Proof-of-Concept Implementation of a Hosted Service for the FutureID Framework

D52.3

Document Identification	
Date	25/09/2015
Status	Final version
Version	1.0

Related SP / WP	SP5/ WP55	**Document Reference**	52.3
Related Deliverable(s)	D52.2, D52.1, D24.1, D44.3, D44.4	**Dissemination Level**	PU
Lead Participant	ATOS	**Lead Author**	Juan Carlos Pérez Baún
Contributors	Nuria Ituarte Aranda Miguel Colomer Pastor Charles Bastos Rodriguez Juan Carlos Pérez Baún Alejandro Stewart de la Fuente Detlef Houdeau Jerzy Szczebak	**Reviewers**	Jessica Schroers (KUL) Alfredo Rial (IBM)

Abstract: This deliverable describes the Proof-of-Concept implementation of a Hosted Service such as Atos e-Learning platform for the FutureID framework.

Not to be distributed outside the FutureID

This document is issued within the frame and for the purpose of the *FutureID* project. This project has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement no. 318424.

| **Reference:** | 52.3 | **Dissemination:** | PU | **Version:** | 1.0 | | **Status**: | Draft |

1 Executive Summary

This deliverable describes the Proof-of-Concept implementation of a Hosted Service such as Atos e-Learning platform for the FutureID framework.

In Section 6 we describe the FutureID Apache specific AIS architecture and detail out its components. The AIS component performs the integration of Atos e-Learning services for enterprises with the FutureID architecture using native Apache modules. This FutureID Apache specific component will be embedded into the Apache server itself and its internal structure is based on the original Apache modules architecture.

FutureID Apache AIS implementation consists of two main parts, the Access Filter (AF) and the Simple Credential Transformer (SCT).

Section 7 describes the AF component which represents a layer for access control to the content it applies to. AF has been implemented as an Apache access handler using Python language, built on top of the mod_python module.

The section describes also how the validity of the session for every request is assessed.

The SCT component described in Section 8 contains a handler that listens for POST requests. The SCT receives an assertion, extracts the required attributes and the identifier sent from the Broker Service. All these data are set in a server session variable.

Section 9 describes the generic AIS configuration, the configuration for the Apache settings, the configuration for SAML verification and the attributes needed by the Atos e-Learning platform.

In Section 10 we summarize the development environment and deployment tasks for Atos e-Learning platform.

Finally, in Section 11 we identify the services relevant to Atos e-Learning, as well as other related services that could benefit from the integration of FutureID components. This section describes also how the integration process was conducted.

2 Document Information

2.1 Contributors

Name	Partner
Nuria Ituarte Aranda Miguel Colomer Pastor Charles Bastos Rodriguez Juan Carlos Pérez Baún Alejandro Stewart de la Fuente	ATOS
Jerzy Szczebak	CA
Detlef Houdeau	IFAG

2.2 History

Version	Date	Author	Changes
0.1	13/10/2014	Juan Carlos Pérez Baún Jerzy Szczebak	Determine TOC and include initial content in all sections
0.2	09/03/2015	Juan Carlos Pérez Baún	Updating figures and including Identity Mapping section
0.3	05/05/2015	Nuria Ituarte Aranda	Updating section 6
0.4	28/08/2015	Alejandro Stewart de la Fuente	Updating section 5, 6, 7, 8 and 9.
0.5	02/09/2015	Juan Carlos Pérez Baún	Updating figures and descriptions throughout the document.
0.6	09/09/2015	Nuria Ituarte Aranda Juan Carlos Pérez Baún	Content on section 10, Executed summary. Updating content and figures.
0.7	10/09/2015	Detlef Houdeau	Content on sections 11.1 and 11.2
0.8	16/09/2015	Nuria Ituarte Aranda Jerzy Szczebak Detlef Houdeau Charles Bastos Rodriguez Juan Carlos Pérez Baún	Editors' review
1.0	25/09/2015	Juan Carlos Pérez Baún	Internal review

2.3 Table of Figures

2.4 Table of Tables

2.5 Table of Acronyms

AF	Access Filter
AIS	Application Integration Service
AJAX	Asynchronous JavaScript And XML
BS	Broker Service
CAD	Computer-aided Design
CORS	Cross-Origin Resource Sharing
ENX	European Network Exchange organization
FAR	FutureID Authentication Request
FC	FutureID Client
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
OEM	Original Equipment Manufacturer
OS	Operating System

PHP	Hypertext Pre-processor (server-side scripting language)
R&D	Research and Development
S&E	Solver and Executor
SCT	Simple Credential Transformer
STORK	Secure idenTity acrOss boRders linked
UA	User Agent

2.6 Referenced Documents

[1] - FutureID_D52.02_WP52_v0.5_Technical Specification including Description of IdP SP and Identity Token Formats, https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=4470073

[2] - FutureID_D52 01_WP52_v1.0_Requirements for FutureID components in Business Scenarios_Final, https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=3858403

[3] - FutureID_D21.04_WP21_v1.1_Reference_Architecture, https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=3841750

[4] - FutureID_D44.03_WP44_v1.1_Technical_Specifications_for_AIS,, https://dms-prext.fraunhofer.de/livelink/livelink.exe/overview/4103780

[5] – FutureID Team Certificate: https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=3792309

3 Table of Contents

4 Project Description

The *FutureID* project builds a comprehensive, flexible, privacy-aware and ubiquitously usable identity management infrastructure for Europe, which integrates existing eID technology and trust infrastructures, emerging federated identity management services and modern credential technologies to provide a user-centric system for the trustworthy and accountable management of identity claims.

The *FutureID* infrastructure will provide great benefits to all stakeholders involved in the eID value chain. Users will benefit from the availability of a ubiquitously usable open source eID client that is capable of running on arbitrary desktop PCs, tablets and modern smart phones. *FutureID* will allow application and service providers to easily integrate their existing services with the *FutureID* infrastructure, providing them with the benefits from the strong security offered by eIDs without requiring them to make substantial investments.

This will enable service providers to offer this technology to users as an alternative to username/password based systems, providing them with a choice for a more trustworthy, usable and innovative technology. For existing and emerging trust service providers and card issuers *FutureID* will provide an integrative framework, which eases using their authentication and signature related products across Europe and beyond.

To demonstrate the applicability of the developed technologies and the feasibility of the overall approach *FutureID* will develop two pilot applications and is open for additional application services who want to use the innovative *FutureID* technology.

Future ID is a three-year duration project funded by the European Commission Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 318424.

5 Introduction

This deliverable describes the Proof-of-Concept implementation of a Hosted Service such as Atos e-Learning platform for the FutureID framework. Two tasks T52.3 and T52.4 were devoted to the creation of D52.3. The first task provided the technical test bed landscape, an implementation of the Apache specific AIS and to demonstrate how to deploy and consume FutureID identity services in this context. The second task integrated the FutureID Application Integration Service (AIS) component into the proof of concept Atos e-Learning platform pilot.

These two tasks are in turn related to T44.4 Implementation of Application Integration Service and the D44.4 document. Both deliverables were produced concurrently and will be available within one month from one another.

Aiming to implement the Apache specific AIS, the starting points are the requirements identified in D52.1 **[2]**, the reference architecture described in document **[3]**, and the technical specifications of Apache specific AIS implementation and the Atos e-Learning integration interface with the FutureID infrastructure established in D52.2 **[1]**.

The AIS plays two major roles in the FutureID architecture **[3]**:

- Intercepting unknown users and requesting the FutureID infrastructure to start the authentication process;
- Receiving and validating credentials in order to set up an authenticated session for users.

In more detail, in the former role, the User Agent (UA) requests a resource provided by the application A. The request is intercepted by Access Filter (AF). The AF uses the Apache Session Environment to determine whether a user is already known. If the user is already known, the request is passed to A. Otherwise, AF issues a FutureID Authentication Request (FAR) through the UA to the Solver and Executor (S&E) component.

In the latter role, the S&E presents a user or session credential to the Simple Credentials Transformer (SCT) capable of validating such credentials. The SCT verifies the credential and, in the case of success, sets a user session. Once an authenticated session is established, the SCT redirects the user to the originally requested resource of A.

6 FutureID Apache specific AIS overview

Section 9.2.2 and 9.2.3 of **[1]** describe FutureID Apache specific AIS architecture and detail its components. This component performs an integration of the Atos e-Learning services for enterprises with FutureID using Apache modules.

The FutureID Apache specific AIS component will be embedded into the Apache server and has its internal structure based on the Apache native modules.

Apache Access Handlers will be developed for the integration and will authenticate the user against the SAML2.0 IdP (role played by the Broker Service). This component will contain the conceptual modules of the AIS, the Simple Credential Transformer (SCT) that is a SAML assertion consumer and the Access Filter (AF), which is an access filter for resources.

6.1 Apache specific AIS Architecture overview

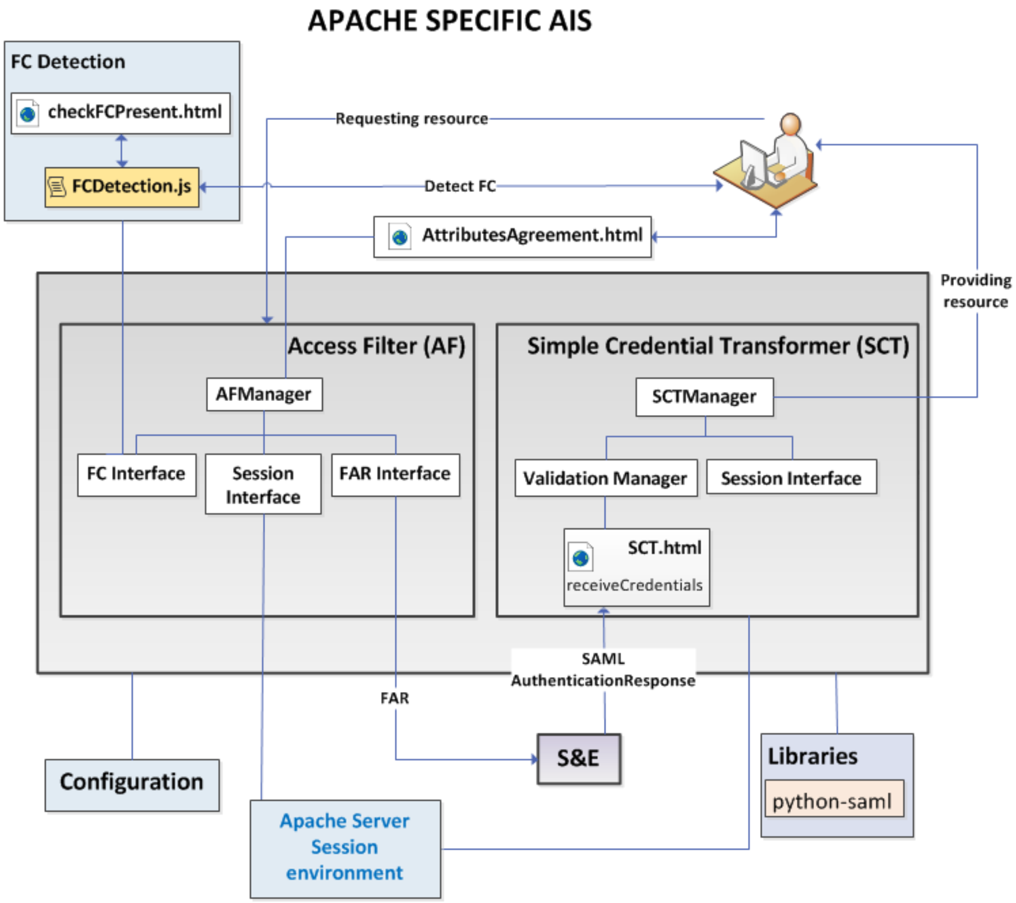

Figure 1: Apache specific AIS architecture

The main building components of the Apache specific FutureID AIS are the AF and the SCT. Both use the same session environment provided by the Apache server, as can be seen in **Figure 1**. In fact the AF sets variables as server variables and then the application gains access to them.

Based on the Apache handlers architecture, the integration of applications such as Atos e-Learning services for enterprises with the FutureID infrastructure can be performed in a straightforward fashion. The AIS component was developed in this manner. It was created to provide authentication using SAML assertions, and to manage the FutureID Authentication Request (FAR) protocol.

The AF and SCT components are implemented in Python language and SAML libraries for Python have been used to manage the SAML assertions.

The communication points and interfaces with the user's application and the S&E component are based on html pages that provide the necessary user interaction facilities and the reception mechanisms of the SAML messages from the FutureID Broker Service component.

6.2 Apache specific AIS flows

The high level overview of the flow of communications is depicted in the picture below.

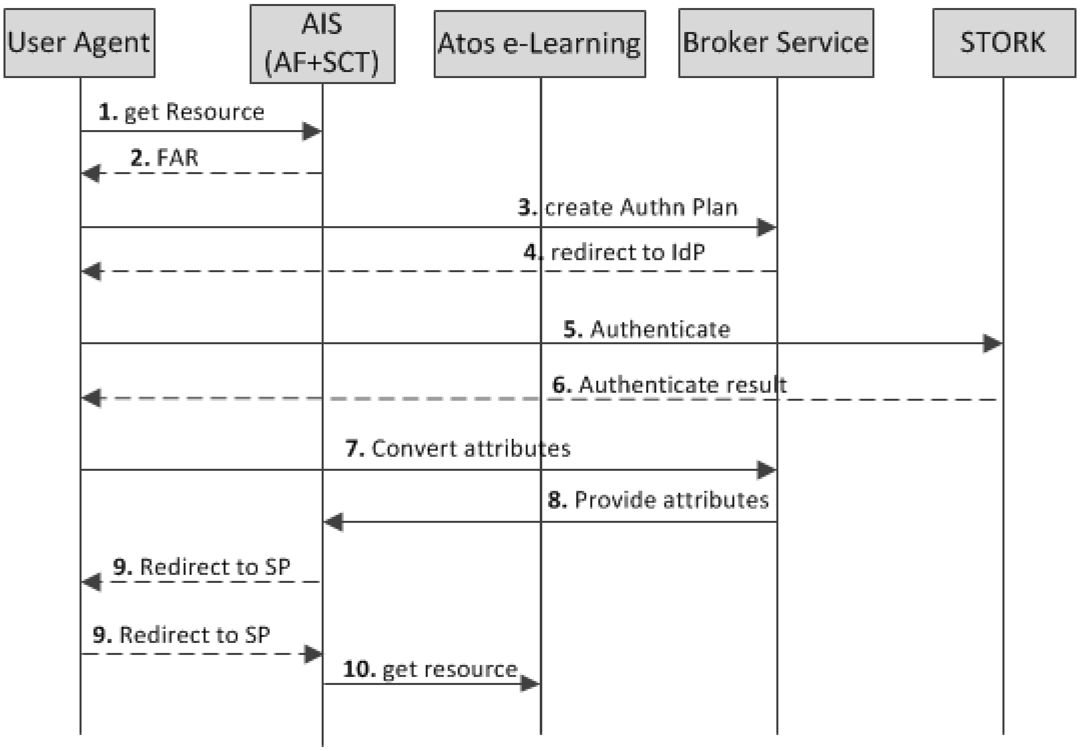

Figure 2: High level overview of the flow of communications.

The detailed description of steps from **Figure 2**:

1. The request to the Application is caught by the Access Filter (AF). If there is no security context for this User Agent or there is an old one, a new one has to be created.
2. The user is redirected to the Solver deployed within the FutureID client or to the remote solver in the case when there is no FutureID client. The Access Filter creates a FutureID Authenticate Request (FAR) that contains information about required user attributes, application service, authorized authentication methods, and preferred intermediary servers.

3-6 The Solver creates authentication plans and the user is redirected via the Broker to the appropriate Identity Provider (STORK in this case) to perform authentication and fetch required attributes.

7-8 After the authentication process is completed the Broker converts user attributes to appropriate data structures and send the attributes to the SCT...

9. The user is redirected to the Application.
10. The Access Filter catches the request and validates the assertions. After successful verification the access to the application is granted and values of the required attributes are made available to the application.

For more detail **Figure 3** depicts the flows of the process when the user requests a resource from the SP, in this case the Atos e-Learning platform.

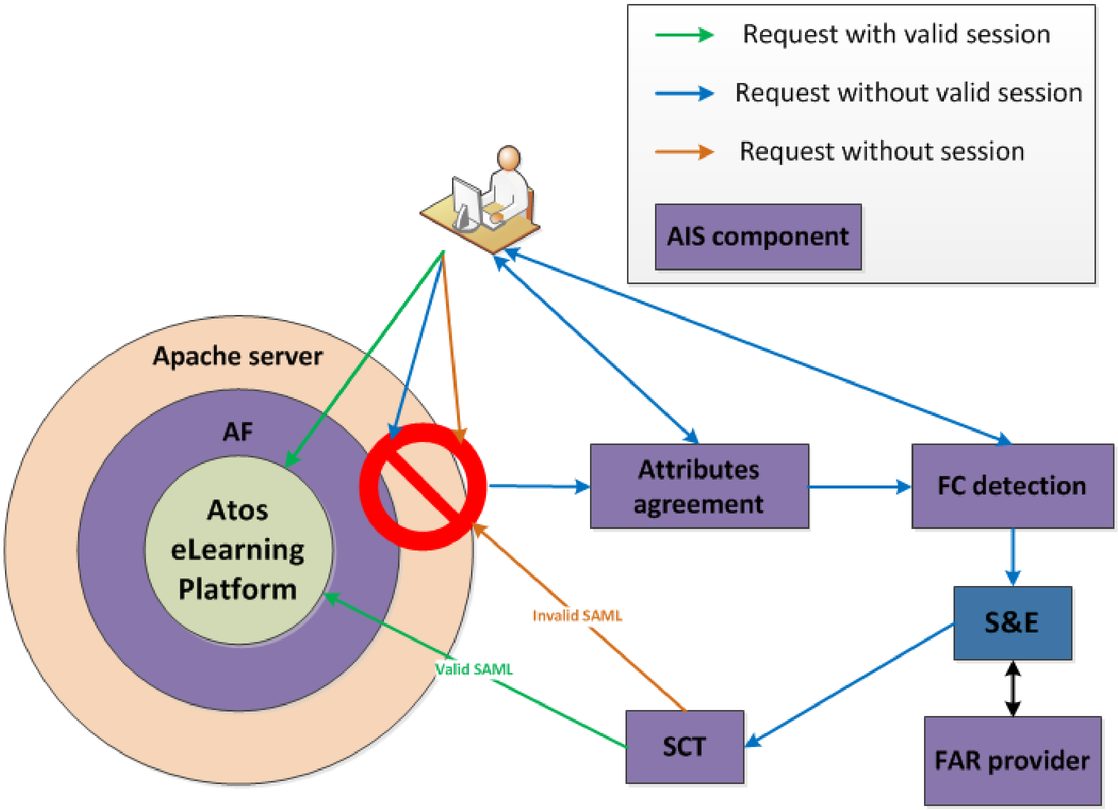

Figure 3: Detailed request flow diagram for Apache AIS.

The process is as follows:

- The AF embedded in the Apache server acts as a filter by checking the FutureID session provided by the user by means of the Apache server session environment;
- If the session is valid the AF provides the requested resource to the user;
- If the session is not provided or the session is not valid the AF asks the user’s approval to provide the attributes needed by the SP.
- Once the user accepts the requested attributes the AF checks if the FC is running on the user device and redirects the user to the trusted S&E.
- The S&E retrieves the FAR message from the AF. This FAR message provides the attributes required and the address to provide the SAML assertion.
- The S&E provides the SAML assertion;
- The SCT component validates the SAML Response;
- The SCT provides the requested resource to the user.

7 Access Filter (AF)

The AF component is a part of the access control layer for the requested content as shown above in the **Figure 3**.

7.1 AF components

AF has been developed as an Apache access handler using a Python implementation built on top of the mod_python module. This module provides access to the low level Apache internals via a series of Python scripts that are configured to work as access and content handlers.

7.1.1 SL Interface

As opposed to JBoss AIS there is neither session library nor Session Library interface. In this case session is just a part of the Apache session environment. The access handler has a direct access to the session environment.

7.1.2 FAR Interface

During the course of the access protocol, the S&E will eventually send a GET request to the server asking for the FAR. This event is managed at /FARLogin.xml or /FARRegister.xml, accessible for all requests. It will return the FAR as “application/xml” content type. This process is managed with a content handler.

The url to get the FAR is sent to the S&E as an argument of the redirect request.

The FAR handler basically dispatches the content of a file based on the url of the request.

There are two types of FAR configurations available for the incoming requests and kept in respective files, one for the login process, and the second one for the registration procedure.

7.1.3 FC Interface

This is a common element for both FutureID AIS’s. Both were developed in a unique way and their integration and deployment will affect both FutureID AIS’s (JBoss specific AIS and Apache specific AIS implementation). JavaScript code is used to check whether the FC is present or not. This check is performed based on the response from a specific URL and the value of a specific parameter. See appendix in D44.3 **[4]**.

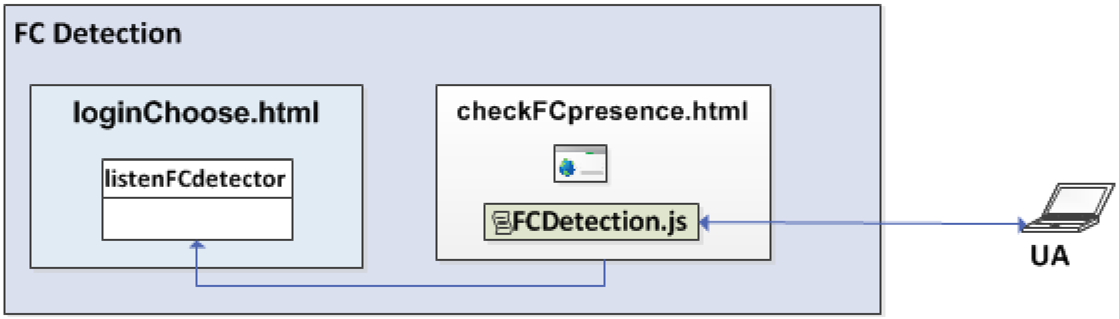

Figure 4: FC detection process

The FC detection will be managed with a handler listening on loginChoose.html. The handler "listenFCdetector" processes the POST request generated and sent by the JavaScript code. Based on the response from the POST request sent by the detectFC it will start the authentication loop with external S&E or give the user a possibility to choose between local (FC) or remote S&E.

As FC detection is based on JavaScript, some interaction with the User Agent (UA) is needed. It means that a page with the special JavaScript code will be sent to the UA.

7.1.3.1 FCDetection.js

The FCDetection.js JavaScript file is a common component for both JBoss specific AIS and Apache specific AIS implementation.

7.2 AF Management

The AF is basically an access handler that checks for every request whether it has a valid session attribute or not. The default behaviour immediately creates a new session for incoming requests (without valid session attribute) so from this point on, the request is known to the server. Later on, once the SCT component has validated the SAML response, the session is activated.

In the case that the session does not have a valid login or login attributes at all, the request is sent to the authentication loop. The first step in this loop is to inform the user about the attributes that are going to be retrieved from the IdP. After this, the FC detection script is executed on the UA. The result of the detection process determines the consecutive authentication path to follow.

The result of the FC detection routine is sent back as a POST request. If the FC is not present the user is sent automatically to the S&E chosen, and if the FC is present, the user has to choose either to identify himself using local S&E running on the FC, or using the remote one.

The S&E will eventually send a GET request to the server asking for the FAR message. This event is managed by FARLogin.xml or FARRegister.xml. Those resources are accessible for all requests. The FAR will be returned as an "application/xml" content type. This process is managed by a content handler.

7.3 AF flow process

The flow process, when the FC is not present, proceeds in more details as follows:

- The UA requests a resource. The AF intercepts the request;
- The AF uses the Session attributes to get the session data from the user in order to determine whether the user is known;
- The Session environment doesn't provide the session data for this user;
- The AF treats the user as an unknown user;
- The AF checks whether the UA has access to the FC, running the detection code;
- The detection code runs on the UA trying to connect to the FC, after the time out[1], the AF assumes that the FC is absent;
- The AF asks the user to choose a remote S&E;
- The user selects a specific remote S&E;
  The AF makes a "self-posting form" sending the FAR to the selected SE through the UA.

7.4 Configuration

The appropriate AF configuration is included in the constants.py file as explained in Section 9.

[1] As FC is an application running on user's device the access to a local port is almost immediate, avoiding network issues.

8 Simple Credential Transformer (SCT)

Once the AIS receives an assertion from the BS, the required attributes needed and the user identifier are set in a server variable as session data. This information will be later used by the Atos e-Learning interface before granting access to the resource requested by the user, see step 6 in **Figure 7** and Section 11.1 for additional information.

8.1 Components and flows inside the SCT

The SCT component consists of a handler listening for a SAML POST request.

Figure 5 depicts the internal structure of the SCT components.

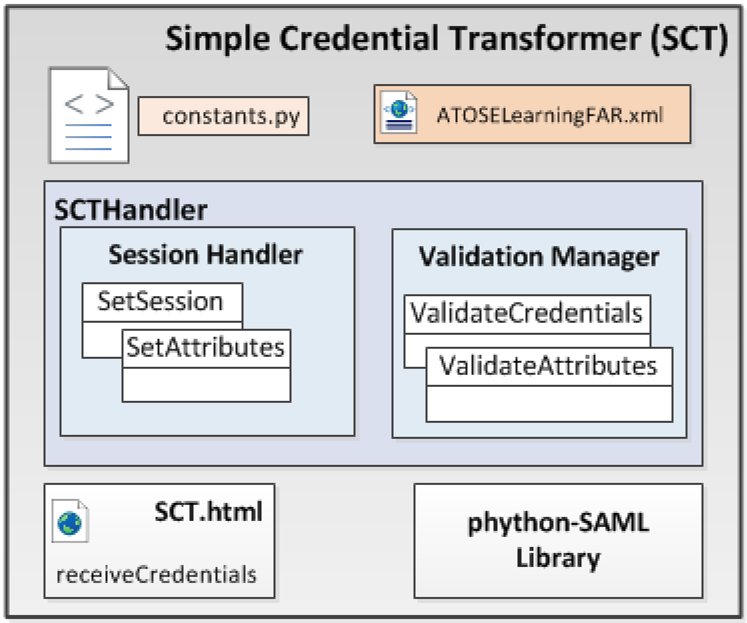

Figure 5: SCT components

The main blocks the SCT component is made of are:

- SCTHandler: this handler is in charge of validating the SAML message and the attributes received. Also, it activates the already created session and sets the attributes on the server session environment;
- constants.py and AtosELearningFAR.xml: these configuration files are used by the SCT handler for validation purposes;
- SCT.html: receives the SAML Response from the S&E;
- Python.SAML library: this library is used for the SCTHandler to validate the SAML message.

The sequence diagram in **Figure 6** shows the process of reception and validation of credentials in order to set up authenticated sessions for users.

The S&E presents a user's session credentials to the suitable SCT. The SCT verifies the credential and, on success, sets a user session by calling the Apache Session Environment. Once an authenticated session is established, the SCT redirects the user to the originally requested resource of A.

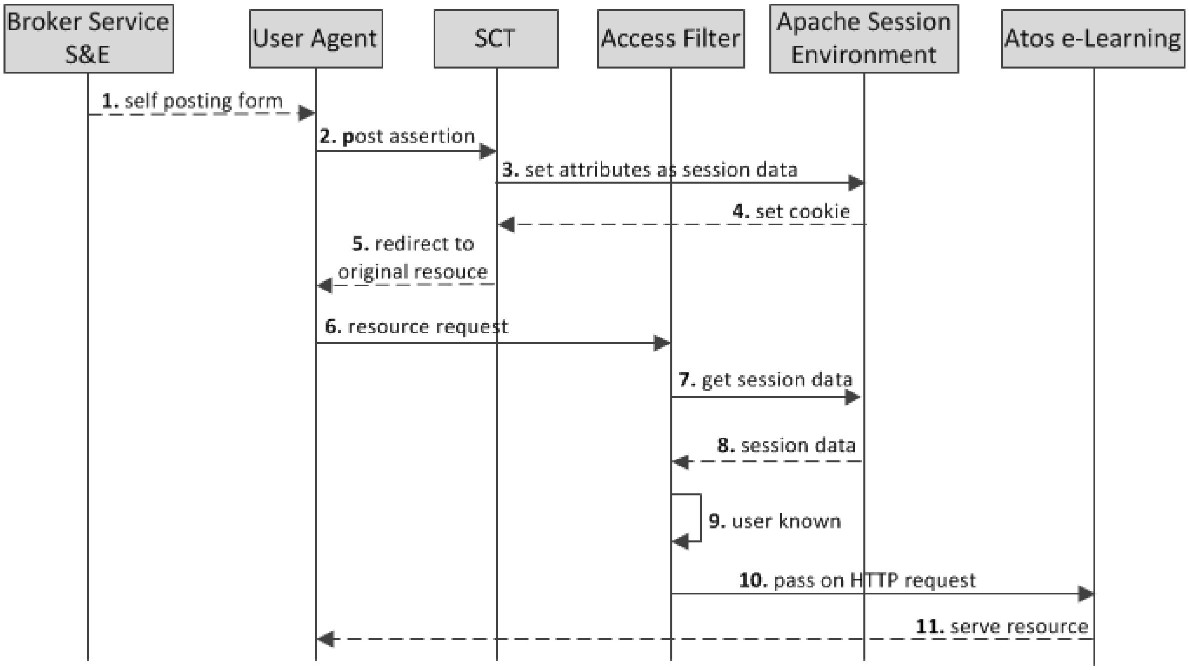

Figure 6: Credential validation

Figure 6 shows following steps:

1-2. The S&E posts a form to SCT through the UA, with the appropriate assertion;

3. The SCT sets the assertion attributes as session data for the user in order to determine whether he or she is known or not;
4. The server sets a cookie to be included also on the user's side;
5. The SCT redirects the user to the original resource;
6. The AF intercepts the UA request for the specific resource;
7. The AF uses the user's session data in order to determine whether the user is known;

8. The Apache session environment provides the session data for the user making a request;
9. The AF treated the user as a known user;
10. The AF grants the user access to the requested resource;
11. The application serves the resource to the UA.

The diagram in **Figure 7** shows the components and the steps inside SCT.

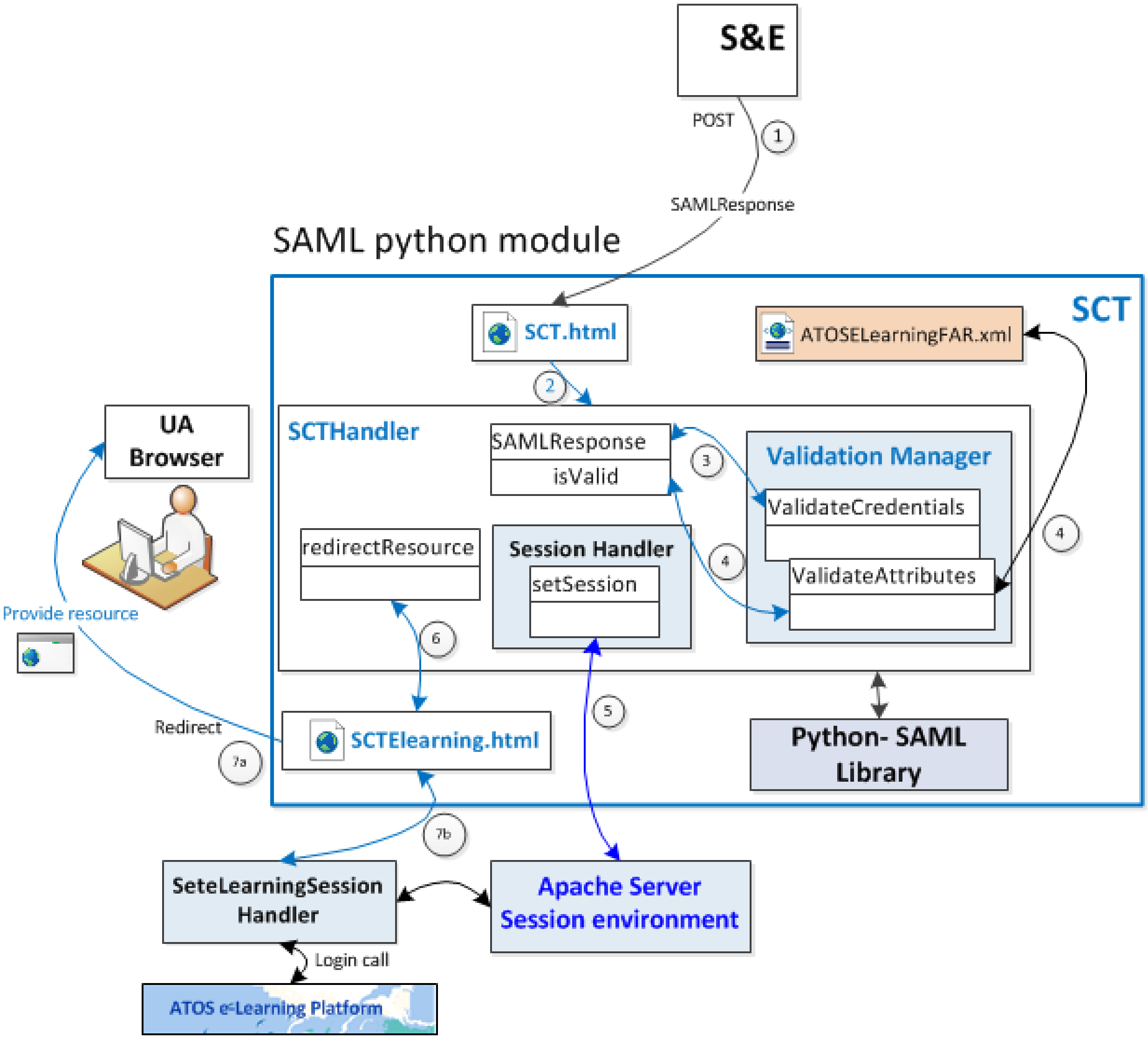

Figure 7: Flows into the SCT component

The following steps describe and provide a short overview of the authentication process. Detailed events inside each component are also summarized below:

- (Step1) The S&E sends a SAML Response (base 64 encoded and signed);
- The SCT handler receives the HTTP POST binding signed request;
  - The SAML Response is extracted and decoded;
  - The URL where the SCT service is placed must be provided to the S&E developer in order for the S&E to be able to make the proper call.
- The SCT will make a call to the python-SAML library. This library is in charge of :
  - The validation of a provided SAML Response by checking credentials (step 3) and attributes attached to the response (step 4).
    - It will validate the signature included in the SAML response against the public keys of the supported Broker Service. S&E only redirects the response without any update;
    - The attribute names will be validated through the AtosELearningFAR.xml. Attributes are pre-configured in AIS and included in the FAR (as required by Atos e-Learning). They are also included in the SAML response.
  - Once the SAMLResponse has been validated, the session is set in the Apache Server Session Environment (step 5);
  - Next the SCT Manager redirects the user to the RelayState stored in the session environment (Step 6 and Step 7). In the particular case of Atos e-Learning platform, this is the SCTElearning.html, where the setElearningSession handler will retrieve the attributes from the session and make the login call into the Atos e-Learning environment.

8.2 SCT Validation

The SCT consists of a handler listening for a specific POST request. The content of the POST form is the base64 encoded SAML response.

The first step is to validate the response, which is done using the “OneLogin library” against the configuration data from settings.json configuration file, which contains the public key names and certificates for the IdP and SP.

Once the response is validated (if it is not valid, an error message will be shown to the user), the SCT sets all the attributes sent within the SAML response into the session environment. In the beginning of the process a check is made on the “accesstype” attribute, in order to determine if the user is trying to login or register.

If the user is trying to log in, the user is redirected to the url address kept in the “relaystate” session variable. If the “entryPoint” variable is set to something different than *“none*”, the user is redirected to the url address set by the “entryPoint” (for example a welcome page). In the Atos e-

Learning platform, the user is sent to a special page that does an automated login process using the attributes set in the session.

If the "accesstype" parameter is set for registration process, the user is redirected to the url stored in the file as "*registrationsURL*" constant. Then the system administrator will be in charge of creation of a handler to manage this process as required by the platform.

9 AIS Configuration

The AIS configuration consists of several configuration files located on three different levels:

- Apache server configuration;
- AIS component configuration;
- Atos e-Learning configuration.

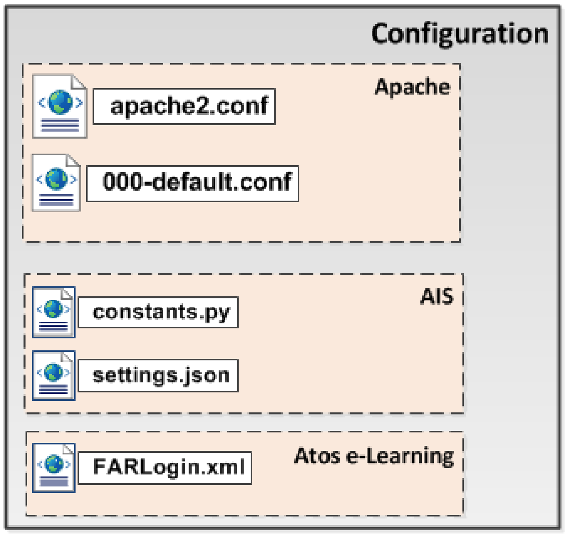

Figure 8: Configuration files for Apache specific AIS.

9.1 Apache server configuration

Apache configuration is based on two configuration files (**apache2.conf**, **000-default.conf**).

The apache2.conf file[2] is used by the Apache Server. It stores information on various functions of the server, which can be edited by removing or adding a number sign "#" at the beginning of the line, thus setting values for each directive.

The configuration related with the server content is managed in **000-default.conf**[3]**.**

This file contains the configuration of the server to operate on port 443 as a https with SSL module, linking to the certificates and keys with the SSL directives:

[2] **etc/apache2/apache2.conf for linux environment**.

[3] **etc/apache2/sites-enabled/000-default.conf for linux environment**.

- *SSLEngine on*
- *SSLCertificateFile pathTo_crt*
- *SSLCertificateKeyFile pathTo_ key*

This configuration has to be set also in */etc/apache2/mods-enabled/ssl.conf* (last two directives only).

To set the path for mod_python to look for the code to use for the handling of the requests, the following directive is set:

- *PythonPath "sys.path+['pathTo_ AIS']"*

The `Alias` directive allows documents to be stored in the local filesystem other than under the `DocumentRoot`. In this case it is used to map the E-Learning environment.

- *Alias /elearning /opt/elearning*

The `SSLVerifyClient` sets the Certificate verification level for the Client Authentication. The `none` level makes that no client Certificate is required at all.

The following sector of the configuration file refers to the per directory and file settings. When the `Directory` directive is used, the settings apply to all the content at any level of depth. When particular settings are needed for a certain file, the `Files` directive applies those settings to the requested file.

Using this configuration feature, the FAR is set as the access handler for the whole protected content, and the necessary files representing functional URL´s are left outside. Among such URLs are SCT.html, checkFCpresent.html or FAR.xml. All of these URL's defined as an exception must be visible for the not logged in users and therefore use their own access and content handlers.

The `Header set Access-Control-Allow-Origin "*"` is used to set by default CORS headers to any content provided within the */var/www/html/* folder. This is required by the S&E for the retrieval of the FAR.

9.2 AIS configuration

Generic AIS configuration is done in **constants.py**, where paths to both FAR files, URL´s, timeout for sessions, secret keys for cookies and SAML configuration file path are set. Besides those the **settings.json** file is used for SAML validation.

9.2.1 SAML Verification configuration

For the purpose of the authenticity verification of the SAML response, a JSON configuration file named **settings.json** is used by the OneLogin library. This file contains the necessary information for the SAML correlation as well as the IdP's (the component acting as IdP in this case is the BS) certificate.

9.3 Atos e-Learning configuration

The Atos e-Learning platform acts as a SP and hence needs some attributes to authenticate the user. The SP trust in the AIS component to provide these needed attributes. The list of required attributes sent by the user through the AIS consists of both the mandatory attributes and the optional attributes selected by the user (see step 3 of Section 6.1of **[2]**). With this aim the SP will provide this list and the AIS will create a file named FARLogin.xml. This file will be provided to the S&E to get the attribute values from the IdP, STORK in this case.

Attribute	Mandatory	Optional
AcademicTitle		X
FirstName	X	
LastName	X	
Street		X
StreetNumber		X
City		X
State		X
Country		X
ZipCode		X
Nationality		X
DateOfBirth		X
IDType		X
IDIssuer		X
IssuingState		X
IDValidUntil		X
eIdentifier	X	
Age		X
AgeVerification		X

Table 1: Attributes of authentication request and response (Broker service)

The AIS will send a list of attributes indicated in the Table 1 to the BS within the FutureID Authentication Request (FAR).

Then this FAR is sent to the BS through UA.

The BS will connect with STORK and the information contained in the attributes received on the response from STORK is mapped to the BS response. Finally the AIS will receive an assertion from the BS. The flow can be seen in Figure 9.

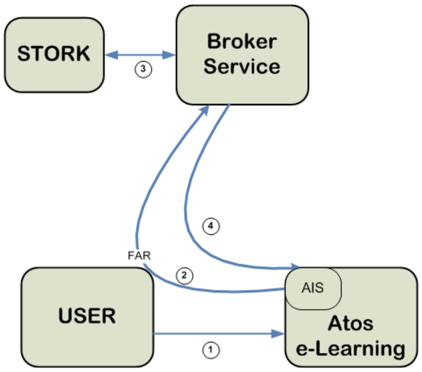

Figure 9: Flow of attributes for authentication process.

9.3.1 FAR

Figure 9 **[4]** depicts how the FAR message is created by the AF component and reaches the BS. After the connection with the IdP the provided final response is not a response as is, but an IdP-initiated SAML response. This case is valid when the FC is absent, then the S&E element is not present on the user side but is deployed within the FutureID infrastructure.

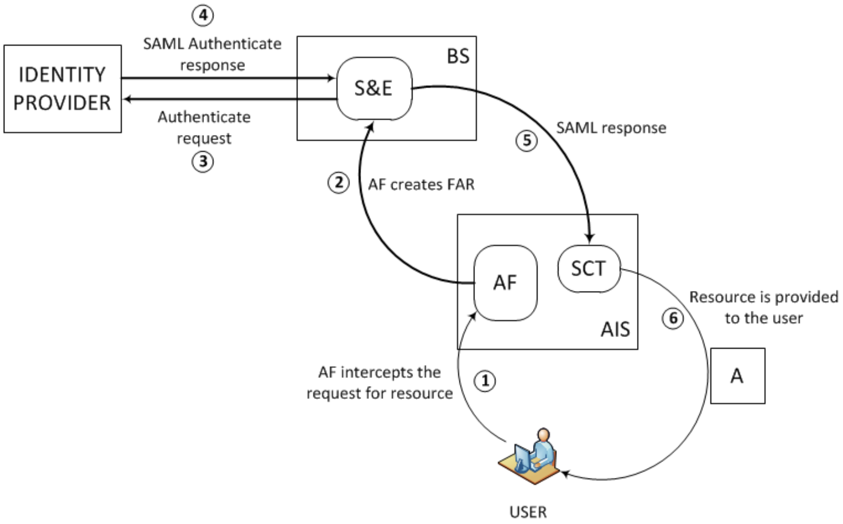
④
SAML Authenticate response
IDENTITY PROVIDER
BS
S&E
Authenticate request
③
⑤
SAML response
②
AF creates FAR
AF
SCT
AIS
Resource is provided to the user
⑥
A
AF intercepts the request for resource
①
USER

Figure 10: FAR message for FC absent

As can be seen in step 2 on the Figure 10 the FAR creation is an independent process. The AF provides the FAR to the S&E component. Then the authentication process is carried out by the Identity Provider (steps 3 and 4). Finally another component of the Broker Service generates the IdP-SAML-Response that reaches the SCT component (step 5). For this reason it is not necessary to include a session identifier in the dynamic part of the FAR.

10 Development environment and deployment

This section will provide some details about the Atos e-Learning platform installation and Apache AIS deployment details.

10.1 AIS Installation

The Apache AIS component has been deployed on Linux as OS, namely on Ubuntu 14.04.3 version. Before deployment the following prerequisites must be fulfilled:

- Installation of Apache 2.4;
- Installation of mod-python in Apache;
- Installation of Python modules "Requests" (for requesting information) and "OneLogin" libraries for SAML;
- Enabling the following Apache Modules: python, SSL and headers.

In order to install the Apache AIS component an installation package containing the following archives will be created:

- AIS_html.zip containing html files;
- AIS_py containing mainly the python files;
- 000-default.conf configuration apache file that will be used as configuration model;

The steps to follow in the AIS installation are:

1. Deploy the content/application to protect, in this case Atos e-Learning (place it in /opt/[name of application])
2. Test the access and running of the deployed application
3. Copy the contents of AIS_html from the installation package in /var/www/html/
4. Copy the contents of AIS_py from the installation package in $HOME/AIS
5. Configuration:
    a. File etc/apache2/sites-enabled/000-default.conf. This file is provided with the installation as example and it should be adapted to the particular application and environment;
    b. Add the following directives to file /etc/apache2/mods-enabled/ssl.conf:

       SSLCertificateFile /home/futureid/AIS/ca.crt

       SSLCertificateKeyFile /home/futureid/AIS/ca.key

6. It is necessary to install the FutureID Team certificate to exchange messages with the S&E **[5]**

11 Integration and Identification of Services

This chapter consists of three sections. Section 11.1 describes the technical integration of the Atos e-Learning platform into the FutureID infrastructure. Section 11.2 lists three typical services working as the combination of the e-Learning platform including the used server, as available from Atos and the relevant FutureID infrastructure and components as developed and implemented along the FutureID project and described in D24.1, D44.3, D44.4, D52.1 and D52.2. In Section 11.3 the focus is set on other possible related services, which can be used in this configuration of e-Learning bridged with the FutureID components.

Two remarks:
The number of e-Learning programs for individual employees in medium and large size companies in Europe increases from year to year and replaces more and more the traditional group seminar learning procedures.
Medium and large companies have increasingly “standard” learning topics, which must be passed by all employees in a yearly mode, such as safety training, compliance knowledge and antitrust training.

These two facts illustrate the increased importance of this pilot case in the enterprise domain in Europe.

11.1 Integration of Atos e-Learning platform into FutureID infrastructure

Apache supports many scripting languages such as PHP, Perl, Tcl, and Python as well as J2EE platform. Atos e-Learning application runs on Apache server and the integration of php applications such as Atos e-Learning platform with FutureID can be performed easily by developing a dedicated Session handler (see **Figure 7** and Section 8.1) and modifying the configuration files described in Chapter 9.

The setElearningSession handler is the bridging component between AIS and Atos e-Learning platform. Its main responsibility is to provide the attributes, acquired from the SCT component, to the Atos e-Learning application and make the login call in the Atos e-Learning environment. The SCT component redirects the user to the requested resource through the SCTElearning.html where the setElearningSession handler is acting.

Figure 11 depicts the connection between the components belonging to AIS and the Atos e-Learning application.

The AF and SCT components included in the AIS Python module will use a common Server Session Environment that will cover the session management (**[3]** and **[4]**).

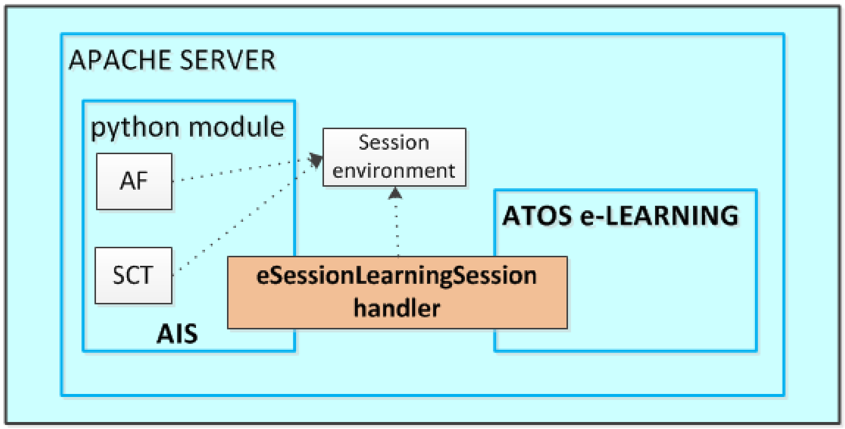

Figure 11: Connection between the Apache AIS implementation components and Atos e-Learning platform

11.1.1 Authentication process

Atos e-Learning will use the FutureID platform for authentication purposes and STORK as external Identity Provider.

11.2 Identification of FutureID services relevant to Atos e-Learning platform

The e-Learning platform was developed to support services for three different kinds of business scenarios

- Traditional e-Learning for the Employee
- e-Learning based on Enterprise Knowledge
- e-Learning based on Enterprise Content

The ***traditional e-Learning for the Employee*** is needed for different purposes, such as distributing the information on new company products to the internal worldwide sales and application engineer team in the company or to the contracted distributors.. Another aspect would be the regular advanced training for the employees. This could be applied to technical aspects, such as new technologies and/or new standards. Furthermore, it can be applied to management aspects, such as project management or innovation management, and standard topics, such as compliance training and refreshment, high performance training and/or safety or security training and many others.

The ***e-Learning based on Enterprise Knowledge*** leverages not only the content used by the content management systems, but also the knowledge that has been captured by using a knowledge management system and stored on the server or inside the cloud. This knowledge can consist of email messages, sales call notes, memos, audio messages, meeting notes, customer support problem reports, personal and strategy changes at customer companies, information on the competitors and many others.

The ***e-Learning based on Enterprise Content*** could be used for example as a content management system to create, store and upgrade the content about new products from engineers and marketing people. Other examples could be a new quality management concept or a new logistic management approach. The number of possible use cases is very high.

Besides this standardized learning programs some new services and business models can be utilized, which use the type of configuration applied by the e-learning platform from Atos in combination with the FutureID infrastructure and selected components. The description of such possible new services and/or business is part of the next section.

11.3 Related FutureID services that could benefit from the integration

Based on the circumstance that the e-learning platform from Atos and the FutureID infrastructure from the FutureID consortium is available in the last year of the project run time, it makes sense to observe and describe a range of other possible services, which can use these two combined infrastructures. Three examples are shown in this section, which mirror cross company access procedures along with this easy use approach.

Example 1: Information exchange in the R&D sector of an automotive OEM with the supplier

In some market domains, like the automotive industry, an ad-hoc joint development team between the supplier and the OEM (the vehicle manufacturer) is temporarily indicated. This development team exchange knowledge on new requirements of vehicule parts for the next generation of vehicles, as defined in the OEM and the development roadmap from the supplier. The outcome would be a new set of specifications and a new CAD drawing. The ad-hoc joint can be done and will be finished if the R&D-part is finished and the result is moved to the commercial departments, such as the purchase department from the OEM and the sales department from the supplier. Background information for this example: 80% of the value in a modern vehicle is created by the supplier.

The main purpose of this shared work regarding the e-learning platform and the FutureID infrastructure is not the learning process itself, but only using the central data management pool and the specific identification and a dedicated access control system in the digital world and access to the sensitive data sets. In this model it is not needed that the ad-hoc joint development team members see or meet each other. Supplier and OEM define components in design, functionality, material and so on. This joint work could create IP, which is extremely sensitive with respect to the competitor. It is of interest for the competitor to get such information as soon as possible.

Remark: Since many years the automotive industry in Europe uses an internal internet network which is based, defined, realized and operated by ENX, placed in Paris, France. ENX stands for European Network Exchange organization. It should be easy to use this framework for dedicated users. The users in this example must register at the e-learning platform and get the access right to the shared data space.

Example 2: Shared industry data set for production machines

The e-learning platform can be used as warehouse for dedicated processes. One example: two companies use punching machines from the same machine vendor. One of the two companies is located in Italy and uses this equipment for punching of alumina foil in the thickness of 120, 150 and 200 micron. The other company is located in Spain and use this equipment for punching a copper foil in the thickness of 50, 100 and 150 micron. All the relevant punching parameter records are stored in the e-learning tool and the related server from Atos.

The company with the alumina punching knowledge in Italy gets an order request from a customer to change the process and the material from alumina to copper and from 200 microns to 150 microns. The management staff from this company has the possibility to buy the required process parameter data sets, like speed, force, tooling and maintenance cycle from the Atos e-learning platform based on the personal access via the FutureID infrastructure. Such user will get an access only to the data room, for which it was requested and paid. In this data room also some practical knowledge on specific maintenance procedures would be available.

The other option could be that the company management decides to start an internal R&D project. This decision needs budget, possibly can stop the current production and introduce technical risks.

This model of shared industrial data set is expected for the Industry 4.0 revolution (i.e. the fourth revolution in the production), which was started in 2012 and offers some new business models and new market places in the near future.

Example 3: Multiple mobile robots should be controlled from a central control station

In modern production environments increasingly robots are involved in the automated manufacturing process. Most of the robots work statically, but more and more of them would be working in the mobile mode. In the case that a large number of robots are in the working process in the full automated manufacturing line, it is necessary to deploy a central control station and to monitor the robots. Such central control station can be linked with the e-learning platform from Atos to share information about the robot's workflow and the work of the super advisory control operator. This approach is also required when new employees need training sessions during the work of such central control station. The secure access of the employee can be defined and organized with the FutureID infrastructure.

On the other hand, the e-learning tool can be used to analyse different scenarios of central control stations and the possibility of operator's interventions, and can be used for searching the best practices.

In this case three elements can be addressed with the e-learning platform combined with the FutureID infrastructure

- Training of operators and new employees
- Analysing of a broad range of interventions of operators along with the impact on the work flow process as well as on the productivity.
- Deploy the best practices and share knowledge with another manufacturing facilities

Remark: In the automotive industry more and more OEMs have more robots in the factory than employees. Having such approach the OEMs can produce vehicles in Europe in a competitiveness model.

12 Conclusions

The focus of the Atos e-Learning platform is on offering enterprise customers a reliable e-Learning enterprise solution, which will lower the training costs for the organizations and provide a faster delivery and a more effective learning for e-Learning users (organization employees and outside organization clients).

FutureID Apache AIS implementation has been developed in order to integrate Future-ID with Atos e-Learning Services for Enterprises. This provides a proof-of-concept that demonstrates the viability of FutureID components in business scenarios, specifically with regards to Internet of Services. Moreover, the implementation has been in such a way that it allows you to easily integrating other kind of platforms, as T44.6 will demonstrate.