Marketplace will be compliant with the API description provided by Application and Services Ecosystem and Delivery Framework chapter. Check for additional details.

4: A context monitor is selected by the PRRS Framework for that client application and security enabler and the validation rules are sent to it.

serviceMonitor

Elements	Description
requestUserMonitorID	User application ID to be monitored.
requestEventURL	URL where the events to be monitored are located.
requestRuleFile	Rules to be checked by the monitor.

serviceMonitorResponse

Elements	Description
responseStatus	It will indicate the error level of the response, where 0 equals to OK and any other value means an error.
responseDetails	It will contain the error description if the previous parameter indicates an error.

5: The selected Optional Security Enabler is deployed and instantiated. Details to interact with it are sent to the end-user application.

PRRS serviceDeploy

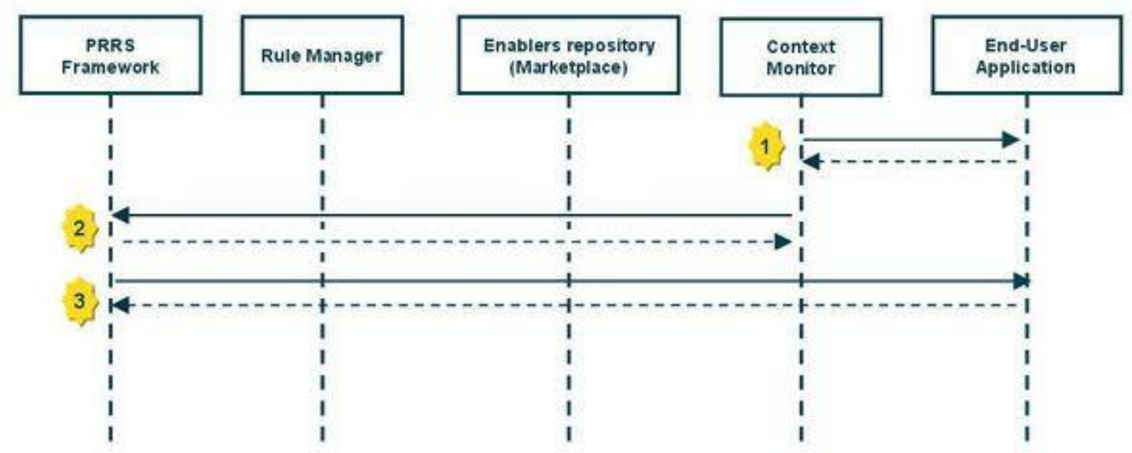
Elements	Description
requestUserID	Assigned request ID.
requestServiceURL	URL of the selected security service.
requestServiceSpec	Access to the USDL-SEC description for the selected security service.

PRRSserviceDeployResponse

Elements	Description
responseStatus	It will indicate the error level of the response, where 0 equals to OK and any other value means an error.

5.7.2 Non Compliance Detection

This section briefly summarizes the steps to be followed in an internal communication between a monitor system and the PRRS Framework component to notify a non-compliance situation. More details about the interface to be implemented will be available for the second FI-WARE release.



Non Compliance Detection Sequence Diagram

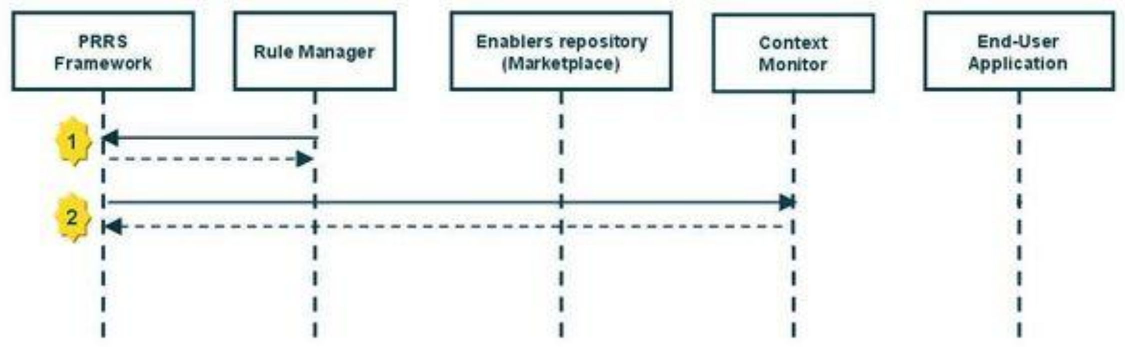
1: A monitor retrieves information from the End-User context both by receiving service event information and by accessing to the deployed security service logs.

2: The information retrieved on step one is compared with the validation rules and the PRRS Framework is triggered in case of a non-compliance situation.

3: The PRRS Framework Manager will take the most suitable recovery action by the reactivation, reconfiguration, deactivation and/or substitution of the deployed security enabler.

5.7.3 Rule Change

This section briefly summarizes the steps to be followed in an internal communication between Rule Repository and the PRRS Framework components to notify a rule change situation. More details about the interface to be implemented will be available for the second FI-WARE release.



Rule Change Sequence Diagram

1: The Rule Manager triggers the PRRS Framework every time a stored rule is modified.

2: The PRRS Framework Manager gets from its internal database the system monitors that are overseeing the context where the rule is applicable and updates them with the associated validation parameters.

5.8 Basic Design Principles

- To provide run-time security support for applications by dynamically deploying and monitoring the End-User applications environment.
- An enhanced security and dependability by supporting automated integration, configuration, monitoring and adaptation.
- Dynamic compliance of software services to business regulations and user requirements than can be easily modeled through the rule repository dashboard.
- The USDL-SEC language, as an extension of the standard USDL language, is going to share their main principles:
    - Uniform Service Descriptions
    - Modular Design
    - Extensibility Principle

5.9 Detailed Specifications

5.9.1 Open API Specifications

- FIWARE.OpenSpecification.Details.Security.Context-based security & compliance_API

5.9.2 Other Relevant Specifications

The data formats for the API rely on the Linked USDL specifications:

- Linked USDL Core Vocabulary
- Linked USDL Pricing Vocabulary
- Linked USDL Service Level Agreements Vocabulary
- Linked USDL Security Vocabulary

FI-WARE Generic Enablers used by the Context-based security & compliance GE are:

- Marketplace GE Open Specification
- Context Broker GE Open Specification

Optional Security Enablers developed in FI-WARE:

- DBAnonymizer GE Open Specification
- Secure Storage Service GE Open Specification

5.10 Re-utilised Technologies/Specifications

The PRRS Framework included in this Generic Enabler is based on the Platform for Runtime Reconfigurability of Security (PRRS) developed within the european Serenity Project (2008 - http://cordis.europa.eu/projects/rcn/78381_en.html).

This platform makes it possible that at design-time developers have the option not to include specific security solutions but rather specify simply the security requirements that the system will have to satisfy during runtime. Consequently, service developers need not to be concerned about the development of a security solution because the PRRS will supply them with the security solution that better fulfils the specifications. Moreover, the client application needs not to be bound to any one particular security solution. By using the PRRS, the system will benefit from a collection of security solutions (which has been developed by security experts and whose security has been previously verified) available during runtime that can be adapted to the context of the system.

The USDL-SEC language developed with this Generic Enabler to describe and register security services, capabilities and compliances rules, is defined as a security oriented module extension of the existing standard USDL 3.0 (http://www.linked-usdl.org).

5.11 Terms and definitions

This section comprises a summary of terms and definitions introduced during the previous sections. It intends to establish a vocabulary that will be help to carry out discussions internally and with third parties (e.g., Use Case projects in the EU FP7 Future Internet PPP). For a summary of terms and definitions managed at overall FI-WARE level, please refer to FIWARE Global Terms and Definitions

- **Attack.** Any kind of malicious activity that attempts to collect, disrupt, deny, degrade, or destroy information system resources or the information itself
- **Authentication protocol**: "Over-the-wire authentication protocols are used to exchange authentication data between the client and server application. Each authentication protocol supports one or more authentication methods. The OATH reference architecture provides for the use of existing protocols, and envisions the use of extended protocols which support new authentication methods as they are defined." (OATH)
- **Access control**: is the prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner. (ITU-T-X-800_Link). More precisely, access control is the protection of resources against unauthorized access; a process by which use of resources is regulated according to a security policy and is permitted by only authorized system entities according to that policy. RFC 2828
- **Account**: A (user) account is “typically a formal business agreement for providing regular dealings and services between principal sand business service providers.” OASIS Security Assertion Markup Language (SAML)
- **Authentication (AuthN)**: We adopted the following definition of authentication from RFC 3588"Authentication is “the act of verifying the identity of an entity (subject)”

  TrustInCyberspace adds the term “level of confidence” to this definition:

  Authentication is the process of confirming a system entity’s asserted identity with a specified, or understood, level of confidence.” This definition holds all necessary parts to examine authentication in broad sense. First of all it does not narrow the authentication to human users, but refers to a generic “system entity”. See authentication reference architecture description for a closer look at different identities that could be authenticated.

Secondly it introduces the often neglected concept of “level of confidence” which applies to each authentication of an identity. No computer program or computer user can definitely prove the identity of another party. There is no authentication method that can be secured against any possible identity-theft attack, be it physical or non-physical. It is only possible to apply one or more tests, which, if passed, have been previously declared to be sufficient to go on further. The problem is to determine which tests are sufficient, and many such are inadequate.

The original Greek word originates from the word 'authentes'='author'. This leads to the general field of claims and trust management, because authentication could also mean to verify the “author” / issuer of any claim.

The confirmation or validation process of authentication is actually done by presenting some kind of proof. This proof is normally derived from some kind of secret hold by the principal. In its simplest form the participant and the authentication authority share the same secret. More advanced concepts rely on challenge/response mechanisms, preventing the secrets to be transmitted. Refer to Authentication Technologies for a detailed list of authentication methods used today.

As stated above, each authentication method assures only some level of trust in the claimed identity, but none could be definite. Therefore it makes sense to distinguish the different authentication methods by an associated assurance level, stating the level of trust in the authentication process.

As this assurance level depends not only on the technical authentication method, but also on the overall computer system and even on the business processes within the organization (provisioning of identities and credentials), there is no ranking of the authentication methods here.

- **Countermeasures**. Action, device, procedure, technique or other measure that reduces the vulnerability of an information system.
- **Cyber attack**. An attack, via cyberspace, targeting an entity (industrial, financial, public...) and using cyberspace for the purpose of disrupting, disabling, destroying, or maliciously controlling a computing environment/infrastructure; or destroying the integrity of the data or stealing controlled information
- **Exploit.** A program or technique that takes advantage of vulnerability in software and that can be used for breaking security, or otherwise attacking a host over the network
- **Federation**: The term federation “is used in two senses - "The act of establishing a relationship between two entities. An association comprising any number of service providers and identity providers.” OASIS Security Assertion Markup Language (SAML)

  “A federation is a collection of realms that have established a producer-consumer relationship whereby one realm can provide authorized access to a resource it manages based on an identity, and possibly associated attributes, that are asserted in another realm.

  Federation requires trust such that a relying party can make a well-informed access control decision based on the credibility of identity and attribute data that is vouched for by another realm.” WS-Federation @ IBM

  Remark: Federation according to WS-Federation @ IBM is similar to the concept of a Circle of Trust.

- **Forensics for evidence.** The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations.
- **Identity**. In the narrow sense, identity is the persistent identifier of users (user name), things or services by which other parties "remember" them and, hence, are able to store or retrieve specific information about them and are able to control their access to different resources. In the wider sense, identity also covers further attributes of users, things and services; e.g. for users, such information may include personal information such as context, group membership and profile.
- **Identity** (Digital): The term identity and its meaning have been discussed controversially in the "identity community" for many years. Until now, there is no commonly agreed definition of that notion. : : The IdM && AAA reference architecture applies the following three definitions of identity.

  The Identity Gang defines the term digital identity as follows:

  A digital identity is "a digital representation of a set of Claims made by one party about itself or another digital subject."

  The following comments were added:

  A digital identity is just one set of claims about a digital subject. For any given digital subject there will typically be many digital identities.

  A digital identity can be created on the fly when a particular identity transaction is desired or persistent in a data store to provide a representation that can be referenced.

  A digital identity may contain claims made by multiple claimants.

  A digital identity may be signed by a digital identity provider to provide assurance to a relying party.

  This definition emphasizes two facts:

  Normally, a principal (subject) has multiple digital identities or personas.

  Identities are made out of attributes (claims).

  Therefore, the scope of identity management in the reference architecture has two viewpoints: For once it focuses on identities and personas itself, and on the other side, it deals with the attributes of these identities and personas.

  The Liberty Alliance Project (LAP) defines digital identity as follows:

  Digital identity is "the essence of an entity. One's identity is often described by one's characteristics, among which may be any number of identifiers. A Principal may wield one or more identities."

  RSA uses the following definition of digital identity:

  "Digital identity consists of an identity assertion and the characteristics, sometimes called attributes that are collected or observed through our computerized relationships. It is often as simple as a user name and password."

  The definition of RSA adds one important aspect to the identity discussion: Even the simplest user name and password combinations without any additional attributes or claims constitute an identity.

- **Identifier**: Identifiers can be understood as a dedicated, publicly known attribute of an identity that refers to that identity only. Typically, identifiers are valid within a specific domain. Special types of identifiers are valid globally, due to the use of popular domain naming and resolution protocols such as DNS, which implies addressing capabilities to the identity. OASIS Security Assertion Markup Language (SAML) defines identifier as follows:

  An identifier is “a data object (for example, a string) mapped to a system entity that uniquely refers to the system entity. A system entity may have multiple distinct identifiers referring to it. An identifier is essentially a "distinguished attribute" of an entity.”

- **Identity context**: is “the surrounding environment and circumstances that determine meaning of digital identities and the policies and protocols that govern their interactions.” (Identity Gang)

- **Identity management (IdM)**: comprises “the management of identity information both internally and when it is passed from one entity to another.” Open Mobile Alliance (OMA)

- **Identity provider**: The Open Mobile Alliance (OMA) defines the term identity provider (IdP) as follows - An identity provider is “a special type of service provider […] that creates, maintains, and manages identity information for principals, and can provide […] assertions to other service providers within an authentication domain (or even a circle of trust).”

  Another notion defines identity provider as “an agent that issues a digital identity [that] is acting on behalf of an issuing Party.” (Identity Gang)

  The following definition of identity provider descends from WS-Federation @ IBM: “An identity provider is an entity that acts as an authentication service to end requestors and as data origin authentication service to service providers […]. Identity providers are trusted (logical) 3rd parties which need to be trusted both by the requestor […] and the service provider which may grant access to valuable resources and information based upon the integrity of the identity information provided by the identity provider.”

  The Identity Provider is part of the Identity Management infrastructure.

- **Impact**. The adverse effect resulting from a successful threat exercise of vulnerability. Can be described in terms of loss or degradation of any, or a combination of any, of the following three security goals: integrity, availability, and confidentiality.

- **Partial identity:** a partial identity is a set of attributes of a user. Thus, an identity is composed of all attributes of a user, a partial identity is a subset of a user's identity. Typically, a user is known to another party only as a partial identity. A partial identity can have a unique identifier. The latter is a strong identifier if it is allows for a strong authentication of the user (holder) of the partial identity, such a cryptographic "identification" protocol

- **Privacy**. Dictionary definitions of privacy refer to "the quality or state of being apart from company or observation, seclusion [...] freedom from unauthorized intrusion" (Merriam-Webster online [MerrWebPriv]). In the online world, we rely on a pragmatic definition of privacy, saying that privacy is the state of being free from certain privacy threats.

- **Privacy threats**. The fundamental privacy threats are: traceability (the digital traces left during transactions), linkability (profile accumulation based on the digital traces), loss of control (over personal data) and identity theft (impersonation).
- **Risk analysis**. The process of identifying security risks, determining their magnitude, and identifying areas needing safeguards. An analysis of an organization's information resources, its existing controls, and its remaining organizational and MIS vulnerabilities. It combines the loss potential for each resource or combination of resources with an estimated rate of occurrence to establish a potential level of damage
- **Security monitoring**. Usage of tools to prevent and detect compliance defaults, security events and malicious actions taken by subjects suspected of misusing the information system.
- **Service impact analysis.** An analysis of a service's requirements, processes, and interdependencies used to characterize information system contingency requirements and priorities in the event of a significant disruption.
- **Single sign-on**: is "From a Principal's perspective, single sign-on encompasses the capability to authenticate with some system entity—[…] an Identity Provider - and have that authentication honored by other system entities, [termed] Service Providers […]. Note that upon authenticating with an Identity Provider, the Identity Provider typically establishes and maintains some notion of local session state between itself and the Principal's user agent. Service Providers may also maintain their own distinct local session state with a Principal's user agent." Liberty Alliance Project (LAP)
- **S&D:** Security and Dependability
- **Threat.** An event, process, activity being perpetuated by one or more threat agents, which, when realized, has an adverse effect on organization assets, resulting in losses (service delays or denials, disclosure of sensitive information, undesired patch of programs or data, reputation...)
- **USDL and USDL-Sec:** The Unified Service Description Language (USDL) is a platform-neutral language for describing services. The security extension of this language is going to be developed FI-WARE project.
- **Vulnerability.** A weakness or finding that is non-compliant, non-adherence to a requirement, a specification or a standard, or unprotected area of an otherwise secure system, which leaves the system open to potential attack or other problem.
- **WS-SecurityPolicy:** It is an extension to SOAP to apply security to web services. It is a member of the WS-* family of web service specifications and was published by OASIS.
- **The protocol** specifies how integrity and confidentiality can be enforced on messages and allows the communication of various security token formats, such as SAML, Kerberos, and X.509. Its main focus is the use of XML Signature and XML Encryption to provide end-to-end security.

6 FIWARE OpenSpecification Details Security Context-based security & compliance_API

You can find the content of this chapter as well in the wiki of fi-ware.

The first version of the Context-based Security & Compliance API is planned to be delivered on the second FI-WARE release.

7 FIWARE OpenSpecification Security Data Handling Generic Enabler

You can find the content of this chapter as well in the wiki of fi-ware.

Name	FIWARE.OpenSpecification.Security.Data Handling Generic Enabler
Chapter	Security,
Catalogue-Link to Implementation	Data Handling
Owner	SAP, Slim Trabelsi

7.1 Preface

Within this document you find a self-contained open specification of a FI-WARE generic enabler, please consult as well the FI-WARE_Product_Vision, the website on http://www.fi-ware.eu and similar pages in order to understand the complete context of the FI-WARE project.

7.1.1 Copyright

- Copyright © 2012 by SAP

7.2 Legal Notice

Please check the following Legal Notice to understand the rights to use these specifications.

SAP strives to make the specifications of this Generic Enabler available under IPR rules that allow for a exploitation and sustainable usage both in Open Source as well as proprietary, closed source products to maximize adoption.

7.3 Overview

The Data Handling GE is a privacy-friendly attribute-based access control system, that targets mainly sensitive data. It permits to store information together with an attached privacy policy, that regulates its usage. Thus, Data Handling GE can reveal certain attributes, according to specific supplied prove conditions. Data Handling GE supports integrated data handling (two-sided detailed data handling), that takes into account specific preferences/policies expressed using the PPL language (Privacy Policy Language)[PPL]. PPL is based on the XACML standard [XACML]. Data usage purpose must always be declared, as it is a relevant part of the policy that must be expressed, as well as downstream usage, i.e., whether one can disclose collected data with third parties. The PPL language supports the enforcement of a number of obligations, that are bound tightly to data. For instance, one can impose a specific retention period, as well as the production of user's notifications and/or logging under certain conditions.

7.3.1 Target usage

The Data Handling GE provides a mechanism for controlling the usage of attributes and data (more precisely, of Personal Identifiable Information or PII) based on the concept of 'sticking' a data usage policy to the data to which it applies. When the data is accessed by an application, an access control technology is then used to verify that the intended use of the data matches the scope defined in the usage policy. Therefore, the Data Handling GE can be used by any application or service that would offer a transparent data handling policy to users and third parties. In the example scenario later proposed, a

7.4 Basic Concepts

7.4.1 Relevant Concepts and Ideas

In this section, the more important concepts shall be presented. The used terminology is coherent with definitions contained in the European Parliament Directive 95/46/EC, "on the protection of individuals with regard to the processing of personal data and on the free movement of such data". More detailed information is provided in the Terminology section.

Data Controller

"Data Controller" indicates the entity which (alone or jointly with others) determines the purposes and means of the processing of personal data.

Data Subject

The Data Subject is the person whose personal data are collected, held or processed by the Data Controller.

The Data Subject has the right to access his data and to require the Controller to rectify without delay any inaccurate or incomplete personal data. The Data Subject has the right to require the Controller to erase data if the processing is unlawful.

Personal Data (Personal Identifiable Information, or PII)

Personal data means any information relating to an identified or identifiable natural person or "Data Subject".

An identifiable person is someone who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his or her physical, physiological, mental, economic, cultural or social identity.

User Agent

A software system (such as a web browser) acting on behalf of a user. The user agent acts on user preferences when dealing with a server acting on behalf of a Data Controller.

7.4.2 PII and PPL

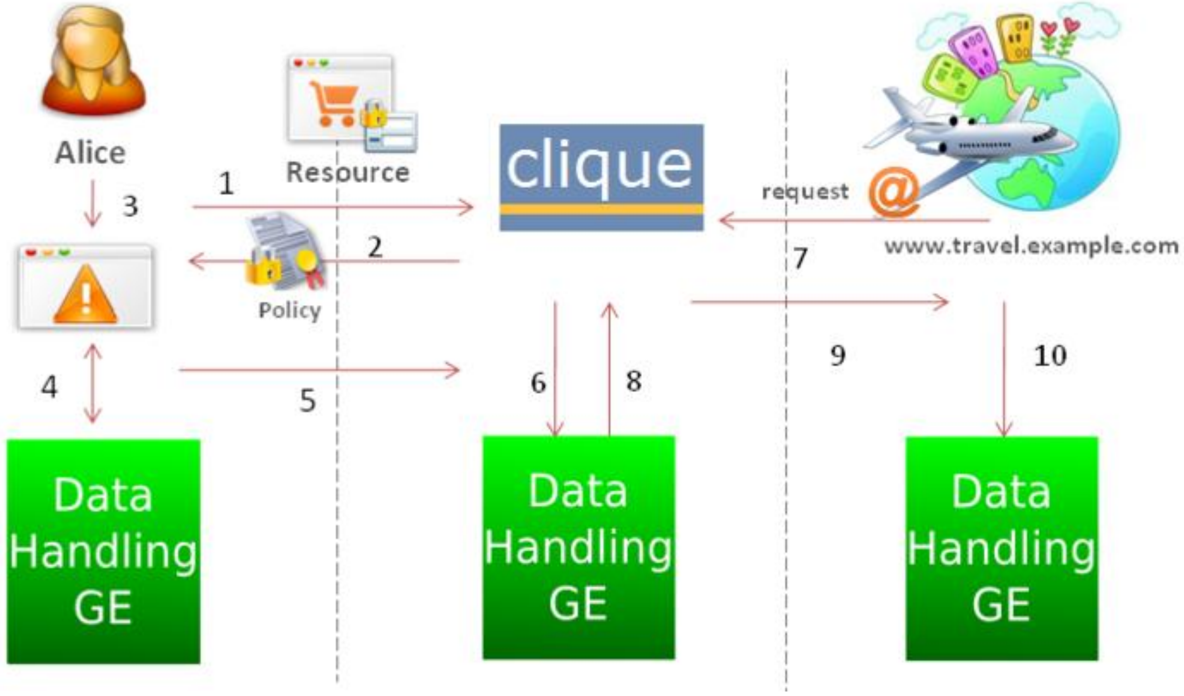
The Data Handling GE regulate the access to sensitive data, collected from users. This can be achieved through the association of a set of preferences/policies to each PII; privacy policies are expressed using the PPL. PPL is used:

1. at PII collection time, as each information that enters the Data Handling GE must come along with a PPL policy;
2. at each data usage, that is regulated according to the associated PPL policy.

In fact, for each data access, the Data Handling GE evaluates its purpose, which must always be declared. Access purpose is a relevant part of the policy that must be expressed, and if and only if there is a compatibility between the data policy and the request policy, the information are provided to the requester. The Data Handling GE is also responsible for fulfilling obligations contained in PPL policies, like for instance, sending an email to the data owner at each access.

7.4.3 Example Scenarios

7.4.3.1 *Use Case: Connection to a Social Network Website*



In this scenario, it is described a subscription to a social network shopping website. The scenario is depicted in the previous picture, and it underlines the different information exchanged between the different parties.

Prerequisite: a different Data Handling GE is available for each peer, and interacts with each subject only using the public interface, so no direct access to stored data is possible.

- *Step 1*: The Data Subject Alice is a privacy-aware user who is quite active in the Web 2.0 websites, but who is concerned about what happens to the data that she provides about herself. Before starting her social networking activity, she has to create an account at an online social network, Clique, an experimental social network. Clique is playing the role of Data Controller.
- *Step 2*: In order to validate her subscription, the website will need to collect some personal information, like her name, her birth date, her e-mail address and her street address... This information is contained in an access control policy on the website

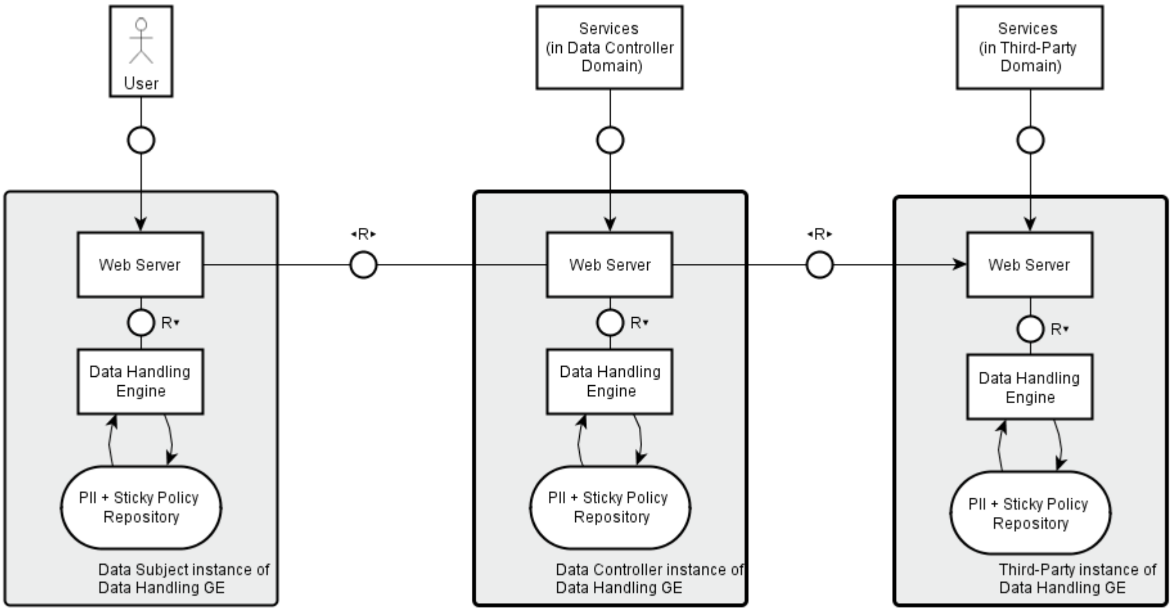
side (i.e., In order to create an account the user has to provide a list of credentials). In order to explain the conditions of usage of such collected information the website sends a privacy policy (written in PPL language).

- *Step 3*: Previously, Alice created privacy preferences related to her personal data.
- *Step 4*: After receiving the website's privacy policy, the Data Handling GE engine of Alice automatically check if the private data requested by the server is stored on Alice's machine. If it is the case, the engine will enforce the access control rules related to the requested data; in other words, Alice's Data Handling GE will check whether Alice's policy and website's policy are compatible (ex. does the domain fi-ware.eu can access my e-mail address?). If the domain is allowed to access this data the engine creates a new privacy policy, combining the privacy policies of both website and Alice. This new policy will comprise the data handling conditions expressed by the two parties.
- *Step 5*: Alice has the possibility to decide if she accepts or refuses to send her data, according to the newly generated privacy policy. If Alice accepts, the new policy gets associated to the requested data, and it becomes a "*sticky policy*". Private Alice's data and sticky policy will be sent at the same time to the Data Handling GE of the website (the Data Controller).
- *Step 6*: The website's Data Handling GE will store the received information, and the website will be able to use this data according to the obligations stated in the sticky policy.
- *Step 7*: The online travel agency www.travel.example.com (third party Data Controller) decided to start an e-mail advertising campaign. In order to target a wide scope of persons, the www.travel.example.com admin asked his partner Clique to provide him with valid e-mail addresses for marketing and statistics purposes. The request will contain a resource query for e-mail and a privacy policy.
- *Step 8*: The policy engine of Clique will compare the privacy policy of travel.example.com with the sticky policy related to Alice's data (and in particular, the obligations on the e-mail address), to verify that the sticky policy allows to forward the protected information for the purpose of statistics. In general, a full matching is performed between travel.example.com's policy and Alice's sticky policy.
- *Step 9 and 10*: The travel.example.com website receives the e-mail address of Alice with a sticky policy (Step 9), that is derived from the original one (see Step 5). The travel agency Data Handling GE configures the actions and the triggers related to the sticky policy obligations, and it stores the e-mail and the sticky policy in the same way as was done by the Clique Data Handling GE (Step 10). In this way, no misuse of data is allowed, as data access is again protected and regulated by the use of privacy policies, as already explained.

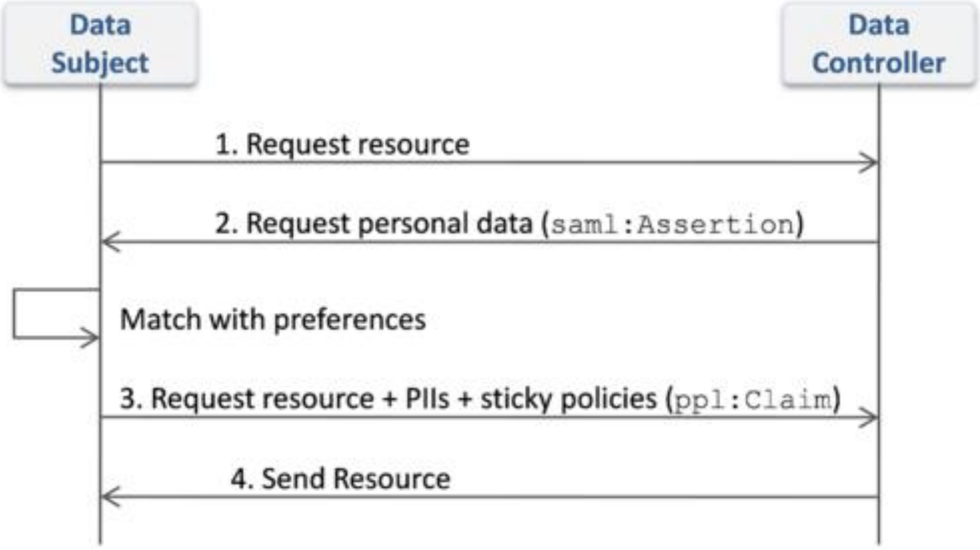
7.5 Main Interactions

7.5.1 Architecture

7.5.1.1 *Block Diagram*



7.5.1.2 *Sequence Diagram*


Data Subject
Data Controller
1. Request resource
2. Request personal data (saml:Assertion)
Match with preferences
3. Request resource + PIIs + sticky policies (ppl:Claim)
4. Send Resource

Description:

This sequence diagram reflects the use case proposed in Use Case. This explanation keeps the same concepts, focusing on detailing Steps 1 to 6, however the numeration of the sequence diagram does not reflect exactly the one of Use Case. *Data Subject* and *Data Controller* are distinguished instances of the Data Handling GE.

1. **Request Resource**: this refers to the access to a resource, that is regulated by the Data Controller. In the Use Case, it refers to Step 1
2. **Request Personal Data**: as the requested resource requires the transmission of PIIs from the Data Subject, the Data Controller invokes the API "Get a PII" of the Data Subject, together with a privacy policy that describes how the PII requested will be used (in a SAML envelope). This reflects Step 2 in the Use Case.
    - **Match With Preferences**: the Data Subject checks whether there is a compatibility (a "match") between user's privacy policy and Data Controller's privacy policy. This is Step 4 in the Use Case.
3. **Request resource + PIIs + sticky policies**: if the two privacy policies match, PIIs are sent to Data Controller, together with the initial request and a number of sticky policies (one for each PII). This is Step 5 in the Use Case.
4. **Send Resource**: Eventually, the resource is sent to the Data Subject. Step 6 of the Use Case.

7.6 Basic Design Principles

The Data Handling GE permits to protect information according to a specific privacy policy. The Data Handling GE safeguards data, storing them together with the respective privacy policies. Any access to the protected resources can happen only declaring explicitly its purposes (using again a description encoded in a privacy policy). The Data Handling GE evaluates the two policies (data and access requests), and transmits the requested information if and only if the policies match.

7.6.1 Detailed Specifications

Following is a list of Open Specifications linked to this Generic Enabler. Specifications labeled as "PRELIMINARY" are considered stable but subject to minor changes derived from lessons learned during last interactions of the development of a first reference implementation planned for the current Major Release of FI-WARE.

7.6.1.1 *Open API Specifications*

- FIWARE.OpenSpecification.Security.DataHandlingGE.Open RESTful API Specification

7.7 Appendix

7.7.1 References

EC Data Protection directive	http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:EN:HTML

PPL Language	http://www.primelife.eu/results/documents/153-534d
XACML	http://xml.coverpages.org/xacml.html

7.8 Re-utilised Technologies/Specifications

The Repository GE is based on RESTful Design Principles. The technologies and specifications used in this GE are:

- RESTful web services
- HTTP/1.1
- JSON and XML data serialization formats

7.9 Terms and definitions

This section comprises a summary of terms and definitions introduced during the previous sections. It intends to establish a vocabulary that will be help to carry out discussions internally and with third parties (e.g., Use Case projects in the EU FP7 Future Internet PPP). For a summary of terms and definitions managed at overall FI-WARE level, please refer to FIWARE Global Terms and Definitions

- **Access Control:** This means to control access to resources such as web pages. This may be on the basis of the identity of the entity requesting access, or more generally the presentation of a set of credentials, and possibly some representation of the purpose for accessing the resource, as well as other contextual information, such as the time of day and properties of the resource itself.
- **Credentials:** A credential is an attestation of qualification, competence, or authority issued to an individual by a third party with a relevant de jure or de facto authority or assumed competence to do so. In this document, we define digital credentials to be lists of attribute-value statements certified by an Issuer. Here we abstract from the concrete mechanism (cryptographic or other) by which the authenticity of the attribute values can be verified. We do not impose any restrictions on which attributes can be contained in a credential, but typically these either describe the identity of the credential's owner or the authority assigned to her.
- **Personal Data (Personal Identifiable Information, or PII):** Personal data means any information relating to an identified or identifiable natural person or "Data Subject". An identifiable person is someone who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his or her physical, physiological, mental, economic, cultural or social identity. The processing of special categories of data, defined as personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, trade-union membership, and of data concerning health or sex life, is prohibited, subject to certain exceptions (see Article 10 of Regulation (EC) 45/2001). *(From the European Directive on the protection of personal data, Regulation (EC) 45/2001,*

article 2. OJ L 281, Nov. 23, 1995, available here)'

- **Data Controller:** The Data Controller means the entity which alone or jointly with others determines the purposes and means of the processing of personal data. The processing of personal data may be carried out by a Data Processor acting on behalf of the Data Controller. This document describes the means for a User Agent acting on behalf of a user to reach an agreement with a Data Controller over the obligations incurred by the controller for any personal data collected about that user.
- **Downstream Data Controller:** When a Data Controller passes personal data to a third party, that third party incurs obligations in respect to the Data Subject, and is referred to in this document as a "downstream data controller".
- **Data Subject:** he Data Subject is the person whose personal data are collected, held or processed by the Data Controller. *The following strictly speaking refers to EC institutions not generally to EU companies etc.* The controller must give the Data Subject the following information about the data being processed:

    1. confirmation as to whether or not data related to him or her are being processed;
    2. information about the purposes of the processing operation, the categories of data concerned, and the recipients or categories of recipients to whom the data are disclosed;
    3. communication of the data undergoing processing and of any available information as to their source;
    4. Knowledge of the logic involved in any automated decision process concerning him or her.

    - The Data Subject has the right to access his data and to require the Controller to rectify without delay any inaccurate or incomplete personal data. The Data Subject has the right to require the Controller to erase data if the processing is unlawful.

Data Subject's privacy preferences: The expectations of a Data Subject in terms of how her personal data should be handled.

Authorization and Obligations: Authorizations and Obligations define how and in which way a Data Subject authorizes a Data Controller to process her personal data. Obligations are negotiated together with Authorizations, and define what operations the Data Controller will perform at each authorized data usage. A specific process enables the Data Controller to propose obligations to the Data Subject, that matches them against her preferences. If the Data Subject is satisfied with the match, she will then authorize the Data Controller to proceed. The Data Controller is then required to implement the agreed obligations in respect to the Data Subject's personal data.
In a variant of this approach, the Data Subject could propose obligations to the Data Controller, who would then match them against her policies, and inform the Data Subject if the proposal is acceptable. The end result is the same — a binding agreement on the obligations for the Data Controller, to handle the Data Subject's personal data.

Sticky Privacy Policy: It is an agreement between Data Subject and Data Controller on the handling of personal data collected from the Data Subject. Sticky policies (as well as privacy preferences and privacy policies) define how data can be handled. A privacy policy becomes "sticky" to the data it regulates, after a specific negotiation process between Data Subject (the data owner) and Data Controller. Sticky policies define different aspects:

0. Authorizations:
    - Usage: what the Data Controller can do with collected data (e.g. use them for a specific purpose).
    - Downstream sharing: under which conditions data can be shared with another Data Controller.
1. Obligations: what the Data Controller must do.

Privacy Policy Language (PPL): The Privacy Policy Language (PPL) expresses access and usage control rules. It is based on the XACML standard. It permits to define privacy policies that regulate and specify obligations for the usage of personal information (PII).

DHP: This term refers to an acronym of Data Handling Policy/Preference. It as a policy configuration file stating the usage condition and handling of a targeted data. In the case of Policy it refers to the description of how the Data Controller will handle the data collected. In the case of Preference, the Data Subject specifies how his data should be handled after being collected.

User Agent: A software system (such as a web browser) acting on behalf of a user. The user agent acts on user preferences when dealing with a server acting on behalf of a Data Controller.

8 FIWARE OpenSpecification Security DataHandlingGE Open RESTful API Specification

You can find the content of this chapter as well in the wiki of fi-ware.

8.1 Introduction to the Data Handling GE API

The FI-WARE Generic Enabler Specification are owned by different partners. Therefore, different Legal Notices might apply. Please check for each FI-WARE Generic Enabler Specification the Legal Notice attached. For this FI-WARE Generic Enabler Specification, this Legal Notice applies.

8.1.1 Data Handling GE API Core

1. *CreatePII* is a RESTful method accessed via HTTP that uses XML-based information to create a new attribute in the Database. The method takes two parameters; one for the attribute name, and the second for the attribute value. For example incerting an e-mail address with the value of test@example.com. the same method can be used with a third parameter in order to attach a sticky policy to the new attribute.
2. *UpdatePII* is a RESTful method accessed via HTTP that uses XML-based information to update an existing attribute entry in the Database. The method takes two parameters; one for the attribute name, and the second for the new attribute value.
3. *DeletePII* is a RESTful method accessed via HTTP that uses XML-based information to delete an attribute from the DB. The method takes one parameter that is the attribute name.
4. *GetAllPII* is a RESTful method accessed via HTTP that uses XML-based information to reteinve all the attributes stored in the DB.
5. *CreatePreferenceGroups* is a RESTful method accessed via HTTP that uses XML-based information to aggregate a set of attributes under a certain group sharing the same Preferences or Sticky Policies. For example all the attribute related to the professional activity of a user can be grouped in a single group with the same privacy preferences.
6. *GetOnePrefeGroup* is a RESTful method accessed via HTTP that uses XML-based information to retreinve all the information related to a specific preference group.
7. *GetAllPrefGroups* is a RESTful method accessed via HTTP that uses XML-based information to retreinve all the preference groups information.
8. *UpdatePrefGroup* is a RESTful method accessed via HTTP that uses XML-based information to update or add new Preferences to the group
9. *DeletePrefGroup* is a RESTful method accessed via HTTP that uses XML-based information to delete a specific prefenrece group.

8.1.2 Intended Audience

This specification is intended for Service Consumers (with development skills), Cloud Providers and reimplementers of this API. For Service Customers, this document provides a full specification of how to interoperate with the Data Handling Service API. For Cloud Providers, this specification indicates the interface to be provided to the client application developers to provide the described functionalities. To use this information, the reader should firstly have a general understanding of the Generic Enabler service Data Handling Generic Enabler . The API user should be familiar with:

- ReSTful web services
- HTTP/1.1
- JSON and/or XML data serialization formats.

8.1.3 API Change History

Current version is: **Version 1.0.0, 27/04/2012**

The most recent changes are described in the table below:

Revision Date	Changes Summary
Apr 27, 2012	• Version 1 of the Data Handling GE API Guide.

8.1.4 How to Read This Document

In the whole document it is taken the assumption that reader is familiarized with REST architecture style. Along the document, some special notations are applied to differentiate some special words or concepts. The following list summarizes these special notations.

- A bold, mono-spaced font is used to represent code or logical entities, e.g., HTTP method (GET, PUT, POST, DELETE).
- An italic font is used to represent document titles or some other kind of special text, e.g., URI.
- The variables are represented between brackets, e.g. {id} and in italic font. When the reader find it, can change it by any value.
- <add any other content that you think that it is relevant>

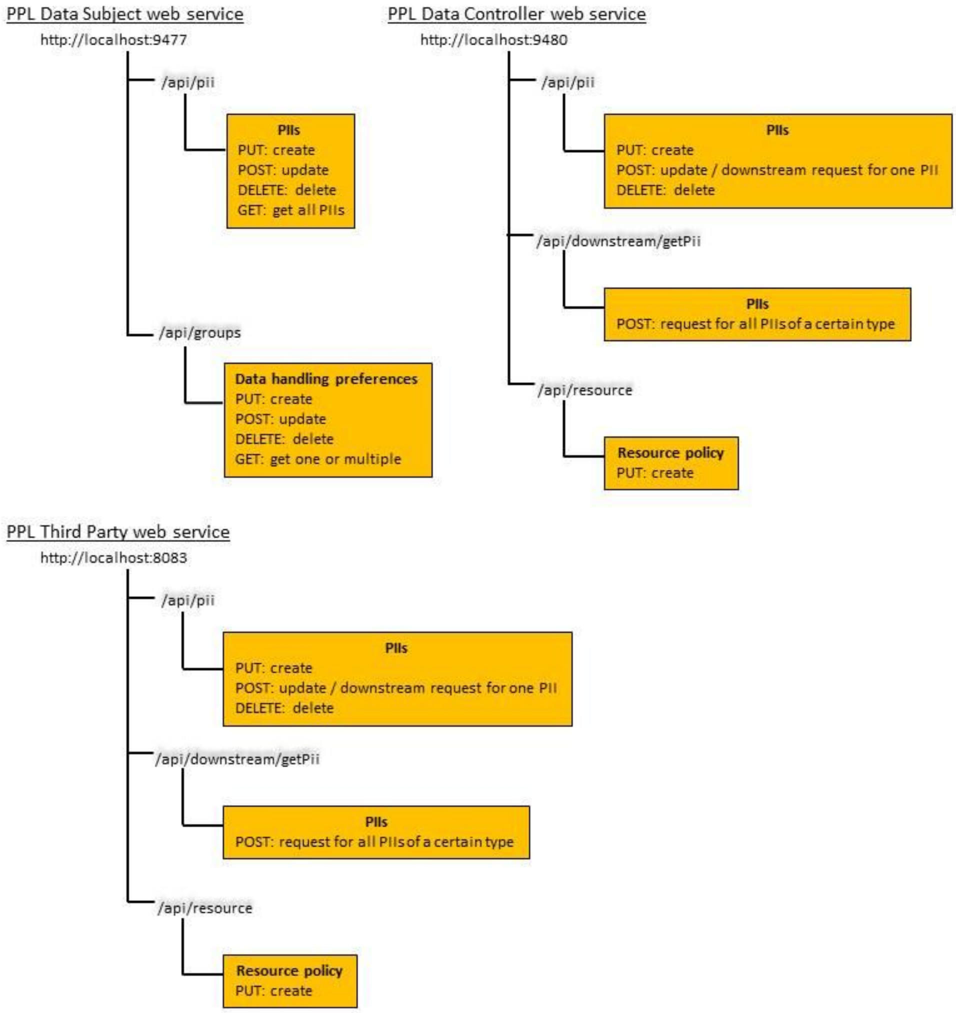
8.1.5 Additional Resources

More documentation related to the architecture and the usecase is available at Data Handling Generic Enabler

8.2 General Data Handling GE API Information

8.2.1 Resources Summary

Graphical diagram in which we can see the different URIs that we can use in the API. It should be something similar to the following one used in the NGSI-10-RestfulBinding-Draft.



8.2.2 Authentication

No additional authentication information is required by the service, except for those foreseen by the hosting platform (if any).

8.2.3 Representation Format

The Data Handling GE API supports the transmission of Strings and XML files. The request format is specified using the Content-Type header and is required for operations that have a request body. The response format is always in plain text ("text/plain").

8.2.4 Representation Transport

Resource representation is transmitted between client and server by using HTTP 1.1 protocol, as defined by IETF RFC-2616. Each time an HTTP request contains payload, a Content-Type header shall be used to specify the MIME type of wrapped representation. In addition, both client and server may use as many HTTP headers as they consider necessary.

8.3 API Operations

8.3.1 Data Subject Functionalities

8.3.1.1 *Create a PII*

URL	/api/pii	Method	PUT
Parameters		**Acceptable values - Description**	
name		string - Name of the PII	
value		string - Value of the PII	
Return value		**Status**	
Empty body		201	
Errors		**Description**	
400		PII with name <name> already exists in the PII store	
400		Missing parameters: name and/or value	

8.3.1.2 *Update PII*

URL	/api/pii	Method	POST
Parameters		**Acceptable values - Description**	
name		string - Name of the PII	
value		string - Value of the PII	
Return value		**Status**	
Empty body		200	

Errors	Description
404	PII with name <name> not found
400	Missing parameters: name and/or value

8.3.1.3 *Delete PII*

URL	/api/pii	Method	DELETE
Parameters		**Acceptable values - Description**	
name		string - Name of the PII	
Return value		**Status**	
Empty body		200	
Errors		**Description**	
404		PII with name <name> not found	
400		Missing parameters: name	

8.3.1.4 *GetAllPII*

URL	/api/pii	Method	GET
Return value		**Status**	
JSON or XML, depends on accept header, by default JSON		200	

8.3.1.5 *Create preference group*

URL	/api/groups	Method	PUT
Body			
String representation of the resource policy in the XML format. The root element of the policy document must be http://www.primelife.eu/ppl:Policy or http://www.primelife.eu/ppl:PolicySet.			
Return value		**Status**	

id in text format	201
Errors	**Description**
500	Preference policy cannot be empty
500	Failed to create a preference group

8.3.1.6 *GetOnePrefGroupbyId*

URL	/api/groups	Method	GET
Parameters		**Acceptable values - Description**	
id		string – identifier of the preference group	
Return value		**Status**	
The preference policy in XML		200	
Errors		**Description**	
404		The preference group <ID> does not exist	

8.3.1.7 *GetAllPrefGroupsId*

URL	/api/groups	Method	GET
Return value		**Status**	
List of groups identifiers in XML format		200	

8.3.1.8 *Update a preference group*

URL	/api/groups	Method	POST
Parameters		**Acceptable values - Description**	
group id		string - identifier of the preference group	
new preference group (optional)		string - identifier of the new preference group	
policy		string - the policy of the resource the user is trying to gain access to	

Return value	**Status**
Preference group successfully updated	200
Errors	**Description**
400	Error during processing
400	Missing parameter preference_group or missing policy

8.3.1.9 *Delete a preference group*

URL	/api/groups	Method	DELETE
Parameters		**Acceptable values - Description**	
id		string - identifier of the preference group	
Return value		**Status**	
Preference group successfully deleted		200	
Errors		**Description**	
404		Preference group not found	
400		Missing parameter id	

8.3.2 API Operations for Data Controller and Third Party

8.3.2.1 *Create PII*

URL	/api/pii	Method	PUT
Parameters		**Acceptable values - Description**	
name		string - Name of the PII	
value		string - Value of the PII	
policy		string - A valid sticky policy	
Return value		**Status**	

id of the PII created in text	201
Errors	**Description**
500	Failed to delete the PII
400	A parameter is missing

8.3.2.2 *Delete PII*

URL	/api/pii	Method	DELETE
Parameters	**Acceptable values - Description**		
id	long – PII identifier		
Return value	**Status**		
Empty body	200		
Errors	**Description**		
500	Failed to delete the PII		
404	PII not found		

8.3.2.3 *Update PII*

URL	/api/pii	Method	POST
Parameters		**Acceptable values - Description**	
piiId		long – PII identifier	
name		string – name of the PII	
value		string – value of the PII	
Body			
String representation of the request policy in the XML format. The root element of the policy document must be: urn:oasis:names:tc:SAML:2.0:assertion:Assertion.			
Return value		**Status**	

PII successfully updated.	200
Errors	**Description**
404	A PII with this ID does not exist
500	Failed to update PII
400	One of the parameters (name, value) is missing

8.3.2.4 *Downstream request for one PII*

URL	/api/pii	Method	POST
Body			
String representation of the request policy in the XML format. The root element of the policy document must be: urn:oasis:names:tc:SAML:2.0:assertion:Assertion.			
Return value		**Status**	
String representation of the claims in the XML format. The root element of the document is http://www.primelife.eu/ppl/claims:Claims.		200	
Errors		**Description**	
500		Downstream usage request for single PII failed	

8.3.2.5 *Downstream request for all PIIs of a certain type*

URL	/api/downstream/getPii	Method	POST
Body			
String representation of the request policy in the XML format. The root element of the policy document must be: urn:oasis:names:tc:SAML:2.0:assertion:Assertion.			
Return value			**Status**
String representation of the claims in the XML format. The root element of the document is http://www.primelife.eu/ppl/claims:Claims.			200

Errors	**Description**
503	An error occurred during the processing
503	No assertion provided

9 FIWARE OpenSpecification Security Optional Security Enablers DBAnonymizer

You can find the content of this chapter as well in the wiki of fi-ware.

Name	FIWARE.OpenSpecification.Security.Optional Security Enablers.DBAnonymizer
Chapter	Security,
Catalogue-Link to Implementation	DB Anonymizer
Owner	SAP, Francesco Di Cerbo

9.1 Preface

Within this document you find a self-contained open specification of a FI-WARE generic enabler, please consult as well the FI-WARE_Product_Vision, the website on http://www.fi-ware.eu and similar pages in order to understand the complete context of the FI-WARE project.

9.2 Copyright

-

9.3 Legal Notice

Please check the following Legal Notice to understand the rights to use these specifications.

SAP strives to make the specifications of this Generic Enabler available under IPR rules that allow for a exploitation and sustainable usage both in Open Source as well as proprietary, closed source products to maximize adoption.

9.4 Overview

Large Organizations hold thousands of terabytes of datasets about their customers or their activities. They often have to release data files containing private information to third parties for data analysis, application testing or support. To preserve individuals' privacy and comply with privacy regulations, part of released datasets have to be hidden or anonymized using various anonymization techniques, before data release.

However, two different problems may arise: first to decide if a piece of data has to be considered private or not, second, to assess whether the exposure of non-private data could be used by correlation algorithms to infer hidden private data. The second task is particularly challenging, and cannot be handled manually for large datasets, where the potential number of combinations of different fields is extremely large. In fact, disclosure policies are typically described by human users (security experts or not) that are not able to predict all the possible combinations of the data that could ease the guess of private data contained in the dataset. In some other cases, policy writers are not necessarily security experts and could expose sensitive data without being aware about the impact of such exposure.

DB Anonymizer is a database risk evaluation and anonymization service; it can be used as a support tool in case of dataset disclosure. In fact, DB Anonymizer analyses whether a shared database is not vulnerable to the re-identification of the non-shared part of the database. In other words, the service allows understanding if a certain policy to be used to anonymize a dataset should be considered safe or not. In particular, the service exposes a function that calculates a value, that represents the likelihood (0->impossibility, 1->certainty) that an attacker can reconstruct exactly a table's content, that is anonymized using a certain obfuscation policy.

9.4.1 Target usage

The service can be used to support these DB administrators to evaluate the disclosure risk for all their types of data; by recommending the safest configurations using a smart bootstrapping system. The service provides the user with an estimation of the re-identification risk when disclosing certain information, and proposes safe combinations in order to help him during the information disclosure. Albeit privacy risk estimators have already been developed in some specific contexts (statistical databases), they have had limited impact, since they are often too specific for a given context, and do not provide the user with the necessary feedback to mitigate the risk. In addition, they can be computationally expensive on large datasets. DB Anonymizer is specifically designed to address all these issues, exposing a simple ReSTful API that can be easily integrated in any application.

9.5 Basic Concepts

The service needs to receive a dump of a MySQL table, containing all data, together with a disclosure policy. Both inputs are mandatory to let the service's algorithm to be able to evaluate the effectiveness of the disclosure policy. Once the policy is evaluated, the table is dropped from the DB and the file dump is erased. The application server encapsulation model permits a complete isolation of each user's data, and any intermediate result created during the algorithm's execution is deleted immediately at the end of the computation.

9.5.1 Input Format

The main function has two parameters:

1. an SQL table dump (for MySQL), containing all information to be disclosed: this file shall contain only a table definition and a set of elements to populate it;
2. a policy file in XML, which describes which information of the previously specified table are going to be disclosed or not: the policy file has the following syntax:

```
<Policy>
  <Column>
    <Name>Gender</Name>
    <Type>identifier</Type>
    <Hide>false</Hide>
  </Column>
  <Column>
    <Name>Wine</Name>
    <Type>sensitive</Type>
    <Hide>false</Hide>
  </Column>
</Policy>
```

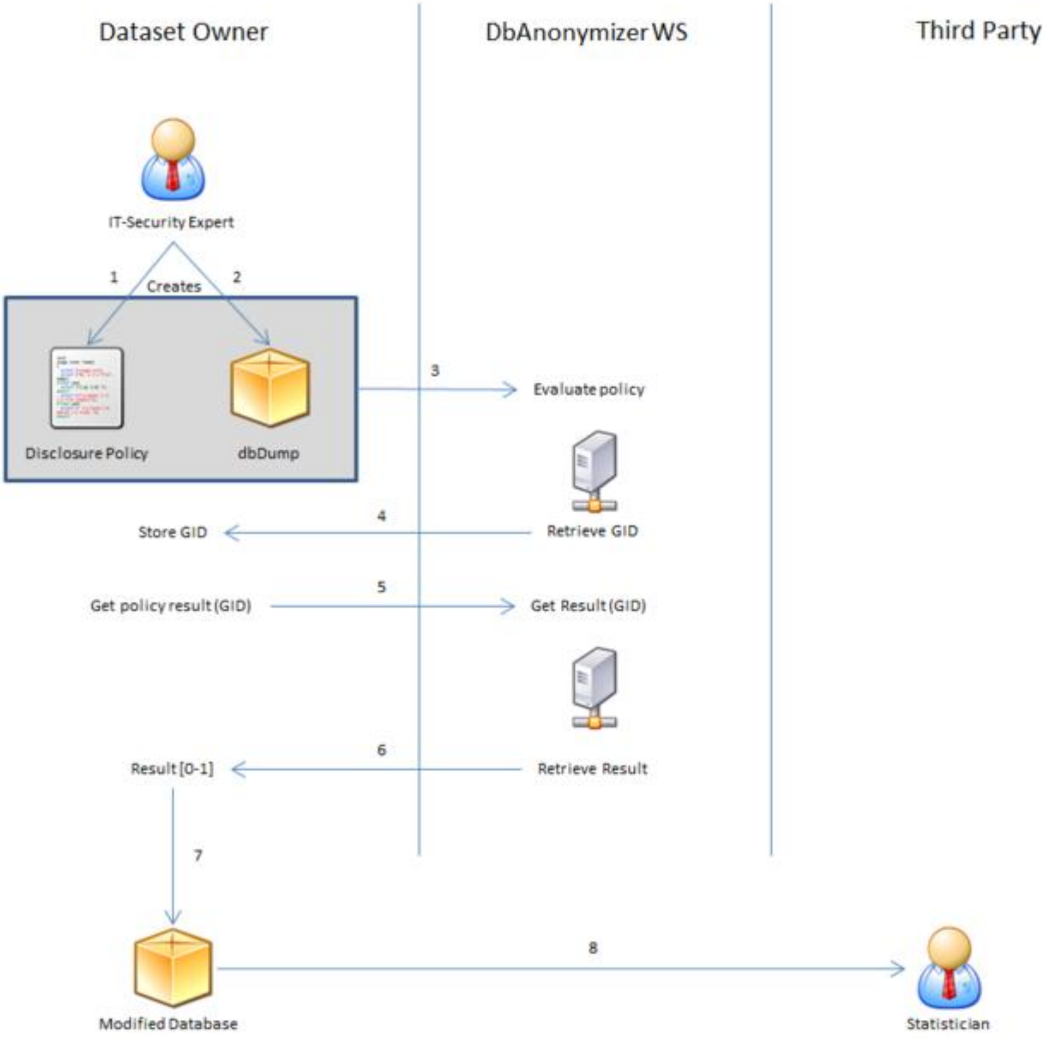
The “Type” information shall be “identifier” or “sensitive”, in order to allow the service algorithm to distinguish them. Please refer to the Glossary (at this link: FIWARE.Glossary.Security.Optional Security Enablers.DBAnonymizer) for an explanation of the two terms.

The service supports MTOM, a W3C standard which allows services to transfer binary data efficiently and conveniently (see http://www.w3.org/TR/soap12-mtom/).

9.5.2 Use Case

A company holds information about people structured in dataset records. Each record has many attributes, such as birthday, address, marital status and occupation, that are useful for company's purposes, but usually are not sensitive, if considered in isolation. Other attributes related to the connection between an individual and the company, such as customer purchases, debts, and credit rating, may be sensitive. Suppose a dataset is released with obvious identifiers, such as social security number, name and address, omitted, some other attributes such as occupation and marital status left intact, and other key and sensitive attributes modified to preserve confidentiality. For example, salaries might be truncated, ages grouped more coarsely, and zip codes swapped on pairs of records. Furthermore, some attributes on some records might be missing or intentionally removed. If the anonymization process is not carefully designed, it could be possible to use techniques to reconstruct the original dataset, as a whole or in parts, also by cross-comparing it with other datasets (e.g., a similar dataset of a competitor). The DB Anonymizer allows to evaluate an anonymization policy, in order to measure its robustness to dataset reconstruction techniques.

Let us consider the following example.

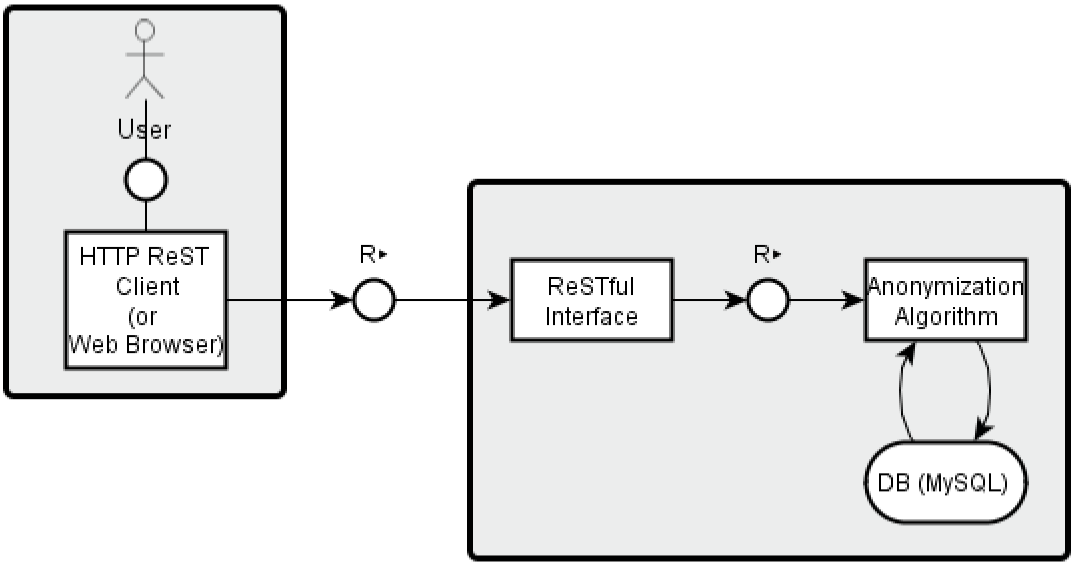


1. The IT-Security Expert creates the Disclosure Policy.
2. The IT-Security Expert creates the DB Dump.
3. DB Dump and Disclosure Policy are sent to the DbAnonymizer WS using the evaluate policy.
4. The DbAnonymizer WS sends back the Result Identifier (GID).
5. The IT-Security Expert ask for the evaluation result.
6. The DbAnonymizer WS sends back the evaluation result.
7. The IT-Security Expert modifies the DB data, according to the accepted policy.
8. The modified DB dump is sent to the Consulting Company.

9.6 Main Interactions

9.6.1 Architecture

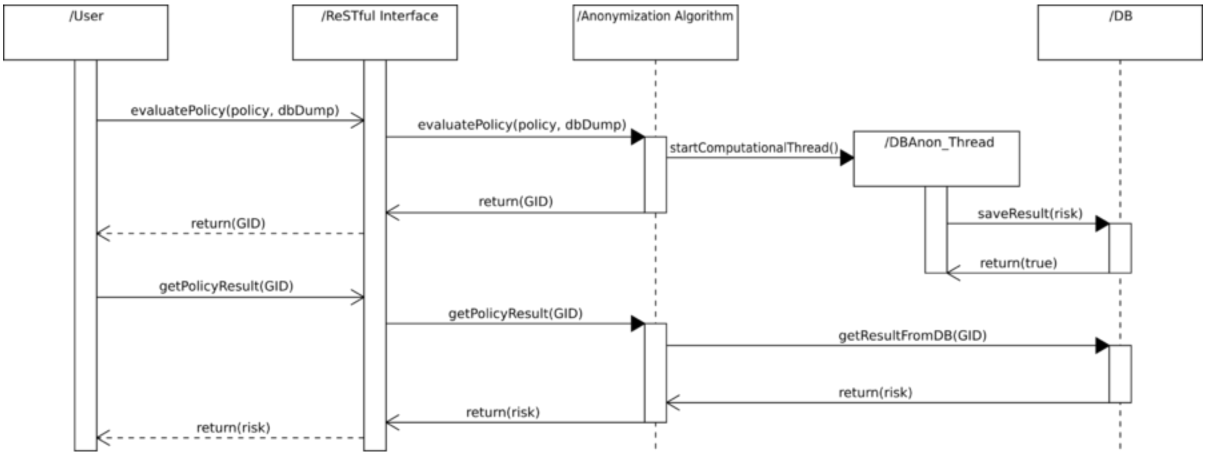
9.6.1.1 *Block Diagram*



The previous block diagram shows the different elements that compose the DB Anonymizer service. Starting from the DB Anonymizer block (on the right side of the diagram), the core of DB Anonymizer is the Anonymization Algorithm component, that interact closely with an internal MySQL database. The Anonymization Algorithm interacts with users through another component, that exposes a ReSTful interface. More precisely, the ReSTful interface component is responsible for invoking the Anonymization Algorighm operations, and to provision them with user inputs.

In the left part of the block diagram, a user is depicted together with a ReSTful client component, with which it is possible to interact with DB Anonymizer ReSTful interface. The ReSTful client can also be implemented by a traditional web browser.

9.6.1.2 *Sequence Diagram*



The previous block diagram shows the order with which DB Anonymizer operations should be invoked; the entities depicted are the same as for the previous block diagram.

The DB Anonymizer API is composed by two methods, to be invoked by users in the following order:

1. evaluatePolicy;
2. getPolicyResult.

The first method allows for starting the analysis of an anonymization policy together with the associated dataset. The ReSTful interface component exposes this method, and any incoming request get routed and served by the Anonymization Algorigthm component, that creates a new computing process. The Anonymization Algorithm component returns immediately a request identifier (GID) to the ReSTful component and thus to the user, which can be used to retrieve the analysis result. Each computation process performs its analysis on the received policy and dataset, and then writes a result to the DB. At that point, the process terminates, deleting any data it used. The second method can be invoked by users to retrieve the result of a computation, identified by a GID. The result of getPolicyResult can be the analysis result when available, otherwise an error code (result is not ready, error in receiving parameters and so on; please refer to the ReSTful API documentation for a detailed error code list and explanation).

9.7 Basic Design Principles

The service manipulates user data in a secure way; dataset and policies are deleted from the application just after their use, to keep confidential any information transmitted. A temporary MySQL table is created at the beginning of the operations, and destroyed just before returning the final result to the invoker. A new process is created for each user request, to ensure data isolation during computation phases.

9.8 Detailed Specifications

Following is a list of Open Specifications linked to this Generic Enabler. Specifications labeled as "PRELIMINARY" are considered stable but subject to minor changes derived from lessons learned during last interactions of the development of a first reference implementation planned for the current Major Release of FI-WARE.

9.8.1 Open API Specifications

- FIWARE.OpenSpecification.Security.DBAnonymizer.Open RESTful API Specification

9.9 Re-utilised Technologies/Specifications

The Repository GE is based on RESTful Design Principles. The technologies and specifications used in this GE are:

- RESTful web services
- HTTP/1.1
- JSON and XML data serialization formats

9.10 Terms and definitions

This section comprises a summary of terms and definitions introduced during the previous sections. It intends to establish a vocabulary that will be help to carry out discussions internally and with third parties (e.g., Use Case projects in the EU FP7 Future Internet PPP). For a summary of terms and definitions managed at overall FI-WARE level, please refer to FIWARE Global Terms and Definitions

- **Data Disclosure:** A release of information to a third party or to public. Data can be confidential, so the operation might require the adoption of techniques that aim at preserving confidentiality and privacy of involved subjects, like, for instance, the hiding a part of the dataset, like for instance names, surnames, social security numbers and so on, generally referred as "identifiers".
- **Identifier:** A piece of information that can identify unambiguously a person or an entity: for instance, names, surnames, social security and passport numbers, and so on.
- **Quasi-Identifier:** Attributes such as birth date, gender and postal code, that cannot identify unambiguously a person or an entity if they are considered in isolation, but they could, if considered aggregated with enough similar attributes.
- **Re-identification risk:** An estimation of the risk that an attacker can reconstruct the contents of a dataset, disclosed without identifier information, by linking it with other external data sources with overlapping attributes with the released dataset.
- **Sensitive information:** A piece of information that reveals "racial or ethnic origin, political opinions, religious or philosophical beliefs or trade union membership, data concerning health or sex life, and data relating to offences, criminal convictions or security measures.", from Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data (OJ L 281, Nov. 23, 1995, available here)

10 FIWARE OpenSpecification Security DBAnonymizer Open RESTful API Specification

You can find the content of this chapter as well in the wiki of fi-ware.

10.1 Introduction to the DB Anonymizer API

The FI-WARE Generic Enabler Specification are owned by different partners. Therefore, different Legal Notices might apply. Please check for each FI-WARE Generic Enabler Specification the Legal Notice attached. For this FI-WARE Generic Enabler Specification, this Legal Notice applies.

10.1.1 DB Anonymizer API Core

The DB Anonymizer API is a RESTful API accessed via HTTP. It uses simple data types or binary files for the information exchange. It offers two main functions:

1. *evaluatePolicy* receives a DB dump (a single table in MySQL) and an obfuscation policy file, to compute the likelihood (0->impossibility, 1->certainty) that an attacker can reconstruct exactly the table's content, if it is anonymized using the obfuscation policy;
2. *getPolicyResult* to retrieve the result of the computation.

10.1.2 Intended Audience

This specification is intended for software developers and reimplementers of this API. For the former, this document provides a full specification of how to interoperate with DB Anonymizer service, that implements DB Anonymizer API.

To use this information, the reader should firstly have a general understanding of the Generic Enabler service (available on DBAnonymizer). The reader should also be familiar with:

- ReSTful web services
- HTTP/1.1
- JSON and XML data serialization formats.

10.1.3 API Change History

Current version is: **Version 1.0.0, 30/4/2012**

The most recent changes are described in the table below:

Revision Date	Changes Summary
Apr 30, 2012	• This is the first version of the DB Anonymizer API Guide.

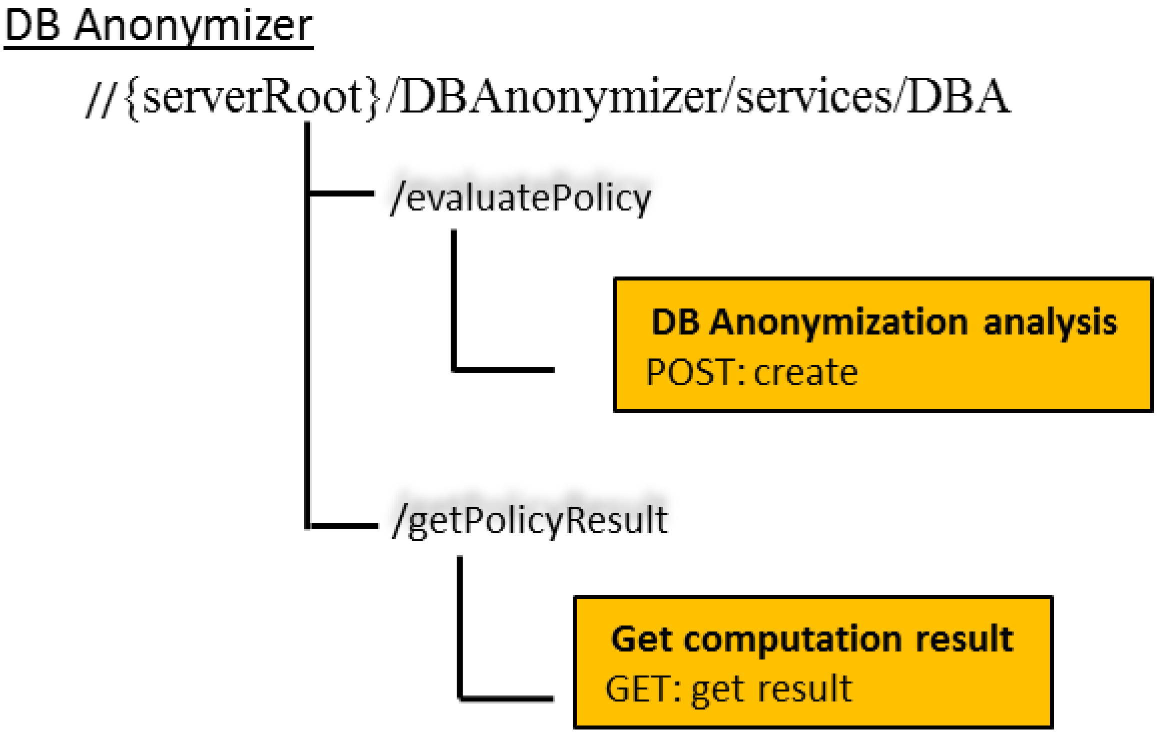
10.1.4 How to Read This Document

In the whole document the assumption is made that the reader is familiarized with REST architecture style. However, the interface was carefully designed to be extremely simple to use, thus to require minimal integration effort from software developers interested in the DB Anonymizer functionalities. Therefore, no special notation or particular constructs were needed in producing this description.

10.2 General DB Anonymizer API Information

10.2.1 Resources Summary

Graphical diagram in which we can see the different URIs exposed in the API.



10.2.2 Authentication

No additional authentication information are required by the service, except for those foreseen by the hosting platform (if any).

10.2.3 Representation Format

The DB Anonymizer API supports the transmission of binary files and strings via HTML FORM. The request format is specified using the Content-Type header and is required for operations that have a request body. The response format is always in plain text ("text/plain").

10.2.4 Representation Transport

Resource representation is transmitted between client and server by using HTTP 1.1 protocol, as defined by IETF RFC-2616. Each time an HTTP request contains payload, a Content-Type header shall be used to specify the MIME type of wrapped representation. In addition, both client and server may use as many HTTP headers as they consider necessary.

10.2.5 Resource Identification

- evaluatePolicy: The following resources must be provided in a HTML FORM ("multipart/form-data")
  - id: "dbDump" --> a zipped DB dump (MySQL), containing a single table called "working_table";
  - id: "policyFile" --> an XML policy file, compliant to this XML Schema, and expressed as explained at: DB Anonymizer Input Format
- getPolicyResult: Just a string containing a long integer is required, in GET or as a URL parameter.

10.2.6 Links and References

Reference to Open Specification, DB Anonymizer

10.2.7 Limits

We can manage the capacity of the system in order to prevent the abuse of the system through some limitations. These limitations will be configured by the operator and may differ from one implementation to the other of the GE implementation.

10.2.7.1 *Rate Limits*

Given the specificity of the service, i.e., analysing database dumps with possibly significant amount of data, it is not foreseen a massive amount of requests per user at the same time. Thresholds might be set, in order to limit the frequencies of new incoming requests. As an estimation, a DB table of ~500 MB gets processed by a single-core 3.0 GHz service in about 3 minutes.

10.2.7.2 *Absolute Limits*

Absolute limits for the service are set in 2GB RAM (aggregating all requests). However, DB table sizes influence the total size in a limited way.

10.2.7.3 *Determining Limits Programmatically*

At present, the API does not allow to retrieve any usage limit programmatically.

10.2.8 Versions

Only one version of the Open Specification is currently supported.

10.2.9 Extensions

DB Anonymizer could be extended in the future. At the moment, we foresee the following resource to indicate a method that will be used in order to allow the extensibility of the API. This allows the introduction of new features in the API without requiring an update of the version, for instance, or to allow the introduction of vendor specific functionality. When extensions will be available, this method will be available. In FI-WARE Release 1, this method is not available.

Applications could recover this information through the following request

Verb	**URI**	**Description**
GET	/extensions	List of all available extensions

10.2.10 Faults

10.2.10.1 *Synchronous Faults*

Fault Element	**Associated Error Codes**	**Expected in All Requests?**	**Return Message**
GET /getPolicyResult	HTTP 204	NO	Error in retrieving the requested result
GET /getPolicyResult	HTTP 400	NO	Error in Request ID
GET /getPolicyResult	HTTP 400	NO	Error: The DB file is not in ZIP format
GET /getPolicyResult	HTTP 400	NO	Error: Problem with input file
GET /getPolicyResult	HTTP 400	NO	Error: Problem with input DB dump
GET /getPolicyResult	HTTP 400	NO	Error: fault in policy parsing and/or setting
GET /getPolicyResult	HTTP 500	NO	Error: DB communication problem
GET /getPolicyResult	HTTP 500	NO	Error: fault in DB setup

Remark: HTTP Status 204 in response to /getPolicyResult indicates that computation result is not yet available (coherently with the HTTP Status definition "No Content").

10.2.10.2 *Asynchronous Faults*

No Asynchronous Faults are used by DB Anonymizer

10.3 API Operations

10.3.1 Operations

Verb	URI	Description
POST	/evaluatePolicy	starts the computation on the input: a MySQL DB table dump and a disclosure policy.
GET	/getPolicyResult/{RequestID}	Retrieve all available information about the context entity (flat, without attribute domains)

Description: *evaluatePolicy*

- Correct Response: HTTP 200
- Input:
    - a zipped MySQL table dump, containing only a single table called "working_table", together with its elements. Allowed SQL commands: CREATE TABLE, INSERT
    - a disclosure policy file, compliant with this XML Schema definition, for example:

```
<Policy>
       <Column>
              <Name>Gender</Name>
              <Type>identifier</Type>
              <Hide>false</Hide>
       </Column>
       <Column>
              <Name>Wine</Name>
              <Type>sensitive</Type>
              <Hide>true</Hide>
       </Column>
</Policy>
```

- Return type: it returns a RequestID (string).

A sample of the required inputs is available on the FI-WARE Catalogue at this link.

Description: *getPolicyResult*

- Correct Response: HTTP 200 .
- Alternative: HTTP 204 (No Content), when computation result is not ready.
- Input: a RequestID (string)

- Return type: the likelihood (0->impossibility, 1->certainty) that an attacker can reconstruct exactly a table's content, that is anonymized using a certain obfuscation policy.

11 FIWARE OpenSpecification Security IdentityManagement

You can find the content of this chapter as well in the wiki of fi-ware.

Name	FIWARE.OpenSpecification.Security.Identity Management Generic Enabler
Chapter	Security,
Catalogue-Link to Implementation	Identity Management
Owner	NSN, Robert Seidl

11.1 Preface

Within this document you find a self-contained open specification of a FI-WARE generic enabler, please consult as well the FI-WARE_Product_Vision, the website on http://www.fi-ware.eu and similar pages in order to understand the complete context of the FI-WARE project.

11.1.1 Copyright

11.1.2 Legal Notice

Please check the following Legal Notice to understand the rights to use these specifications.

11.1.3 Overview

On the one hand the ever-growing tsunami of today's shore-bound technologies can often overwhelm the user, significantly affecting his daily life. On a daily basis, he is forced to depend on his technological competence. The smooth running of his affairs depends on user's ability to handle a whole raft of often transient technologies. On account of very intensive, at times forced usage of the Internet and diverse services, the user encounters the need to transfer his "network-duties" to the networks as much as possible.

In other words, he seeks to find a convenient problem solver, which will allow him to cope easily and securely with services. Thus, the need arises for a clever composed Identity Management system, which will address the users' requirements.

Identity Management (IdM) encompasses a number of aspects involved with users' access to networks, services and applications, including secure and private authentication from users to devices, networks and services, Authorization & Trust management, User Profile management, Single Sign-On (SSO) to service domains and Identity Federation towards applications.

An IdM system is intended to undertake the complex task of handling, communicating with and coordinating between the slew of today's diverse technologies. Provide user-friendly technologies, putting the end user and his needs squarely at centre of the architecture (user-centric approach) whilst protecting the users' privacy.

On the other hand the computing resources are being actively exploited by the Enterprises lately through the use of cloudification and virtualization technologies. Nevertheless, with regard to such an evolution on the Web, the Enterprises still have to keep in mind the Identity Management issues and should be able to deliver such technologies to their customers. Thus, the Identity Management Enabler could also deliver a multi-tenant user and profile management solution that allows Enterprises to manage consumers of their (Web based) services in the Cloud securely. Instead of developing and operating the user and profile management by themselves, it can be hosted in the Cloud as a tenant instance and will be delivered on demand.

11.1.3.1 ***Target usage***

This enabler provides authentication/access control and identity/attribute assertions as a service to relying parties. The relying parties are typically service providers that provide easy and secure access to their services to users/IoT/other services for instance by means of SSO and that rely on (personal user) attributes (e.g. preferences, location, home address, etc). The users need easy access (SSO) to the growing number of services, and many of them also prefer their personal/identity attributes to be maintained by a trusted party which also protects the users' privacy. The Identity Management core generic enabler can be used by such a trusted party which we also call an identity provider (for SSO) and attribute broker. The Identity Management GE is a core Security GE that provides services to its relying parties via open protocols such as OAuth [OAuth] and OASIS SAML v2.0 [Saml] (Security Assertion Markup Language). Motivated by the IoT, the enabler also covers new user attributes such as things, as well as it manages the identity of things themselves (attributes, current users, location, use history, etc). The large number of sensors and mobile devices poses new challenges; identity federation and single-sign-on support ease of use. Furthermore, the authentication feature of the enabler also covers the authentication of things for services, other objects or users as relying parties, and the authentication of users, services and other things for things as relying parties. It also supports user SSO across multiple things. Motivated by Cloud computing, the enabler can be run in the cloud as well; when doing so. Special care is taken so that the sensitive data is not exposed to the threats related to the nature of clouds (e.g. deployment in a public cloud).

11.1.4 Basic Concepts

Identity Management encompasses a number of aspects involved with users' access to networks, services and applications, including secure and private authentication from users to devices, networks and services, Authorisation & Trust management, User Profile management, Single Sign-On (SSO) to service domains and Identity Federation towards applications. The Identity Manager is the central component that provides a bridge between IdM systems at connectivity-level and application-level.

Identity Management is used in multiple scenarios spanning from Operator oriented scenarios towards Internet Service Providers (ISP). End users benefit from having simplified and easy access to services (User Centric Identity Management). In the following basic concepts supporting the above mentioned features are described:

11.1.4.1 *User Life-Cycle Management*

The IdM offers tools for administrators to support the handling of user life-cycle functions. It reduces the effort for account creation and management, as it supports the enforcement of policies and procedures for user registration, user profile management and the modification of user accounts. Administrators can quickly configure customized pages for the inclusion of different authentication providers, registration of tenant applications with access to user profile data and the handling of error notifications. For end users, the IdM provides a convenient solution for registering with applications since it gives them a means to re-use attributes like address, email or others, thus allowing an easy and convenient management of profile information. Users and administrators can rely on standardized solutions to allow user self-service features like:

- User registration and login resp. logout,
- Checks for password strength,
- Password reset or renewal procedures or
- Secured storage of user data.

11.1.4.2 *Flexible Authentication Providers*

In addition to providing a native login, the Identity Provider (IdP) supports the integration of multiple 3rd party authentication providers. Foremost, it supports in a first step the configuration of preferred identity providers to lower entry barriers for a native user registration to administrators and on user side to link a preferred 3rd party IdP as alternative authentication provider to a native account.

11.1.4.3 *3rd Party Login*

3rd party login supports customers of the IdM to enhance the reach of their websites by means of attracting users without forcing them to register new user accounts on their sites manually. 3rd party login allows users to register to the customers' sites with already existing digital identities from their favourite 3rd party identity providers, such as e.g. Google, Facebook or Yahoo. Thus, 3rd party login lowers the obstacles of registration processes and increases the number of successful business flows on the customers' sites.

11.1.4.4 *Web Single Sign-On*

As it is possible to configure several applications that shall be linked to his IdM, the main benefit for users is a single sign-on (SSO) to all these applications.

11.1.4.5 *Hosted User Profile Management*

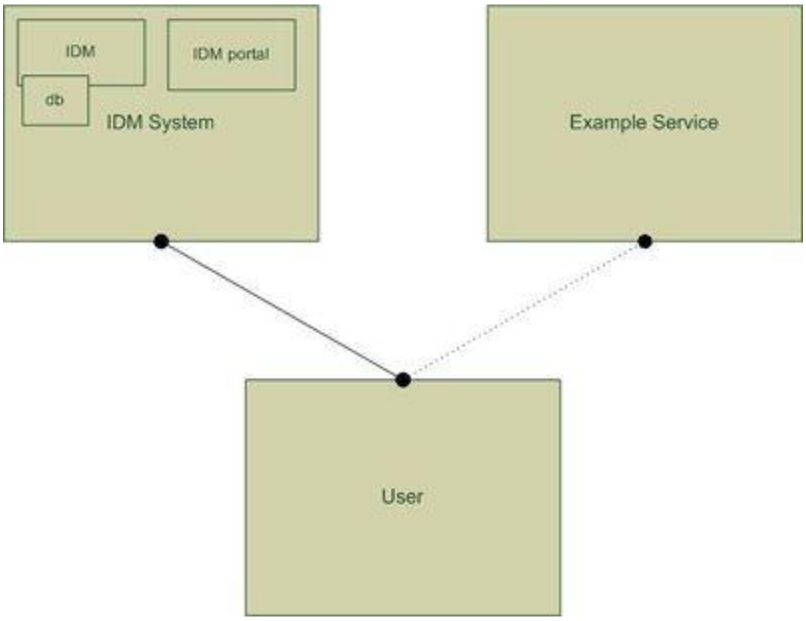
The IdM offers hosted user profile storage with specific user profile attributes. Applications do not have to run and manage their own persistent user data storages, but instead can use the IdM user profile storage as a SaaS offering.

11.1.4.6 *Multi-Tenancy*

A multi-tenancy architecture refers to a principle in software architecture where a single software instance runs on a server, serving multiple client organizations/customers (tenants). Multi-tenancy is contrasted with a multi-instance architecture where separate software instances (or hardware systems) are set up for different client organizations. With a multi-tenant architecture, a software application is designed to virtually partition its data and configuration, and each client organization works with a customized virtual application instance. In a multi-tenancy environment, multiple customers share the same application, running on the same operating system, on the same virtualized hardware, with the same data storage mechanism. The distinction between the customers is achieved during application design, thus customers do not share or see each other's data. The concept allows each tenant to apply their own branding to login or registration UIs or for user self-services to create a user experience that is aligned with the one offered in a tenant application.

11.1.5 Main Interactions

11.1.5.1 *Example Architecture*



Identity Generic Enabler - High Level Architecture

11.1.5.2 *Modules and Interfaces*

The Identity Management System consists of the following building blocks:

- *IDM Portal*

Providing the interface to the user / application. The functionality includes user profile management and modification of user accounts (e.g. password settings, secured storage of user data).

- IDM system

The core component handling the authentication requests of the users, by providing e.g. federated IDM for Web-SSO or basic authentication features for devices and services.

- Authentication framework

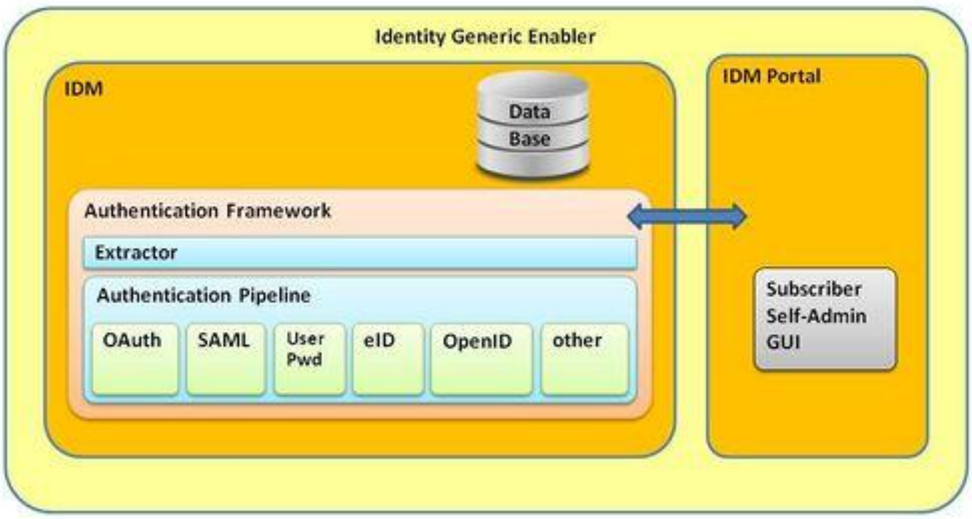
The authentication framework consists of the Extractor and the Authentication Pipeline. The Extractor extracts the authentication data from different sources. Each one of them is specialized in extracting a special kind of data. There exists a pipeline of authentication data extractors.

- Database

Central repository that stores the user data, profiles and preferences as well as the service provider preferences. This could be implemented as a distributed storage system, depending on the usage scenario.

- Supported Authentication Methods

> - OAuth stack
>
> Open Authorization Protocol is an open standard for authorization.
>
> - SAML stack
>
> Security Assertion Markup Language is an XML-based standard for exchanging authentication and authorization data between security domains
>
> - OpenID stack
>
> Open standard that describes how users can be authenticated in a decentralized manner
>
> - eID support
>
> The Generic enabler will support European Identity Cards of multiple memebr states.
>
> - user name / password
>
> As well basic authentication mechanisms like user name / password are provided.



Identity Generic Enabler - Core Components

Beyond that there may be some additional network security components necessary (e.g. Firewall, router, …). Next to the Public Key Infrastructer (PKI) this is a prerequisite for the Generic Identity Enabler.It supports the following authentication methods:

1. SAML (in a later version)

2. OAuth

3. OpenID

4. Username / Password

5. eID - cards

11.1.5.3 *Interface Descriptions*

Different authentication mechanisms are offered by the Identity Generic Enabler. It supports standardized interfaces as well as proprietary once. In the following the interfaces are described with the help of message flows and reference code examples, by thus offering an easy implementation and usage of the Generic Enabler.

SAML

The Identity Management GE makes use of SAML (Security Assertion Markup Language) 2.0 for authenticating federated relying parties and, after authenticating the Users on behalf of the federated relying parties in a second step, for informing them that these Users are authorized to access their services.

The advantages of SAML 2.0

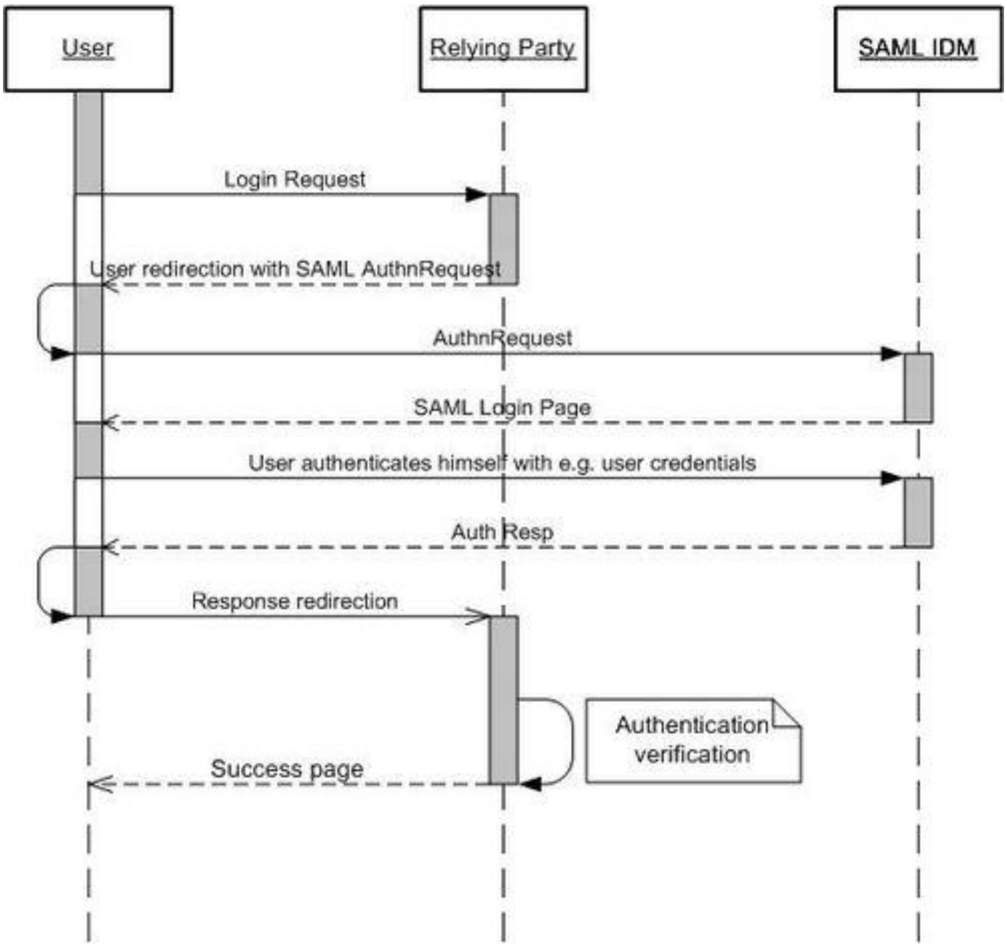
• Provides a means of exchanging data between security domains (i.e. the Identity Management GE and its federated service providers (relying parties))

• Provides the SSO feature for the federated service providers to the Users

• Service providers do not need to authenticate users themselves

• Provides security features such as digital signatures to certify the integrity of the exchanged data (and certified attributes)

• Standardized, non-proprietary protocol (e.g. also supported by Google)


User
Relying Party
SAML IDM
Login Request
User redirection with SAML AuthnRequest
AuthnRequest
SAML Login Page
User authenticates himself with e.g. user credentials
Auth Resp
Response redirection
Authentication verification
Success page

Identity Generic Enabler - SAML Authentication Flow

Authentication Request

```
<samlp:AuthnRequest
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  ID="aaf23196-1773-2113-474a-fe114412ab72"
  Version="2.0"
  IssueInstant="2004-12-05T09:21:59Z"
  AssertionConsumerServiceIndex="0"
  AttributeConsumingServiceIndex="0">
  <saml:Issuer>https://sp.example.com/SAML2</saml:Issuer>
  <samlp:NameIDPolicy
```

```
    AllowCreate="true"
    Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient"/>
</samlp:AuthnRequest>
```

Authentication Response

```
  <samlp:Response
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
ID="identifier_2"
InResponseTo="identifier_1"
Version="2.0"
IssueInstant="2004-12-05T09:22:05Z"
    Destination="https://sp.example.com/SAML2/SSO/POST">
    <saml:Issuer>https://idp.example.org/SAML2</saml:Issuer>
    <samlp:Status>
      <samlp:StatusCode
        Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </samlp:Status>
    <saml:Assertion
      xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
      ID="identifier_3"
      Version="2.0"
      IssueInstant="2004-12-05T09:22:05Z">
      <saml:Issuer>https://idp.example.org/SAML2</saml:Issuer>
      <ds:Signature

xmlns:ds="http://www.w3.org/2000/09/xmldsig#">...</ds:Signature>
      <saml:Subject>
        <saml:NameID
          Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:transient">
          3f7b3dcf-1674-4ecd-92c8-1544f346baf8
        </saml:NameID>
        <saml:SubjectConfirmation
          Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
          <saml:SubjectConfirmationData
            InResponseTo="identifier_1"
            Recipient="https://sp.example.com/SAML2/SSO/POST"
            NotOnOrAfter="2004-12-05T09:27:05Z"/>
        </saml:SubjectConfirmation>
      </saml:Subject>
      <saml:Conditions
        NotBefore="2004-12-05T09:17:05Z"
        NotOnOrAfter="2004-12-05T09:27:05Z">
        <saml:AudienceRestriction>

<saml:Audience>https://sp.example.com/SAML2</saml:Audience>
        </saml:AudienceRestriction>
      </saml:Conditions>
      <saml:AuthnStatement
        AuthnInstant="2004-12-05T09:22:00Z"
        SessionIndex="identifier_3">
        <saml:AuthnContext>
          <saml:AuthnContextClassRef>

urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
         </saml:AuthnContextClassRef>
        </saml:AuthnContext>
      </saml:AuthnStatement>
    </saml:Assertion>
  </samlp:Response>
```

OAuth

Contrary to SAML, OAuth has nothing to do with SSO. Main focus is not on attributes, but ‘resources’ of the user which a consumer wishes to access.

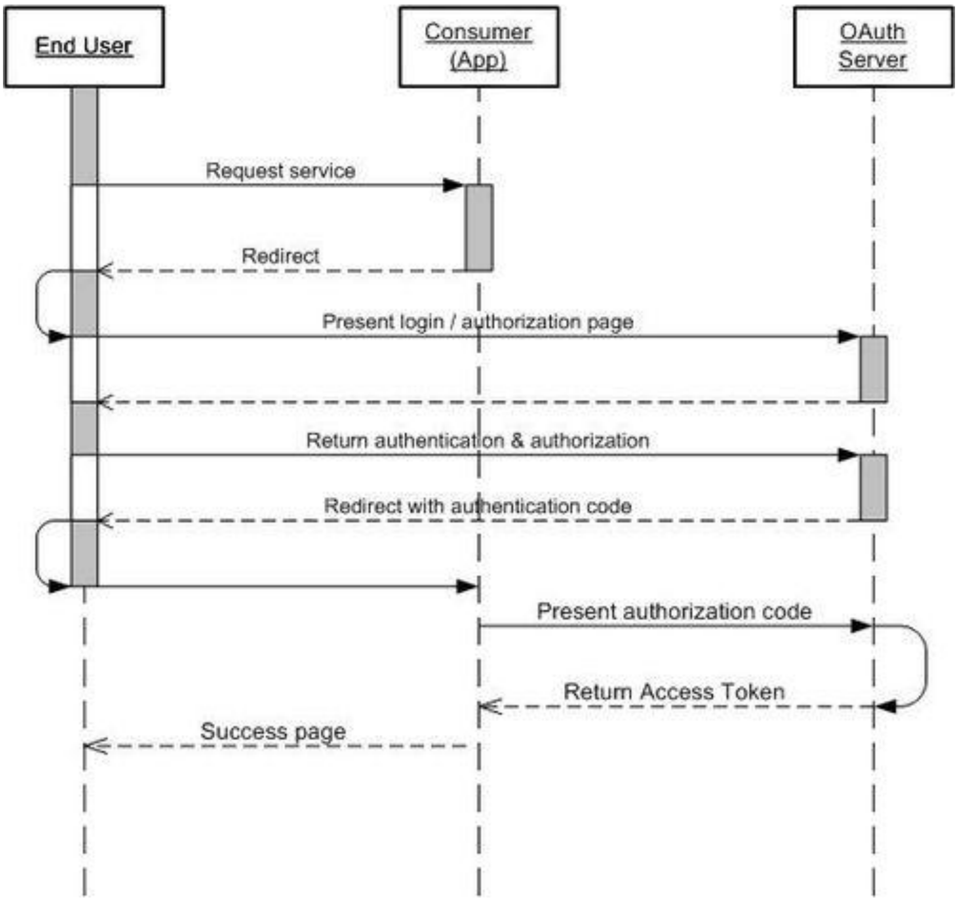
The advantages of OAuth 2.0

• A standardized protocol supported by a wider set of Service Providers (Facebook, Google,LinkedIn, …)

• The user grants access for a consumer to a specific resource by providing an access token to the consumer. The user need not to be online, when the consumer accesses the resource, i.e. actually makes use of the access token.

• The consumer always requires the consent of the user for receiving the access token.

• The user is in full control of who can access his resources.


End User
Consumer (App)
OAuth Server
Request service
Redirect
Present login / authorization page
Return authentication & authorization
Redirect with authentication code
Present authorization code
Return Access Token
Success page

Identity Generic Enabler - OAuth 2.0 Authentication Flow

Get Request Token

```
https://api.login.<xyz.com>/oauth/v2/
   get_request_token?oauth_nonce=ce2130523f788f313f76314ed3965ea6
   &oauth_timestamp=1202956957
   &oauth_consumer_key=123456891011121314151617181920
   &oauth_signature_method=plaintext
   &oauth_signature=abcdef
   &oauth_version=1.0
   &xoauth_lang_pref="en-us"
   &oauth_callback="http://yoursite.com/callback"
```

Get Request Token Response

```
oauth_token=z4ezdgj
&oauth_token_secret=47ba47e0048b7f2105db67df18ffd24bd072688a
&oauth_expires_in=3600

&xoauth_request_auth_url=https%3A%2F%2Fapi.login.<xyz.com>%2Foauth%2
Fv2%2Frequest_auth%3Foauth_token%3Dz4ezdgj
&oauth_callback_confirmed=true
```

Get Access Token

```
https://api.login.<xyz.com>/oauth/v2/request_auth?oauth_token=j5nyp6
```

Get Access Token Response

```
http://yoursite.com/callback?oauth_verifierer=svmhhd
```

Exchange Token Request

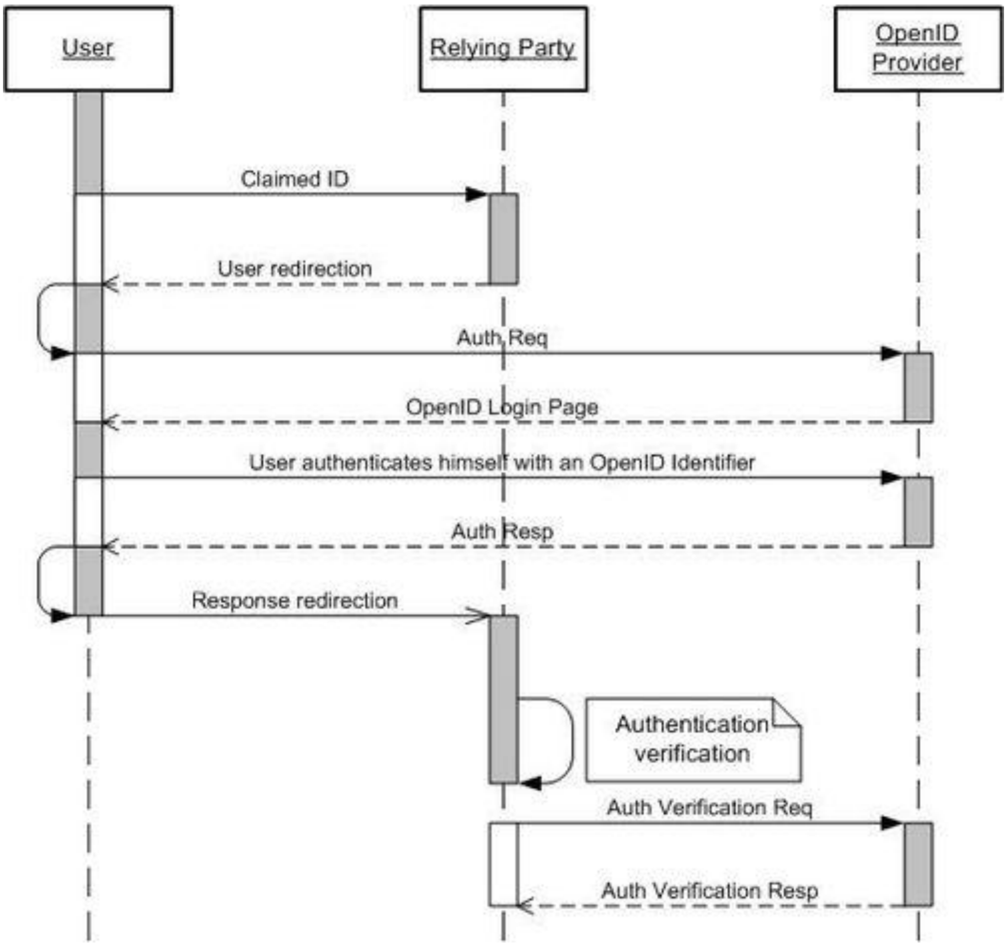
```
https://api.login.<xyz.com>/oauth/v2/get_token?oauth_consumer_key=dj
0yJmk9NG5USlVvTlZsZEpnJmQ9WVdrOVQwa


zFPRUozTkc4bWNHozlNVE13TXprM01UUTBNZy0tJnM9Y29uc3VtZXJzZWNyZXQmeD1kN
g--
&oauth_signature_method=PLAINTEXT
&oauth_version=1.0
&oauth_verifier=svmhhd
&oauth_token=gugucz&oauth_timestamp=1228169662
&oauth_nonce=8B9SpF


&oauth_signature=5f78507cf0acc38890cf5aa697210822e90c8b1c%261fa61b46
4613d0d32de80089fe099caf34c9dac5
```

Exchange Token Response

```
oauth_token=A%3DqVDHXBngo1tEtzox.JMhzd91Rk99.39Al7hos3J80mm1j_3nGP4B
iilL777vUj2rsPLj1cZw.srbisvw.cz42Lzmlxt


H0Kk9mkXilvS1ll5lNoMKXO5zy5YG4vO3fbGKewp7IESYMIdEi4Md7SroYiv6kBCEjqB
4jXr0.8XsMvOlQgZ.aKNKXwc2sv3n4BOZxs


54tzXV6rGNpEHZUaj9CovPUo44isTgs9FnLIKpXFCU4Jq1BB3_IOTFBNf1vtf5vSxaxe
_L5dUhr.i15Hx0LTZ2tlsWeDcActSGGBWVc


vytPF3cK9mDWy44baBgCVI3AEbGCqg.NGhDPqOh1ZHfKFtYlBZfG4xf2n..CdxcM5x4I
NxnVz2.biMkfhfkw8haJuR0RaUY37lBxZ9z
```

OpenID

In case if a user wants to login to the service of a Service Provider (SP) [referred to as “Relying Party” (RP) in the OpenId protocol] a “Claimed Identifier” is being presented by the user . The SP analyses the Identifier and redirects the user to his OpenId Provider (OP) with an OpenID Authentication request, in consequence of which the user authenticates himself against this OP. Nevertheless it is important to mention that the authentication process between the user and the OpenID server is not part of the OpenId protocol. After successful authentication procedure the OP redirects the user back to the RP with an assertion, indicating that the authentication was successful (Positive Assertion), else with the information that authentication failed (Negative Assertions). In the positive case the RP verifies the received information by sending a return URL, the discovered information, the nonce and the signature within a request to the OP. Alternatively the OP and the RP can establish an association after the RP received the Claimed Identifier by using the key exchange protocol standardized by the IETF.



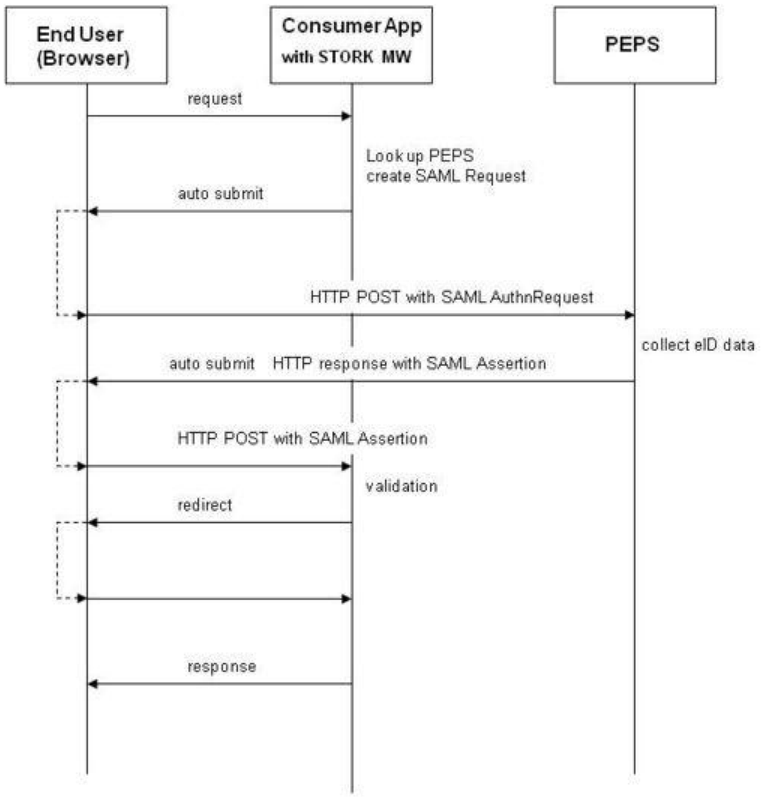
Identity Generic Enabler - OpenID Authentication Flow

Username / Password

A user can be identified by providing a <user name> and a <pass word>, which will be verified against data base.

eID – card

Alternatively to <user name> and <pass word> a user can identify himself/herself with an European eID (electronic identity). This is not possible with all eIDs of all European countries, but with those, supported by the STORK project. European countries have a variety of different and incompatible eID solutions. STORK has developed an integrative wrapper, allowing a uniform, country-independent access. In the Generic Identity Enabler Scenario the IDM will delegate the user authentication to an external eID service. The information exchange with the eID Service is SAML based.



Identity Generic Enabler - Access Portal with eID Authentication

11.1.5.4 *Detailed Specification*

- Identity Management Generic Enabler API Specification

11.1.6 References

[SAML]	http://code.google.com/intl/de-DE/googleapps/domain/sso/saml_reference_implementation.html http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security#samlv20
[OAuth]	http://oauth.net http://en.wikipedia.org/wiki/OAuth
[OpenID]	http://openid.net/specs/openid-authentication-2_0.html current version http://openid.net/connect/ next version based on oauth2
[e-ID]	http://www.eid-stork.eu

11.2 Re-utilised Technologies/Specifications

The Identity Management GE is based on RESTful Design Principles. The technologies and specifications used in this GE are:

- RESTful web services
- HTTP/1.1
- JSON and/or XML data serialization formats.
- SOAP 1.2 (NSN)
- SAML 2.0 (NSN)
- OAuth 2.0
- OpenID (DT)

11.3 Terms and definitions

This section comprises a summary of terms and definitions introduced during the previous sections. It intends to establish a vocabulary that will be help to carry out discussions internally and with third parties (e.g., Use Case projects in the EU FP7 Future Internet PPP). For a summary of terms and definitions managed at overall FI-WARE level, please refer to FIWARE Global Terms and Definitions

- **Access control**: is the prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner. (ITU-T-X-800_Link). More precisely, access control is the protection of resources against unauthorized access; a process by which use of resources is regulated according to a security policy and is permitted by only authorized system entities according to that policy. RFC 2828
- **Account**: A (user) account is “typically a formal business agreement for providing regular dealings and services between principal sand business service providers.” OASIS Security Assertion Markup Language (SAML)
- **Authentication (AuthN)**: We adopted the following definition of authentication from RFC 3588"Authentication is “the act of verifying the identity of an entity (subject)”

  TrustInCyberspace adds the term “level of confidence” to this definition:

  Authentication is the process of confirming a system entity’s asserted identity with a specified, or understood, level of confidence.” This definition holds all necessary parts to examine authentication in broad sense. First of all it does not narrow the authentication to human users, but refers to a generic “system entity”. See authentication reference architecture description for a closer look at different identities that could be authenticated.

  Secondly it introduces the often neglected concept of “level of confidence” which applies to each authentication of an identity. No computer program or computer user can definitely prove the identity of another party. There is no authentication method that can be secured against any possible identity-theft attack, be it physical or non-physical. It is only possible to apply one or more tests, which, if passed, have been previously declared to be sufficient to go on further. The problem is to determine which tests are sufficient, and many such are inadequate.

  The original Greek word originates from the word 'authentes'='author'. This leads to the general field of claims and trust management, because authentication could also mean to verify the “author” / issuer of any claim.

The confirmation or validation process of authentication is actually done by presenting some kind of proof. This proof is normally derived from some kind of secret hold by the principal. In its simplest form the participant and the authentication authority share the same secret. More advanced concepts rely on challenge/response mechanisms, preventing the secrets to be transmitted. Refer to Authentication Technologies for a detailed list of authentication methods used today.

As stated above, each authentication method assures only some level of trust in the claimed identity, but none could be definite. Therefore it makes sense to distinguish the different authentication methods by an associated assurance level, stating the level of trust in the authentication process.

As this assurance level depends not only on the technical authentication method, but also on the overall computer system and even on the business processes within the organization (provisioning of identities and credentials), there is no ranking of the authentication methods here.

- **Authentication protocol**: "Over-the-wire authentication protocols are used to exchange authentication data between the client and server application. Each authentication protocol supports one or more authentication methods. The OATH reference architecture provides for the use of existing protocols, and envisions the use of extended protocols which support new authentication methods as they are defined." (OATH)
- **Federation**: The term federation “is used in two senses - "The act of establishing a relationship between two entities. An association comprising any number of service providers and identity providers.” OASIS Security Assertion Markup Language (SAML)

  “A federation is a collection of realms that have established a producer-consumer relationship whereby one realm can provide authorized access to a resource it manages based on an identity, and possibly associated attributes, that are asserted in another realm.

  Federation requires trust such that a relying party can make a well-informed access control decision based on the credibility of identity and attribute data that is vouched for by another realm.” WS-Federation @ IBM

  Remark: Federation according to WS-Federation @ IBM is similar to the concept of a Circle of Trust.
- **Identifier**: Identifiers can be understood as a dedicated, publicly known attribute of an identity that refers to that identity only. Typically, identifiers are valid within a specific domain. Special types of identifiers are valid globally, due to the use of popular domain naming and resolution protocols such as DNS, which implies addressing capabilities to the identity. OASIS Security Assertion Markup Language (SAML) defines identifier as follows:

  An identifier is “a data object (for example, a string) mapped to a system entity that uniquely refers to the system entity. A system entity may have multiple distinct identifiers referring to it. An identifier is essentially a "distinguished attribute" of an entity.”
- **Identity** (Digital): The term identity and its meaning have been discussed

controversially in the “identity community” for many years. Until now, there is no commonly agreed definition of that notion. : : The IdM && AAA reference architecture applies the following three definitions of identity.

The Identity Gang defines the term digital identity as follows:

A digital identity is “a digital representation of a set of Claims made by one party about itself or another digital subject.”

The following comments were added:

A digital identity is just one set of claims about a digital subject. For any given digital subject there will typically be many digital identities.

A digital identity can be created on the fly when a particular identity transaction is desired or persistent in a data store to provide a representation that can be referenced.

A digital identity may contain claims made by multiple claimants.

A digital identity may be signed by a digital identity provider to provide assurance to a relying party.

This definition emphasizes two facts:

Normally, a principal (subject) has multiple digital identities or personas.

Identities are made out of attributes (claims).

Therefore, the scope of identity management in the reference architecture has two viewpoints: For once it focuses on identities and personas itself, and on the other side, it deals with the attributes of these identities and personas.

The Liberty Alliance Project (LAP) defines digital identity as follows:

Digital identity is “the essence of an entity. One’s identity is often described by one’s characteristics, among which may be any number of identifiers. A Principal may wield one or more identities.”

RSA uses the following definition of digital identity:

“Digital identity consists of an identity assertion and the characteristics, sometimes called attributes that are collected or observed through our computerized relationships. It is often as simple as a user name and password.”

The definition of RSA adds one important aspect to the identity discussion: Even the simplest user name and password combinations without any additional attributes or claims constitute an identity.

- **Identity context**: is “the surrounding environment and circumstances that determine meaning of digital identities and the policies and protocols that govern their interactions.” (Identity Gang)
- **Identity management (IdM)**: comprises “the management of identity information both internally and when it is passed from one entity to another.” Open Mobile Alliance (OMA)
- **Identity provider**: The Open Mobile Alliance (OMA) defines the term identity provider (IdP) as follows - An identity provider is “a special type of service provider […] that creates, maintains, and manages identity information for principals, and can provide […] assertions to other service providers within an authentication domain (or even a

circle of trust)."

Another notion defines identity provider as "an agent that issues a digital identity [that] is acting on behalf of an issuing Party." (Identity Gang)

The following definition of identity provider descends from WS-Federation @ IBM: "An identity provider is an entity that acts as an authentication service to end requestors and as data origin authentication service to service providers […]. Identity providers are trusted (logical) 3rd parties which need to be trusted both by the requestor […] and the service provider which may grant access to valuable resources and information based upon the integrity of the identity information provided by the identity provider."

The Identity Provider is part of the Identity Management infrastructure.

- **Single sign-on**: is "From a Principal's perspective, single sign-on encompasses the capability to authenticate with some system entity—[…] an Identity Provider - and have that authentication honored by other system entities, [termed] Service Providers […]. Note that upon authenticating with an Identity Provider, the Identity Provider typically establishes and maintains some notion of local session state between itself and the Principal's user agent. Service Providers may also maintain their own distinct local session state with a Principal's user agent." Liberty Alliance Project (LAP)

12 Identity Management Generic Enabler API Specification

You can find the content of this chapter as well in the wiki of fi-ware.

12.1 Introduction to the Identity Management GE API

Please check the FI-WARE Open Specifications Legal Notice to understand the rights to use FI-WARE Open Specifications.

12.1.1 Identity Management GE API Core

- There are two Identity Management Generic Enablers where the GE of NSN provides API's following the OAuth and SAML standard and the GE of Deutsche Telekom (DT) the OAuth and OpenID standard.
- On the GE of NSN Authenticate endpoint is a method accessed via HTTP(s) that uses XML based information to start the authentication process for a user. The end point is typically accessed via browser request containing the SAML authentication request either in body (POST) or parameters (GET).
- On the GE of Deutsche Telekom Authenticate endpoint is a method accessed via HTTPS that uses JSON based information to start the authentication process for a user. The end point is typically accessed via browser request containing the OpenID authentication request (POST).
- Attribute request endpoint is a SOAP method accessed via HTTP(s). The only parameter is the standard SAML attribute request.
- The OAuth endpoint is a RESTfull method accessed via HTTP(s) containing client authentication in the HTTP header and the standard OAuth parameters in the URL or body.

12.1.2 Intended Audience

This specification is intended for Service Consumers (with developement skills) and Cloud Providers. For the former, this document provides a full specification of how to interoperate with the Identity Management Service API. For the latter, this specification indicates the interface to be provided to the client application developers to provide the described functionalities. To use this information, the reader should first have a general understanding of the Generic Enabler service

(FIWARE.ArchitectureDescription.Identity_Management_Generic_Enabler)

The API user should be familiar with:

- RESTful web services
- HTTP/1.1
- JSON and/or XML data serialization formats.
- SOAP 1.2 (NSN)
- SAML 2.0 (NSN)

- OAuth 2.0
- OpenID (DT)

12.1.3 API Change History

Current version is: Version 1.0.0, 2012.04.30.

The most recent changes are described in the table below:

Revision Date	Changes Summary
2012.04.30	• This is the first version of the Identity Management API Specification

12.1.4 How to Read this Document

The following list summarizes special notations for the current and future versions of this document.

- A bold, mono-spaced font is used to represent code or logical entities, e.g., `HTTP method (GET, PUT, POST, DELETE).`
- An italic font is used to represent document titles or some other kind of special text, e.g. *URI*.
- The variables are represented between brackets, e.g. {*id*} and in italic font. When the reader find it, can change it by any value.

12.1.5 Additional Resources

Detailed specification and descriptions can be found here:

- SAML

  http://code.google.com/intl/de-DE/googleapps/domain/sso/saml_reference_implementation.html

  http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security#samlv20
- OAuth

  http://oauth.net

  http://en.wikipedia.org/wiki/OAuth
- OpenID

  http://openid.net/specs/openid-authentication-2_0.html current version

  http://openid.net/connect/ next version based on oauth2
- Examples

  FIWARE.ArchitectureDescription.Identity Management Generic Enabler

12.2 General *Identity Management Generic Enabler* API Information

12.2.1 Resources Summary

- The endpoint URL for SAML authentication request is:
  http://85.182.208.140:8080/IDM/SamlSsoAaServlet
- The endpoint URL for SOAP SAML attribute request is:
  http://85.182.208.140:8080/IDM/services/AttributeAuthorityService
- The OAuth authorisation endpoint URL is:
  http://85.182.208.140:9090/server/oauth2/authorize
- The OAuth token endpoint URL is:
  http://85.182.208.140:9090/server/oauth2/token
- The URLs to the Deutsche Telekom IdM will be provided when a tenat instance is booked.

12.2.2 Authentication

At service side one of the herein specified authentication protocols (SAML, OAuth, OpenID) should be implemented.

12.2.3 Representation Format

In case of SAML always XML is used, whereas for OAuth no body is defined.

In case of the OpenID Authentication the attributs are as JSON string formated.

12.2.4 Representation Transport

Resource representation is transmitted between client and server by using HTTP 1.1 protocol, as defined by IETF RFC-2616. Each time an HTTP request contains payload, a Content-Type header shall be used to specify the MIME type of wrapped representation. In addition, both client and server may use as many HTTP headers as they consider necessary.

12.2.5 Resource Identification

The resources are uniquely identified by their URI.

12.2.6 Links and References

n.a.

12.2.7 Paginated Collections (Optional)

n.a.

12.2.8 Efficient Polling with the Changes-Since Parameter (Optional)

n.a.

12.2.9 Limits

Resources are only limited by hardware resources.

12.2.9.1 *Rate Limits*

n.a.

12.2.9.2 *Absolute Limits*

tbd.

12.2.9.3 *Determining Limits Programmatically*

n.a.

12.2.10 Versions

n.a.

12.2.11 Extensions

The SAML protocol supports extensions, however this is not supported at the moment. Maybe this will be considered in the future.

Verb	**URI**	**Description**
GET	/extensions	List of all available extensions

12.2.12 Faults

12.2.12.1 *Synchronous Faults*

- GE of NSN

Synchronous faults will be sent via HTTP, usual error code is 500.

Response Code	**Description**

HTTP 200	successful
HTTP 500	fault

- GE of DT

Synchronous faults will be sent via HTTP. A detailed description of errors are provided with the API documentation.

Response Code	**Description**
HTTP 200	successful
HTTP 400	fault with requestmethod
HTTP 404	resource not found

12.2.12.2 *Asynchronous Faults*

n.a.

13 FIWARE OpenSpecification Security Optional Security Enablers SecureStorageService

You can find the content of this chapter as well in the wiki of fi-ware.

Name	FIWARE.OpenSpecification.Security.SecureStorageService
Chapter	Security,
Catalogue-Link to Implementation	Secure Storage Service
Owner	Thales, Lucie Gaspard

13.1 Preface

Within this document you find a self-contained open specification of a FI-WARE generic enabler, please consult as well the FI-WARE Product Vision, the website on http://www.fi-ware.eu and similar pages in order to understand the complete context of the FI-WARE project.

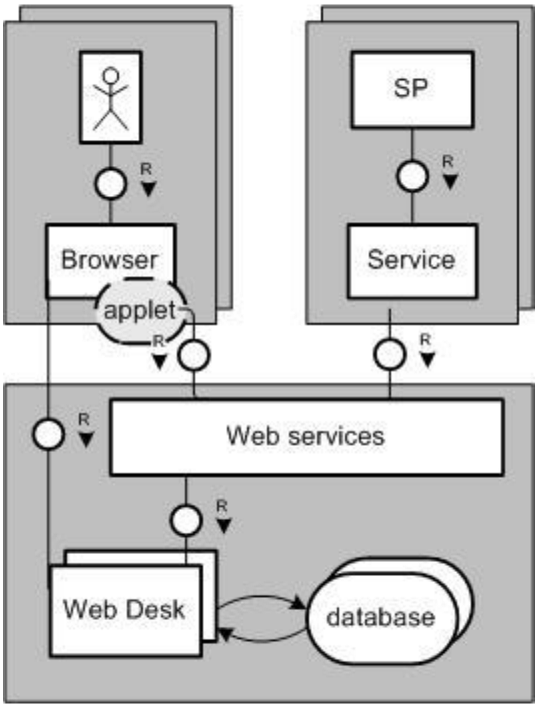
13.2 Copyright

13.3 Legal Notice

Please check the following Legal Notice to understand the rights to use these specifications.

13.4 Overview

The Secure Storage Service provides a storage for labelled (i.e. XML DSIG protected) data. It comes with an application-level filter which authorizes read access in function of the identity of the authenticated requester (for example, a service provider) and in function of the sensitivity of the data.



Glossary

SSS Secure Storage Service

SP Service Provider

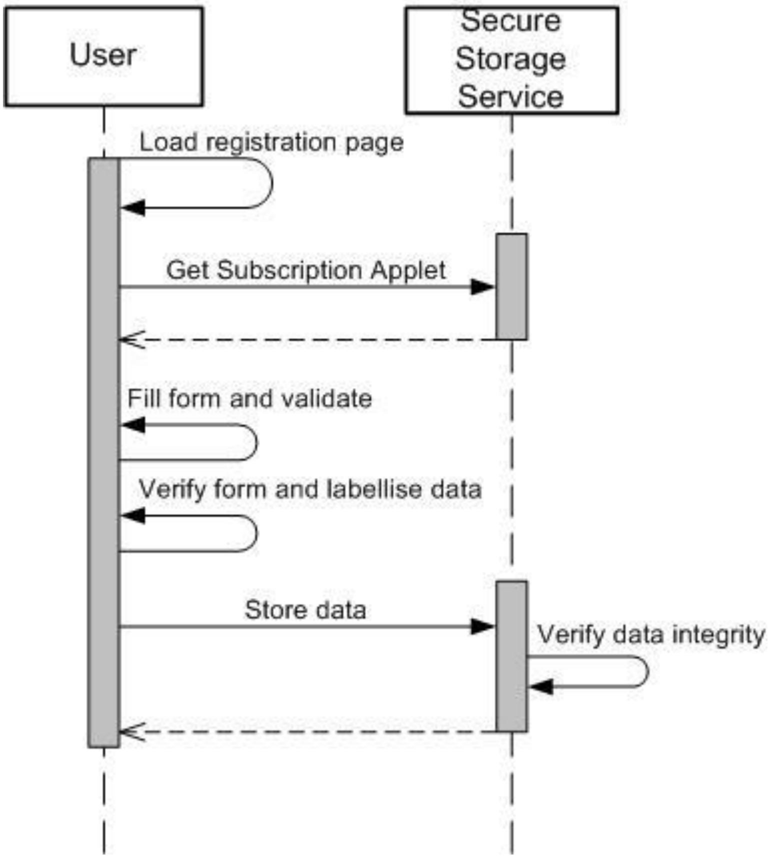
13.5 Basic Concepts

The data is labelled before being stored, i.e. it is previously protected by its owner. Moreover, the owner himself has initialised the sensitivity level of the different fields of his data (for example : mail address > private, main interest > public, job > public, etc...). Once the data are stored by SSS, the public fields (i.e. the fields that have been tagged 'public') can be read by anyone. The private one can be read by trusted service providers (SP) only. A trusted service is a service which is authenticated by a certificate which has been delivered by a dedicated Certification Authority.

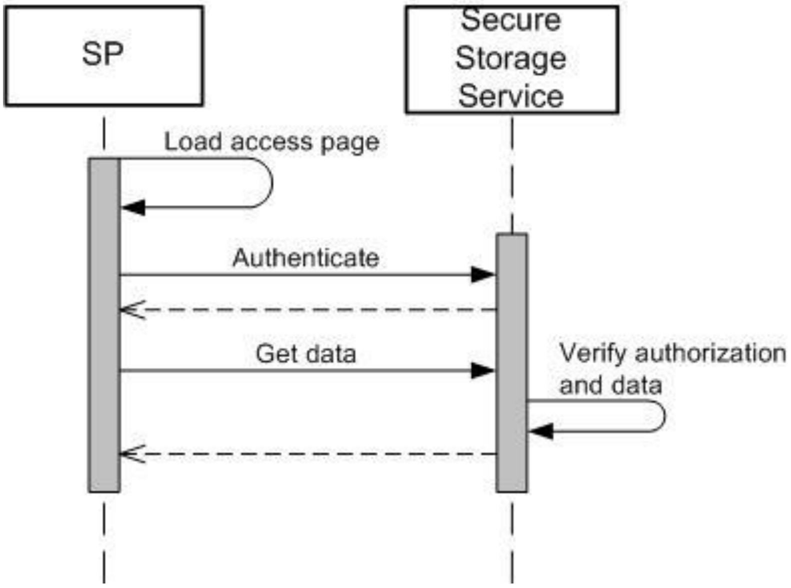
13.6 Main Interactions

The interactions between the SSS and a user or a SP are made through Web Services developed with the SOAP/WSDL technology. The user needs to connect through a web page which downloads an signed applet on the secured Web Desk. This applet labellizes the data according to the level of sensitivity and sends the data to the SSS via the WS available. The SP, authenticated and authorized, accesses the data in the SSS via the WS available.

Sequence diagram : Saving data



Sequence diagram : Retreiving data



13.7 Basic Design Principles

The functions (WS) available for the data owners are:

- Data injection
- Data update
- Data deactivation
- Get the list of SPs who had access to the profile and when, and what usage they did of the data

The functions (WS) available for the service providers:

- Data request
- Security Attribute request

NB :

- These functions are defined according to European rules : 95/46/CE, 2002/58/CE, COM(2010) 609
- WSDL available

13.8 Re-utilised Technologies/Specifications

The Repository GE is based on SOAP/WSDL Design Principles. The technologies and specifications used in this GE are:

- SOAP/WSDL web services
- HTTPs/1.1
- XML and XML DSig data serialization formats

13.9 Terms and definitions

This section comprises a summary of terms and definitions introduced during the previous sections. It intends to establish a vocabulary that will be help to carry out discussions internally and with third parties (e.g., Use Case projects in the EU FP7 Future Internet PPP). For a summary of terms and definitions managed at overall FI-WARE level, please refer to FIWARE Global Terms and Definitions

- **SSS:** Secure Storage Service, the service provided to store securely data.
- **Credentials:** A credential is an attestation of qualification, competence, or authority issued to an individual by a third party with a relevant de jure or de facto authority or assumed competence to do so. In this document, we define digital credentials to be lists of attribute-value statements certified by an Issuer. Here we abstract from the concrete mechanism (cryptographic or other) by which the authenticity of the attribute values can be verified. We do not impose any restrictions on which attributes can be contained in a credential, but typically these either describe the identity of the credential's owner or the authority assigned to her.

- **Data:** Data means any information stored by a user.
- **SP:** Service Provider, any authorized ans authentified service needing some of the data stored.
- **WSDL:** Web Services Description Language, an XML-based language that is used for describing the functionality offered by a Web service. A WSDL description of a web service (also referred to as a WSDL file) provides a machine-readable description of how the service can be called, what parameters it expects, and what data structures it returns..
- **Applet:** any small application that performs one specific task that runs within the scope of a larger program, often as a plug-in. An applet typically also refers to Java applets, i.e., programs written in the Java programming language that are included in a web page.

14 FIWARE OpenSpecification Security Privacy Generic Enabler

You can find the content of this chapter as well in the wiki of fi-ware.

***[DISCLAIMER]:** The first version of the generic enabler described below is planned to be delivered on the second FI-WARE release. Since the GE is still under the development, we only briefly sketch the main functionality provided by the Privacy GE and refer for the full specification to the second version of this deliverable. .*

Name	FIWARE.OpenSpecification.Security.Privacy Generic Enabler
Chapter	Security,
Catalogue-Link to Implementation	[- scheduled for 2nd release only]
Owner	IBM, Anja Lehmann

14.1 Preface

Within this document you find a self-contained open specification of a FI-WARE generic enabler, please consult as well the FI-WARE_Product_Vision, the website on http://www.fi-ware.eu and similar pages in order to understand the complete context of the FI-WARE project.

14.2 Copyright

14.3 Legal Notice

Please check the following Legal Notice to understand the rights to use these specifications.

14.4 Overview

The Privacy enabler provides trustworthy, yet privacy-friendly authentication, using privacy-enhanced attribute-based credentials (Privacy-ABCs). In a nutshell, the User first obtains credentials, which are certified attribute-value pairs, from an Issuer who vouches for the correctness of the certified attributes. The User can subsequently authenticate towards a Verifier by sending a presentation token which is derived from her credentials. A single presentation token can selectively reveal attribute values from one or more credentials. It can also prove that a given predicate over one or more attributes holds without revealing the full attribute values, e.g., that the birthdate is before January 1st, 1994, or that the name on the User’s credit card matches that on her driver’s license.

For an easy integration of Privacy-ABCs in various applications and systems, we consider a mechanism-independent ABC Engine layer on top of the core Cryptographic Engines, which will provide simple and well-defined APIs.

14.5 Detailed Specifications

14.5.1 Open API Specifications

- FIWARE.OpenSpecification.Details.Security.Privacy_Generic_Enabler_API

14.5.2 Other Relevant Specifications

Available in Release 2:

14.6 Terms and definitions

This section comprises a summary of terms and definitions introduced during the previous sections. It intends to establish a vocabulary that will be help to carry out discussions internally and with third parties (e.g., Use Case projects in the EU FP7 Future Internet PPP). For a summary of terms and definitions managed at overall FI-WARE level, please refer to FIWARE Global Terms and Definitions

14.7 Privacy Glossary

- **Attribute** A piece of information, possibly certified by a credential, describing a characteristic of a natural person or entity, or of the credential itself. An attribute consists of an attribute type determining the semantics of the attribute (e.g., first name) and an attribute value determining its contents (e.g., John).
- **Certified pseudonym** verifiable pseudonym based on a user secret that also underlies an issued credential. A certified pseudonym is established in a presentation token that also demonstrates possession of a credential bound to the same User (i.e., to the same user secret) as the pseudonym.
- **Credential** A list of certified attributes issued by an Issuer to a User. By issuing a credential, the Issuer vouches for the correctness of the contained attributes with respect to the User.
- **Credential specification** A data artifact specifying the list of attribute types that are encoded in a credential.
- **Data Controller** “'Controller' shall mean the natural or legal person, public authority, agency or any other body which alone or jointly with others determines the purposes and means of the processing of personal data...”, Art. 2 (d) of Directive 95/46/EC. In the area of Privacy-ABCs the Issuer, Verifier, the Revocation Authority and the Inspector are Data Controllers with the respective duties arising from the law.
- **Data Processor** “'Processor' shall mean a natural or legal person, public authority, agency or any other body which processes personal data on behalf of the controller“, Art. 2 (e) of Directive 95/46/EC. Data Controllers processes personal data on behalf of the data Controller.
- **Data Subject** A data subject is an identified or identifiable natural person, Art. 2 (a) of Directive 95/46/EC. In the area of Privacy-ABCs the User and any other national person of which personal data is processes is a data subject. Data subjects have data subjects‟ rights assigned such as the right of access, rectification, erasure and blocking, Art. 12 of Directive 95/46/EC.
- **Device binding** An optional credential feature whereby the credential is bound to a strong secret embedded in a dedicated hardware device so that any presentation token involving the credential requires the presence of the device.
- **Inspection** An optional feature allowing a presentation token to be de-anonymized by a dedicated Inspector. At the time of creating the presentation token, the User is aware (through the presentation policy) of the identity of the Inspector and the valid grounds for inspection.
- **Inspection grounds** The circumstances under which a Verifier may ask an Inspector to trace the User who created a given presentation token.
- **Inspection Requester** Entity requesting an inspection from the Inspector, asserting that inspection is compliant with the inspection grounds specified or is legally required. In most cases this will be the Verifier, but also may be the police, or other legally authorised entity.
- **Inspector** A trusted entity that can trace the User who created a presentation token by revealing attributes from the presentation token that were originally hidden from the Verifier.

- **Issuance key** The Issuer‟s secret cryptographic key used to issue credentials.
- **Issuer** The party who vouches for the validity of one or more attributes of a User, by issuing a credential to the User.
- **Issuer parameters** A public data artifact containing cryptographic and other information by means of which presentation tokens derived from credentials issued by the Issuer can be verified.
- **Linkability** See unlinkability.
- **Personal data** "„Personal data' shall mean any information relating to an identified or identifiable natural person ('data subject'); an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity", Art. 2 (a) of Directive 95/46/EC. Within this deliverable personal data is the terminology used for legal considerations. See also Personally * **Identifiable Information.**
- **Personally Identifiable Information (PII)** Personally Identifiable Information is defined as any information about an individual maintained by an [entity], including any information that can be used to distinguish or trace an individual„s identity, such as name, social security number, date and place of birth, and any other information that is linked or linkable to an individual ([NIST10] p. 2-1). PII is a widely used terminology for personal data in the domain of information security. Within this document PII is used in relation to information security.
- **Presentation policy** A policy created and published by a Verifier specifying the class of presentation tokens that the Verifier will accept. The presentation policy contains, among other things, which credentials from which Issuers it accepts and which information a presentation token must reveal from these credentials.
- **Presentation token** A collection of information derived from a set of credentials, usually created and sent by a User to authenticate to a Verifier. A presentation token can contain information from several credentials, reveal attribute values, prove that attribute values satisfy predicates, sign an application-specific message or nonce or support advanced features such as pseudonyms, device binding, inspection, and revocation. The presentation token consists of the presentation token description, containing a technology-agnostic description of the revealed information, and the presentation token evidence, containing opaque technology-specific cryptographic parameters in support of the token.
- **Pseudonym** See verifiable pseudonym.
- **Pseudonym scope** A string provided in the Verifier‟s presentation policy as a hint to the User which previously established pseudonym she can use, or to which a new pseudonym should be associated. A single User (with a single user secret) can generate multiple verifiable or certified pseudonyms for the same scope string, but can only generate a single scope-exclusive pseudonym.
- **Revocation** The act of withdrawing the validity of a previously issued credential. Revocation is performed by a dedicated Revocation Authority, which could be the Issuer, the Verifier, or an independent third party. Which Revocation Authorities must be taken into account can be specified by the Issuer in the issuer parameters (Issuer-driven revocation) or by the Verifier in the presentation policy (Verifier-driven revocation).

- **Revocation Authority** The entity in charge of revoking credentials. The Revocation Authority can be an Issuer, a Relying Party, or an independent entity. Multiple Issuers or Verifiers may rely on the same Revocation Authority.
- **Revocation information** The public information that a Revocation Authority publishes every time a new credential is revoked or at regular time intervals to allow Verifiers to check that a presentation token was not derived from revoked credentials.
- **Revocation parameters** The public information related to a Revocation Authority, containing cryptographic information as well as instructions where and how the most recent revocation information and non-revocation evidence can be obtained. The revocation parameters are static, i.e., they do not change every time a new credential is revoked or at regular time intervals like the revocation information and non-revocation evidence (may) do.

- **Revocation Authority** The entity in charge of revoking credentials. The Revocation Authority can be an Issuer, a Relying Party, or an independent entity. Multiple Issuers or Verifiers may rely on the same Revocation Authority.
- **Revocation information** The public information that a Revocation Authority publishes every time a new credential is revoked or at regular time intervals to allow Verifiers to check that a presentation token was not derived from revoked credentials.
- **Revocation parameters** The public information related to a Revocation Authority, containing cryptographic information as well as instructions where and how the most recent revocation information and non-revocation evidence can be obtained. The revocation parameters are static, i.e., they do not change every time a new credential is revoked or at regular time intervals like the revocation information and non-revocation evidence (may) do.

- **Non-revocation evidence** The User-specific or credential-specific information that the user agent maintains, allowing it to prove in presentation tokens that the credential was not revoked. The non-revocation evidence may need to be updated either at regular time intervals or when new credentials are revoked.
- **Scope** See pseudonym scope.
- **Scope-exclusive pseudonym** A certified pseudonym that is guaranteed to be cryptographically unique per scope string and per user secret. Meaning, from a single user-bound credential, only a single scope-exclusive pseudonym can be generated for the same scope string.
- **Traceability** See untraceability.
- **Unlinkability** The property that different actions performed by the same User, in particular different presentation tokens generated by the same User, cannot be linked to each other as having originated from the same User.
- **Untraceability** The property that an action performed by a User cannot be traced back to her identity. In particular, the property that a presentation token generated by a User cannot be traced back to the issuance of the credential from which the token was derived.
- **User** The human entity who wants to access a resource controlled by a verifier and obtains credentials from Issuers to this end.
- **User agent** The software entity that represents the human User and manages her credentials.

- **User binding** An optional credential feature whereby the credential is bound to an underlying user secret. By requiring multiple credentials to be bound to the same user secret, one can prevent Users from “pooling” their credentials.
- **User secret** A piece of secret information known to a User (either a strong random secret or a human-memorizable password or PIN code) underlying one or more issued credentials or pseudonyms. A presentation token involving a pseudonym or a user-bound credential implicitly proves knowledge of the underlying user secret.
- **Verifiable pseudonym** A public identifier derived from a user secret allowing a User to voluntarily link different presentation tokens created by her or to re-authenticate under a previously established pseudonym by proving knowledge of the user secret. Multiple unlinkable pseudonyms can be derived from the same user secret.
- **Verifier** The party that protects access to a resource by verifying presentation tokens to check whether a User has the requested attributes. The Verifier only accepts credentials from Issuers that it trusts.

14.8 Privacy Abbreviations

- **ABCs** Attribute Based Credentials
- **CA.** Certificate Authority.
- **CE** Crypto Engine.
- **HTTP** Hypertext Transfer Protocol.
- **HTTPS** HyperText Transfer Protocol Secure (HTTP secured by TLS or SSL).
- **Idemix** IBM Identity Mixer.
- **IdM** Identity Manager.
- **IdSP** Identity Service Provider.
- **PET** Privacy Enhancing Technology.
- **PRIME** Privacy and Identity Management for Europe.
- **PIN** Personal Identification Number
- **RP** Relying Party
- **SCI** Smartcard Interface
- **SSL** Secure Sockets Layer
- **STS** Secure Token Service
- **TLS** Transport Layer Security
- **URI** Uniform Resource Identifier
- **XML** eXtensible Markup Language

14.9 Security Terms

- **Attack.** Any kind of malicious activity that attempts to collect, disrupt, deny, degrade, or destroy information system resources or the information itself
- **Authentication protocol**: "Over-the-wire authentication protocols are used to exchange authentication data between the client and server application. Each authentication protocol supports one or more authentication methods. The OATH reference architecture provides for the use of existing protocols, and envisions the use of extended protocols which support new authentication methods as they are defined." (OATH)
- **Access control**: is the prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner. (ITU-T-X-800_Link). More precisely, access control is the protection of resources against unauthorized access; a process by which use of resources is regulated according to a security policy and is permitted by only authorized system entities according to that policy. RFC 2828
- **Account**: A (user) account is “typically a formal business agreement for providing regular dealings and services between principal sand business service providers.” OASIS Security Assertion Markup Language (SAML)
- **Authentication (AuthN)**: We adopted the following definition of authentication from RFC 3588"Authentication is “the act of verifying the identity of an entity (subject)”

  TrustInCyberspace adds the term “level of confidence” to this definition:

  Authentication is the process of confirming a system entity’s asserted identity with a specified, or understood, level of confidence.” This definition holds all necessary parts to examine authentication in broad sense. First of all it does not narrow the authentication to human users, but refers to a generic “system entity”. See authentication reference architecture description for a closer look at different identities that could be authenticated.

  Secondly it introduces the often neglected concept of “level of confidence” which applies to each authentication of an identity. No computer program or computer user can definitely prove the identity of another party. There is no authentication method that can be secured against any possible identity-theft attack, be it physical or non-physical. It is only possible to apply one or more tests, which, if passed, have been previously declared to be sufficient to go on further. The problem is to determine which tests are sufficient, and many such are inadequate.

  The original Greek word originates from the word 'authentes'='author'. This leads to the general field of claims and trust management, because authentication could also mean to verify the “author” / issuer of any claim.

  The confirmation or validation process of authentication is actually done by presenting some kind of proof. This proof is normally derived from some kind of secret hold by the principal. In its simplest form the participant and the authentication authority share the same secret. More advanced concepts rely on challenge/response mechanisms, preventing the secrets to be transmitted. Refer to Authentication Technologies for a detailed list of authentication methods used today.

  As stated above, each authentication method assures only some level of trust in the claimed identity, but none could be definite. Therefore it makes sense to distinguish the different authentication methods by an associated assurance level, stating the level of trust in the authentication process.

  As this assurance level depends not only on the technical authentication method, but also on the overall computer system and even on the business processes within the organization (provisioning of identities and credentials), there is no ranking of the authentication methods here.

- **Countermeasures**. Action, device, procedure, technique or other measure that reduces the vulnerability of an information system.
- **Cyber attack**. An attack, via cyberspace, targeting an entity (industrial, financial, public...) and using cyberspace for the purpose of disrupting, disabling, destroying, or maliciously controlling a computing environment/infrastructure; or destroying the integrity of the data or stealing controlled information
- **Exploit.** A program or technique that takes advantage of vulnerability in software and that can be used for breaking security, or otherwise attacking a host over the network
- **Federation**: The term federation "is used in two senses - "The act of establishing a relationship between two entities. An association comprising any number of service providers and identity providers." OASIS Security Assertion Markup Language (SAML)

  "A federation is a collection of realms that have established a producer-consumer relationship whereby one realm can provide authorized access to a resource it manages based on an identity, and possibly associated attributes, that are asserted in another realm.

  Federation requires trust such that a relying party can make a well-informed access control decision based on the credibility of identity and attribute data that is vouched for by another realm." WS-Federation @ IBM

  Remark: Federation according to WS-Federation @ IBM is similar to the concept of a Circle of Trust.
- **Forensics for evidence.** The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations.
- **Identity**. In the narrow sense, identity is the persistent identifier of users (user name), things or services by which other parties "remember" them and, hence, are able to store or retrieve specific information about them and are able to control their access to different resources. In the wider sense, identity also covers further attributes of users, things and services; e.g. for users, such information may include personal information such as context, group membership and profile.
- **Identity** (Digital): The term identity and its meaning have been discussed controversially in the "identity community" for many years. Until now, there is no commonly agreed definition of that notion. : : The IdM && AAA reference architecture applies the following three definitions of identity.

  The Identity Gang defines the term digital identity as follows:

  A digital identity is "a digital representation of a set of Claims made by one party about itself or another digital subject."

  The following comments were added:

  A digital identity is just one set of claims about a digital subject. For any given digital subject there will typically be many digital identities.

A digital identity can be created on the fly when a particular identity transaction is desired or persistent in a data store to provide a representation that can be referenced.

A digital identity may contain claims made by multiple claimants.

A digital identity may be signed by a digital identity provider to provide assurance to a relying party.

This definition emphasizes two facts:

Normally, a principal (subject) has multiple digital identities or personas.

Identities are made out of attributes (claims).

Therefore, the scope of identity management in the reference architecture has two viewpoints: For once it focuses on identities and personas itself, and on the other side, it deals with the attributes of these identities and personas.

The Liberty Alliance Project (LAP) defines digital identity as follows:

Digital identity is “the essence of an entity. One’s identity is often described by one’s characteristics, among which may be any number of identifiers. A Principal may wield one or more identities.”

RSA uses the following definition of digital identity:

“Digital identity consists of an identity assertion and the characteristics, sometimes called attributes that are collected or observed through our computerized relationships. It is often as simple as a user name and password.”

The definition of RSA adds one important aspect to the identity discussion: Even the simplest user name and password combinations without any additional attributes or claims constitute an identity.

- **Identifier**: Identifiers can be understood as a dedicated, publicly known attribute of an identity that refers to that identity only. Typically, identifiers are valid within a specific domain. Special types of identifiers are valid globally, due to the use of popular domain naming and resolution protocols such as DNS, which implies addressing capabilities to the identity. OASIS Security Assertion Markup Language (SAML) defines identifier as follows:

  An identifier is “a data object (for example, a string) mapped to a system entity that uniquely refers to the system entity. A system entity may have multiple distinct identifiers referring to it. An identifier is essentially a "distinguished attribute" of an entity.”

- **Identity context**: is “the surrounding environment and circumstances that determine meaning of digital identities and the policies and protocols that govern their interactions.” (Identity Gang)

- **Identity management (IdM)**: comprises “the management of identity information both internally and when it is passed from one entity to another.” Open Mobile Alliance (OMA)

- **Identity provider**: The Open Mobile Alliance (OMA) defines the term identity provider (IdP) as follows - An identity provider is “a special type of service provider […] that creates, maintains, and manages identity information for principals, and can provide […] assertions to other service providers within an authentication domain (or even a circle of trust)."

Another notion defines identity provider as "an agent that issues a digital identity [that] is acting on behalf of an issuing Party." (Identity Gang)

The following definition of identity provider descends from WS-Federation @ IBM: "An identity provider is an entity that acts as an authentication service to end requestors and as data origin authentication service to service providers […]. Identity providers are trusted (logical) 3rd parties which need to be trusted both by the requestor […] and the service provider which may grant access to valuable resources and information based upon the integrity of the identity information provided by the identity provider."

The Identity Provider is part of the Identity Management infrastructure.

- **Impact**. The adverse effect resulting from a successful threat exercise of vulnerability. Can be described in terms of loss or degradation of any, or a combination of any, of the following three security goals: integrity, availability, and confidentiality.
- **Partial identity:** a partial identity is a set of attributes of a user. Thus, an identity is composed of all attributes of a user, a partial identity is a subset of a user's identity. Typically, a user is known to another party only as a partial identity. A partial identity can have a unique identifier. The latter is a strong identifier if it is allows for a strong authentication of the user (holder) of the partial identity, such a cryptographic "identification" protocol
- **Privacy**. Dictionary definitions of privacy refer to "the quality or state of being apart from company or observation, seclusion [...] freedom from unauthorized intrusion" (Merriam-Webster online [MerrWebPriv]). In the online world, we rely on a pragmatic definition of privacy, saying that privacy is the state of being free from certain privacy threats.
- **Privacy threats**. The fundamental privacy threats are: traceability (the digital traces left during transactions), linkability (profile accumulation based on the digital traces), loss of control (over personal data) and identity theft (impersonation).
- **Risk analysis**. The process of identifying security risks, determining their magnitude, and identifying areas needing safeguards. An analysis of an organization's information resources, its existing controls, and its remaining organizational and MIS vulnerabilities. It combines the loss potential for each resource or combination of resources with an estimated rate of occurrence to establish a potential level of damage
- **Security monitoring**. Usage of tools to prevent and detect compliance defaults, security events and malicious actions taken by subjects suspected of misusing the information system.
- **Service impact analysis.** An analysis of a service's requirements, processes, and interdependencies used to characterize information system contingency requirements and priorities in the event of a significant disruption.
- **Single sign-on**: is "From a Principal's perspective, single sign-on encompasses the capability to authenticate with some system entity—[…] an Identity Provider - and have that authentication honored by other system entities, [termed] Service Providers […]. Note that upon authenticating with an Identity Provider, the Identity Provider typically establishes and maintains some notion of local session state between itself and the Principal's user agent. Service Providers may also maintain their own distinct

local session state with a Principal’s user agent.” Liberty Alliance Project (LAP)

- **S&D:** Security and Dependability
- **Threat.** An event, process, activity being perpetuated by one or more threat agents, which, when realized, has an adverse effect on organization assets, resulting in losses (service delays or denials, disclosure of sensitive information, undesired patch of programs or data, reputation...)
- **USDL and USDL-Sec:** The Unified Service Description Language (USDL) is a platform-neutral language for describing services. The security extension of this language is going to be developed FI-WARE project.
- **Vulnerability.** A weakness or finding that is non-compliant, non-adherence to a requirement, a specification or a standard, or unprotected area of an otherwise secure system, which leaves the system open to potential attack or other problem.
- **WS-SecurityPolicy:** It is an extension to SOAP to apply security to web services. It is a member of the WS-* family of web service specifications and was published by OASIS.
- **The protocol** specifies how integrity and confidentiality can be enforced on messages and allows the communication of various security token formats, such as SAML, Kerberos, and X.509. Its main focus is the use of XML Signature and XML Encryption to provide end-to-end security.

15 FI-WARE Open Specifications Legal Notice

You can find the content of this chapter as well in the wiki of fi-ware.

15.1.1.1 *General Information*

"FI-WARE Partners" refer to Parties of the FI-WARE Project in accordance with the terms of the FI-WARE Consortium Agreement"

15.1.1.2 *Use Of Specification - Terms, Conditions & Notices*

The material in this specification details a FI-WARE Generic Enabler Specification (hereinafter "Specification") in accordance with the terms, conditions and notices set forth below. This Specification does not represent a commitment to implement any portion of this Specification in any company's products. The information contained in this Specification is subject to change without notice.

15.1.1.3 *Copyright License*

Subject to all of the terms and conditions below, the copyright holders in this Specification hereby grant you, the individual or legal entity exercising permissions granted by this License, a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide license, royalty free (without the right to sublicense) under its respective copyrights incorporated in the Specification, to copy and modify this Specification and to distribute copies of the modified version, and to use this Specification, to create and distribute special purpose specifications and software that are an implementation of this Specification.

15.1.1.4 *Patent Information*

The FI-WARE Project Partners shall not be responsible for identifying patents for which a license may be required by any FI-WARE Specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. FI-WARE specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

15.1.1.5 *General Use Restrictions*

Any unauthorized use of this Specification may violate copyright laws, trademark laws, and communications regulations and statutes. This Specification contains information which is protected by copyright. All Rights Reserved. This Specification shall not be used in any form or for any other purpose different from those herein authorized, without the permission of the respective copyright owners.

This Specification shall not be used in any form or for any other purpose different from those herein authorized, without the permission of the respective copyright owners.

For avoidance of doubt, the rights granted are only those expressly stated in this Section herein. No other rights of any kind are granted by implication, estoppel, waiver or otherwise

15.1.1.6 *Disclaimer Of Warranty*

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE FI-WARE PARTNERS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, WARRANTY OF NON INFRINGEMENT OF THIRD PARTY RIGHTS, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE.

IN NO EVENT SHALL THE FI-WARE PARTNERS BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this Specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this Specification.

15.1.1.7 *Trademarks*

You shall not use any trademark, marks or trade names (collectively, "Marks") of the FI-WARE Partners or the FI-WARE project without prior written consent.

15.1.1.8 *Issue Reporting*

This Specification is subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Procedure described on the web page http://www.fi-ware.eu.

16 Open Specifications Interim Legal Notice

You can find the content of this chapter as well in the wiki of fi-ware.

16.1.1.1 *General Information*

FI-WARE Project Partners refers to Parties of the FI-WARE Project in accordance with the terms of the FI-WARE Consortium Agreement.

16.1.1.2 *Use Of Specification - Terms, Conditions & Notices*

The material in this specification details a FI-WARE Generic Enabler Specification (hereinafter “Specification”) in accordance with the terms, conditions and notices set forth below. This Specification does not represent a commitment to implement any portion of this Specification in any company's products. The information contained in this Specification is subject to change without notice.

16.1.1.3 *Copyright License*

Subject to all of the terms and conditions below, the copyright holders in this Specification hereby grant you, the individual or legal entity exercising permissions granted by this License, a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide license (without the right to sublicense) under its respective copyrights incorporated in the Specification, to copy and modify this Specification and to distribute copies of the modified version, and to use this Specification, to create and distribute special purpose specifications and software that are an implementation of this Specification, and to use, copy, and distribute this Specification as provided under applicable law.

16.1.1.4 *Patent License*

“Specification Essential Patents” shall mean patents and patent applications, which are necessarily infringed by an implementation of the Specification and which are owned by any of the FI-WARE Project Partners. “Necessarily infringed” shall mean that no commercially reasonable alternative exists to avoid infringement.

Each of the FI-WARE Project Partners, jointly or solely, hereby agrees to grant you, on royalty-free and otherwise fair, reasonable and non-discriminatory terms, a personal, nonexclusive, non-transferable, non-sub-licensable, royalty-free, paid up, worldwide license, under their respective Specification Essential Patents, to make, use sell, offer to sell, and import software implementations utilizing the Specification.

The FI-WARE Project Partners shall not be responsible for identifying patents for which a license may be required by any FI-WARE Specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. FI-WARE specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

16.1.1.5 *General Use Restrictions*

Any unauthorized use of this Specification may violate copyright laws, trademark laws, and communications regulations and statutes. This Specification contains information which is protected by copyright. All Rights Reserved. This Specification shall not be used in any form or for any other purpose different from those herein authorized, without the permission of the respective copyright owners.

For avoidance of doubt, the rights granted are only those expressly stated in this Section herein. No other rights of any kind are granted by implication, estoppel, waiver or otherwise

16.1.1.6 ***Disclaimer Of Warranty***

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE FI-WARE PARTNERS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, WARRANTY OF NON INFRINGEMENT OF THIRD PARTY RIGHTS, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE.

IN NO EVENT SHALL THE FI-WARE PARTNERS BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. The entire risk as to the quality and performance of software developed using this Specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this Specification.

16.1.1.7 ***Trademarks***

You shall not use any trademark, marks or trade names (collectively, "Marks") of the FI-WARE Project Partners or the FI-WARE project without prior written consent.

16.1.1.8 ***Issue Reporting***

This Specification is subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Procedure described on the web page http://www.fi-ware.eu.