ETSI GS INS 001 V1.1.1 (2011-03)

Group Specification

Identity and access management for Networks and Services; IdM Interoperability between Operators or ISPs with Enterprise

Disclaimer

This document has been produced and approved by the Identity and Access Management for Networks and Services (ETSI INS) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
DGS/INS-001

Keywords
access, ID, interoperability, management,
network, service, use case

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status.
Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2011.
All rights reserved.

DECT™, **PLUGTESTS**™, **UMTS**™, **TIPHON**™, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.
3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.
LTE™ is a Trade Mark of ETSI currently being registered
for the benefit of its Members and of the 3GPP Organizational Partners.
GSM® and the GSM logo are Trade Marks registered and owned by the GSM Association.

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification (ISG) Identity and access management for Networks and Services (INS).

Introduction

In the present document we present an architecture and its instantiation for use cases where interoperability exists between Operators and Enterprises in terms of authentication and attribute exchange. Historically both domains were seen as separated, without any kind of interactions. The demand for new scenarios, i.e. Software as a Service, implies that some interactions need to be in place. This cooperation can be achieved either by exchanging data about the user or reusing the authentication context.

The first part of the present document provides a brief overview of the actual authentication and attributes exchange within the Operator and the Enterprise. Next, a set of use cases which demand for cooperation between Enterprise and Operators are presented. These use cases are the ground to collect the requirements and the impact of such requirements in the actual architectures.

The second part of the present document presents the architecture in terms of functions and its relationships, which answers the collected requirements. Moreover it describes the interfaces and the protocols such interfaces can use. Finally two examples of its instantiation are presented.

1 Scope

The present document presents a set of use-cases where the interoperability between Operators and Enterprise allows authentication reuse and attributes exchange. It identifies and describes the requirements imposed by such scenarios, derives an architecture and describes a set of interfaces or operations between architectural elements.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

Not applicable.

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] Cantor, S., Kemp, J., Philpott, R., and Maler, E.: "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0". March 2005.

NOTE: Available at http://docs.oasis-open.org/security/saml/v2.0/.

[i.2] OpenID Foundation.

NOTE: Available at http://openid.net/.

[i.3] S. Cantor, J. Kemp, and D. Champagne (editors): "Liberty ID-FF bindings and profiles specification --- 1.2-errata-v2.0, 2004. Liberty Alliance Project".

[i.4] S. Cantor, J. Kemp, R. Philpott and E. Maler (editors): "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS - December 2009.

[i.5] S. Cantor, J. Kemp, R. Philpott, E. Maler and P. Mishra (editors): "Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS - March 2005.

[i.6] S. Cantor, J. Kemp, R. Philpott, E. Maler and F. Hirsch (editors): "Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS - March 2005.

[i.7] J. Hughes, S. Cantor, J. Hodges, P. Mishra, R. Philpott, E. Maler and F. Hirsch (editors): "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS - December 2009.

[i.8] S. Cantor, J. Moreh, R. Phipott and E. Maler (editors): "Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS - December 2009.

[i.9] F. Hirsch, R. Philpott and E. Maler (editors): "Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS - March 2005.

[i.10] P. Mishra, R. Philpott and E. Maler (editors): "Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS - March 2005.

[i.11] J. Hodges, R. Philpott and E. Maler (editors): "Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS - March 2005.

3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AAA	Authentication Authorization and Accounting
AAAS	Authentication, Authorization and Accounting Server
AKA	Authentication and Key Agreement
CPE	Customer Premises Equipment
DB	DataBase
DHP	Data Handling Policy
GSM	Global System for Mobile Communications
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
ID	IDentity
IdM	Identity Management
IdP	Identity Provider
IMSI	International Mobile Subscriber Identity
ISP	Internet Service Provider
LDAP	Lightweight Directory Access Protocol
ME	Mobile Equipment
MO	Mobile Operator
NAS	Network Access Server
PKI	Public Key Infrastructure
PW	Password
QoS	Quality of Service
SaaS	Software as a Service
SAML	Security Assertion Markup Language
SIM	Subscriber Identity Module
SP	Service Provider
SSO	Single Sign-On
U(SIM)	Universal Subscriber Identity Module
UE	User Equipment
USB	Universal Serial Bus
VoD	Video on Demand
VPN	Virtual Private Network
w.r.t.	with respect to
WS	Web Service
XaaS	X as a Service

4 IdM Overview: authentication and attribute exchange

4.1 Operators/ISPs

This clause briefly describes the authentication and attributes exchange mechanisms and architecture for an operator.

4.1.1 Authentication

Authentication defines the process where one entity (commonly named server) verifies another entity's claim to holding a specific digital identity (commonly named client). Authentication is accomplished using two different processes:

i) Implicit authentication (device authentication) - relies only on an implicit authentication through physical or logical identity on the layer 2 transport layer.

ii) Explicit authentication (user authentication) - relies on an explicit signaling between the client (UE) and the authentication server. The UE sends its corresponding credentials to the server where they must be validated. Different types of credentials could be used. Examples are passwords, digital certificates, or one-time tokens.

The network registration involves the authentication and authorization procedures between a client (UE) and the server which controls the access to the access network based on the credentials the UE sent and the policies.

In order to provide interoperability between various network equipments, protocols for various segments of the authentication model have been standardized.

Figure 1 provides an abstract architecture for the typically network authentication mechanisms.

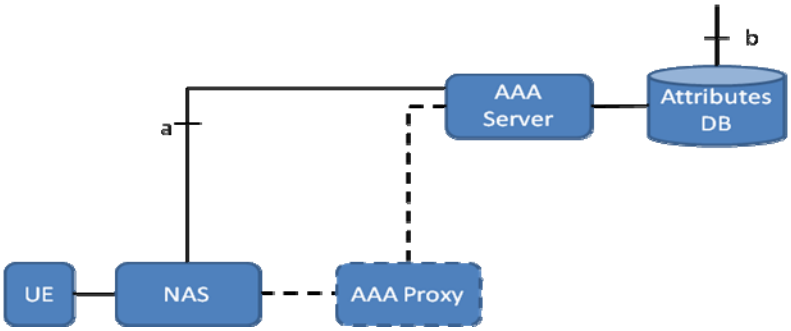

Figure 1: Network authentication abstract architecture

The architecture is composed by different entities, presented next:

1) User Equipment (UE) - represents the entity which needs to be authenticated, during the network registration. The result of a successful authentication procedure results in network access.

2) Network Access Server (NAS) - This entity translates the network access requests issued by the UE and forwards it to the AAA (Authentication, Authorization and Accounting) Server.

3) AAA Server - represents the server which performs the user authentication and authorization for network access. This entity retrieves authentication data from a database that could, or not, be collocated with the AAA server. After a successfully authentication the AAA server produces a set of authorization rules that are enforced in the NAS entity. Those rules are typically, operator's defined and stored in the Attributes Database (DB).

4) AAA Proxy - the AAA Proxy is a typical AAA server which acts as a proxy for the user's requests. Upon the reception of user's requests it forwards them to the AAA server in charge of the authentication procedure. The AAA server location is one of the AAA proxy functionalities. Responses received back from the AAA server will be returned to the NAS.

5) Attributes DB - this entity contains the user authentication data (user identity, key materials, etc.). It could also contain information related with operator's authorization and accounting rules, as well as user specific configuration data. Dynamic information related with the user is also maintained in this DB. An example is the user's location in moment he authenticates and accesses the network.

4.1.2 Attribute Exchange

One of the components in an operator infrastructure is the Attribute Database (see figure 1). This entity stores static information provisioned by the operator. An example of such information is the user's profile. On the other hand, more dynamic information could also be stored, e.g. user's location. Both types of information can be exchanged with other network entities through interfaces "a" and "b".

Interface "a" is used during the network registration process for different proposes:

i) transport the necessary information for the authentication procedure;

ii) transport the user's profile (or part of it) to the NAS. The goal of such action is to provide NAS with the necessary information to correctly configure the filters (e.g. QoS, or firewall) that will support the entrance of the new user. That information is stored in the Attributes DB;

iii) Provide user's dynamic information to the AAA server (in fact that information will populate the Attributes DB, that could or not be co-located with the AAA server). Example of such information is user's context information.

Another interface, "b", exists that provides a way for Application to retrieve information, related with the user that it is stored in the Attribute DB. The use of this interface is not limited to any of the two information types stored within the Attributes DB. It is possible to retrieve user's profile (static) information or user's generated information (e.g. context information). The attributes collected from the Attributes DB will feed operator's internal or external applications.

The exchange of the information between the Attributes DB and the appropriate counterpart are based on established standard protocols.

4.2 Enterprise (and Home Network)

This clause describes the common technology used for authentication and attribute exchange in the Internet Services domain, the Enterprise and home networks belong to. From a network layer point of view, the Internet Services reside in the application layer, this is a higher layer than the basic network authentication mentioned in the previous clause.

4.2.1 Authentication

Authentication information in the internet services of the user domain can be transferred by different protocols (e.g. SAML [i.1], OpenID [i.2], ID-FF [i.3]). The most common authentication protocol in the Enterprise is SAML.

The fundamental SAML architecture consists of:

- a User (subject in SAML terms), who wishes access to a certain resource or service;
- a Service Provider, who provides the desired resource or service; and
- an Identity Provider, providing application layer authentication of the User (e.g. based on additional authentication providers in other layers) and issues assertions about Users to Service Providers. It plays a key role in federating identities.

The SAML architecture can easily be mapped to the Operator/ISP - Enterprise interworking setting. The result of this mapping is shown in figure 2, in which interface 'd' is for example instantiated with the SAML protocol. A conceptual difference is that in SAML a subject (user) can have several Identifiers of different type (NameID-Type) that will depend on the authentication (service) context.

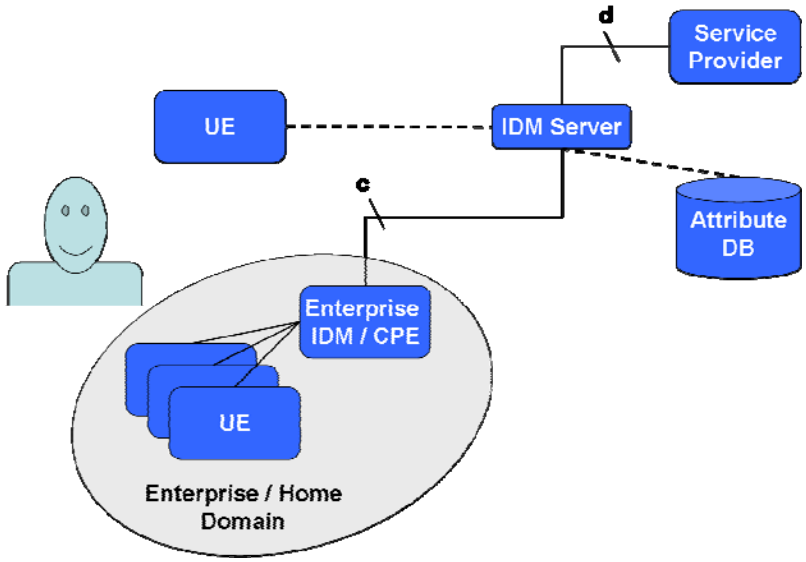

Figure 2: Service Provider abstract architecture

The current SAML version 2.0 consists of the following specifications, all of which are available on [i.4]. Table 1 summarizes the several SAML specifications.

Table 1: SAML core specification summary

Specification	Summary
Core [i.4]	XML schemas for assertions and SAML request-response protocols
Authentication Context [i.5]	Includes the actual authentication method used (eg, Password, Smartcard, Kerberos)
Bindings [i.6]	Define how SAML request-response messages can be mapped onto standard messaging protocols such as HTTP or SOAP
Profiles [i.7]	Define rules for using and restricting SAML's syntax for conveying security information to solve specific business problems (eg, web SSO exchange)
Metadata [i.8]	Defines how a SAML entity can describe its configuration data (e.g. service endpoint URLs, key material for verifying signatures) in a standard way for consumption by partner entities
Security and privacy considerations [i.9]	Security and privacy considerations w.r.t. security techniques, bindings, profiles
Conformance [i.10]	Describes features that are mandatory and optional for implementations claiming conformance to SAML
Glossary [i.11]	Defines terms used throughout SAML specifications and related documents

4.2.2 Attribute Exchange

Attribute exchange works similar to authentication and is defined in the Core SAML specification [i.1].

5 Operator/ISP-Enterprise Use Cases

5.1 SSO for small enterprises and home network users

5.1.1 Description

The user from his enterprise or home network wants to login to a SP.

5.1.2 Actors

- User.
- Operator/ISP IDP.
- Home IDP.
- SP.

5.1.2.1 Actors specific Issues

- User:
  - Wants to utilize his Enterprise SSO for Web Services.
- Operator:
  - Provides service of IDP.
- Home IdM:
  - Provides support functions (e.g. id mapping, or local authentication) for the operator IDP.
- Web Service:
  - Relies on SSO provided by operator.

5.1.2.2 Actors specific benefits

- User:
  - Receives a convenient experience of SSO across enterprise and web service.
  - Does not need to consider roles and different account names.
- Operator:
  - Provides service of IDP.
- Home IdM:
  - No direct benefit, but Home IdM is necessary to provide mapping.
- Web Service:
  - Benefits by easier accessibility.

5.1.3 Pre-Condition

- The Home IdM has to be registered with the Telco IdM.

- The Home IdM can authenticate the user.
- the user has the Telco IdM set as his IDP.
- the SP relies on Telco IDP as SSO authentication source.

5.1.4 Post-Condition

- User is logged in to the web service without performing additional authentication.

5.1.5 Normative Flow

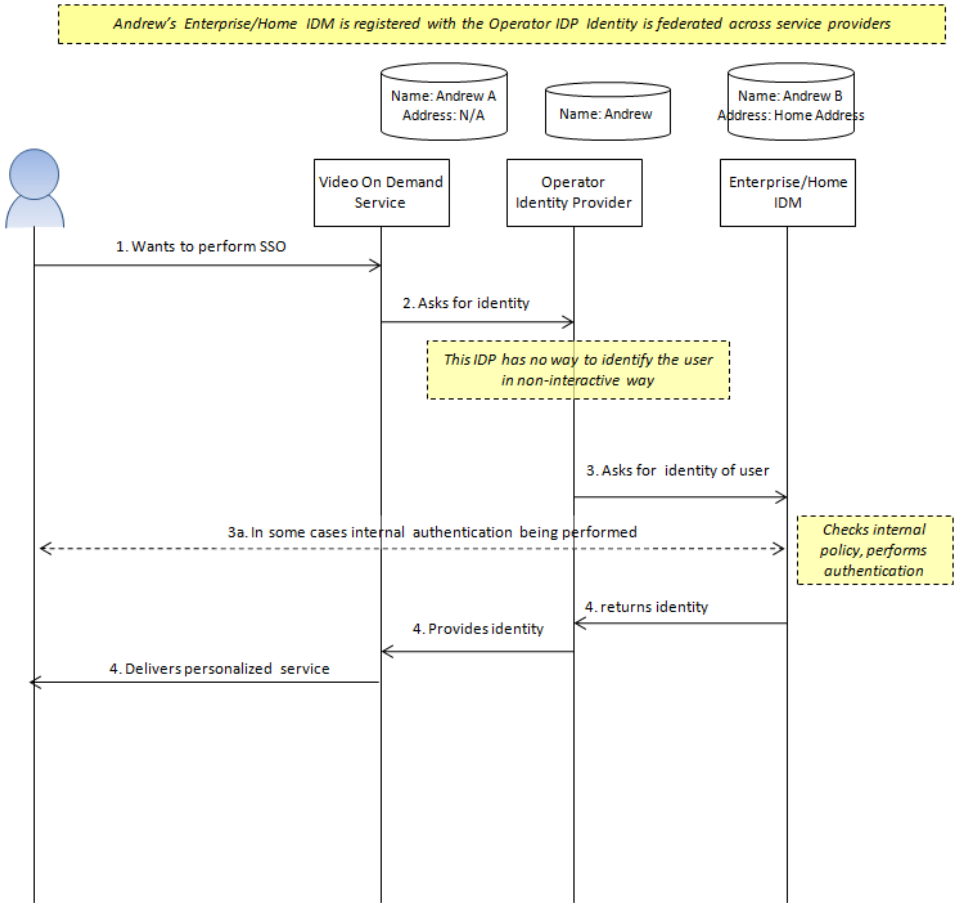

Figure 3: SSO for small enterprises and home network users

Flow description:

1) The user, upon arriving on the web site of the SP, clicks on the "Telco ID" login button.

2) Then the SP queries the Telco IDP for authentication.

3) The Telco IDP identifies that the user is at home, without direct Network Access means of identifying. He queries the users Home IdM for authentication.

4) The User Home IdM utilizes his own policy to perform authentication and returns a SAML assertion to the Telco IDP.

5) The Telco IDP provides the SAML assertion about the user identity to the SP.

5.2 Attribute Sharing between Operator and Web Enterprise

5.2.1 Description

User requests e.g. a video-on-demand (VoD) provided by a web enterprise over the Internet. He changes his default screen resolution and likes this value to be used across different web enterprise applications. For this the attribute has to be stored at the identity provider.

5.2.2 Actors

- User.
- Operator/ISP (IDP).
- Web enterprise (SP).

5.2.2.1 Actors specific Issues

- User:
  - Wants to have his default values automatically used by different web enterprise services.
- Operator:
  - Provides service of IDP.
- Web Enterprise:
  - Provide a service to the customer.
  - Utilizes different attributes to customize service.

5.2.2.2 Actors specific benefits

- User:
  - Needs to do customization only once.
- Operator:
  - Provides service of attribute sharing to different web enterprises.
  - Is trusted partner of web enterprise.
- Web Enterprise:
  - Can customize service according to needs of user, regardless if first time at service or frequent user.

5.2.3 Pre-Condition

- The operator/ISP has stored User identity attributes such as "screen resolution" (of the User's display on which the User wants to watch the VoD) and "available bandwidth" (a value that characterizes the bandwidth available to the User).
- The User wants the operator/ISP to act as IDP on behalf of the User.

5.2.4 Post-Condition

- Attribute of user is stored at IDP by web enterprise.

5.2.5 Normative Flow

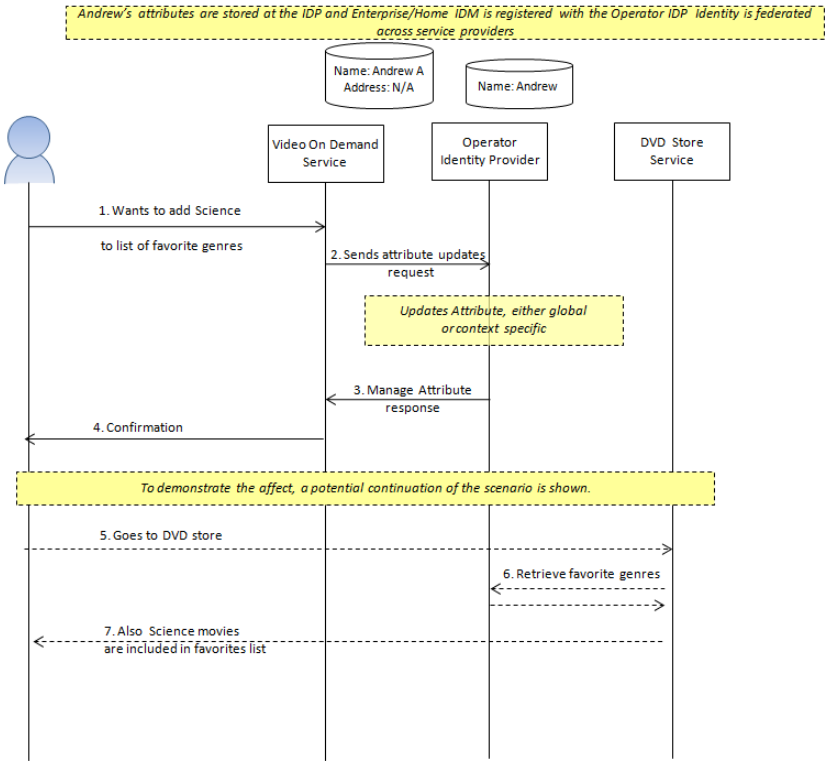

Figure 4: Attribute Sharing between Operator and Web Enterprise

Flow description:

1) The User requests a video-on-demand (VoD) provided by a web enterprise (SP) over the Internet.

2) The web enterprise asks the operator/ISP for User identity attributes such as "screen resolution" and "available bandwidth".

3) The operator/ISP sends the values of the requested User identity attributes to the web enterprise.

4) The user performs some changes to his user profile, e.g. update of phone number, or change in layout.

5) The SP pushes a new value of the attribute to the Telco IdM.

6) The Telco IdM stores the new value.

NOTE: There are a lot of variations of this use case if you replace the attributes "screen resolution" and "available bandwidth" by, e.g. "bank account number", "credit card number", or "email address".

5.3 Outsource billing to operator

5.3.1 Description

The User wants to access a commercial service for which he has to be charged. In this use case the charging element is outsourced from the Service Provider to the Operator's Identity Provider. An already established legal relation may exist between the Service Provider and the Operator.

5.3.2 Actors

- User.
- Operator/ISP Identity Provider.

- Service Provider.

5.3.2.1 Actors specific Issues

- User:
  - Wants to access a service.
  - The bill is paid through the Operator/ISP.
- Operator /ISP Identity Provider:
  - Has a pre-established contract with the SP.
  - Provides outsource billing services to 3rd party services providers.
- Service Provider:
  - The entity which provides the commercial content to the user.
  - Relies on external billing services.

5.3.2.2 Actors specific benefits

- User:
  - Receives a service without having a formal contract with the service provider (user's privacy is enhanced).
  - The service is charged directly on the user's account in the operator.
- Operator:
  - Provides service of billing to 3rd party providers.
- Service Provider:
  - Increase of potential users once there is no need to explicit create a contract with the user.
  - All the Operators users are potential user's for the service.

5.3.3 Pre-Condition

- A contract exists between the User and the Operator.
- A contract exists between the Operator and the Service Provider.

5.3.4 Post-Condition

- User consumes the service without performing additional registration with the SP.
- Payment is transferred from the Operator to the SP.

5.3.5 Normative Flow

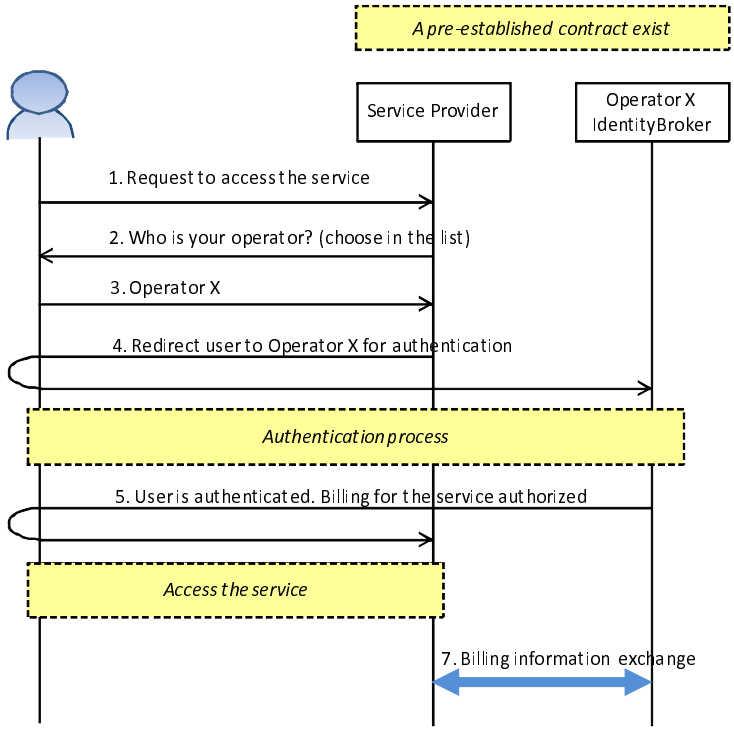

Figure 5: service access

Flow description:

1) The user requests a certain service form the Service Provider.
2) A list of supported operators are presented to the user.
3) The user chooses his operator (the one he has a pre-established contract).
4) The Service Provider redirects the user to the selected operator, for user authentication.
5) If user's authentication is successful the Service Provider receives a "billing authorization" from the Operator.
6) The User access the service.
7) The Service Provider contacts the Operator in order to receive billing for the transaction.

5.4 Integration of XaaS and multi-stage IdM systems

5.4.1 Description

This use case demonstrates how a hierarchical authentication and a hierarchical attributed exchange can be done among XaaS, which is operated by Operator or ISP, and Enterprise.

An Employee X is going to use XaaS service, where the contract has been signed by Employee X's Company for enterprise use. To make use of services for Employee X, traditionally XaaS provider has to set up ID/PW for Employee X but actually XaaS provider does not require authentication of individual user. It only needs to confirm the user is an Employee at the Employee X's Company. Thus in this user case, Enterprise: Employee X's Company is used for authentication of Employee X. When Employee X accesses to XaaS service, XaaS provider requires Employee X to input his company code to identify Enterprise. XaaS provider redirects Employee X to Employee X's Company with authentication request. After Employee X's Company authenticates Employee X by its own authentication scheme (e.g. ID/PW, PKI, etc.), it re-redirects Employee X to XaaS service with authentication assertion. XaaS service verifies the assertion and allows Employee X to use the service. XaaS provider manages identity of Employee X's Company. Employee X is managed as a virtual identity of Employee X's Company. When XaaS service requires Employee X's attribute to provide customized service to Employee X, XaaS provider requests attributes to Employee X's Company.

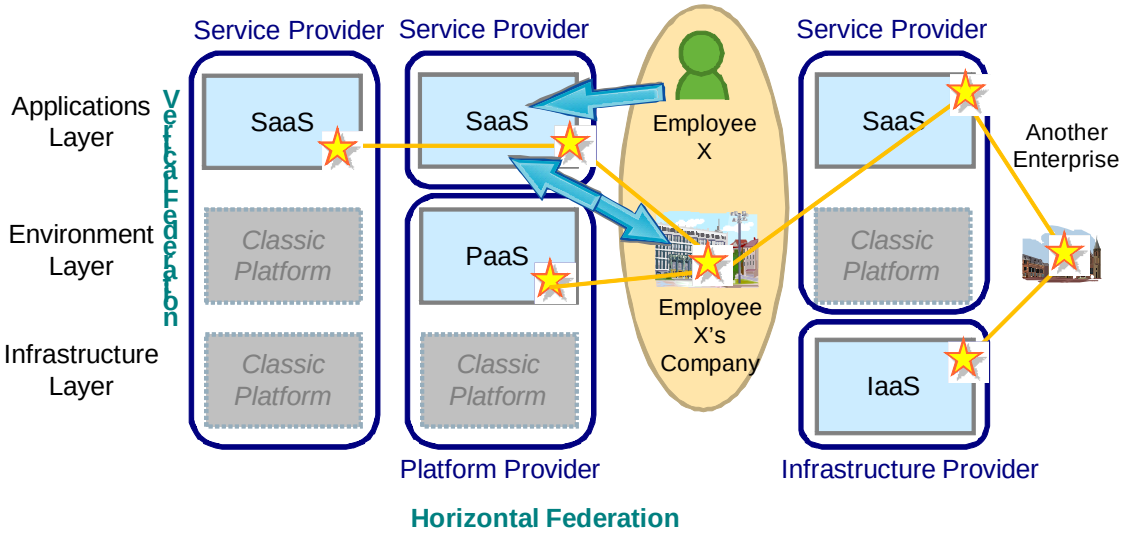

Figure 6: Use Case Integration of XaaS and Enterprise system

Figure 6 shows the overview idea of federation among XaaS and Enterprise.

5.4.2 Actors

- Employee X.
- Employee X's Company.
- Operator (XaaS Services Provider).

5.4.2.1 Actors specific Issues

- Employee X:
  - Desires XaaS services usage but does not have an account at XaaS provider.
- Employee X's Company:
  - Stores data related to their employees.
  - Acts and an IdP for its employee's.

- Has signed contract of XaaS services.

- Operator (XaaS Services Provider):
  - Maintains and operates the XaaS infrastructure and services.
  - Has a trust relation with Employee X's company.
  - Acts as a service provider (regarding the authentication request and attribute request to the Company).

5.4.2.2 Actors specific benefits

- Employee X:
  - Does not need to have and remember another ID/PW for the usage of XaaS services.
- Employee X's Company:
  - Maintains control over employee authentication and attributes provision.
  - Can provide XaaS service usage to any employee with a single agreement with the Operator, which offers XaaS services.
- Operator:
  - Can provide various XaaS services to several Companies.
  - Does not have to manage users' credentials or attributes.

5.4.3 Pre-Conditions

- Employee X's Company and Operator, which is providing XaaS services, have already agreed and signed the contract.
- Employee X is registered as an Employee at Identity Management system at Employee X's Company.

5.4.4 Post-Condition

- In addition to authentication and attribute provision, access control can also be forced at Employee X's Company.
- Operator might customize its XaaS service based on user's attributes, which is retrieved from Employee X's Company.

5.4.5 Example Flow

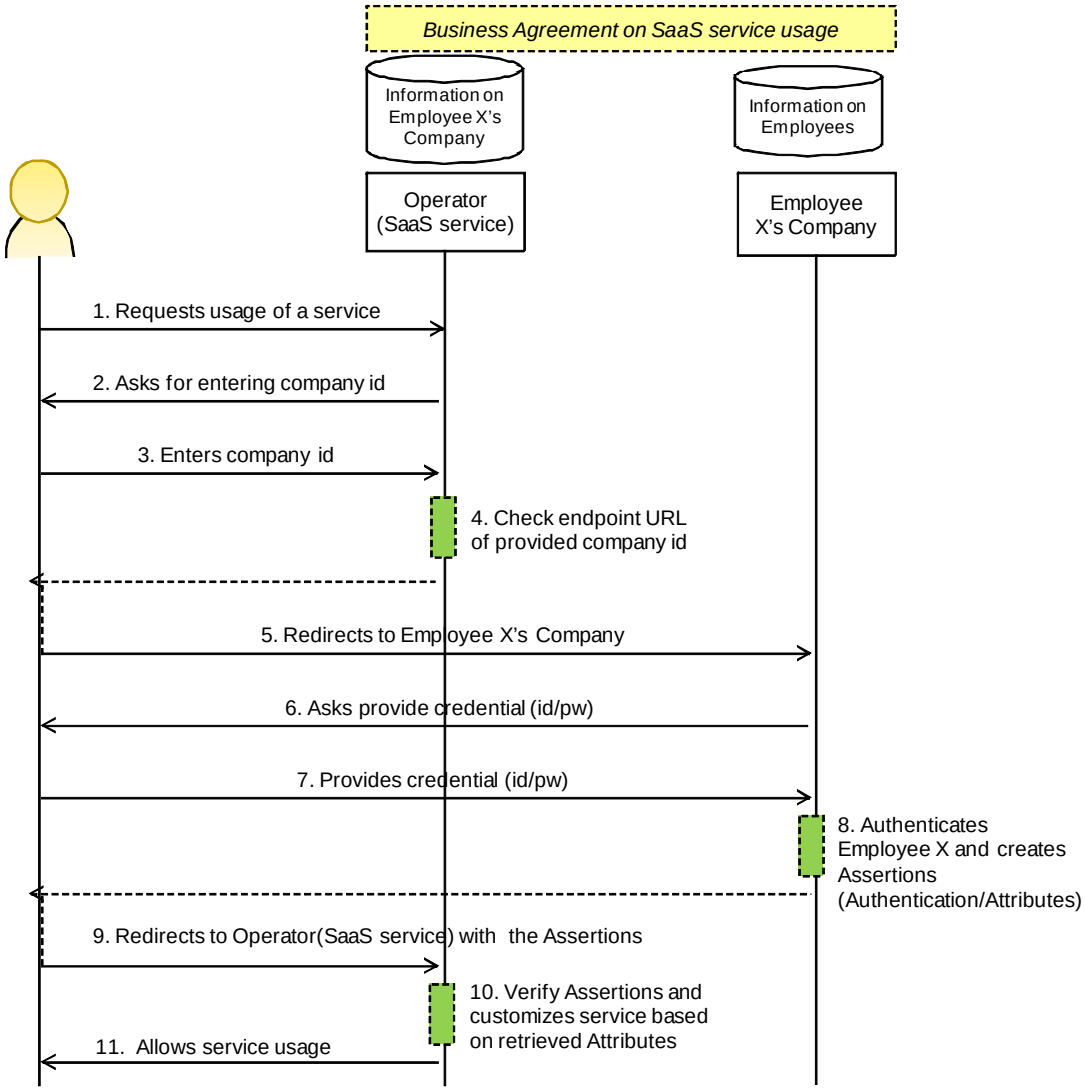

Figure 7: SaaS service access example flow

Flow description:

1) Employee X accesses SaaS service run by Operator.

2) Operator asks Employee X to enter his company's ID.

3) Employee X enters his company's ID.

4) Operator checks endpoint URL of provided company ID.

5) Employee is redirected to Employee X's Company portal to perform authentication.

6) Employee X's Company requests Employee's credential.

7) Employee X provides his credential.

8) Employee X's Company authenticates Employee X and creates Assertions (Authentication Assertion and Attribute Assertion).

9) Employee X's Company redirects Employee X to Operator with created Assertions.

10) After the Assertions are transmitted back to Operator, Operator verifies Assertions and customizes services based on retrieved Employee X's attributes.

11) Operator grants the access of SaaS service to Employee X.

5.5 Authentication as a service

5.5.1 Description

This use case shows how mobile operators can provide authentication as a service offered to enterprises by extending their current authentication functions based on (U)SIM or by offering other authentication schemes like One-time password, PKI, etc.

An enterprise is currently providing access to content for users having an account at its Web site but finds the authentication using passwords too weak. Since most of people have nowadays a mobile phone and a related subscription it is quite convenient and cost efficient to use the mobile phone in the authentication for enterprise access. Technically speaking, it is desirable to reuse the authentication mechanism on the SIM or USIM or other authentication schemes for the access to the enterprise Web site. Seen from the operator authentication will be a new service that brings revenue.

To illustrate how the authentication service works let consider the case of a user X that browses on his/her PC and attempts to log in at a Web site of an enterprise E. After he/she has enter his/her user name and click on return he will be redirected to the operator for authentication. The operator will carry on authentication towards his/her mobile phone. The user will receive a prompt on the mobile phone and will have to enter a Pin code (or simply press return) to confirm that he/she is in the possession of the mobile phone. The authentication is now performed with the participation of the HLR (Home Location Register) or HSS (Home Subscriber Server) or other authentication server and the (U)SIM on the user's mobile phone connected to the PC via Bluetooth or on a USB dongle. If successful the user's browser will be redirected back to the Web site of the enterprise with an authentication token. The Web site verifies the authentication token and grants access to the user.

Another case is the one of a user X who is employee of enterprise E. He/she wants to establish a VPN (Virtual Private Network) between his/her PC. He /she enters his/her user name to the VPN client and presses return. The operator will be triggered either by the VPN client or the VPN server and initiate authentication towards the U(SIM). If successful, a ciphering key generated by the GSM or AKA ciphering will be delivered to both the VPN client and VPN server that use it to establish a secure channel between them.

In both cases the pre-conditions are:

- The mapping (federation) of the two user's identities, namely the user name and the IMSI (International Mobile Subscriber Identity) must be done by the user and verified by the enterprise, i.e. the enterprise must be sure that the mapping is done by the user and the mobile subscription is the right one.
- Trust is established between the enterprise and the operator.

5.5.2 Actors

- User X.
- Enterprise E.
- Mobile Operator.

5.5.2.1 Actors Specific Issues

- User X:
  - Wants to get granted access from anywhere anytime in a simple way.
- Enterprise E:
  - Wants to have stronger but simple and affordable authentication.

- Mobile Operator W:
  - Has the infrastructure for the strong authentication using (U)SIM.
  - Has a trust relation with the user X.
  - Acts as a subcontractor that performs authentication on behalf of the enterprise.

5.5.2.2 Actor Specific Benefits

- User X:
  - Can get granted access in a simple way and without having to carry any additional authentication token such as smart card, one-time password generator, etc.
- Employee E:
  - Obtains a stronger authentication which is both more cost-efficient and more user-friendly.
- Mobile Operator W:
  - Establishes a new source of revenue.
  - Gets improved loyalty and reduced churn because the user has now a new valuable service.

5.5.3 Pre-conditions

- User X has a subscription at the mobile operator.
- The user has the SIM installed on a mobile phone capable of communicating with the mobile phone via Bluetooth, on a SIM slot in the PC, on a USB dongle or on 3G broadband card.
- User X has mapped (federated) his/her account at enterprise E with his/her mobile identity.
- The mapping has been verified and approved by the enterprise E.
- Trust relation has been established between the enterprise E and the mobile operator W.

5.5.4 Post-conditions

- User X gets granted access to Enterprise's system.

5.5.5 Example Flow

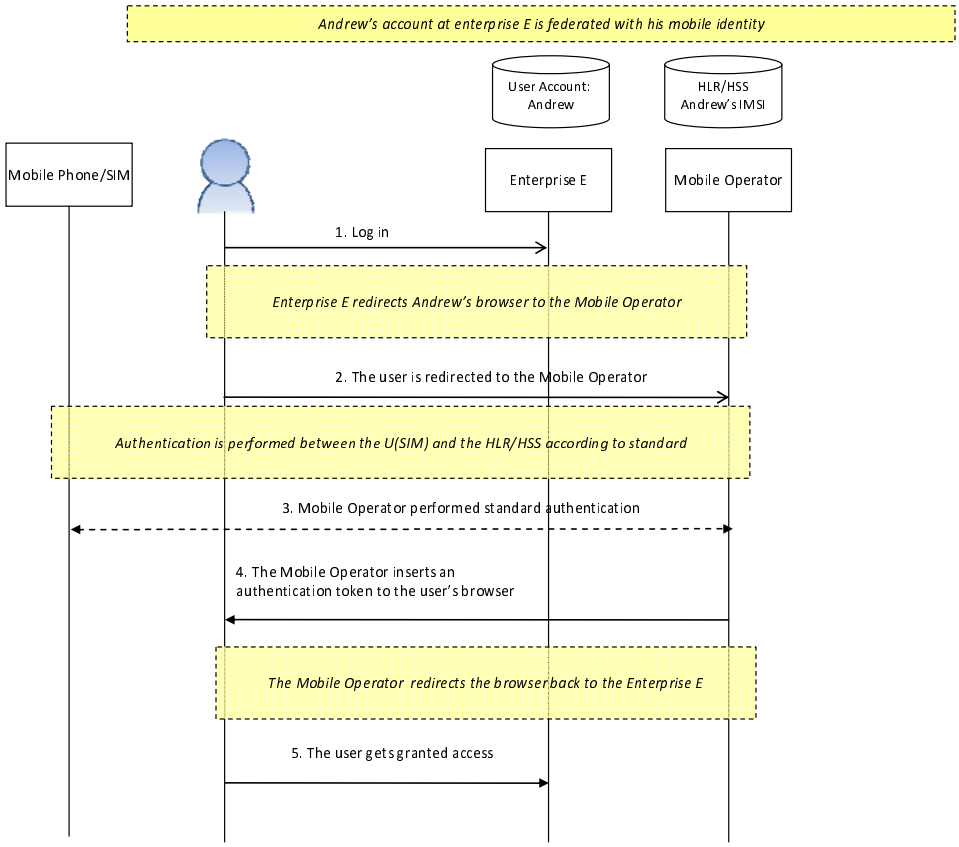

Figure 8: Authentication as service example flow

1) User X attempts to log in at the Web site of Enterprise E.

2) User X is redirected to the mobile operator for authentication.

3) The Mobile operator initiates the authentication towards the SIM using standard authentication.

4) After a successful authentication the Mobile operator inserts an authentication token on the user's browser and redirects it back the enterprise's Web site.

5) The Enterprise's Web site verifies the authentication token and grants access to the user.

5.6 Summary Table of Use Cases

Table 2: Use cases summary

ID	Summary
UC1	The user from an enterprise or home network wants to login to a SP, using the credential he/she typically uses to access the enterprise or home network.
UC2	In this use case a user requests a certain service (e.g. a video-on-demand). When changing an attribute (e.g. default screen resolution), it should be reflected across all services. For this the attribute has to be stored at the identity provider.
UC3	The User accesses a commercial service for which he has to be charged. In this use case the charging element is outsourced from the Service Provider to the Operator's Identity Provider.
UC4	This use case demonstrates how a hierarchical authentication and a hierarchical attributed exchange can be done among XaaS, which is operated by Operator or ISP, and Enterprise.
UC 5	This use case shows how mobile operators can provide authentication as a service offered to enterprises by extending their current authentication functions based on (U)SIM or by offering other authentication schemes like One-time password, PKI, etc.

6 Functional requirements

The architecture presented in the present document aims to present an IdM framework which provides consistent crucial services to both the network and the services. A set of requirements in terms of authentication and attribute exchange reuse, were gathered from the use cases presented previously, those are enumerated next.

Table 3: Functional Requirements

Number	Requirement	Map to Use Cases
6.1 Authentication		
R1	User agent should be able to authenticate with an authentication server, either directly or by means of implicit authentication.	**UC1,UC3**
R2	User's authentication request should be able to be forwarded to the correct authentication server (e.g. home domain).	**UC1, UC2, UC3**
R3	Different types of authentication technologies or protocols can be supported.	**UC1,UC3, UC4, UC5**
R4	An authentication server function should exist and should be able to create assertions about the user's identity.	**UC5, UC1**
R5	Authentication assertions should be augmented with additional information pertaining to the user's authentication (authentication context).	**UC5**
R6	The entity which initiates the authentication process should decide on user's access based on the authentication assertion and authentication context.	**UC1, UC4, UC5**
R7	Authenticity, integrity and confidentiality should exist between the different entities.	**UC1, UC2, UC3, UC4, UC5**
R17	Non-repudiation mechanisms must be supported when appropriate.	**UC1, UC2, UC3, UC4, UC5**
R14	The IdM should support network and application level authentication.	**UC5**
6.1.1 Attribute Exchange		
R8	End to end security should be enabled in all interfaces.	**UC1, UC2, UC3, UC4, UC5**
R9	Interfaces for attribute retrieval should be available intra- and inter-domain.	**UC2**
R10	Attribute retrieval should support pull and push models.	**UC2, UC4**
R11	User defined policies should govern the exposure of attributes to 3rd Party.	**UC1, UC2, UC4**
R12	An attribute could just be an (opaque) reference to an identity stored at a different attribute server.	**UC2**
R13	The attribute management entity(ies) may only store references for the attributes.	**UC2, UC4**
R15	Attribute updates should be possible.	**UC2**
R18	Mutual trust/authentication is required between the entity requesting user's identity attributes and the one responding to the request.	**UC1, UC2, UC3, UC4, UC5**

7 Functional Requirements: Impact on current architectures

The requirements (clause 6) and use cases identified (clause 5) require the state of the art architecture (clause 4) to be adapted to facilitate current developments in merging Network Operator and Internet Application domains.

The goal of the present document is to propose a way in which the Enterprise/Home domains interact with the operator's domain, providing new and innovative services to the customer. Such interactions make necessary the introduction of new element, the IdM Server with interfaces toward the Attribute DB, the Enterprise or CPE is the new entity in the architecture. The IdM Server main functionality is on the application layer toward web applications located in the Internet or Enterprise network.

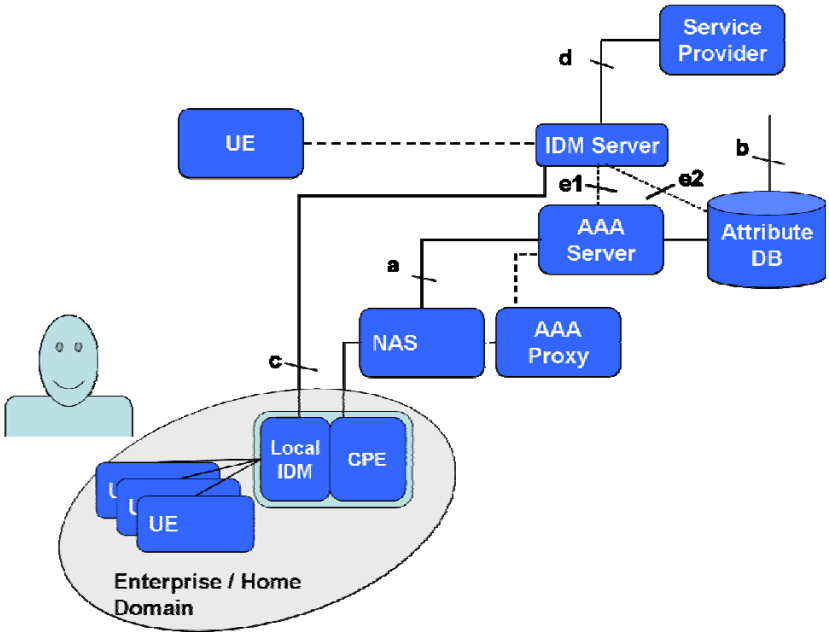

Figure 9: Interfaces of the IdM Server

Figure 9 depicts the new integration (with the IdM Server), Solid lines depict mandatory interfaces, dashed lines optional interfaces.

Moreover it shows the new interfaces briefly described next:

(c) is a bidirectional interface between the IdM Server and a Local IdM, such as Enterprise IdM or Home IdM (in the latter case it is collocated on a CPE). The IdM Server can issue authentication assertions (e.g. if the authentication is being derived from the SIM card) or take into account authentication assertions by the respective "local IdM".

(d) is an interface toward internet application, this is, a standard interface deployed already today in Web SSO or Attribute sharing.

(e1) and (e2) provide the same functionality, namely, attribute retrieval by the IdM server but different protocols would typically be used to achieve this functionality. See clause 8.1.6 for more details on suitable interface protocols.

There are two advantages that the filled gaps enable the operator to do, namely:

- application layer authentication can be derived from network layer authentication; and
- SSO of enterprise users into internet applications facilitated by the operator.

Use cases 1 and 5 reflect these two advantages.

8 Functional architecture definition

Based on the requirements and scenarios/use-cases presented previously, this clause introduces the functional architecture to support the operator/enterprise interoperability scenarios.

8.1 General

Figure 10 presents the set of functional elements and interfaces that are part of the functional architecture that provides the mechanisms to answer the interoperability scenarios presented previously. Interfaces marked as red are outside the scope of the present document. The Service Provider entity is outside of the scope, as well, once it could be implemented by any party and is identified as the consumer of the user's identity or authentication information. Dashed lines represent different possibilities depending on the implementation scenario.

In terms of implementation, different functional entities may be collocated in a single physical entity. In the case one functional entity is implemented by to physical entities, the interface between these physical entities is outside the scope of the present document.

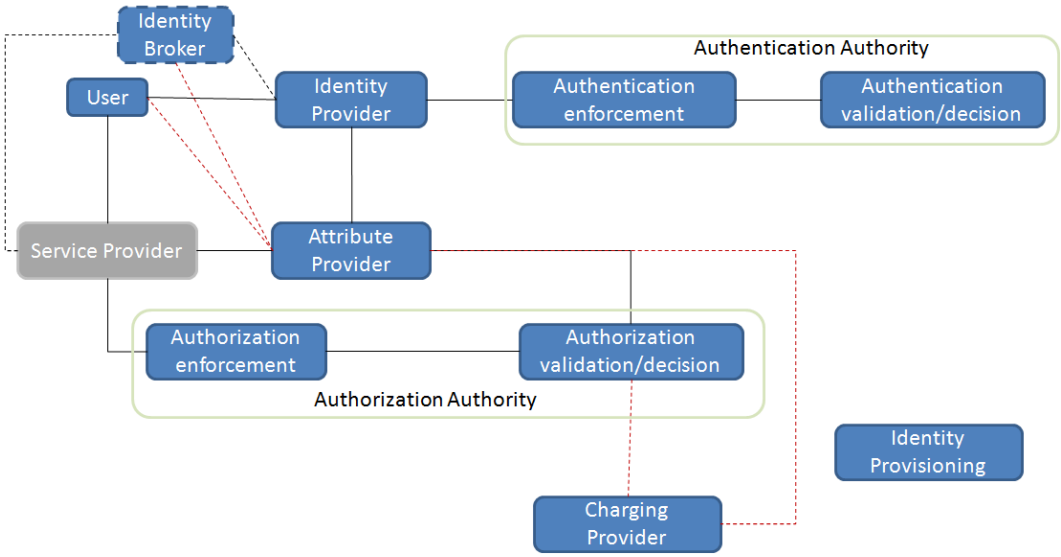

Figure 10: Functional architecture

The proposed architecture was designed to answer the requirements previously presented, having in mind the Telco Operator and the Enterprise as end consumers of it. The main difference of the proposed architecture, with the ones presented in the past, and the ones that make part of the actual standards is the clear separation of the Authentication, Authorization and Identity Management functionalities. Even if the three concepts (or functions) are closely related, they have differences that need to be taken into consideration in order to support and implement future internet scenarios, where interoperability is the key point.

The different functional entities are detailed in clause 8.1.3.

8.1.1 Authentication relationship

Figure 11 shows the high level relations between the different functional elements for the authentication process. Simultaneously it is shown the information exchanged during an authentication procedure. Dashed lines present alternative links between the elements. Those alternatives depend on the scenario where the functional elements are deployed.

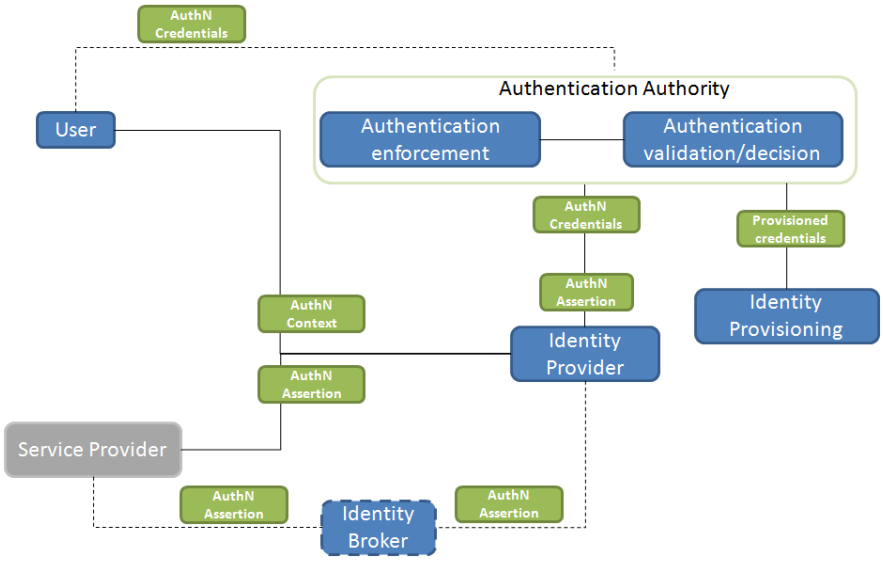

Figure 11: Authentication Authority relation

The main task of the authentication authority is to enforce authentication and to validate claimed identities of entities. The user (principal or supplicant) supplies authentication credentials to the authentication authority either directly or via the Identity Provider, which in turn relays them to the Authentication Decision/Validation element.

After a successfully authentication, the Authentication Decision/Validation generates authentication assertions carrying information about successful authentication and returns it to the Identity Provider. Those assertions are combined with an authentication context describing the authentication characteristics - authentication assurance could be part of the context. Both authentication assertions and authentication context are then used within the Service Provider business logic to decide on the service access

The lifecycle of identity attributes is controlled by the Identity Provisioning element which can be instantiated either as a unique element within the Operator/Enterprise domain's or it could be instantiated in every element that needs to be provisioned.

8.1.2 Attribute exchange relationship

Figure 12 depicts the relations and the exchanged information between the different functions during an attribute exchange process. Dashed lines present alternative links between the elements. Those alternatives depend on the scenario where the functional elements are deployed.

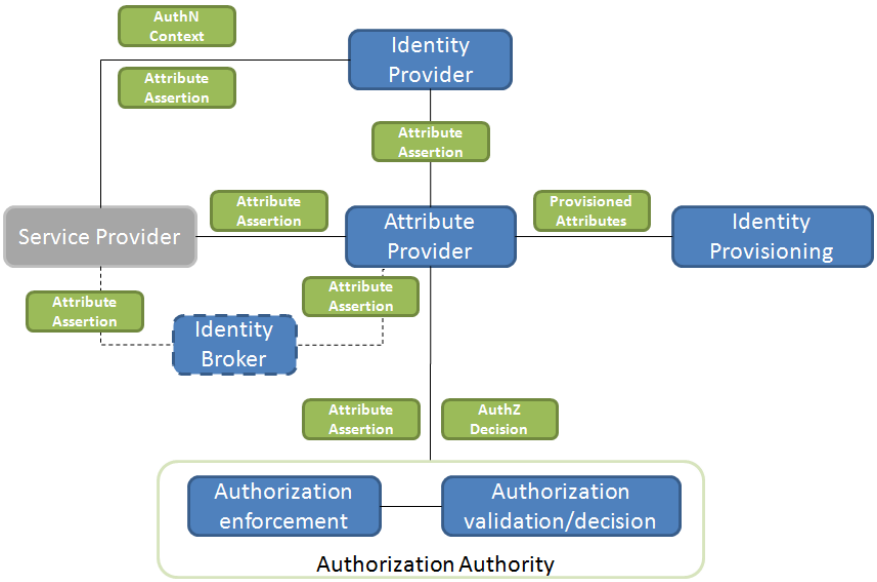

Figure 12: Attribute Provider relations

The Attribute Provider main task is to manage the identity attributes of the user. The Service Provider requires identity attributes on request or receive them automatically if a specific event, which was previously configured, occurs. Whenever the case is, the Attribute Provider issues identity attribute assertions that contains a set of information related with the attributes itself, for example the quality of the attribute.

The Attribute Provider controls access to identity attributes by means of invoking the authorization authority, which returns authorization decisions.

The lifecycle of identity attributes is controlled by the Identity Provisioning element which can be instantiated as a unique element within the Operator/Enterprise's or it could be instantiated in every element that needs to be provisioned.

8.1.3 Functional elements description

8.1.3.1 Identity Provider

The Identity Provider functional element is a service that acts as a mediator party between the requesting entities (i.e. the Service Providers) and the different authorities involved in the user authentication and authorization. It can mediate on any identity management process (i.e. authentication or/and authorization). To do that, the Identity Provider is able to map the different identifiers the user uses on each one of these entities, making them interoperate with a minimum coupling requirements, favoring this way their interoperability. This also provides an additional level of anonymity to users, since their identifiers are not disclosed to any additional entity but their Identity Provider. This would also provide an additional level of privacy to users since access to any remote service (SP or Attribute Provider) is granted based on a subset of user's identity attributes attached to a transient pseudonymous identifier, rather than on a user's permanent identifier. Moreover no entity other than the Identity Provider is able to map the different user's identifiers even if they collude with others.

8.1.3.2 Attribute Provider

The Attribute Provider functional element is a service which provides attributes about a specific user - or principal. The attributes may contain dynamic data such, location, presence or reputation, among others, but they can also hold static information about a user, such as address, birth date, etc.

Despite the dynamism of the attributes the sensitive of the information stored and provided by this element may vary from non-sensitive to sensitive assured by 3rd parties.

The attribute provider should support two different models, i) a request-response communication model to implement interactions with a requestor (e.g. an identity provider) and ii) an event driven communication model. In both models the attribute provider replies requests for attributes by issuing attribute tokens by means of attribute assertion. The assertion should contain sufficient information related with the attribute quality. That information is the foundation of an evaluation mechanism that enables business logic within the attribute requestors.

The attributes life-cycle is managed by the Identity Provisioning function and is the responsibility of the domain's operator.

The generation and issuance of attribute assertions is subject to several types of policies, the data disclosure policy (DHP) among others. The DHP in particular are the result of a negotiation/ agreement between the data owner and the data requester.

The restricted access to, and disclosure of, user's identity attributes is performed by the Attribute Provider and in accordance with rules specified within DHP. The underlying (privacy-preserving) access control functionalities are for that matter provided by an Authorization Authority, which invoked by the Attribute Provider returns authorization decisions. Another type of policies are those modelled to support trust negotiation (i.e. the process by which two entities exchange and mutually validate their credentials) which is obviously required before the start of any authorization process.

8.1.3.3 Authorization Authority

The Authorization Authority functional element provides access control functionality to the Attribute Provider returning authorization decisions. It is composed by the Authorization Enforcement and the Authorization Validation/Decision.

8.1.3.3.1 Authorization Enforcement

The authorization enforcement functional element is responsible for the Service Provider to be authorized and the associated authorization policies to be enforced. The authorization policy contains a set of acceptable authorization protocols and methods. This set could be used to negotiate and agree on an authorization protocol and method with the client. Based on these terms, the authorization enforcement point gathers user identity, the service provider and user's attributes from the Attribute Provider, and sends them to the Authorization Validation/Decision function.

8.1.3.3.2 Authorization Validation/Decision

The Authorization Validation/Decision comprises all functionalities required for access control and usage directives enforcement. The Authorization Validation/Decision point validates authorization request of the privacy policy enforcement with respect to usage directives submitted within the request and with regards to the attribute provider's privacy policy. The Authorization Validation/Decision point may have the need for additional attributes in order to decide on a request. In that case the attribute processing is invoked.

8.1.3.4 Authentication Authority

The Authentication Authority is responsible for the identity to authentication functional element is a system entity that produces authentication assertions. It is composed by the Authentication Enforcement and the Authentication Validation/Decision

8.1.3.4.1 Authentication Enforcement

The authentication enforcement functional element is responsible for the identity to be authenticated and the associated authentication policies to be enforced.

An authentication policy contains a set of acceptable authentication protocols and methods incl. privacy-preserving methods. This set could be used to negotiate and agree on an authentication protocol and method with the client.

Based on the established authentication protocol and methods, the authentication enforcement function gathers the authentication credentials from the client and sends them to the Authentication Validation/Decision function.

8.1.3.4.2 Authentication Validation/Decision

The authentication validation functional element is responsible for validating the authentication credentials, gathered by the enforcement point, for service access. For this purpose, it contains a credential store that holds the server-side secrets and credential data to be used to validate the authentication credentials supplied by the supplicant. During the authentication process a set of policies can be applied to the specific supplicant. Those policies can for example be a set of acceptable authentication protocols and methods. This set could be used to negotiate and agree on an authentication protocol and method with the client. Those policies are part of the authentication context created by this entity and sent to the authentication client - Service Provider. The credential store and policies are provisioned by the Identity Provisioning function.

The Authentication Validation/Decision can also act as a proxy. When acting as a proxy the Authentication Validation/Decision can locate and communicate with the Authentication Validation/Decision acting as a server which contains the server-side secrets and credential data that will be used to validate the authentication credentials supplied.

The authentication methods to be used are specific of the scenario and are not specified within the present document. Different methods could co-exist within the same scenario.

8.1.3.5 Charging Provider

The Charging Provider functional element is a service which provides accounting data sets based on service and customer specific tariff parameters. Different cost metrics may be applied to the same accounting records even in parallel. Charging Provider uses information related to chargeable events and the rating function to derive costs for chargeable events and it adjusts the charging accounts accordingly in order to make it possible to determine usage for which the charged party may be billed. The charging provider usually implements ratings and charging functions. The rating function refers to the process of pricing the mediated usage records from the mediation system (accounting provider). The pricing/rating is done according to a rating policy and may have any appropriate target unit type compatible to a given customer account. Charging refers to the actual act of debiting/crediting a customer's credit account or applying a reservation. In case of prepaid accounts the charging transaction is directly settled via the available funds on the account. In case of post-paid accounts the settlement usually happens.

8.1.3.6 Identity Provisioning

This functional element is the unique contact point between the Operator/Enterprise administrator and the entire functional elements. Its main responsibility it to store, push or be able to receive requests about domain specific policies, attributes, authentication methods and credentials to the correct elements within the Operator's/Enterprise's domain.

Different implementation possibilities exist, depending on the implementation scenario. As example, it could be seen as an unique instantiation in all the domain, or it can be instantiated co-located with the other elements - for example the Identity Provider.

The interfaces to the other components are outside the scope of the present document.

8.1.3.7 Identity Broker

The Identity Broker functional element provides the user with the ability to easily manage her accounts on different Identity Providers. It provides a homogeneous way of accessing services, allowing for example automatic identity negotiation, autonomous policy management based on device capabilities. It also simplifies the policy management from the point of view of the user, becoming into a single configuration point for identity management. Depending on the operation mode, different amounts of trust are placed at the Identity Broker. If the ID-Broker only redirects requests, the burden of policy control is redirected to the identity providers. But when applying access control rules directly on the mapping and content access, pointing to different locations depending on the requesting entity, it becomes a more crucial architecture component.

The Identity Provider, User and the Identity Manager trust the ID-Broker to hold the mapping metric, the access control policies and the links to the different locations securely. Moreover they trust the broker to not divulge such sensitive information to anyone except under the instructions of the User. Thus the latter is able to specify, depending on the Service Provider, which Identity Providers can "vouch" for his/her identity in a federated environment.

8.2 Interfaces

To support the use cases and the architecture described previously, a set of interfaces have to be specified. All the interfaces require mutual authentication between the components exposing the interface and the consumer. The following interfaces are supported:

- IdentityResolution interface - to retrieve another Identifier from a known identifier of a given Identity.
- IdentityManagement interface - to create, register, revoke and delete an identifier of a given Identity.
- AttributeManagement interface - to retrieve, create, delete an attribute for a given Identity.
- IdentityAuthentication interface - to (de-)authenticate a given Identity.
- IdentityCharging interface - to bill a service on a given Identity.

Next clauses present, for each interface, the supported operations.

8.2.1.1 IdentityResolution

The IdentityResolution interface provides the operations to retrieve a certain identifier from a known identifier. It is composed by a unique operation **GetIdentifier** that returns the target identifier and accept an identifier which is known to the element.

Table 4: GetIdentifier input parameters

Argument	Description
KnownIdentifier	The identifier known to the application which should be resolved to another one of the same Identity.
TargetIdentifier	An Identifier of the target towards the known Identifier should be resolved with another one.

The invocation of ResolveIdentifier requests to resolve the given Identifier of an Identity into another Identifier. The TargetIdentifier parameter indicates the target to which the new Identifier of the Identity will be exposed.

8.2.1.2 IdentityManagement

The IdentityManagement interface provides the operations to external providers to create and register as well as to delete an identifier of a given identity. Associated with the created identifier, or an existing identifier, it is also provided the way to set policies that will govern the access to such identifier.

Operation **CreatePseudonym** - This operation creates a new identifier to "masquerade" a certain identifier and returns it to the caller.

Table 5: CreatePseudonym input parameters

Argument	Description
KnownIdentifier	The identifier known to the application which should be resolved to another one of the same Identity.
Policy	The policy that will govern the access to the automatically created pseudonym.

Operation **DeletePseudonym** - This operation requests deletion of a "masquerading" identifier.

Table 6: DeletePseudonym input parameter

Argument	Description
KnownIdentifier	The identifier known to the application which should be resolved to another one of the same Identity.

8.2.1.3 AttributeManagement

The AttributeManagement interface provides the operations to create, retrieve, modify and delete attributes for a certain identity. Attributes can be created and managed individually or in groups. Moreover, it provides the operations to set policies over each individual attribute or groups of attributes. A service provider can only modify/delete his own attributes, it is the task of the IdM to do proper context based authorization control.

Operation **CreateAttribute** - This operation creates an attribute, associates it with a given identifier with a given policy that will govern the usage of the attribute. Table 7 presents the input parameters to the operation.

Table 7: CreateAttribute input parameters

Argument	Description
ExistingIdentifier	The identifier known to the application for which the attribute will be linked, in case the NewIdentifier argument is not present. If the NewIdentifier is present this argument is only used to identify the Identity within the application.
NewIdentifier (*optional)*	If present, the new created attribute will be linked to this identifier.
AttributeName	The name for which the attribute will be known.
AttributeType	The type of the attribute.
Value (*optional)*	The value provisioned for the created attribute, defined in the AttributeName argument. If not present the value will be set to null.
Policy	Policy that will govern the usage of the attribute.

Operation **DeleteAttribute** - This operation deletes a certain attribute given an identifier and the attribute name. If applicable, the policy that governed the usage of the attribute in question should be deleted, too.

Table 8: DeleteAttribute input parameters

Argument	Description
ExistingIdentifier	The identifier known to the application for which the attribute is linked.
AttributeName	The name for which the attribute is known.

Operation **GetAttribute** - This operation gets the value of a given attribute. Based on the policies that apply to this attribute (or group of attributes to which this attribute belongs), the returned value could not be the real value but some token, or related information.

Table 9: GetAttribute input parameters

Argument	Description
ExistingIdentifier	The identifier known to the application for which the attribute is linked.
AttributeName	The name for which the attribute is known. This argument is optional, once the policy can be applied to a group of attributes and not to a particular attribute An omitted AttributeName argument indicates that the sender requests to receive all attributes linked to the ExistingIdentifier - in accordance to the applicable policies governing usage of these attributes.

Operation **ModifyAttribute** - This operation can be used to modify the value of a given attribute.

Table 10: ModifyAttribute input parameters

Argument	Description
ExistingIdentifier	The identifier known to the application for which the attribute is linked.
AttributeName	The name for which the attribute, whose value shall be modified, is known.
Value	The new value provisioned for the attribute identified in the AttributeName argument.

Operation **SetAttributePolicy** - This operation sets the attribute, or group of attributes, disclosure policy. It states for example for each Service Providers the attribute value can be released and in which conditions.

Table 11: SetAttributePolicy input parameters

Argument	Description
ExistingIdentifier	The identifier known to the application for which the attribute is linked.
AttributeName (*optional*)	The name for which the attribute is known. This argument is optional, once the policy can be applied to a group of attributes and not to a particular attribute.
AttGroupName (*optional)*	The group name where the attribute belongs. This argument is optional and if present means that the policy should be applied to a group of attributes.
Policy	Specifies the policy that governs the disclosure of the attribute or group of attributes. Where applied to a group all the attributes within the group will inherit the policy.

8.2.1.4 IdentityAuthentication

A 3rd party entity (e.g. a service provider on the web) can use the IdentityAuthentication interface to request from the recipient (e.g. an IdM Server) to authenticate a User or other entity whose identity is specified by means of an existing identifier. In addition, the requester can also ask the recipient to return authorization policies associated to the authenticated identity.

Operation **AuthenticateIdentity -** This operation provides the way for 3rd party entities to request a User or entity authentication and, optionally, authorization policies assocated to the authenticated User or entity.

Table 12: AuthenticateIdentity input parameters

Argument	Description
ExistingIdentifier	The identifier known to the application for which the sender requests authentication.
AuthorizationPolicies	Optional argument that, if present, indicates that the sender requests from the recipient not only to authenticate the User or entity in question but also to return authorization policies associated to that authenticated identity.

8.2.2 IdentityCharging interface

The IdentityCharging interface provides the operations to charge a certain identity with the costs associated with a certain transaction. Using this interface, Service Providers may bill services via MNO billing. For example, refer to use case 5.3. The description of the operations in this clause should be seen as rudimentary pointers to more elaborate accounting and charging mechanisms, as defined by the RADIUS and DIAMETER specifications, for example.

Operation **Reserve -** this operation reserves a certain amount of credit for a service provider specific action that needs to be paid.

Table 13: Reserve input parameters

Argument	Description
ExistingIdentifier	The identifier known to the application for which the reserve is linked.
Amount	The amount of credit that needs to be reserved.

Operation **GetAuthorization** - This operation requests an authorization token for charge a certain identity. A Service Provider can use this operation to verify if the payment for the service can be done by the operator.

Table 14: GetAuthorization input parameters

Argument	Description
ExistingIdentifier	The identifier known to the application for which the authorization is required.
Amount	The amount of credit that is planned to be charged to the identity.

8.3 Protocols

This clause proposes protocols suitable for achieving the desired functionality of the interfaces defined in clause 7.

8.3.1 Interface c

A Local IdM and an IdM Server communicate over interface c in order to request and return authentication assertions. A suitable protocol for interface c would be SAML.

8.3.2 Interface d

Interface d connects Service Providers and IdM Servers for SSO and attribute sharing purposes. Like in the case of interface c, these functions can be implemented by means of SAML.

8.3.3 Interface e1

Interface e1 is the interface between the IdM Server and the AAAS Server. Its function is to retrieve attributes from the IdM Server. Protocols suitable to achieve this function are RADIUS and DIAMETER. Interface e1 is expected to require further standardization beyond existing RADIUS and DIAMETER specifications.

8.3.4 Interface e2

Interface e2 is the interface between the IdM Server and an Attribute Database. Its function is the same as in the case of interface e1, namely, attribute retrieval. In contrary to e1, however, interface e2 would usually be implemented by means of LDAP. Like for interface e1, interface e2 is expected to require further standardization beyond what LDAP already provides.

9 Operator/ISP-Enterprise IdM Interoperability instantiation

Within this clause the instantiation of two of the use cases presented in clause 4 are described. The instantiations presented in this clause are possibilities and should be seen as informative. Different instantiations are possible depending on the specificities of the scenarios where the use cases are applied.

9.1 Instantiation SSO for small enterprises and home network users

This clause describes a possible instantiation of the use case described in clause 5.1. ("SSO for small enterprises and home network users"). Apart from a UE, the functional architecture entities present in this use case are the following:

- Video On Demand System.
- Local IdM (e.g. Home or Enterprise IdM).
- Operator IdM.

The subsequent three paragraphs overview instantiations of these functional architecture elements. Clause 9.1.4 points to interface protocols and messages that can be used to achieve the desired SSO functionality.

9.1.1 Instantiation Video On Demand System

The Video on Demand System has all interfaces and components of the service provider. It may include attribute enforcement.

9.1.2 Instantiation Local IdM (e.g. Home or Enterprise IdM)

Home IdM/Enterprise IdM is the Local IdM system, which needs to provide interface c toward the IdM server. In small networks it will be co-located with the CPE, in large enterprises it will be instantiated using the existing Enterprise IdM system.

The following components are included in the Local IdM:

- Identity Provider (toward the Operator IdM).
- Optionally, an Attribute Provider.

9.1.3 Instantiation Operator IdM

In general the Operator IdM will depend on different identity providers. In this use case, the Local IdM will be the Identity Provider in question. The Operator IdM will have additional attribute sources along with attributes provided by the Local IdM.

The following components are included in the Local IdM:

- Identity provider for the Video on Demand Service.
- Attribute provider for the Video on Demand Service.

9.1.4 Use of Interfaces

Figure 13 depicts interface protocols that can be used to instantiate the use case described in clause 5.1 to achieve the desired SSO functionality (dashed lines around system entities indicate optional components).

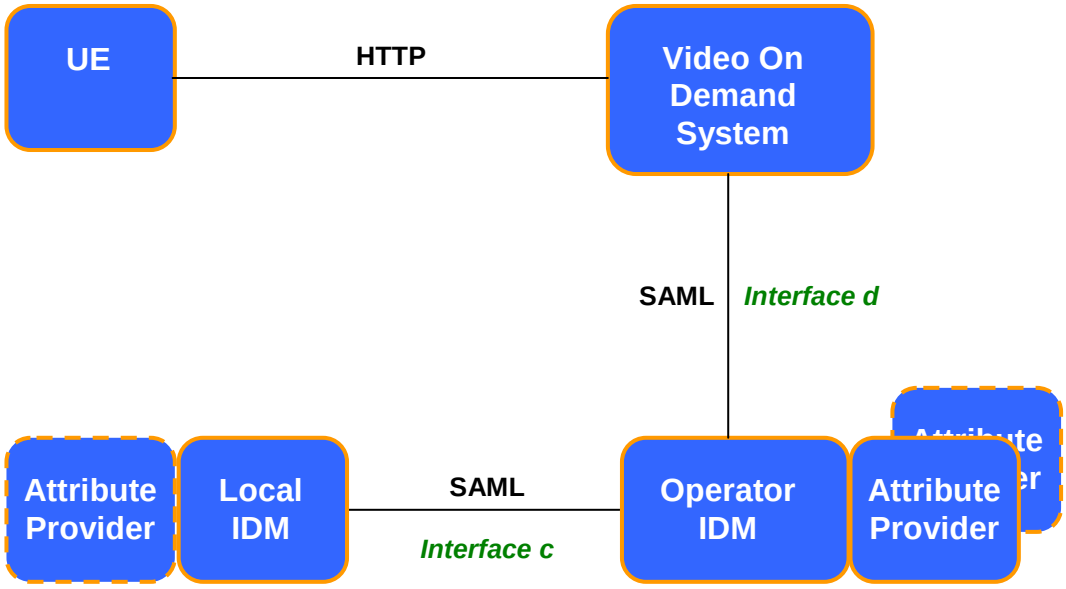

Figure 13: Instantiation of Use Case "SSO for small enterprises and home network users"

The User can access the Video On Demand Service Provider via HTTP and clicks on an "Telco ID" button. To ask the Operator IdM for User authentication, the Security Assertion Markup Language (SAML) and its protocols can be employed over interface d. The use case description in clause 5.1 assumes that the Operator IdM detects that the User currently does not have any direct network access means of identification. Therefore, over interface c, the Operator IdM requests the Local IdM to authenticate the User. To this end, SAML can be used, too. According to its own policies, the Local IdM authenticates the User and returns a SAML assertion back to the Operator IdM. Over interface d, the Operator IdM in turn sends this SAML assertion to the Video On Demand System. If the Video On Demand System accepts this SAML assertion after validation, the SSO functionality desired in this use case will be accomplished.

9.2 Instantiation Authentication as a Service

This clause describes an envisioned instantiation of the use case described in clause 5.5. ("Authentication as a Service"). The following functional entities are present in this use case:

- User.
- Enterprise.
- Mobile Operator.

Next clauses provide the instantiations of these functional elements. The User is not instantiated once none of the functions presented in the architecture are part of the UE. Moreover clause 9.2.4 provides a graphical presentation of the next clause instantiation and points to interface protocols and messages that can be used to achieve the desired SSO functionality.

9.2.1 Instantiation Enterprise

The Enterprise provides a service to the User, based on strong authentication mechanism that relies on Operator's authentication functionalities; in that sense it behaves as a Service Provider. The following components are included within the Enterprise.

Table 15: architectural elements within the Enterprise element

Functional element	Rational
Authentication Enforcement	The enterprise should, based on the policies received from the Identity Provider enforce the access to the resource (in this particular case within the Enterprise).
Authentication Decision	Based on the information obtained from the Identity Provider, the Enterprise has to decide if the authentication, successfully done on the Operator should be accepted (from example the Level of Assurance of the authentication could not be enough).
Authorization Enforcement	Based on the rules and attributes received after a successful user authentication, the Enterprise should be able to enforce the service access authorization.
Authorization Decision	After the authentication process the Enterprise receives a set of attributes and rules that will be used to perform a decision in terms of authorization to the service.

9.2.2 Instantiation Mobile Operator

The Mobile Operator (MO) provides the authentication service to other actors. Along with the authentication, the MO could also provide attributes and policies that will help the authentication requester decide on the authentication. Furthermore, the MO can play the role of Identity Provider, mediating the requesting entities and authorities involved in the user's authentication and authorization processes.

The following components are included within the MO.

Table 16: Architectural elements within the Mobile Operator element

Functional element	Rational
Authentication Decision	The MO behaves as an Authentication Decision functions. It verifies if the information provided by the user (or any other entity in is name) is correct. In this scenario it verifies if the SIM Card of the user is correctly authenticated with the network.
Attribute Provider	After the authentication process the MO provides a set of user's attributes to the Enterprise. Such attributes will be used to assist the Enterprise to authorize the access to the resource.
Identity Provider	The MO is also the mediator between the requester (the Enterprise) and the ME function responsible for the authentication process (i.e. HSS).
Identity Provisioning	For this particular use-case, the user has to be a MO customer. This implies the provisioning of an identity and a set of attributes for the customer.
Identity Broker (optionally)	The MO can also play the role of Identity Broker if it provides the technology and the applications needed for the customer to manage his identities (spread between different Identity Providers).

9.2.3 Use of Interfaces

Figure 14 depicts interface protocols that can be used to instantiate the use case described previously. It also presents the architectural elements distribution between the different use-case actors.

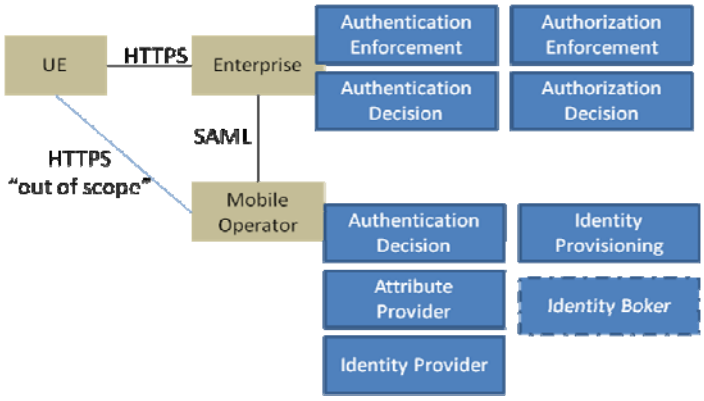

Figure 14: Instantiation of Use Case "Authentication as a Service"

The interface between the UE and the Mobile Operator is out of the scope of the goals of the present document. The interface is the one used actually in mobile networks to perform authentication.

Annex A (informative): Authors and contributors

The following people have contributed to the present document:

Rapporteur:

- Ricardo Azevedo, Portugal Telecom Inovação.

Other contributors:

- Antonio Skarmeta, University of Murcia.
- Hervais Simo Fhom, Fraunhofer Institute for Secure Information Technology.
- Kpatcha Bayarou, Fraunhofer Institute for Secure Information Technology.
- Joerg Abendroth, Nokia Siemens Networks.
- Naoko Ito, NEC.
- João Girão, NEC.
- Pedro Santos, Portugal Telecom Inovação.
- Peter Scholta, Deutsche Telekom.
- Van Thanh Do, Telenor.
- Wolfgang Steigerwald, Deutsche Telekom.

History

Document history		
V1.1.1	March 2011	Publication