Ecosystem for COllaborative Manufacturing PrOceSses – Intra- and Interfactory Integration and AutomaTION
(Grant Agreement No 723145)

D4.1 Design of Security Framework I

Date: 2017-08-31

Version 1.0

Published by the COMPOSITION Consortium

Dissemination Level: Public

Co-funded by the European Union's Horizon 2020 Framework Programme for Research and Innovation under Grant Agreement No 723145

Document control page

Document file: D4.1 Security Framework I.docx
Document version: 1.0
Document owner: Atos

Work package: WP4 - Secure Data Management and Exchange in Manufacturing
Task: T4.1 – Security by design for cloud-based data exchange
T4.4 – Cyber Security for Factories
Deliverable type: R

Document status: ☒ Approved by the document owner for internal review
☒ Approved for submission to the EC

Document history:

Version	Author(s)	Date	Summary of changes made
0.1	Javier Romero (Atos)	2017-08-10	Deliverable structure / architecture definition
0.2	Javier Garcia Robles (Atos)	2017-08-14	Authorisation service
0.3	Javier Romero (Atos)	2017-08-21	Authentication service
0.4	Dawid Machnicki (Atos)	2017-08-28	Executive Summary, Introduction, XL-SIEM, Summary
0.5	Javier Romero (Atos)	2017-08-29	Changes architecture section, minor changes and improvements
0.6	Dawid Machnicki (Atos)	2017-08-29	Editorial work
0.7	Javier Romero (Atos)	2017-08-29	Complete abbreviations table. Authentication process diagrams to Authentication section. Final version for internal review.
0.8	Javier Romero (Atos)	2017-09-05	Reviewer's comments addressed
1.0	Javier Romero (Atos)	2017-09-05	Final version submitted to the European Commission

Internal review history:

Reviewed by	Date	Summary of comments
Jovana Milenkovic (ATL)	2017-08-31	Minor comments on formatting. Extend some sections in the sense of presentation, adding minor detailed explanation for better understanding.
Alexandros Nizamis, Vagia Rousopoulou (CERTH)	2017-09-01	Comments on format and wording. Missing references table.

Index:

Abbreviations

Acronym	Meaning
AMQP	Advanced Message Queuing Protocol
DRPC	Distributed Remote Procedure Call
DSS	Decision Support System
EPL	Event Processing Language
HTTP	Hypertext Transfer Protocol
IPR	Intellectual Property Rights
JSON	JavaScript Object Notation
JWT	JSON Web Token
MB	Message Broker
OAuth	Open Authorization
OIDC	Open ID Connect
OSSIM	Open Source Security Information Management
PAP	Policy Administration Point
PEP	Policy Enforcement Point
PDP	Policy Decision Point
PIP	Policy Information Point
PRP	Policy Retrieval Point
REST	Representational State Transfer
SAML	Security Assertion Mark-up Language
SIEM	Security Information and Event Management
SPI	Service Provider Interface
SQL	Structured Query Language
SSL	Secure Sockets Layer
STIX	Structured Threat Information eXpression
TLS	Transport Layer Security
XACML	eXtensible Access Control Mark-up Language
XML	eXtensible Markup Language

1 Executive Summary

The aim of this deliverable is to define, propose a design and develop a core set of security measures that will conform the first version of COMPOSITION Security Framework. The purpose of this framework will be to guarantee security, confidentiality, integrity and availability of managed information for all authorized stakeholders in the supply chain.

Some of the modules proposed in this deliverable ensure trusted and secure collaboration, at the same time guarantee confidentiality and integrity of the information transmitted addressing end-to-end security across all layers of the system integrating in a seamless manner three major groups of security mechanisms: authentication, access control and transport security; while other components ensure protection against cyber-attacks and provide security monitoring.

The architecture is based on well established guidelines and best practices but also includes innovative and experimental solutions that will guard the COMPOSITION system against unknown threats.

At this point, this deliverable will not provide a detailed description of the integration of the security mechanisms with the COMPOSITION system; however those will be the result of feature experiments and tests and will presented in the deliverable D4.2 Security Framework II due in month 18.

2 Introduction

Deliverable D4.1 Security Framework I reports the results of the Security Framework design activities for the COMPOSTION platform. It describes the current architecture design and components that conforms it for the MS5 milestone in project month 12. This deliverable will be followed by deliverable D4.2 Security Framework II in project month 18 providing an updated version of the security framework. No component from the Security Framework using blockchain technology is described on this deliverable but on D4.3 The Composition Blockchain due on month 30.

2.1 Purpose, context and scope of this deliverable

The purpose of the deliverable is to propose a first design of a security framework that will ensure trusted and secure cooperation providing protection and monitoring against cyber-attacks. A set of components have been envisioned based on the following needs and requirements:

- Well-established authentication mechanism along with a multi-stakeholder attribute based access control mechanism. This combination should provide, based on a security token included within a submitted request and the evaluation of security policies, fine-grained access control to the data.
- guarantee the confidentiality and integrity of data in motion with the use of cryptographic mechanisms at transport layer
- ensure the security monitoring and protection against potential threats identified in collaborative manufacturing and logistics ecosystems

The solutions provided are an integral part of the COMPOSITION ecosystem and build upon the architecture specification provided in deliverable D2.3 The COMPOSITION architecture specification I.

2.2 Content and structure of this deliverable

This deliverable is structured as follows:

Section 2 - Introduction: serves as introduction and identifies the purpose, scope and context of this deliverable.

Section 3 - Security Framework Architecture: focuses on the architecture general overview giving brief description of current envisioned components. It provides a context for Chapter 4 which focuses on particular components themselves.

Section 4 - Security Framework Components: deliberates about the components but also resembles the steps through which the actors have to go through in order to use the system, which are: authentication, authorization and behavioural analysis.

Section 5 - Next Steps: gives an overview on the future work.

3 Security Framework Architecture

Composition security framework consists of initially five components, each with a task to fulfil: an authentication service, an authorization service, an authentication and authorization service for COMPOSITION message broker, XL-SIEM which is a Security Information and Event Management system (SIEM) with additional functionalities and a reverse proxy. Each component is briefly described below and a more detailed description of each of them can be found in the following chapters of this document.

Authentication service

The responsibility of this component is to provide the authentication mechanisms for users, applications, services and devices. It uses standard protocols such as OpenID Connect, OAuth 2.0 and SAML.

Authorization service

This component is responsible for providing authorization and privacy access control to resources based on policies. In this particular case the policies are sets of conditions that define whether a user should be permitted or denied access to a protected resource. It´s based on XACML 3.0 and provides two different functionalities:

- Policy management: Ability to manage policies, which means generating, storing, removing and modifying policies.
- Policy enforcement: Enforce that a given access request for a specific resource fulfils the requirements of the policies applicable to the resource trying to be accessed.

Message Broker authentication and authorization service

This component enables the use of COMPOSITION Authentication and Authorization services, overriding the use of the message broker built-in mechanisms for authentication and authorization. The reason behind this service is to have identity, access management and authorization policies centralised in COMPOSITION Authentication and Authorization services instead of having them scattered. This approach will be used on any other COMPOSITION component with custom authentication and/or authorization mechanisms, if possible.

XL-SIEM

This component, along with cyberagents, provides capabilities of a SIEM solution with the advantage of being able to handle large volumes of data and raise security alerts from a business perspective, thanks to analysis and event processing in Storm cluster. The main functionalities can be summarized by the following points:

- Real-time collection and analysis of security events.
- Prioritization, filtering and normalization of the data gathered from different sources.
- Consolidation and correlation of security events to carry out a risk assessment and generation of alarms and reports.

Reverse proxy

The main task of this component is to direct client requests to the appropriate backend server and also enable the use of TLS (Transport Layer Security), which is a cryptographic protocol that provides security over a computer network, and aims primarily to provide privacy and data integrity between two communicating applications. It will also provide an additional defence layer against security attacks by protecting identities of servers and services.

The diagram below (Figure 1) from D2.3 The COMPOSITION architecture specification provides a high-level view of COMPOSTION components and shows the interaction between the different components including the security framework.

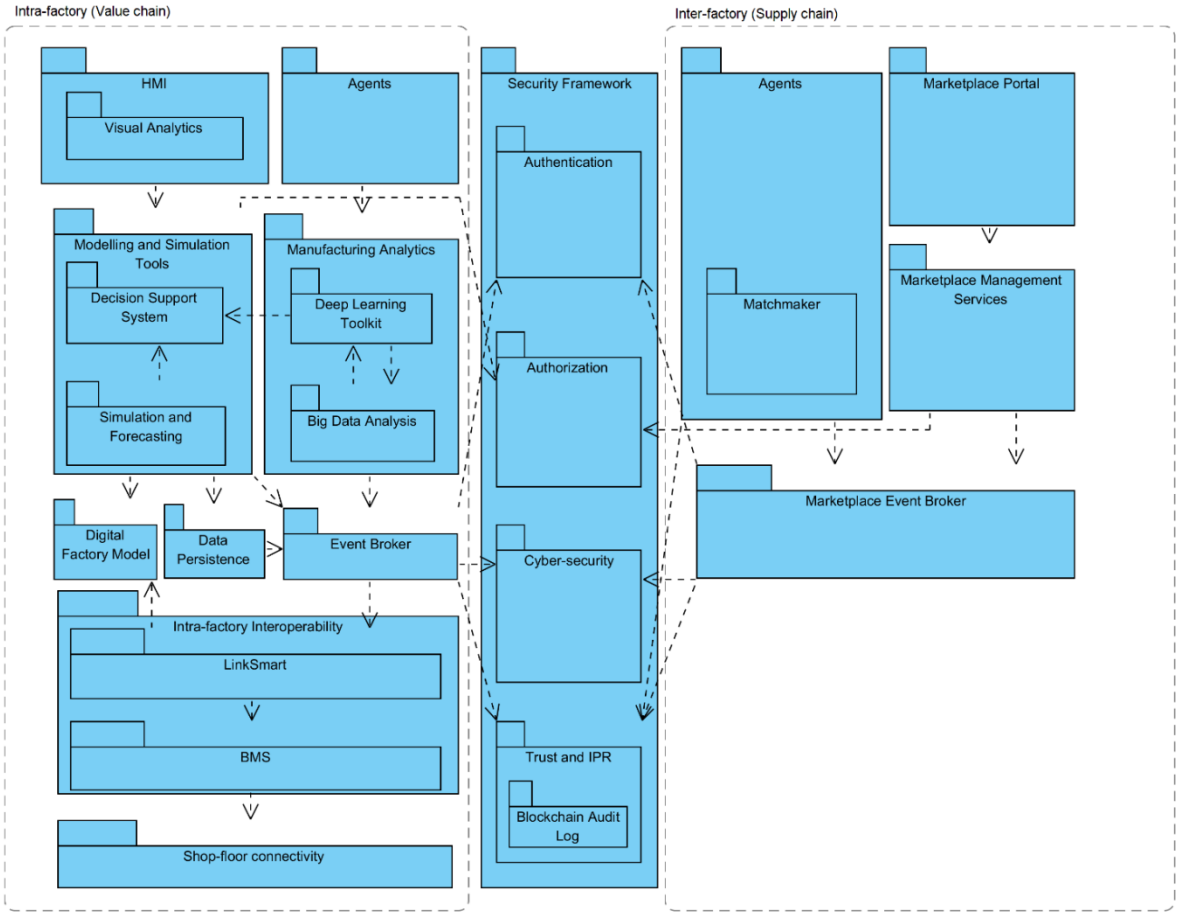

Figure 1: High-level functional view of COMPOSITION architecture

The following diagram presented at the Figure 2 provides an overview of the initial security components and the relationships between them and with other Composition components.

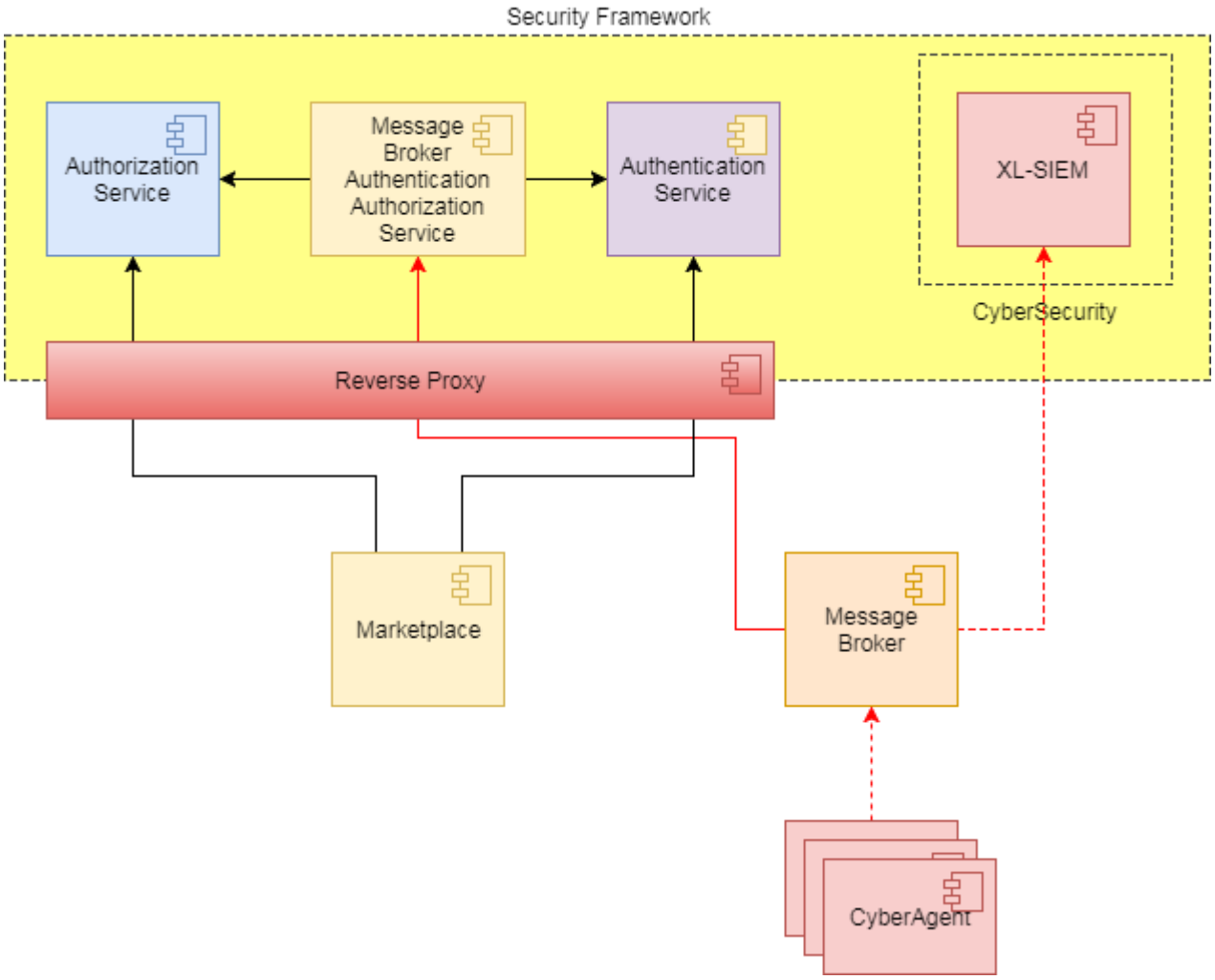

Figure 2: Security framework overview

The process to get access to a resource is described on the following flowchart diagram (Figure 3). In the diagram it can be seen how to get access to a resource in the COMPOSITION Marketplace, first a request to it need to be done. If the user is logged-in the Marketplace the next step is to check if the user is allowed to access the resource; in case the Authorization service grants access to the resource the Marketplace returns the resources, otherwise a denied message is returned. In case the user it´s not logged-in or the session is no longer valid the user must enter valid credentials which are verified by the Authentication service. If the credentials are not valid the user must retry and provide valid credentials to continue; if the credentials are valid the process continues by checking if the user is allowed to access the requested resource, as explained before.

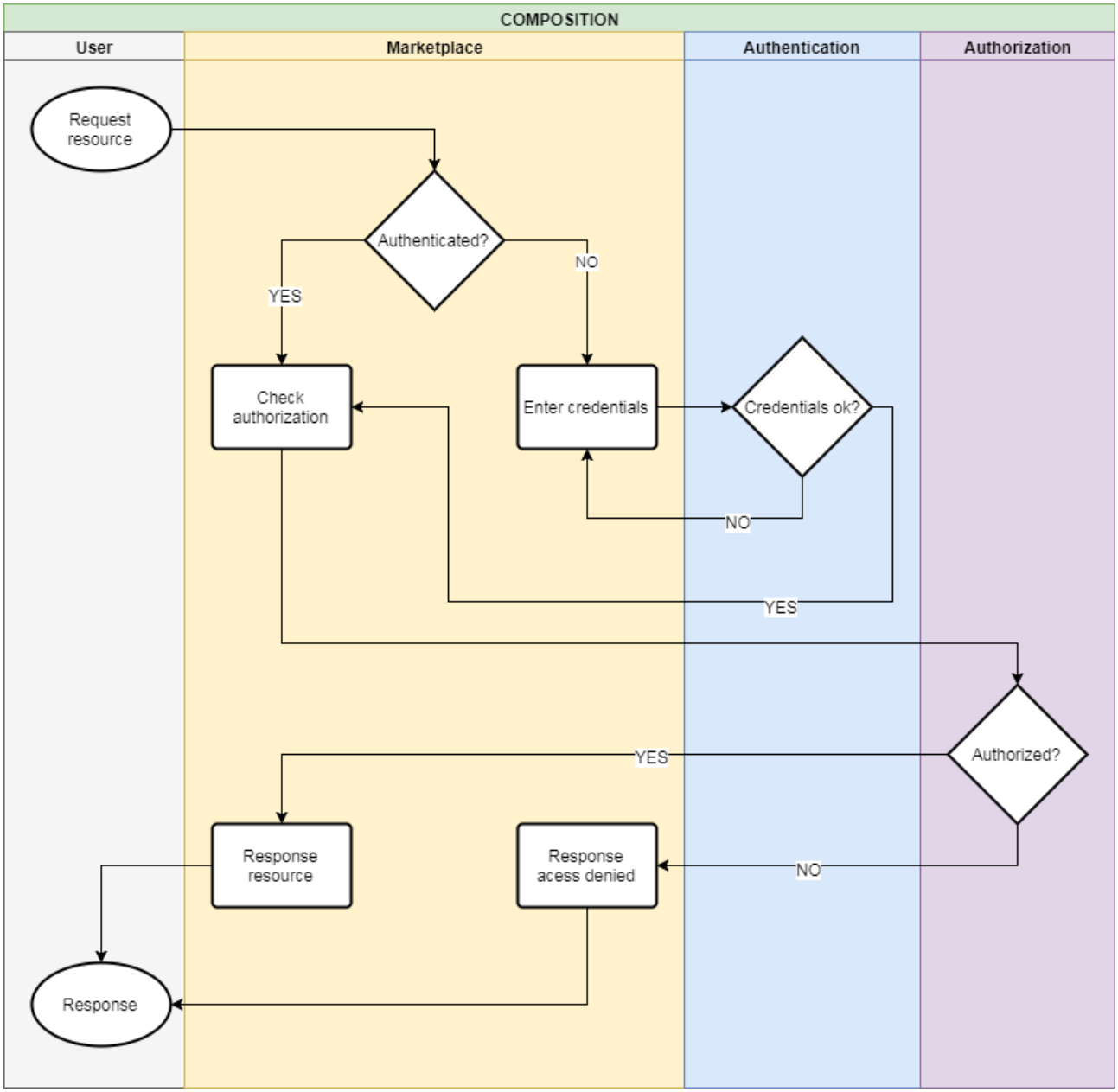

Figure 3: Request resource process

All data transfers within COMPOSITION components will be encrypted and TLS security protocol will be used. In case of the Security Framework a Reverse Proxy will be used on top of all components to provide TLS encryption instead of being provided by each component separately, as can be seen in the previous diagram.

In the case of XL-SIEM, although it provides different ways to communicate with the Cyber-Agents, in this specific case it has been decided to make use of the COMPOSTION Message Broker for such task. The COMPOSITION Message Broker will provide encrypted communication through TLS protocol, as well.

Any COMPOSTION application/service will use of Authentication service to authenticate and identify the users and/or external clients. Also, authorization service will be used by COMPOSITION applications/services to grant /deny access to specific resources requested by users and/or external clients.

The most generic procedure is when an application wants to authenticate a user using the username and password as credentials. The application exchanges the credentials with the Authentication Service and it returns a JWT (JSON Web Token) containing a set of tokens asserting a number of claims in case the authentication succeeded, otherwise an error message.

When a user wants to access any resource, the application validates the access using the Authorization Service. In this case, the application exchanges the id/name of the resource and the token obtained from the Authentication Service with the Authorization Service that will return if the user is allowed or not to access the required resource.

The following diagram presented at Figure 4 gives an overview of the processes of authentication and authorization as well as the components involved:

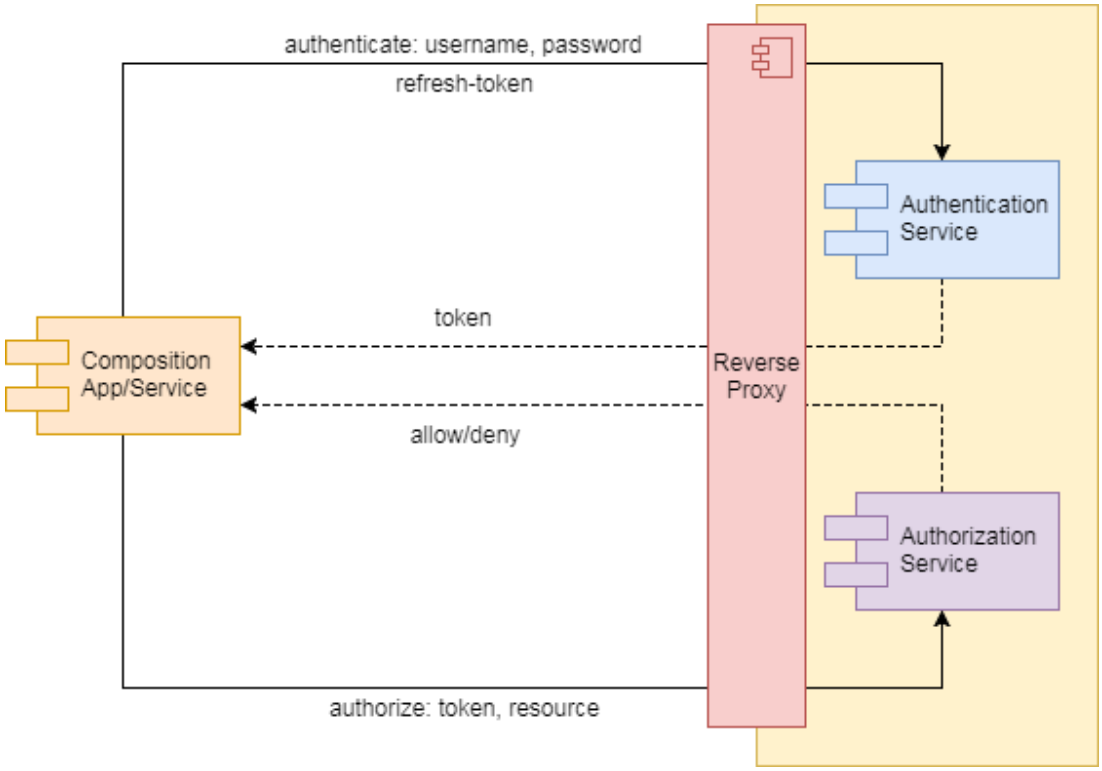

Figure 4: General authentication and authorization

In the case of COMPOSITION Message Broker (see Figure 5), authentication and authorisation processes are quite similar to what previously is described for a generic COMPOSITION application/service, except for the use of an added component that will override Message Broker built-in authentication and authorization mechanisms.

The following diagram gives an overview of the authentication and authorization processes for the Message Broker as well as the components involved.

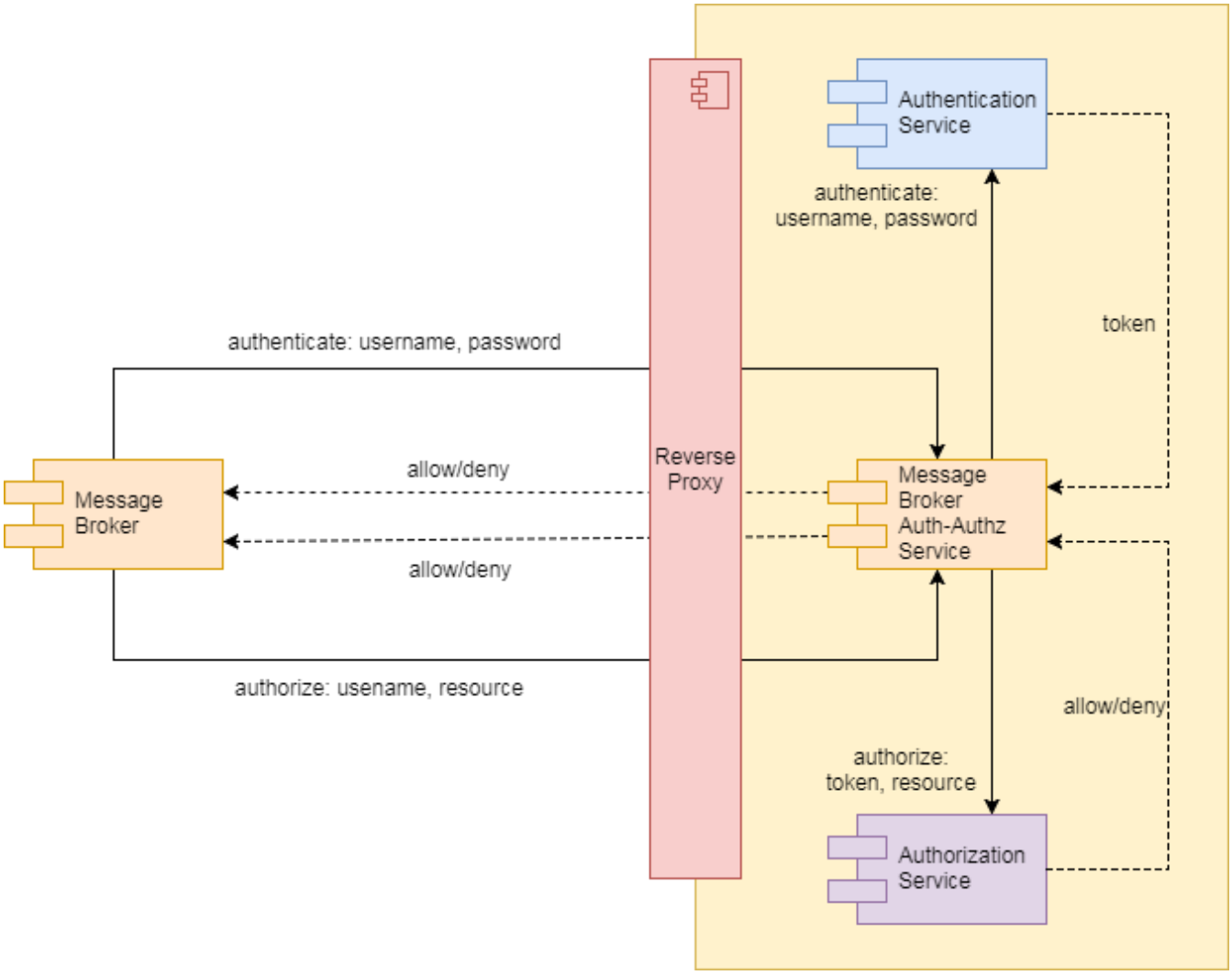

Figure 5: Message Broker authentication and authorization

4 Security Framework Components

4.1 Authentication Service

This component is the responsible for providing the authentication mechanisms for users, applications, services and devices within COMPOSITION. From the available solutions and based on the authorization needs and requirements, a solution capable of securing applications and services, extensible if needed, lightweight, scalable and supporting standard protocols is needed; the preferred option to become COMPOSITION Authentication Service was Keycloak[1].

The main features provided by the solution from which COMPOSITION security framework will take advantage:

- Standard protocols – The following authentication standard protocols are supported:
  - OAuth 2.0: It´s the industry-standard protocol for authorization. Makes heavy use of the JSON Web Token (JWT) set of standards[2].
  - Open ID Connect (OIDC): Authentication protocol based on OAuth 2.0. Unlike OAuth 2.0 OIDC is an authentication and authorization protocol.

[1] http://www.keycloak.org/
[2] https://tools.ietf.org/html/rfc7519

    - SAML 2.0: Authentication protocol similar to OIDC, but older, that relies on the exchange of XML documents between the authentication server and the application.

- Extensibility – Although almost all use-cases are covered out-of-the-box there is the possibility to customize the Authentication Service through the Service Provider Interface (SPI) framework which offers the possibility to implement custom providers or override built-in ones. There is also the possibility to extends core functionalities, like adding custom REST endpoints or add custom SPI.
- Client Adapters – A wide range of libraries to secure applications and services are available for most platforms/programming languages.
- Themes – Integration with other COMPOSITION web apps is possible though the use of Themes, which is provided for web pages and emails, this allows customizing the look and feel of end-user pages. Different types of customization are available:
    - Welcome page
    - Login forms
    - Administration console
    - User account management
    - Emails
- Social Login – Built-in support for the most common social networks is provided which allows delegating authentication to a semi-trusted and respected entity where the user already has an account. Most common social networks are supported, like Google, Facebook, Twitter, Github, LinkedIn, Microsoft and StackOverflow.

From the standard protocols available COMPOSITION Authentication Service will make use of OIDC as its stands as the most used authentication protocol nowadays.

The next UML diagram (Figure 6) shows the process of authenticating a user accessing a web application within COMPOSITION. Whenever a user requests a page and if the user is not authenticated or the session is not valid the user is redirected to the Authentication service which returns the login page to enter the credentials. After the credentials are entered they are submitted to the Authentication service and if they are valid the user is redirected again the application that delivers the page requested by the user.

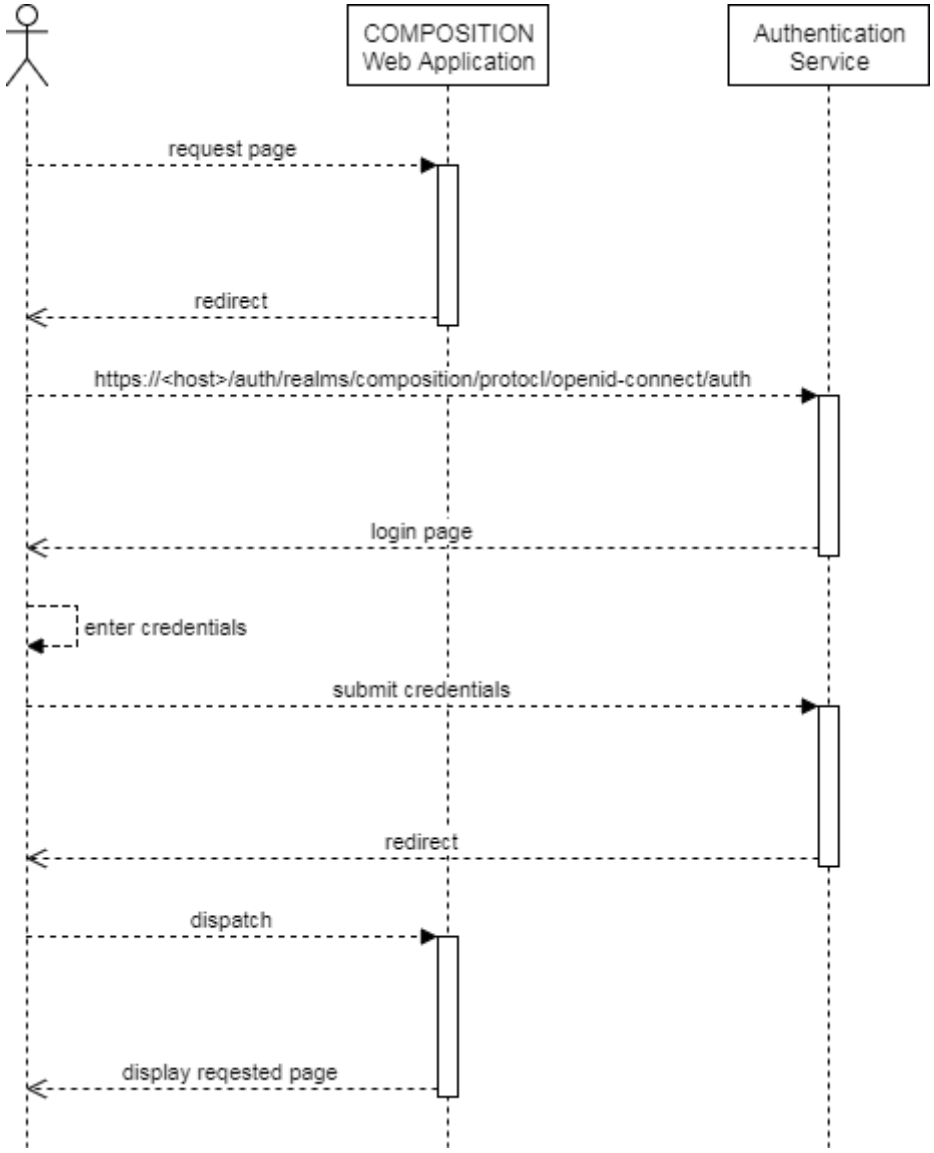

Figure 6: Web application authentication

The following Figure 7 describes the process to authenticate a web service client connecting to a COMPOSITION web service. In this case whenever a client wants to access to a web service, it should first obtain a valid token from the Authentication service by providing the valid credentials for the client. When the token it´s obtained the client can start making the requests to the web service but only for the time the token it´s valid. Once the token it´s not valid anymore a new one should be obtained.

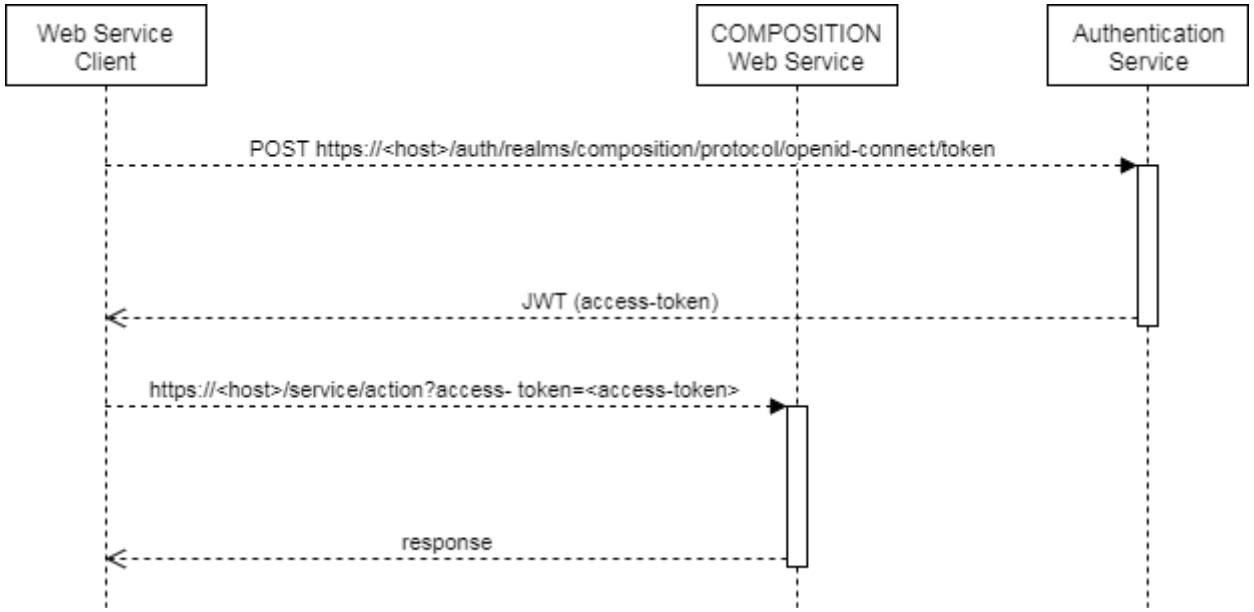

Figure 7: Web service authentication

Any application or service that needs to be secured can make use of the any the available client adapters depending on the platform, but in case no client adapter is used the following endpoints are available to talk OIDC with the authentication server:

- **/auth/realms/composition/protocol/openid-connect/token**

  URL endpoint for obtaining a temporary code or obtaining a token

- **/auth/realms/composition/protocol/openid-connect/userinfo**

  URL endpoint for the User Info service described in the OIDC specification

- **/auth/realms/composition/protocol/openid-connect/logout**

  URL endpoint for performing logouts

4.2 Authorization Service

Authorization service is provided by EPICA, a component based on XACML v3.0[3] that provides an Attribute-based access control mechanism. It provides the means to define policies used to protect resources, then any request to access a protected resource will first be evaluated against the policies and the evaluation result will be enforced depending on the outcome.

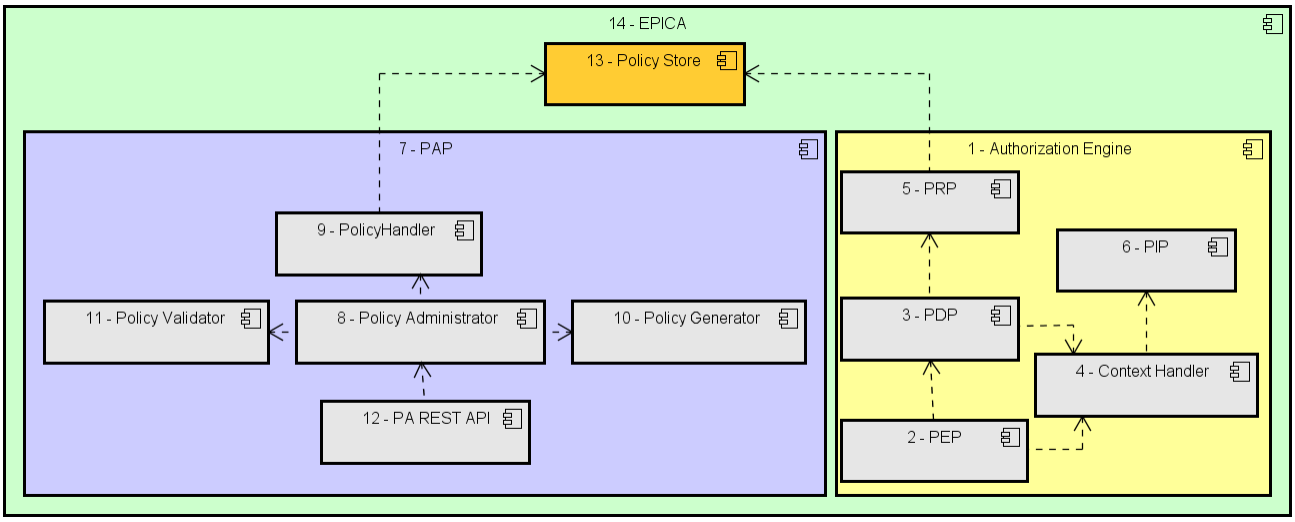

Figure 8: Architecture of EPICA

As Figure 8 displays EPICA is divided into two main subcomponents: the Authorization engine and the Policy Administration Point (PAP).

Authorization engine contains all the components according to the XACML architecture, the Policy Enforcement Point (PEP) is the entry point which receives the request in the native format of the system (e.g. HTTP), then it contacts the Context Handler to transform this request into a XACML request, for this it is also required to contact the Policy Information Points (PIPs) that are active, since these could provide extra information to the request. Once the XACML request is complete the Policy Decision Point (PDP) receives it and finds the policies that are applicable to this request through the Policy Retrieval Point (PRP), then it evaluates the request against the policies and provides the result to the PEP, which will enforce it accordingly.

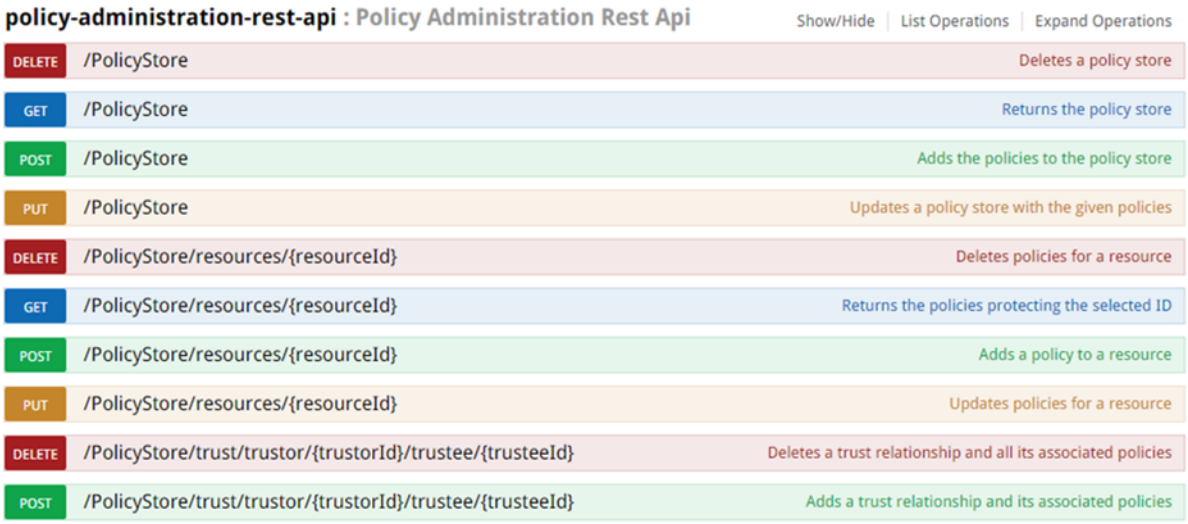

Figure 9: Policy administration REST API

[3] https://www.oasis-open.org/committees/xacml/

PAP is in charge of managing the policies that are used to protect resources. It is done through a RESTful API, shown in Figure 9, which provides different methods. When creating or updating a policy the Policy Generator is invoked to create it, then it is validated by the Policy Validator and lastly the Policy Handler is the tool that interacts directly with the Policy Store to create, update, retrieve and delete policies.

As show in the picture, Policy Store is a component used by both Authorization Engine and PAP. This component is not exactly a part provided by EPICA, it is just used to represent where the policies are stored. Currently the policies are stored directly on the filesystem following a folder structure that depends on the configuration selected and on the kind of resources that are being used.

EPICA is highly configurable in order to be able to adapt to different environments. For instance in a cloud environment it can be configured to work with one or with multiple tenants. Meanwhile depending on the kind of resources being protected it can configure the policy store accordingly to work in a more efficient manner. Depending on these options there are some components of EPICA that are adaptable, in particular the Policy Handler and the PRP will vary depending on the way the policies are stored.

Furthermore, since the PEP is the main entry point of the application it varies depending on the system, since it can be adapted to work as a deployed web service accepting HTTP requests or it could be deployed to be accessible only locally by invoking a java binary file.

4.3 Message Broker Authentication/Authorization Service

This service is developed to facilitate the use of COMPOSITION Authentication and Authorization services, instead of using the built-in mechanisms of COMPOSITION Message Broker.

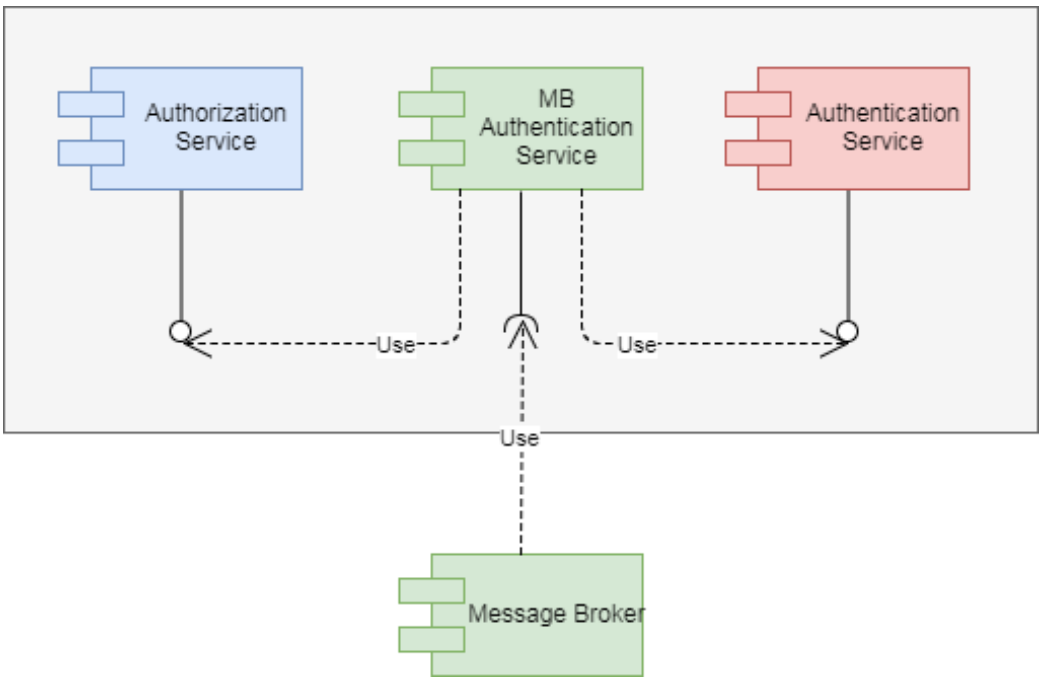

Figure 10: Message Broker Authentication/Authorization diagram overview

In order to achieve this, COMPOSITION Message Broker will be configured to override the built-in mechanisms for authentication and authorization and will use an available plugin to connect to the COMPOSITION MB Authentication/Authorization service. The service will provide the following web interface methods to enable communication with the Message Broker:

- http(s)://server/auth/user
- http(s)://server/auth/vhost
- http(s)://server/auth/resource
- http(s)://server/auth/topic

The service will be responsible to authenticate against COMPOSTION Authentication service obtaining a token from when authentication succeeds and using this token to check against Authorization service when access to a resource is requested from the client. It will be also responsible to check the validity of the tokens and renew them in case they are no longer valid, as well as verify the signature of the tokens, revoking the access if it is not valid.

Figure 5 offers a detailed view of all security components related with it and interactions between them.

In order to communicate with the Authentication service, the following endpoint (see Table 1), already described in section 4.1, will be used in both actions: login and refresh token.

Table 1: Login and Refresh token actions

/auth/realms/composition/protocol/openid-connect/token	
action	**parameters**
login (authenticate user and get access- token)	grant_type=password client_id=rabbitmq username=xxx password=xxx client_secret=xxx response_type=token
refresh-token (obtain new acces-token when current one expired)	grant_type=refresh_token client_id=rabbitmq client_secret=xxx refresh_token=xxx

4.4 XL-SIEM

The development of XL-SIEM (Cross-Layer SIEM) started in context of the European initiative FIWARE[4] as part of the Security Monitoring Generic Enabler (part of the FIWARE Platform) with the aim to get around limitations of open source SIEMs available in the market. In particular, to extend their capabilities and enhance their performance, allowing processing larger amounts of data and add correlation of events at different layers with more complex rules. XL-SIEM has been subsequently improved and validated throughout different projects such as ACDC (Advanced Cyber Defence Centre)[5], FI-XIFI[6], SAGA (Secured Grid metering Architecture), RERUM (Reliable, Resilient and secUre IoT for sMart city applications)[7] or WISER (Wide-Impact cyber Security Risk framework)[8].

It integrates a set of Java processes, including the high-performance correlation engine Esper[9] library, packaged into a topology to be deployed in an Apache Storm cluster. Apache Storm[10] is an open source distributed real-time computation system for processing large volumes of data.

The architecture of XL-SIEM is presented on Figure 11Figure 11: XL-SIEM architecture The SIEM Agents are responsible for data collection and are deployed within the monitored infrastructure. In case of any event occurrence, they are sent to XL-SIEM core where they are processed and correlated. The OSSIM is responsible for storing gathered events and eventual alarms that were produced during the correlation process. The OSSIM has also visualisation capabilities enabling data inspection.

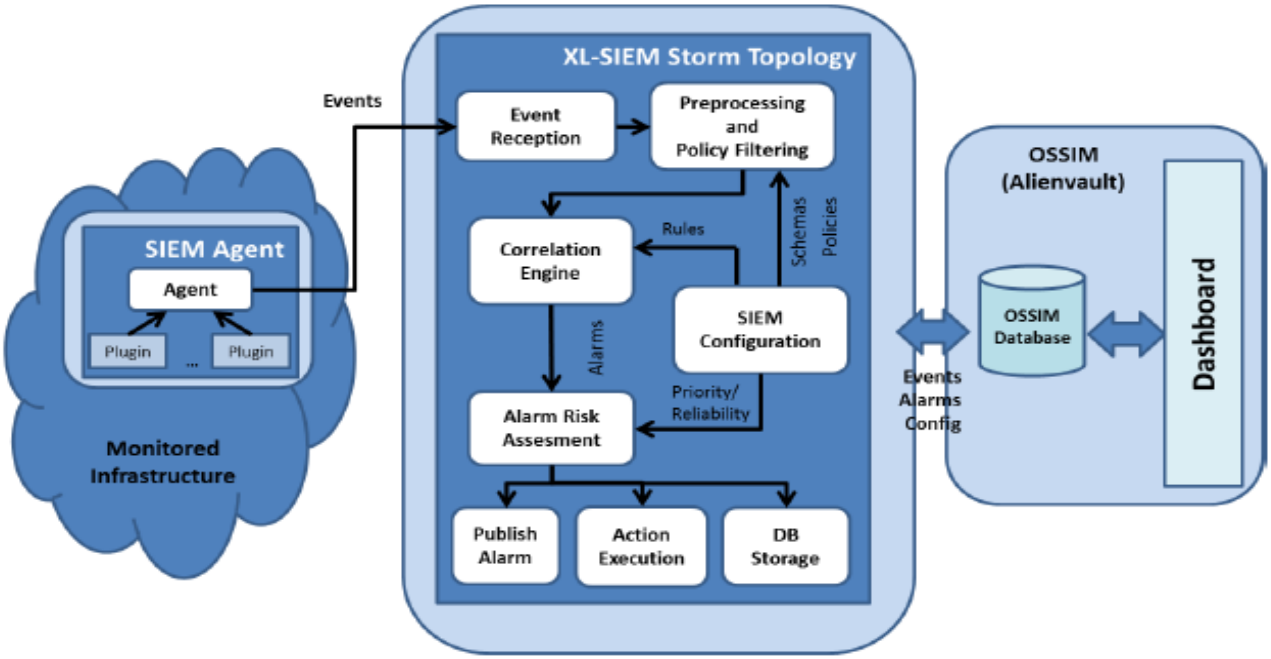

Figure 11: XL-SIEM architecture

4.4.1 Data sources supported

Data sources supported by OSSIM are also supported by XL-SIEM. Additionally XL-SIEM was extended with the following:

- STIX (Structured Threat Information eXpression)[11] format data: Cyber-threat observations, represented using this type of structured language for cyber threat intelligence, are supported through a STIX plug-in developed by Atos in the project ACDC and added to the SIEM agents. This

[4] https://www.fiware.org/
[5] https://www.acdc-project.eu/
[6] https://www.fi-xifi.eu/home.html
[7] https://ict-rerum.eu/
[8] https://www.cyberwiser.eu/
[9] http://www.espertech.com/products/esper.php
[10] http://storm.apache.org/
[11] https://stixproject.github.io/

plug-in parses STIX data and generates its representation in the OSSIM normalized event format used in the XL-SIEM.

- JSON format data: JSON format is supported, while received from the open source message broker RabbitMQ[12] that implements the Advanced Message Queuing Protocol (AMQP).

Additional features added to SIEM agents include secure transmission, where normalized events generated by agents can be sent encrypted from the monitored infrastructure to XL-SIEM server using the TLS (Transport Layer Security) protocol, as well as, data anonymization, where user has the possibility of defining which fields in an event type are sensitive and need to be transmitted and stored anonymized. In the case of IP addresses, it is done as pseudo-anonymization where the original IP addresses are stored in a database together with the anonymized IP in order to be able of recovering the original one if required. For the rest of fields in an event they are anonymized using a salt (random data) which is added to the cryptographic hash function used for encryption.

4.4.2 Data storage capabilities

The format used by XL-SIEM for events and alerts storage can be summarized in the following points:

- Data is stored in MySQL relational databases.
- There is a separate database for historical data.
- Currently, integration with cloud storage services is not supported.

Data storage exists in a different machine than the one where event processing takes place improving performance, or increasing storage capacity.

XL-SIEM takes advantage of the OSSIM data storage capabilities in order to allow integration between detection done by the open source SIEM and the one done by the XL-SIEM correlation processes. For this reason, data storage capabilities included in the OSSIM version are also available in XL-SIEM.

Besides, the XL-SIEM includes the capability to store both, events gathered by the agents and alarms generated by the server, for example in an external database using a RabbitMQ server. The format supported to send events and alarms in this case is JSON.

4.4.3 Processing capabilities

One of the advantages of XL-SIEM architecture is the use of a high-performance correlation engine running in an Apache Storm cluster for the processing of the incoming security events.

Thanks to Apache Storm running together with Apache ZooKeeper[13] and ZeroMQ[14], distributed and real-time processing of events along with scalability could have been achieved. Apache Strom distributes the load between different servers supporting to increase the number of machines in the cluster. Apache ZooKeeper provides distributed synchronization across the Storm cluster maintaining centralized the configuration information. Communication among concurrent processes in the Storm cluster is done using the asynchronous distributed messaging ZeroMQ, without dedicated message broker and reducing the latency.

Storm cluster is capable of running multiple correlation processes defined through XL-SIME graphical interface, each of which can have a different set of rules and “parallelism” set up.

XL-SIEM includes the possibility of defining different data schemas through the graphical interface. When configured a new correlation process, one can use one of the existing data schemas or create a new of according to the expected format of new events. Additionally, to reduce the number of events arriving to a specific correlation engine various filtering policies can be applied in the chain. These filters also enable XL-SIEM to support multi-tenant processing capabilities. That allows each client having separated policies based on the agents belonging to a specific organization/system where the data will be collected.

As a consequence, this architecture is capable of real-time distribution dispatching among different machines not only the correlation processes but also the support of different filtering policies, rules and data schemas associated to each correlation process.

[12] https://www.rabbitmq.com/
[13] https://zookeeper.apache.org/
[14] http://zeromq.org/

4.4.4 Flexibility in security directives

XL-SIEM provides the ability to configure different correlation processes. For each correlation process, the user has flexibility to configure:

- Filtering policies that will be applied before the events reach the correlation engine.
- Actions that will be executed when an alarm is triggered.
- Fields grouping to be used with the events arriving to a specific correlation process. Apache Storm includes a feature called “stream grouping” which allows configuring how the incoming stream to a process (in this case, the set of fields included the normalized XL-SIEM event arriving to the correlation process) will be partitioned among its different tasks (in this case, if parallelism of correlation process is higher than one, there will be a task by each processing thread). This is an advanced feature that probably only will be used by expert users but can improve the performance in case the rules are based on some specific field.
- The rules or security directives that will be enabled in that correlation engine to trigger alarms.

XL-SIEM supports two ways of defining the security directives:

- **Pre-configured categories of rules –** The user can select one or several directive categories to be included in each correlation process. The categories are: scans behaviours, malware detection, denial of service attacks, brute force attacks or network attacks. Each of these categories includes a pre-configured set of rules or security directives related to that different topic that will be enabled in the correlation engine.
- **User custom rules –** The user can also configure his/her own rules or security directives to be used in each correlation process. In order to correlate events and generate alarms, the open-source Java-based Esper is used in the XL-SIEM. It implies that security rules are expressed in Event Processing Language (EPL).

EPL is a declarative programming language similar to SQL (Structured Query Language) which allow expressing security directives with rich event conditions and patterns in a simple way. The usage of SQL clauses, such as select, order by, group by or where, in the definition of the rules that will trigger the alarms as well as SQL concepts such as joins or filtering through sub-queries, add a business perspective to the definition of security directives.

Through XL-SIEM graphical interface, the user can setup three levels in the definition of security directives:

- **EPL directives:** Those trigger alarms in XL-SIEM. Each rule specifies a listener mechanism that is added automatically to Esper correlation engine instance that notifies a pattern occurrence as soon as it occurs. The user has flexibility to select the reliability and priority values associated to the alarm related to the definition of each security directive. These EPL directives can use EPL variables and EPL statements to make them clearer from a business perspective or for non-expert users. Another feature included in the XL-SIEM is the possibility of generating alarms based on previous alarms (cross-alarms). That is, to define security directives based on previous security directives.
- **EPL variables:** These are values to be used in different EPL statements or directives that can change in the time (for example, to define a timeout). In this way, the user does not need to modify each security directive to change the value of a variable used in several of them.
- **EPL statements:** In case it will be necessary to forward events from one stream to another, for instance to provide a filtering in the incoming events, these EPL statements will no longer trigger alarms but serve as input to another EPL statements or directives.

4.4.5 Risk analysis capacity

XL-SIEM includes a risk assessment procedure analogous to the one provided by AlienVault SIEM solutions. This risk analysis takes into consideration the following aspects: Reliability, Priority and Asset relevance.

- Priority: How urgently an event should be investigated; ranging 0 to 5, 0 being the least important and 5 the most important.
- Reliability: The chance an event is a false positive; ranging 0 to 10, 0 being the least important and 10 the most important.
- Asset relevance: Relevance of the asset (piece of equipment on the network that bears a unique IP address) being protected; ranging 0 to 5, 0 being the least important and 5 the most important

The final risk assessment to be stored with the alarm generated will be the result of the following formula:

Risk = (priority * reliability * asset) / 25

The resulting value can be mapped to the following Risk Categories:

0, 1, 2 = Low

3, 4 = Precaution

5, 6 = Elevated

7, 8 = High

9, 10 = Very High

4.4.6 Exposed APIs

Apart from APIs inherited from the fact that XL-SIEM deployment integrates the open source OSSIM, the following ones are available in XL-SIEM:

- **Alarms via RabbitMQ:** Alarms generated by XL-SIEM can be also sent in JSON format to a RabbitMQ server. This output supports the use of TLS protocol.
- **Alarms view DDS:** XL-SIEM supports to send the alarms generated using the Data Distribution Service (DDS)[15], commonly used for real-time systems.
- **Events via RabbitMQ:** The events generated by SIEM agents from collected data by sensors can be sent in JSON format to a RabbitMQ server. This output supports the use of TLS protocol.
- **DRPC service to provide network topology JSON:** XL-SIEM includes a Distributed Remote Procedure Call (DRPC)[16] service that can be invoked to get a JSON which includes the monitored network topology with the alarms associated to each node. This JSON can be used e.g. to generate a graphical representation with the status of the network.

4.4.7 Resilience

Resilience capabilities included in the XL-SIEM are the ones provided by the setup of a supervisory process called Daemontools[17] (although any other supervisor could be used) to monitor Apache Storm (where XL-SIEM processes are running) and Apache Zookeeper processes.

Apache Strom is fault-tolerant[18] in the sense that if one of the worker processes running in a node is not responding, it will be automatically restarted by the storm daemon called supervisor. If a node will stop to respond, the workers running on it will be automatically restarted on another node in the cluster. This reassignment is done by the storm daemon called nimbus thanks to a heartbeat system and the coordination service provided by Apache Zookeeper.

However, Storm is also a fail-fast system and therefore in case of any unexpected error, the processes will automatically halt. Consequently, if there is only one nimbus instance running on the cluster, the fault tolerance provided by Apache Storm is not enough. Running the Storm daemons under supervision using

[15] http://portals.omg.org/dds/
[16] http://storm.apache.org/releases/1.0.0/Distributed-RPC.html
[17] http://cr.yp.to/daemontools.html
[18] http://storm.apache.org/releases/current/Fault-tolerance.html

Daemontools assures they will be automatically restarted in case of failure. And for the same reason, Apache Zookeeper also needs to be run under supervision.

4.4.8 Security event management and visualization capabilities

XL-SIEM web graphical interface presented at the Figure 12, is built on top of OSSIM dashboard and consequently it integrates its visualization capabilities. It facilitates usage of the XL-SIEM by providing high-level charts and diagrams, through various diagrams. The flexibility of the diagrams allows adaptation based on the user needs.

Examples of the provided dashboards:

- **Executive dashboard –** it shows a color-coded high-level plot relevant for the C-level administrator, for instance the current threat level of the monitored system.
- **Operational dashboard –** This dashboard is aimed at decision taking administrators indicating things such as top 5 incidents, identified hosts and the source of the security alarms.
- **Situational Awareness dashboard –** This dashboard represents a color-coded monitored network topology, including the number of alerts detected in each component. It takes advantage of the DRPC service.

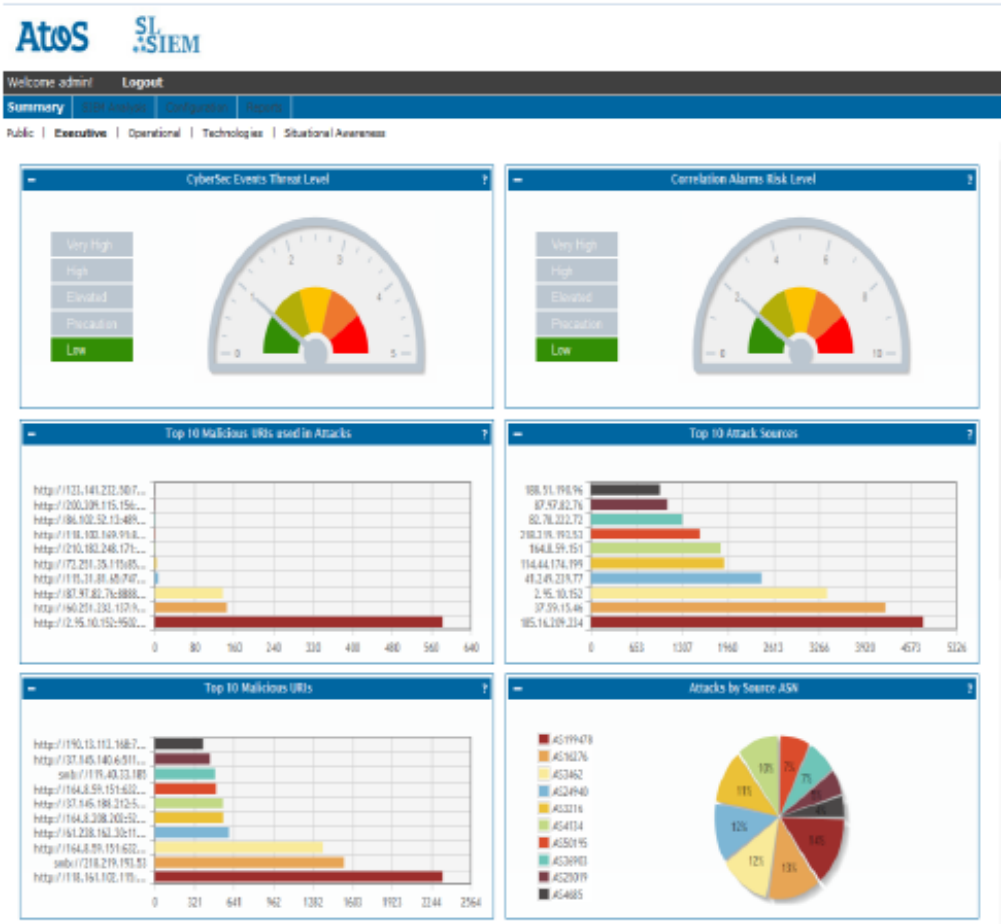

Figure 12: XL-SIEM web graphical interface

4.4.9 Reaction capabilities

The reaction capabilities include various notification channels such as creation of tickets or email notification. These mechanisms also allow executing third party scripts.

Associating actions with different correlation processes allows execution of those actions on specific alerts, and not only associate them to specific, defined policies. It is also possible to increase performance (reducing the reaction time) by taking advantage of distributed Storm topology.

Additionally, XL-SIEM provides a Decision Support System (DSS) to help the user to analyse the risks detected and select suitable mitigation measures. Based on business data provided by the user and a set of mitigation measures associated to the different risks, this component offers an analysis of the societal impact of the risks as well as an analysis of costs and benefits of the mitigation measures proposed.

4.5 Reverse Proxy

A reverse proxy (see diagram at Figure 13) is a type of proxy server which accepts client requests and direct to the appropriate backend server behind it. Reverse proxies servers add an additional level of abstraction providing a single point of access and control.

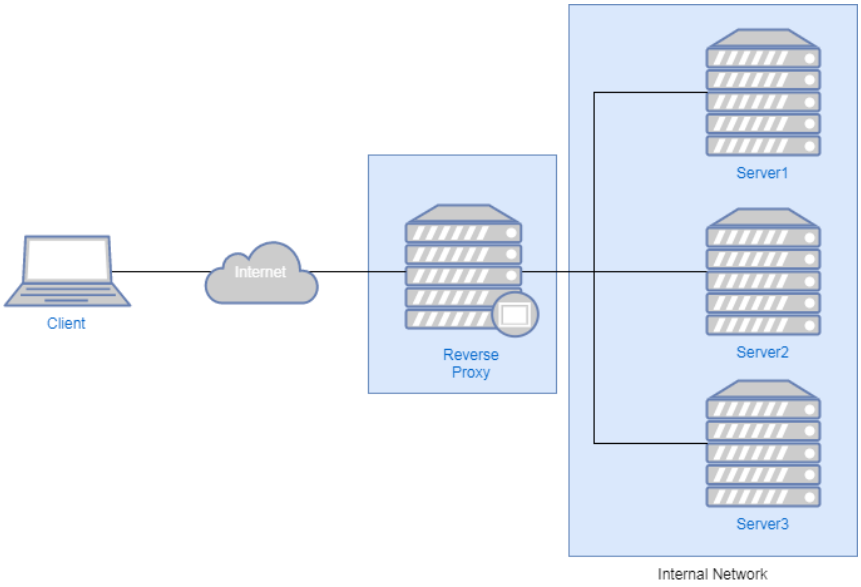

Figure 13: Reverse proxy diagram

From common uses of a reverse proxy, COMPOSITION security framework reverse proxy will be used mainly to provide security and anonymity; and if needed as a web accelerator. Currently it is not envisioned it´s used as a load balancing server.

Security and anonymity

Identities and characteristics of backend servers as well as topology of the network will be protected, as all requests headed to the servers will be intercepted by the reverse proxy. The reverse proxy will be the only one single point of access to the framework and its servers. It will also ensure that multiple servers can be accessed from a single URL, regardless of the structure of the network.

TLS/SSL encryption will be also provided by the reverse proxy, instead of each server; this will ease the management and will also improve performance.

Web acceleration

The ability to compress inbound and outbound data, as well as cache content, will bring benefit on the communication between clients and servers by speed increase in data flow. A boost in performance will be also appreciated by the fact of providing TLS/SSL encryption in the reverse proxy instead of the servers.

From the available solutions NGINX[19] has been chosen as the reverse proxy in COMPOSITION security framework. Apart from the fact that it´s open source, its ability to scale easily on minimal hardware and it´s light weight resource utilization make it the preferred solution. NGINX also offers a rich feature set.

[19] https://nginx.org/

5 Next Steps

To validate the proposed design and components that form the COMPOSITION Security Framework the following steps will take place in the time being

- Dockerize Authorization Service and Reverse Proxy
- Deploy Authorization Service, Reverse Proxy and XL-SIEM
- Implementation and Integration of the live anomaly detection module with the XL-SIEM.
- Implementation of the GUI interface for systems security state monitoring.
- Integration of the XL-SIEM with the collaborative manufacturing and logistics ecosystem.
- Configure Authentication Service; create first set of clients, roles and users.
- Configure Authorization Service; create first set of rules.
- Configure Reverse Proxy
- Create second prototype of Message Broker Authentication/Authorization Service integrating with the COMPOSITION Authorization Service
- Initial tests and validation of the framework

Any new components added to the Security Framework or changes on current design will be reflected on the upcoming D4.2 Design of the Security Framework II due M18.

6 Summary

This deliverable introduced the security services to be implemented and that will define the Security Framework in the COMPOSTION environment, along with an overview of the authentication and authorization mechanisms and processes. These services supplement each other, creating together a full-scale security framework capable of performing authentication, authorization and monitoring of system components, ensuring not only protection against unauthorized access to the system resources, but also defend against various attacks on the system, whether from the outside or the inside.

A high level overview of the authentication and authorization mechanisms, procedures and interfaces is also presented; while a detailed description of them will be reflected on D4.2 Design of the Security Framework II.

The deliverable presents a generalised integration with the COMPOSITION environment as it is intended to give the reader an overview of the security mechanisms rather than a detailed description of interactions and integration. Those will be the consequence of future comprehensive tests and experiments with the aim of achieving maximum effectiveness in terms of provided security, at the same time meeting the performance requirements of the system and will be reflected on D4.2 Design of the Security Framework II due M18

The outcome of this deliverable affects directly to the work done in WP6 COMPOSITION Collaborative Ecosystem and in some way to the work done in WP5 Key Enabling Technologies for Intra- and Interfactory Interoperability and Data Analysis as it lays the foundation of the security services, methodologies and procedures to be used to secure both components and data transmissions within COMPOSITION system.

7 List of Figures and Tables

7.1 Figures

7.2 Tables

8 References

RFC-6749 The OAuth 2.0 Authorization Framework - https://tools.ietf.org/html/rfc6749

RFC-7515 JSON Web Signature (JWS) - https://tools.ietf.org/html/rfc7515

RFC-7516 JSON Web Encryption (JWE) - https://tools.ietf.org/html/rfc7516

RFC-7519 JSON Web Token (JWT) - https://tools.ietf.org/html/rfc7519

OpenID Connect specification - http://openid.net/specs/openid-connect-core-1_0-final.html

XACML 3.0 committee specification 01 - https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml#CURRENT
SAML V2.0 specification - http://saml.xml.org/saml-specifications
RCF-5246 The Transport Layer Security (TLS) Protocol Version 1.2 - https://tools.ietf.org/html/rfc5246