D3.3

Specification of remote and D2D IDM mechanisms for HRs Interoperability - V1

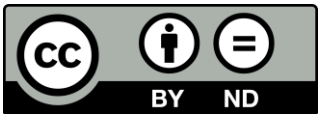
ABSTRACT

This deliverable provides the first version of the specification of remote and D2D Identity Management (IDM) including authentication mechanisms in InteropEHRate. This document also provides a detailed technical background, which is a necessary step to move forward. The final and more detailed specification will be provided in the second forthcoming deliverable.

Delivery Date	3rd July 2019
Work Package	WP3
Task	3.2
Dissemination Level	Public
Type of Deliverable	Report
Lead partner	UBIT

InteropEHRate project is co-funded by the European Union (EU) Horizon 2020 program under Grant number 826106.

This document has been produced in the context of the InteropEHRate Project which is co-funded by the European Commission (grant agreement n° 826106). All information provided in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose.



CONTRIBUTORS

	Name	Partner
Contributors	Sofianna Menesidou	UBIT
	Alessio Graziani	ENG
	Simona Bica	SIVECO
Reviewers	Gábor Bella, Simone Bocca	UNITN
	Sébastien Hannay	A7

LOGTABLE

Version	Date	Change	Author	Partner
0.1	09-05-2019	First draft of ToC	Sofianna Menesidou	UBIT
0.2	13-05-2019	Feedback for structure	Alessio Graziani	ENG
0.3	13-05-2019	Feedback for structure	Simona Bica	SIVEO
0.4	28-05-2019	Input at Chapter 2 and 3	Sofianna Menesidou	UBIT
0.5	01-06-2019	Input at Chapter 1 and 2	Sofianna Menesidou	UBIT
0.6	03-06-2019	Input at Chapter 2	Sofianna Menesidou	UBIT
0.7	05-06-2019	Input at Chapter 2	Sofianna Menesidou	UBIT
0.8	11-06-2019	Input at Chapter 2	Sofianna Menesidou	UBIT
0.9	16-06-2019	Input at Chapter 2	Sofianna Menesidou	UBIT
0.10	17-06-2019	Input at Chapter 3 and 4	Sofianna Menesidou	UBIT
0.11	18-06-2019	Input at Chapter 2	Alessio Graziani	ENG
0.12	19-06-2019	Input at Chapter 2	Alessio Graziani	ENG
0.13	22-06-2019	Input at Chapter 3	Sofianna Menesidou	UBIT
0.14	23-06-2019	Input at Chapter 3	Sofianna Menesidou	UBIT
1.0	25-06-2019	Internal review	Gábor Bella, Simone Bocca	UNITN

1.1	25-06-2019	Internal review	Sébastien Hannay	A7
1.2	26-06-2019	Changes based on reviewers' comments	Sofianna Menesidou	UBIT
1.3	30-06-2019	Completed quality check	Argyro Mavrogiorgou	UPRC
Vfinal	03-07-2019	Final version for submission	Laura Pucci	ENG

ACRONYMS

Acronym	Description
2FA	Two-Factor Authentication
B2C	Businesses and Consumers
B2E	Businesses and Employees
BLE	Bluetooth Low Energy
CEF	Connecting Europe Facility
eHDSI	eHealth Digital Service Infrastructure
eID	Electronic identification
ETL	European Trust Lists
FHIR	Fast Healthcare Interoperability Resources
G2C	Governments and Citizens
HSM	Hardware Security Modules
IDM	Identity Management
LSP	Large Scale Pilots
MS	Middleware-Service
NFC	Near-Field Communication
NCP	National Contact Point
OIDC	OpenID Connect
OCSP	Online Certificate Status Protocol
PKI	Public Key Infrastructure
QC	Qualified Certificates
QSCD	Qualified Signature Creation Device
SAML	Security Assertion Markup Language
SP	Service Provider
TSP	Trusted Service Provider

U2F	Universal 2nd Factor Authentication
UAF	Universal Authentication Framework
USB	Universal Serial Bus
WebAuthN	Web Authentication API
WS-Federation	Web Services Federation
XACML	eXtensible Access Control Markup Language

TABLE OF CONTENT

LIST OF FIGURES

1. INTRODUCTION

Electronic identification (eID) within the eHealth domain is the cornerstone of patient safety. Improvements in electronic identification systems and processes have emerged as a fundamental concern for European states in their pursuit of better relationships and interactions between governments and citizens (G2C), businesses and consumers (B2C), and businesses and employees (B2E) [KAI2009]. Specific regulations are being established to ensure that electronic signatures provide a legal standing equivalent to that of handwritten signatures. eIDAS [EE2017] and NIST-DSS [NISTDSS] for the EU and USA respectively are examples of such regulations.

The eIDAS [EE2017] Regulation basically establishes a cross-border and cross-sector legal framework that covers electronic signatures, electronic documents, electronic time stamps, electronic seals, certificate services, and electronic registered delivery services in relation to eIDs and EU-based trust service providers. Trust forms the heart of the eIDAS Regulation, where eID is short for electronic ID. The regulations for electronic IDs has come into force from autumn of 2018, which is closely related to the cross-border acceptance of eIDs in public services [NGUYEN2018]. The European Commission is using eIDAS as a toolbox to ensure the trustworthiness of various online services that fall under the control of the Commission.

This report provides the first version of the specification of remote and D2D Identity Management (IDM) including authentication mechanisms in InteropEHRate considering the regulations and the state-of-the-art mechanism for interoperable eIDs focus on the objectives of the first year.

1.1. Scope of the document

The main goal of the present document is to describe the InteropEHRate specification of remote and D2D Identity Management (IDM) and authentication mechanisms focused on the objectives of the first year of the project the D2D scenario.

1.2. Intended audience

The document is intended to security engineers, policy makers, architects, developers and all the project participants and partners interested to have an overview of how the InteropEHRate support the Identity Management and authentication in the remote and D2D protocols.

1.3. Structure of the document

This deliverable is structured as follows. This Chapter explains the goal and structure of the document. In Chapter 2, we describe and review the research background regarding IDM and authentication, starting by a general overview and then focusing on other related European research initiatives. The overall IDM in terms of InteropEHRate is presented in Chapter 3, where it is analysed in detail for both the remote and the D2D protocols. Finally, Chapter 4 concludes the deliverable.

1.4. Updates with respect to previous version (if any)

Not applicable.

1.5. Relation to other deliverables

Similarly to other reports of the InteropEHRate project, this document present just a first draft of the specification of remote and D2D IDM mechanisms for HRs Interoperability. One more version of this

document is planned. The final version will be more detailed based on the new knowledge acquired from the experience of development during the first year. This first version of the deliverable D3.3 considers the work of WP2 regarding the architecture, user requirements and the interoperability profile and serve as a basis for the WP4 interoperability protocols. Figure 1 below presents the main relation with the other deliverables.

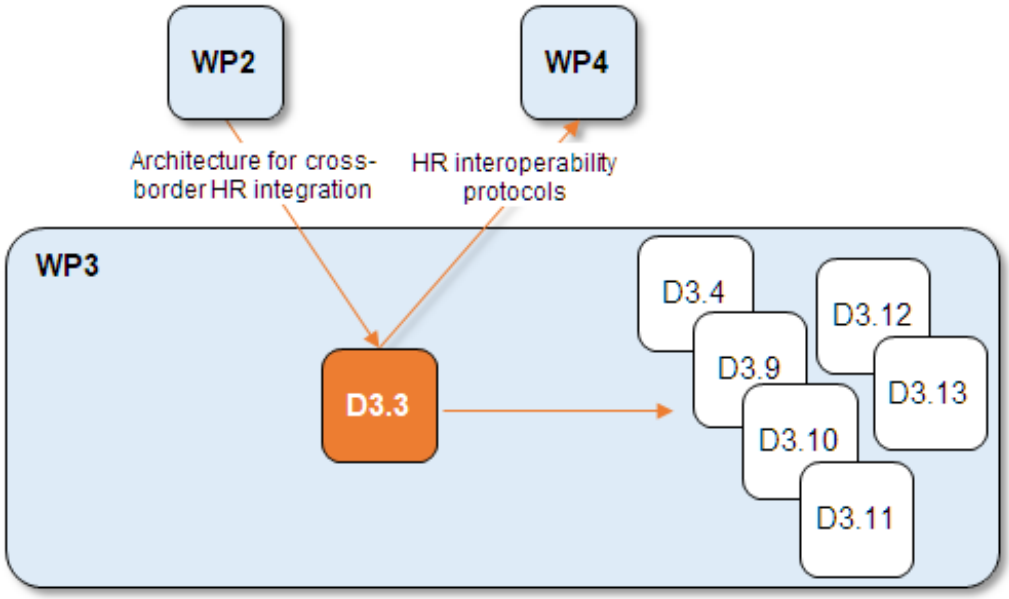


Figure 1 - Relation with other deliverables

2. TECHNICAL BACKGROUND

This chapter presents an understanding of the state-of-the-art concerning identity management (IDM) including authentication mechanisms for interoperability in healthcare domain, based on a review of the current literature. A detailed technical background is necessary to understand the area and move a step forward.

2.1. State of the art in IDM

A Digital Identity is the information used to represent an entity in an ICT system [IT2011]. An entity may be a person, an organization, a device, an application, etc. Electronic Identification provides the proper authentication strength for patients when seeking health care in a cooperating EU member state, as well as safeguarding their fundamental access rights.

Identity management (IDM) is the mechanism or objects used by entities to manage the claims about their digital identities. Working on identity management in the health area is not reduced to unique identification of citizens/patients, but also of healthcare professionals and health institutions. Personal health data are handled as explicit sensitive data and the definition and management of rights is essential for reaching a status which conforms to the legal systems in the member states [EU2009].

Authentication is a security mechanism that allows systems to identify the user as a registered user by providing information to prove the user is who he/she claims to be. There are several authentication mechanisms in the literature based on biometrics, usernames and passwords, certificates, tokens, etc. The most common mechanism is the combination of username and password. Some alternatives are HTTP-based authentication by using HTTP headers and other more modern approaches include two-factor authentication and password-less mechanisms. We will analyse the most important of them in the sub-section below.

The architecture of identity management (IDM) systems can be divided in two distinct categories according to [SCUDDER2010]:

- **Network Based IDM** - In this category, the attributes are stored at the identity provider and users authorize a request by the relying party to access to the attributes. The relying party can then access the data from the identity provider.
- **Claim Based IDM** - In this category, the attributes are stored at the user. A relying party can request the user to show the possession of these claims. The user can then use these claims to directly interact with the relying party, without any additional interaction with the issuer.

The most known identity management models are the isolated, centralized, and the federated model, which are briefly detailed below [CARRETERO2018]:

- **Isolated Model** - In this model the Service Provider (SP) and Identity Provider (IdP) are combined in a single server. However, this model is a very simple approach and may cause many problems [JOSANG2005].
- **Centralized Model** - This model consists of centralizing the identity storage while separating the services. Multiple SPs have to authenticate their users against a central IdP. The most extended implementation is the single sign-on (SSO) authentication method, which enables the user to access several SPs with a single identification instance. This model reduces the usability problems

derived from the Isolated Model, but has a clear reliability problem, as it depends on a single point of failure [CARRETERO2018].

- **Federated Model** - In this model, the parties involved in the identity management system establish an agreement on which entities are part of the system, how entities are going to be referred to, and the configuration parameters of the participating system parties. A Service Provider in one domain can grant authorized access to a resource it manages based on the exchange of identity, attribute, authentication and authorization assertions with an Identity Provider in another domain.

2.1.1. Identity Federations

There are several Identity Federations in the education, government and research sectors as well as the general public. According to [EMTG2017] the most known hybrid Identity Federations in the literature are the STORK and eIDAS. Identity federations are based on the establishment of trust agreements between organizations. Thus, any user in the federation will be able to access resources and services of any federated organization based on a unique digital identity, which is common to the whole federation. This federated identity, has two benefits a) simplifies the credential control by the user and b) the user management by service providers [EMTG2017]. Within identity federations, there are different types of entities that interact with end users the Service Providers and Identity Providers. In general, Identity Providers include the Authentication Provider and the Attribute Provider [EMTG2017].

eIDAS Regulation [EE2017] on electronic identification and trust services for electronic transactions in the European internal market is based on the work done along the STORK and STORK 2.0 projects and it has been designed as an evolution of both of them. The eIDAS Regulation was published in 2014 as a regulatory environment that guarantees people and services the use of their national eIDs to use public services in all European countries with the same legal reliability as traditional paper based procedures. eIDAS promotes and facilitates the use of cross-border electronic identification and trust services, and guarantees transparency and accountability. The objective is to extend and popularize the use of eID among citizens of the European Union in their relations with institutions as well as in the private sector.

2.1.2. Electronic Identification (eID) and the eHealth domain

eIDAS defines citizens as persons and organisations that seek online services from any EU member state using their domestic eID with assured security, cost- and time-efficiencies, and usability [KENNEDY2016]. eIDAS eID consolidates the independent national eID schemes by streamlining their output through Nodes and Connectors. The proprietary national input is mapped and conditioned through the eIDAS Node in Country-A to an interoperable transport form, the eIDAS SAML Assertion. Such assertions can be requested during an authentication request by a Service Provider (SP) through an eIDAS Connector in Country-B [ESENS2017].

A typical eID ecosystem comprises of [KENNEDY2016]:

- **member states**,
- **node operators** or connection points,
- **attribute and identity providers** that provide information related to electronic identities and that verify user identities,
- **service providers** that offer online services whose access is authenticated through eID, and
- **citizens**.

Electronic identification within the eHealth domain is motivated by two primary goals a) **patient safety** and b) **protection against illegitimate disclosure of medical data**, while it is identified as a difficult task. The work in [KATEHAKIS2017] already summarizes the main difficulties for eID, such as technical issues that hinder an efficient eID due to the need of hardware devices and middleware, the inability to reach foreign security services, or the inability to deploy non-certified software onto highly regulated medical systems are some of the identified issues and the maintenance of the legitimate-use of a national eID means across borders. The use of cross-border authentication through the eIDAS Network provides a reliable, responsible and convenient manner for online-services to identify their users.

Important groundwork on the interoperability of EHRs was carried out in the framework of the project epSOS, and with the support of the EXPAND project paved the way to roll out of the eHealth Digital Service Infrastructure (eHDSI). Currently, the most important initiative for interoperability at European level is eHealth Digital Service Infrastructure (eHDSI or eHealth DSI), that offers initial services for cross-border health data exchange under the Connecting Europe Facility (CEF). These services are limited to the exchange of four kinds of documents namely ePrescription and eDispensation, Patient Summary, European Reference Networks, and Patient Registries.

2.1.3. eIDAS Infrastructure

The eIDAS interoperability framework comprises two different authentication models [BERBECARU2019].

- In the **proxy model**, each country adhering to this model has to run a single national bridge called eIDAS node. This element is actually composed of two logical subcomponents:
  - an eIDAS-Proxy-Service (eIDAS Proxy), which is in charge of communicating with the National eID scheme to which the citizen will be authenticated; and
  - one eIDAS-Connector (Connector), which is in charge of communicating with the national SPs.
- The **middleware model** (adopted by Germany) does not exploit a national bridge:
  - the eIDAS Connector (in the other countries) communicates with a country-specific Middleware-Service (MW) to allow SPs to provide eIDAS-enabled services to German citizens. Citizen authentication is delegated from an SP to its national Connector, which acts as a gateway and subsequently forwards the authentication request to the eIDAS Proxy of the country selected by the citizen (or to the MW). The authentication request is further handled by the eIDAS Proxy according to Member State (MS)-specific approach.

Most countries follow the traditional approach, in which a new authentication request is constructed by the eIDAS Proxy and is sent (through the user's browser) to the national IdP (part of the National eID scheme). At the IdP, the citizen is asked to authenticate with a national eID. If this operation completes successfully, an authentication response containing also the eIDAS attributes that have been requested are returned through the eIDAS infrastructure back to the requesting SP.

Each eIDAS node has a Specific part used to communicate with the national SPs and IdPs and a Generic part used to communicate with the other eIDAS nodes via the eIDAS communication protocol, which is based on SAML 2.0 WebSSO Profile to transfer authentication data and eIDAS attributes between the eIDAS nodes.

According to the eIDAS specification, the eIDAS nodes may exchange only a restricted set of personal attributes, named eIDAS minimum data set (MDS) for natural persons, containing the person's current **family name**(s), the current **first name**(s), the **date and place of birth**, an **eIDAS unique identifier**, the current **address**, and the **gender** of a person. The attributes are either mandatory or optional.

2.1.4. Identity and Authentication Standards

In general, there are two types of standards: "build it and they will come" standards, and "let's work together so we don't all do something different" standards. The most successful standards typically fall into the latter category. Below we cover the most known and used identity and authentication standards. A timeline of the standards is also presented in Figure 2 below, in order to have a more holistic view.

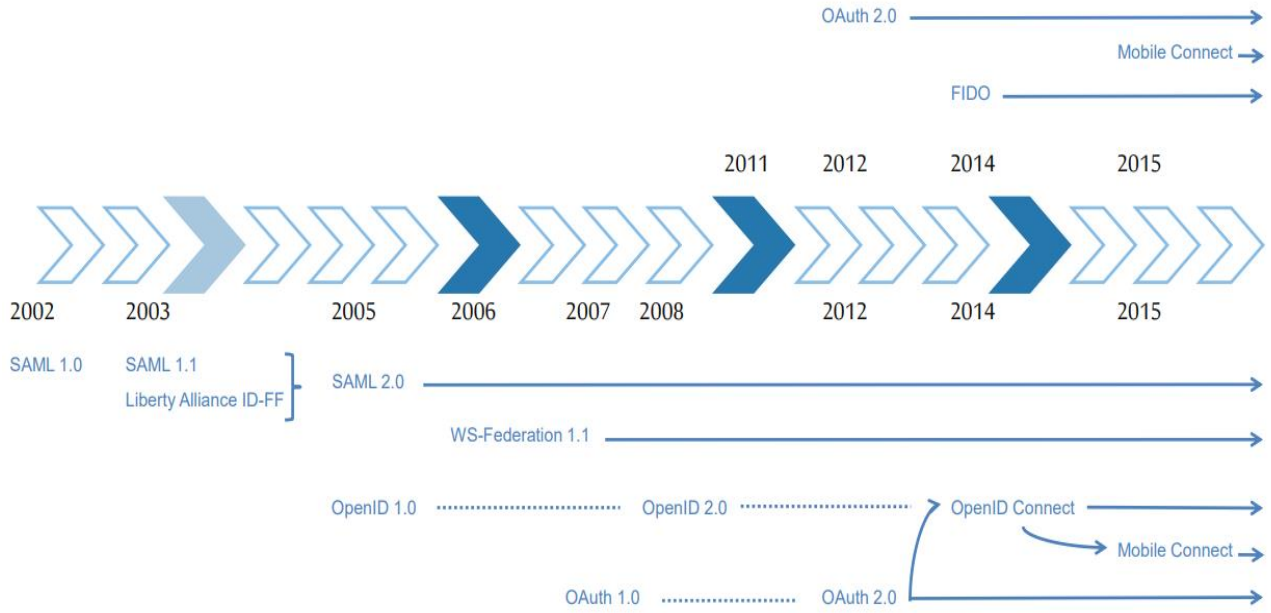


Figure 2 - Standards Timeline

SAML

Security Assertion Mark-up Language (SAML) is an XML-based open standard for exchanging authentication and authorization data between parties, in particular, between an identity provider and a service provider. This security information is expressed in the form of portable SAML assertions that applications working across security domain boundaries can trust. The OASIS SAML standard defines precise syntax and rules for requesting, creating, communicating, and using these SAML assertions. The four main components of the standard are the Assertions, Protocols, Bindings and Profiles.

SAML is one of the most important web-based federated identity standards. It's the most widely supported standard by SaaS providers who want to accept credentials from large enterprise customers. Like most other federated identity standards, it is based on redirecting a person's browser to a website maintained by their home organization. Assuming the website is trusted, the home organization then returns information about the person to the original website [SAML2005]. In this standard, the Identity providers pass identity information to service providers through digitally signed XML documents.

The SAML specification defines three roles: a) the principal (e.g. user), the identity provider (IdP), and the service provider (SP). In the primary use case addressed by SAML, the principal requests a service from the service provider. The service provider requests and obtains an authentication assertion from the identity provider. On the basis of this assertion, the service provider can make an access control decision, that is, it can decide whether to perform the service for the connected principal. In SAML, one identity provider may provide SAML assertions to many service providers. Similarly, one SP may rely on and trust assertions from many independent IdPs. The SAML Web Browser SSO profile was specified and standardized to promote interoperability.

A "SAML assertion" is a statement written in XML and issued by an "identity provider" about a "subject" (person) for a "relying party" (the recipient of the assertion) who is normally a "service provider" (website). Identity provider is abbreviated simply as "IDP" and service provider as "SP". SAML is a mature standard, and it's been successfully deployed to solve many business challenges. SAML uses public key cryptography to sign or encrypt messages and documents. The use of such keys enables the parties to protect and verify the integrity of information [SAML2].

Assertions contain the information that a web application needs from the Identity Provider about the person accessing the site. A typical example of a SAML assertion presented below.

```
<saml:Assertion
   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   ID="b07b804c-7c29-ea16-7300-4f3d6f7928ac"
   Version="2.0"
   IssueInstant="2004-12-05T09:22:05Z">
   <saml:Issuer>https://idp.example.org/SAML2</saml:Issuer>
   <ds:Signature
     xmlns:ds="http://www.w3.org/2000/09/xmldsig#">...</ds:Signature>
   <saml:Subject>
     <saml:NameID
       Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">
       3f7b3dcf-1674-4ecd-92c8-1544f346baf8
     </saml:NameID>
     <saml:SubjectConfirmation
       Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
       <saml:SubjectConfirmationData
         InResponseTo="aaf23196-1773-2113-474a-fe114412ab72"
         Recipient="https://sp.example.com/SAML2/SSO/POST"
         NotOnOrAfter="2004-12-05T09:27:05Z"/>
     </saml:SubjectConfirmation>
   </saml:Subject>
   <saml:Conditions
     NotBefore="2004-12-05T09:17:05Z"
     NotOnOrAfter="2004-12-05T09:27:05Z">
     <saml:AudienceRestriction>
       <saml:Audience>https://sp.example.com/SAML2</saml:Audience>
     </saml:AudienceRestriction>
   </saml:Conditions>
```

```
    <saml:AuthnStatement
      AuthnInstant="2004-12-05T09:22:00Z"
      SessionIndex="b07b804c-7c29-ea16-7300-4f3d6f7928ac">
      <saml:AuthnContext>
        <saml:AuthnContextClassRef>

urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
        </saml:AuthnContextClassRef>
      </saml:AuthnContext>
    </saml:AuthnStatement>
    <saml:AttributeStatement>
      <saml:Attribute
        xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
        x500:Encoding="LDAP"
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
        Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
        FriendlyName="eduPersonAffiliation">
        <saml:AttributeValue
          xsi:type="xs:string">member</saml:AttributeValue>
        <saml:AttributeValue
          xsi:type="xs:string">staff</saml:AttributeValue>
      </saml:Attribute>
    </saml:AttributeStatement>
  </saml:Assertion>
```

SAML Architecture

The core SAML specification defines the structure and content of both a) assertions and b) protocol messages used to transfer this information [SAML2005].

- SAML assertions carry statements about a principal that an asserting party claims to be true. The valid structure and contents of an assertion are defined by the SAML assertion XML schema.
- SAML protocol messages are used to make the SAML-defined requests and return appropriate responses. The structure and contents of these messages are defined by the SAML-defined protocol XML schema.

SAML profiles are defined to satisfy a particular business use case, for example the Web Browser SSO profile. Profiles typically define constraints on the contents of SAML assertions, protocols, and bindings in order to solve the business use case in an interoperable fashion. There are also Attribute Profiles, which do not refer to any protocol messages and bindings, that define how to exchange attribute information using assertions in ways that align with a number of common usage environments. Figure 3 illustrates the relationship between these basic SAML concepts [SAML2005].

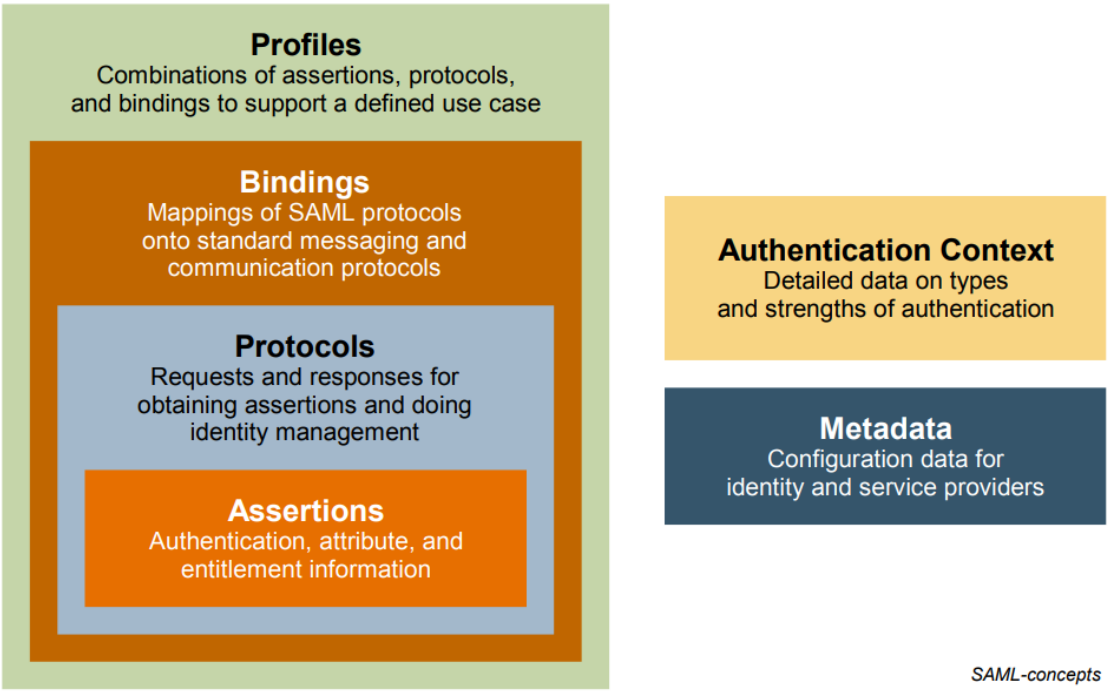


Figure 3 - Basic SAML Concepts

Two other SAML concepts are useful for building and deploying a SAML environment are a) Metadata and b) Authentication Context. Metadata defines a way to express and share configuration information between SAML parties, while a SAML authentication context is used in an assertion's authentication statement to carry information regarding the type and strength of authentication that a user employed when they authenticated at an identity provider.

The Liberty Alliance Project released frameworks for federation, identity assurance, an Identity Governance Framework, and Identity Web Services. Liberty endorses SAML2 as its identity federation solution and provides interoperability and conformance testing.

WS-Federation

WS-Federation (Web Services Federation) is an Identity Federation specification, developed by a group of companies: BEA Systems, BMC Software, CA Inc. (along with Layer 7 Technologies now a part of CA Inc.), IBM, Microsoft, Novell, HP Enterprise, and VeriSign. Part of the larger Web Services Security framework, WS-Federation defines mechanisms for allowing different security realms to broker information on identities, identity attributes and authentication [WSFEDERATION].

WS-Security, WS-Trust, and WS-SecurityPolicy provide a basic model for federation between Identity Providers and Relying Parties. These specifications define mechanisms for codifying claims (assertions) about a requestor as security tokens which can be used to protect and authorize web services requests in accordance with policy. WS-Federation extends this foundation by describing how the claim transformation model inherent in security token exchanges can enable richer trust relationships and advanced federation of services. This enables high value scenarios where authorized access to resources managed in one realm can be provided to security principals whose identities and attributes are managed in other realms. WS-Federation includes mechanisms for brokering of identity, attribute discovery and retrieval, authentication

and authorization claims between federation partners, and protecting the privacy of these claims across organizational boundaries.

A federation is a collection of realms (security domains) that have established relationships for securely sharing resources. A Resource Provider in one realm can provide authorized access to a resource it manages based on claims about a principal (such as identity or other distinguishing attributes) that are asserted by an Identity Provider (or any Security Token Service) in another realm.

The value of establishing a federation is to facilitate the use of security principal attributes across trust boundaries to establish a federation context for that principal. A Relying Party can then use this context to grant/deny access to a resource. Establishing a federation context when Identity and Resource Providers operate in different realms requires agreement between these parties on what claims are required and frequently requires agreement on mechanisms for securely transporting those claims over unprotected networks. This provides the basis for interoperability. In general it is necessary for participants in a federation to communicate these requirements over a wide variety of trust and communication topologies. Supporting different topologies requires the exchange of metadata describing endpoint references where services may be obtained, plus the potential security policies and communication requirements that must be observed when accessing those endpoints. The exchange of this metadata can be further complicated because the participants in a single federation may have different policies and service providers may participate in multiple federations.

OAuth 2.0

OAuth was introduced to allow a user to grant access to private resources connected to their identity and is a standard for authorization and a set of defined process flows for “delegated authorization”. OAuth 2.0 is a specification as to how to issue access tokens. It is defined in RFC 6749 (The OAuth 2.0 Authorization Framework) [RFC6749]. This is done without sharing private identity details or passwords between services. A long-lasting access token is specified by the protocol, which can be used by entities for continued access to user resources [OAUTH2010].

OAuth is distinct from OpenID, as although it shares the common architecture of redirection for obtaining authorization, it only manages the access control of resources. OAuth and its updated standard OAuth 2.0 are both still in active use by many social networks and dependent applications. It uses JSON as the data format, and RESTful APIs to enable a person (or organization) to authorize access to resources. OAuth is a delegated authorization protocol, not an authentication protocol. OAuth is used in a wide variety of applications, including providing mechanisms for user authentication. This has led many developers and API providers to incorrectly conclude that OAuth is itself an authentication protocol and to mistakenly use it as such.

OAuth 2.0 can be extended by implementing custom grant types and/or token types. Some of the profiles built on top of the core framework are:

- SAML 2.0 Bearer Assertion Profile - used to exchange SAML assertions for access tokens
- User Managed Access Profile - enables the resource owner to define and manage multiple access policies for his protected resources in a single place
- Chain Grant Type Profile - enables a resource service to use the received access token

- Token Introspection Profile - allows clients to request metadata regarding a token
- Token Revocation Profile - used to revoke a token
- Dynamic Client Registration Profile - allows clients to register with an authorization server and retrieve their client ID and secret dynamically

OAuth Protocol Flow

The abstract OAuth 2.0 flow illustrated in the Figure below, which describes the interaction between the four roles and includes the following steps [RCF6749]:

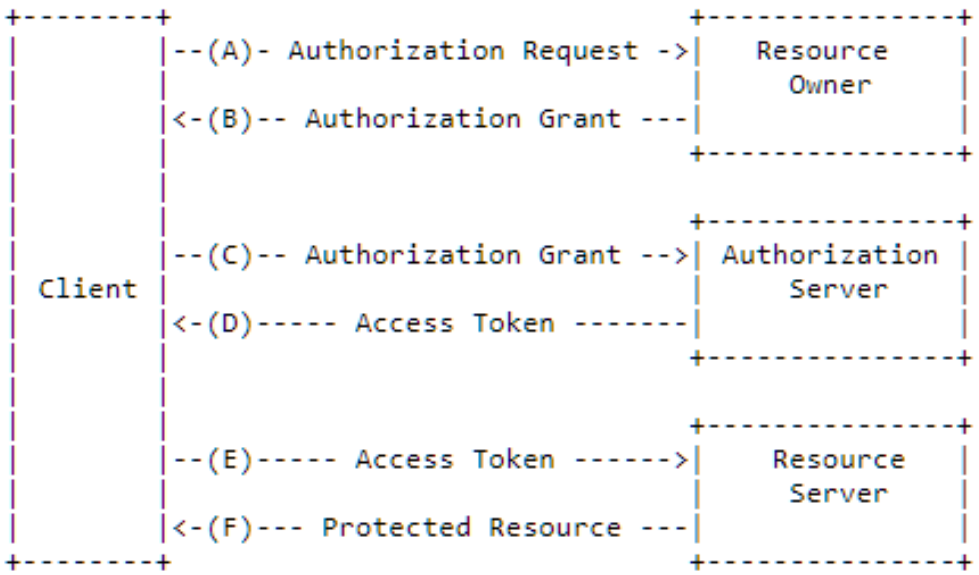


Figure 4 - Abstract Protocol Flow

A. The client requests authorization from the resource owner.
B. The client receives an authorization grant, which is a credential representing the resource owner's authorization, expressed using one of four grant types defined in this specification or using an extension grant type.
C. The client requests an access token by authenticating with the authorization server and presenting the authorization grant.
D. The authorization server authenticates the client and validates the authorization grant, and if valid, issues an access token.
E. The client requests the protected resource from the resource server and authenticates by presenting the access token.
F. The resource server validates the access token, and if valid, serves the request.

OpenID / OpenID Connect (OIDC)

OpenID is an open standard for authentication, promoted by the non-profit OpenID Foundation. There are over a billion OpenID-enabled accounts on the internet, and organizations such as Google, WordPress, Yahoo, and PayPal use OpenId to authenticate users. OpenID allows users to create an account with an identity provider that supports the standard, known as the OpenID Provider [FITZPATRICK2005]. A user

must obtain an OpenID account through an OpenID identity provider (e.g. Google). The user will then use that account to authenticate - sign into any website that accepts OpenID authentication, without managing multiple usernames and passwords. The OpenID standard provides a framework for the communication that must take place between the identity provider and the relying party.

In OpenID, authentication is delegated: server A wants to authenticate user U, but U's credentials (e.g. U's name and password) are sent to another server, B, that A trusts (at least, trusts for authenticating users). Indeed, server B makes sure that U is indeed U, and then tells to A: "ok, that's the genuine U". Basically, OpenID is about verifying a person's identity. OpenID removes the requirement for remembering passwords across many sites, but still leaves the users trusting their OpenID identity provider with important data. This inherent centralisation, coupled with the fact that users are forced to rely on an abstract identity system, eventually caused OpenID to lose prominence on the web [OPENID2011].

The latest version of OpenID is OpenID Connect, which combines OpenID authentication and OAuth2 authorization. OpenID Connect combines the features of OpenID 2.0, OpenID Attribute Exchange 1.0, and OAuth 2.0 in a single protocol. It allows an application to use authority. a) to verify the end user's identity, b) to fetch the end user's profile info, and c) to gain limited access to the end user's stuff. Is an open standard for authentication and a set of defined process flows for “federated authentication”. OpenID Connect implements an authentication layer on top of the OAuth 2.0 protocol and employs REST/JSON for messaging. It is a “profile” of OAuth 2.0 specifically designed for attribute release and authentication. It allows clients of all types, including Web-based, mobile, and JavaScript clients, to verify the identity of the end-user based on the authentication performed by an Authorization Server, as well as to request and receive information about authenticated session. OpenID Connect is a suite of lightweight specifications that provide a framework for identity interactions via REST like APIs. OpenID Connect Clients use the scope values as defined in OAuth 2.0 to specify what access privileges are requested for Access Tokens. The scopes associated with Access Tokens determine what resources will be available when they are used to access OAuth 2.0 protected endpoints [ALTICELABS2014]. OpenID Connect has many parallels to SAML. The equivalent of the SAML assertion is an id_token , a signed JSON Web Token, or JWT that contains very similar information.

The OpenID Connect specification defines three roles:

- The **end user** or the entity that is looking to verify its identity
- The **relying party** (RP), which is the entity looking to verify the identity of the end user
- The **OpenID Connect provider** (OP), which is the entity that registers the OpenID URL and can verify the end user’s identity

OpenID Connect Authorization Code Flow

The diagram below depicts at a high level the Authorization Code Flow.

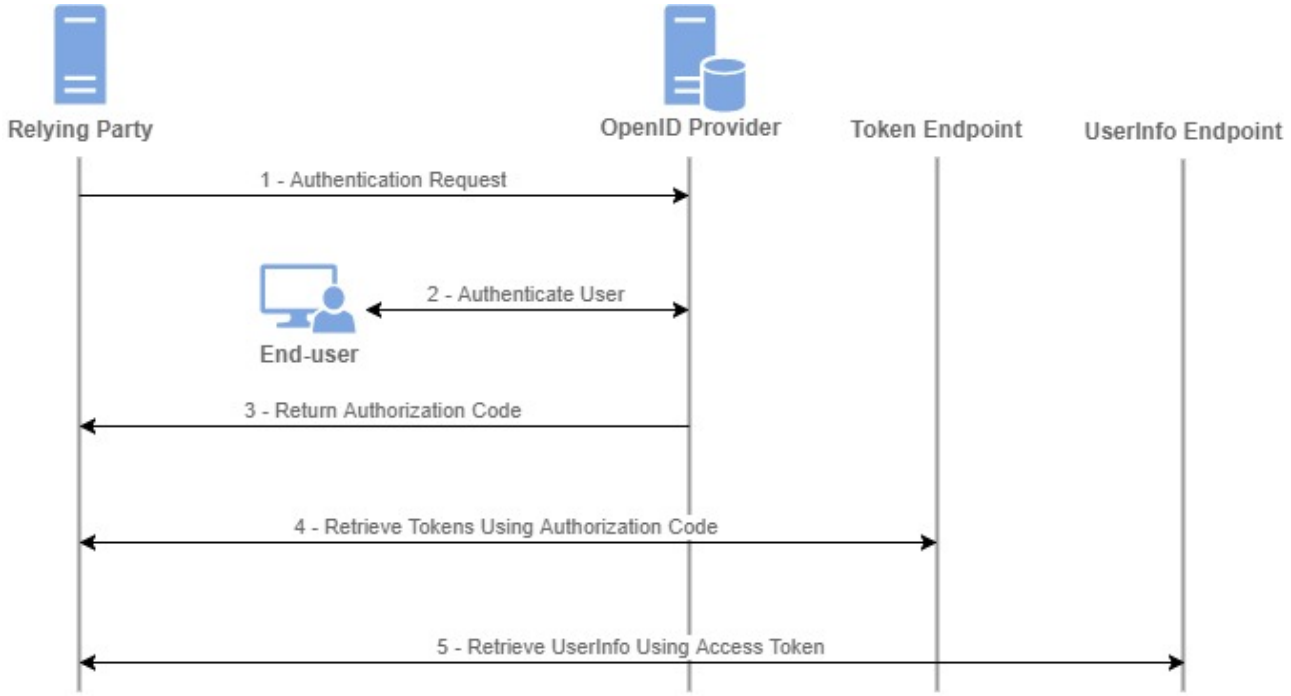


Figure 5 - High-level Authorization Code Flow

1. The Relying Party sends a request to the OpenID Provider to authenticate the End-User.
2. The OpenID Provider authenticates the end-user using one of the methods available to it and obtains authorization from the End-user to provide the requested scopes to the identified Relying Party.
3. Once the End-User has been authenticated and has authorized the request the OpenID Provider will return an authorization code to the Relying Party's server component.
4. The Relying Party's server component contacts the token endpoint and exchanges the authorization code for an id token identifying the end-user and optionally access and refresh tokens granting access to the userinfo endpoint.
5. Optionally the Relying Party may request the additional user information from the userInfo endpoint by presenting the access token obtained in the previous step.

FIDO Universal Authentication Framework (UAF)

The FIDO (Fast Identity Online) UAF strong authentication framework hosted by FIDO Alliance and enables online services and websites, whether on the open Internet or within enterprises, to transparently leverage native security features of end-user computing devices for strong user authentication and to reduce the problems associated with creating and remembering many online credentials [UAF2017]. More precisely, FIDO UAF supports a passwordless experience.

The user carries a device with a FIDO UAF stack installed. Users can then register their device to the online service by selecting a local authentication mechanism such as swiping a finger, looking at the camera, speaking into the mic, entering a PIN, etc. The FIDO UAF protocol allows the service to select which mechanisms are presented to the user. Once registered, the user simply repeats the local authentication action whenever they need to authenticate to the service. The user no longer needs to enter their

password when authenticating from that device. FIDO UAF also allows experiences that combine multiple authentication mechanisms such as fingerprint and PIN.

FIDO UAF High-Level Architecture

The FIDO UAF Architecture is designed to meet the FIDO goals and yield the desired ecosystem benefits [UAF2017]. It accomplishes this by filling in the status-quo's gaps using standardized protocols and APIs. The following Figure summarizes the reference architecture and how its components relate to typical user devices and Relying Parties.

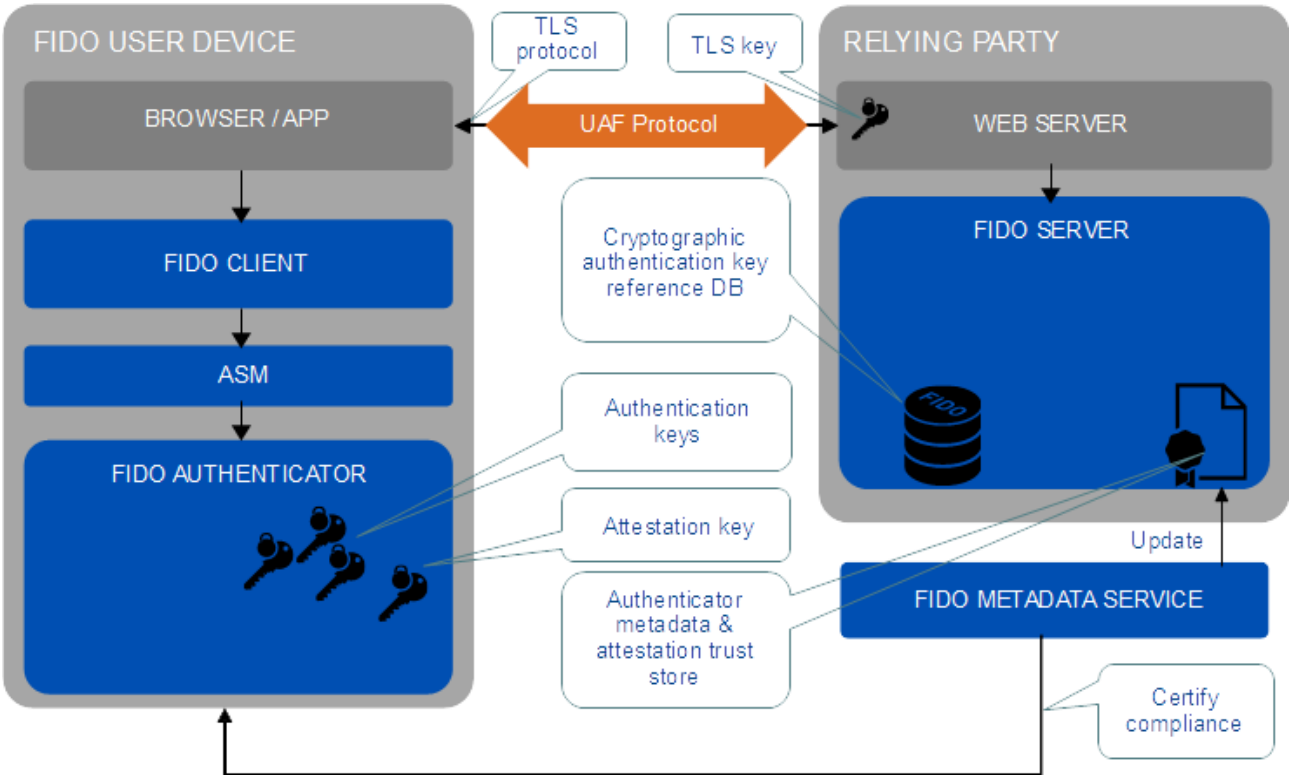


Figure 6 - FIDO UAF High-Level Architecture

A **FIDO UAF Client** implements the client side of the FIDO UAF protocols, and is responsible for a) interacting with specific FIDO UAF Authenticators using the FIDO UAF Authenticator Abstraction layer via the FIDO UAF Authenticator API and b) interacting with a user agent on the device (e.g. a mobile app, browser) using user agent-specific interfaces to communicate with the FIDO UAF Server.

A **FIDO UAF Server** implements the server side of the FIDO UAF protocols and is responsible for a) interacting with the Relying Party web server to communicate FIDO UAF protocol messages to a FIDO UAF Client via a device user agent, b) validating FIDO UAF authenticator attestations against the configured authenticator metadata to ensure only trusted authenticators are registered for use, c) manage the association of registered FIDO UAF Authenticators to user accounts at the Relying Party and d) evaluating user authentication and transaction confirmation responses to determine their validity.

A **FIDO UAF Authenticator** is a secure entity, connected to or housed within FIDO user devices, that can create key material associated to a Relying Party. The key can then be used to participate in FIDO UAF strong authentication protocols.

FIDO UAF Protocol Message Flows

The FIDO UAF Protocols carry FIDO UAF messages between user devices and Relying Parties. There are protocol messages addressing a) Authenticator Registration, b) User Authentication, c) Secure Transaction Confirmation and d) Authenticator Deregistration. Figure 7, below presents the UAF Authentication Sequence Diagram.

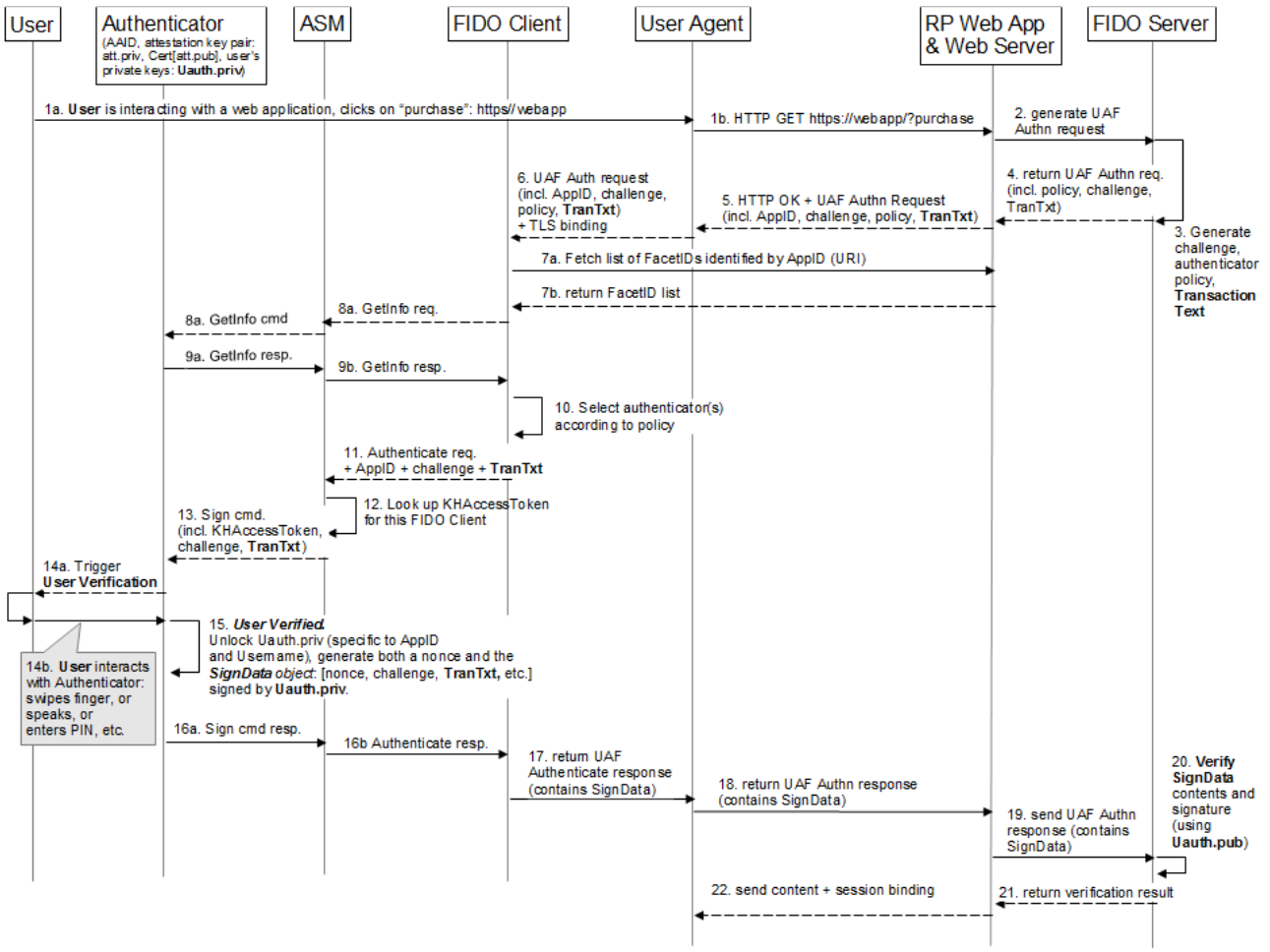


Figure 7 - UAF Authentication Sequence Diagram

This overall scenario will vary slightly depending upon the type of FIDO UAF Authenticator being employed. Some authenticators may sample biometric data such as a face image, fingerprint, or voice print. Others will require a PIN or local authenticator-specific passphrase entry. Still others may simply be a hardware bearer authenticator.

FIDO 2nd Factor Authentication (U2F)

Universal 2nd Factor (U2F) is an open authentication standard that strengthens and simplifies two-factor authentication (2FA) using specialized Universal Serial Bus (USB) or near-field communication (NFC) devices based on similar security technology found in smart cards. Initially the standard developed by Google and Yubico with contributions from NXP Semiconductors, but now hosted by the FIDO Alliance.

The FIDO U2F protocol enables relying parties to offer a strong cryptographic 2nd factor option for end user security. The relying party's dependence on passwords is reduced. The password can even be simplified to a four digit PIN. End users carry a single U2F device which works with any relying party supporting the protocol. The user gets the convenience of a single 'keychain' device and convenient security [U2F2017]. FIDO U2F allows online services to augment the security of their existing password infrastructure by adding a strong second factor to user login. The user logs in with a username and password as before. The service can also prompt the user to present a second factor device at any time it chooses. During registration and authentication, the user presents the second factor by simply pressing a button on a USB device or tapping over NFC or BLE. The user can use their FIDO U2F device across all online services that support the protocol leveraging built–in support in web browsers.

The U2F eco-system is designed to provide strong authentication for users on the web while preserving the user's privacy. The user carries a 'U2F device' as a second factor. When the user registers the U2F device at an account at a particular origin the device creates a new key pair usable only at that origin and gives the origin the public key to associate with the account. When the user authenticates to the origin, in addition to username and password, the origin can check whether the user has the U2F device by verifying a signature created by the device. The user is able to use the same device across multiple sites on the web - it thus serves as the user's physical web keychain with multiple (virtual) keys to various sites provisioned from one physical device. Using the open U2F standard, any origin will be able to use any browser (or OS) which has U2F support to talk to any U2F compliant device presented by the user to enable strong authentication [U2F2017].

The U2F device can be embodied in various form factors, such as standalone USB devices, standalone Near Field Communication (NFC) device, standalone Bluetooth Low Energy (BLE) devices, built-on-board the user's client machine/mobile device as pure software or utilizing secured crypto capabilities. It is strongly preferable to have hardware backed security, but it is not a requirement. However, as we shall see the protocol provides an attestation mechanism which allows the accepting online service or website to identify the class of device and either accept it or not depending on the particular site's policy.

Figure 8, below presents the s the basic process flow of U2F:

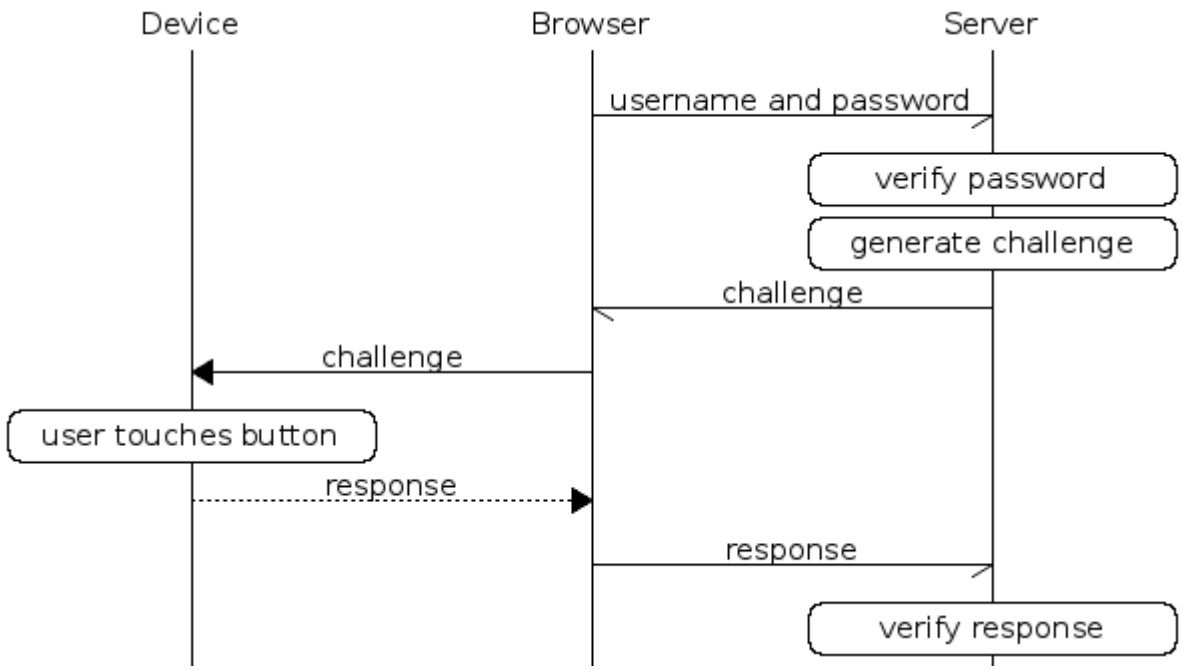


Figure 8 - U2F Basic Flow Diagram

U2F Registration

Below diagram shows the working of Registration with Attestation details.

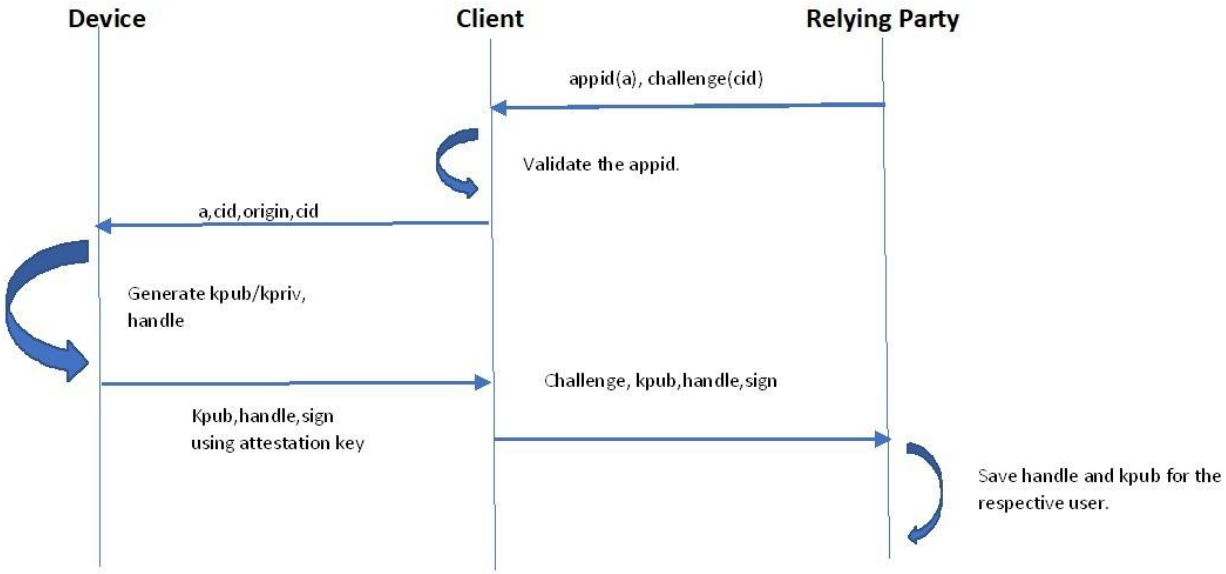


Figure 9 - U2F Registration

Each device class is provisioned with the attestation certificate (burnt into the device). The attestation public key can be found In the FIDO meta data server. The attestation private key is used to sign the registration response. Signing with the attestation private key gives a way to securely register the KPub/Key-handle with the relying party.

Flow of events:

- User clicks on Register, and the Relying party initiates the request with appid + challenge. Appid uniquely identifies the relying party and challenge is a unique random string.
- Client validates the appid before forwarding the request to device. If we look at the Google's reference code, the validation involves checking the origin.
- Device receives the request and generates a pair of Kpub/Kpriv and key handle per RP per user on successful user presence check.
- User presence check is done on yubikeys by tapping on the u2f keys when it blinks.
- The attestation private key is used to generate the digital signature of the response containing challenge, key handle, public key of the device etc.
- When the relying party receives the response validates the response signature using attestation public key. Attestation public key can be found in FIDO Metadata server.
- The relying party saves the key handle and Kpub for the respective user.

U2F Authentication

During the registration process we saw that the Kpub was registered with the Relying party. Now in the authentication flow we will see how the Kpub will be used.

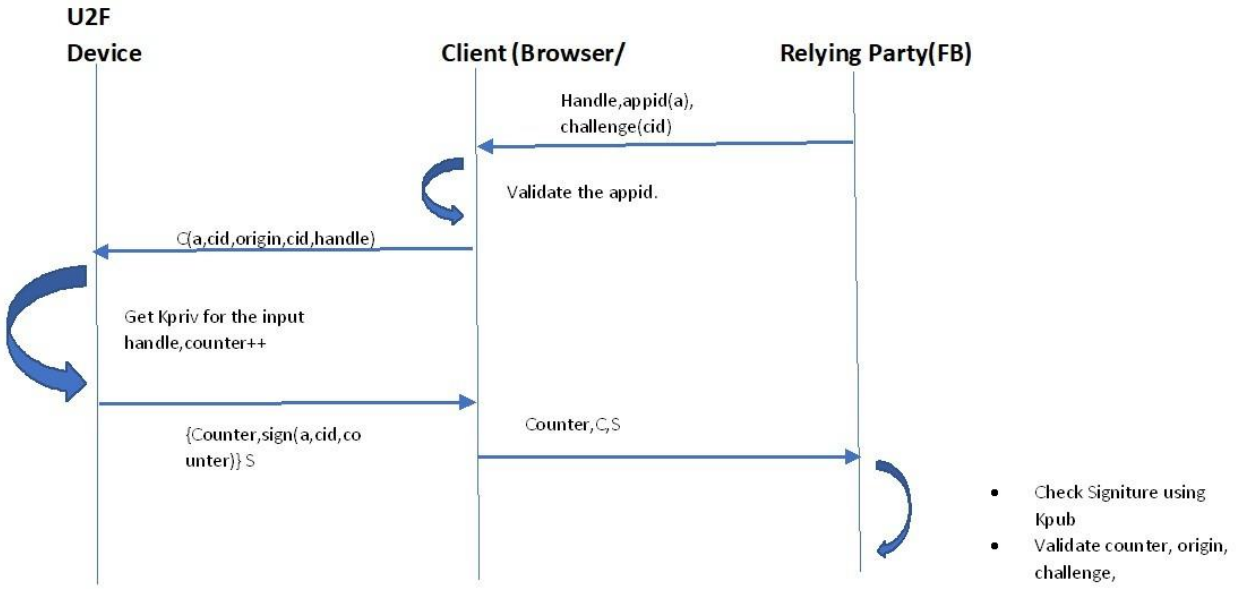


Figure 10 - U2F Authentication

Flow of events:

- User clicks on Login and enters the first factor. On successful authentication with the first factor; the second factor comes into picture.
- RP sends the Key handle/Challenge for the user trying to login to client.
- The client-side application validates the appid and forwards it to device.

- Device gets the Kpriv by looking at the key handle and increments the counter to mitigate replay attack on successful user presence check.
- Device sends back the response signed by KPriv.
- Relying party verifies the signature of the response using KPub the second factor and validates the counter, origin and challenge.

FIDO Web Authentication API (WebAuthN)

WebAuthn defines a standard web API that is being built into browsers and platforms to enable support for FIDO Authentication. Password-Free FIDO is a mechanism that allows users to log in into systems without the need of user and password credentials. FIDO was developed by companies that are part of the World Wide Web Consortium (WC3). The main idea is to improve the user experience and create a robust and secure mechanism for authenticating users in a system. The WebAuthn specification also defines a series of extensible points, such as the ability to add new attestation formats and the ability to add new extensions to the protocol and define their processing rules.

FIDO is using the pre-existent specifications: FIDO (Fast Identity Online), Universal 2nd Factor Authentication (U2F) and Universal Authentication Framework (UAF) for verifying user identities. To login into the system the user provides password information and as a second authentication mechanism biometrics or other mechanisms.

- **CTAP2** - allows the use of external authenticators (FIDO Security Keys, mobile devices) for authentication on FIDO2-enabled browsers and operating systems over USB, NFC, or BLE for a passwordless, second-factor or multi-factor authentication experience.
- **CTAP1** - The new name for FIDO U2F, CTAP1 allows the use of existing FIDO U2F devices (such as FIDO Security Keys) for authentication on FIDO2-enabled browsers and operating systems over USB, NFC, or BLE for a second-factor experience.

FIDO Authentication Flow

1. Initiate authentication with Relying Party
2. FIDO Server sends authentication challenge and preferences for the authenticators or credentials to be used
3. Authenticator performs user verification on device to signal the user's consent to authenticate with the service
4. The authenticator uses the service's origin to look up the private key for authentication and uses the private key to sign the challenge from the server. The server sends an authentication response: challenge + signature.
5. The server retrieves the public key for the user and validates the signature on the challenge.

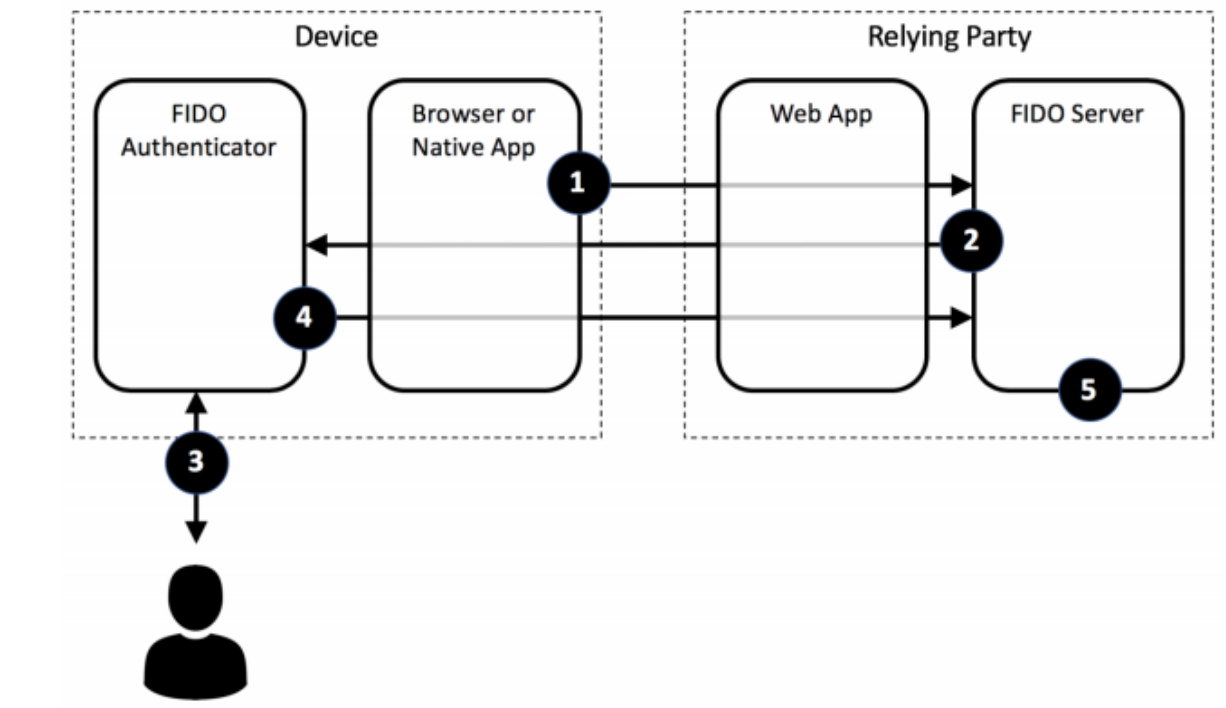


Figure 11 - FIDO Authentication Flow

Mobile Connect

Mobile Connect is the mobile operator-facilitated secure universal identity solution developed by the GSMA in collaboration with Mobile Operators. The GSMA represents the interests of mobile operators worldwide spanning more than 220 countries and unites nearly 800 of the world's mobile operators, as well as more than 230 companies in the broader mobile ecosystem. To-date there are more than 470 million active Mobile Connect users via over 70 operators covering more than 40 countries and reaching more than 3 billion people [MOBILECONNECT].

Mobile Connect is a portfolio of mobile-enabled services that can be integrated into a Service Provider's application to support access to services provided by the Service Provider. Mobile Connect provides strong customer authentication, authorisation, and permissioned access to a User's identity and contextual network attributes.

Mobile Connect uses a distributed architecture in which each Mobile Operator deploys Mobile Connect services for its particular user base, but with all deployments abiding by a strict set of technical standards to ensure that from a Service Provider's perspective, the experience of consuming Mobile Connect services from any of the Mobile Operators is consistent

Mobile Connect is based upon the OpenID Connect protocol. It allows Users to be identified by their MSISDN (or a related Pseudonymous Customer Reference) and to be authenticated securely via their mobile device with the SIM providing security. Mobile Connect defines two profiles of OIDC to support Device-Initiated and Server-Initiated requests for authentication, authorisation or permissioned access to User attributes.

The serving Mobile Operator supports and selects an appropriate Authenticator to present the authentication and authorisation requests to the User on their mobile device to which the User responds. The Authenticator may also be used to seek User consent for the serving Operator to share or validate User attributes with the Service Provider. The Authenticator is selected based on Operator policy, device capability and the Level of Assurance required.

Mobile Connect also meets the eIDAS technical specification and interoperability requirements for integration with national ID as designed by EU Member States eIDAS Nodes in collaboration with the European Commission CEF project. An example reference architecture of eIDAS for the integration with Mobile Connect is shown in the following Figure.

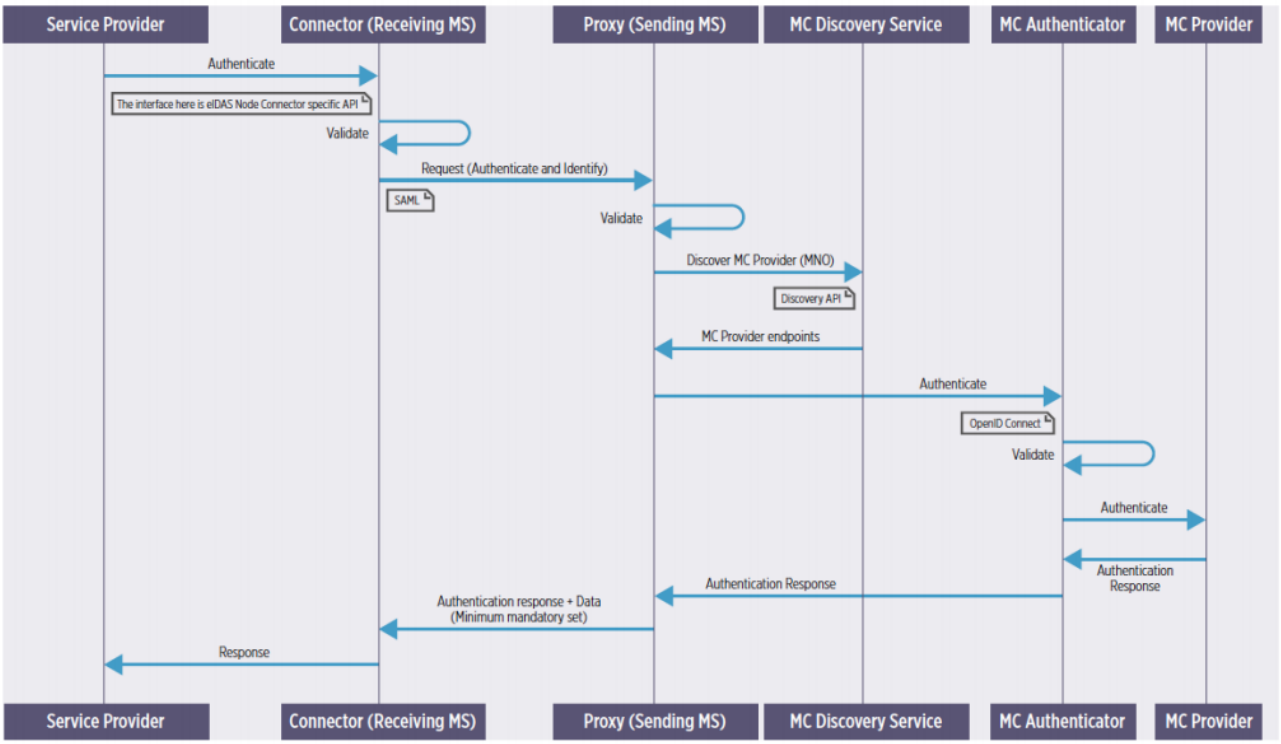


Figure 12 - Mobile Connect and eIDAS technical flow

2.2. Relation with other research projects

In this section we summarize some related research projects and Large Scale Pilots (LSP) that tried to solve the issue of interoperability and cross-border identification and authentication. The most known are listed below:

STORK

The goal of Secure idenTity acrOss boRders linked (STORK) project is to establish the cross-border recognition and authentication of e-IDs issued by other member states. The authorized use of e-IDs, secure access to work stations, and confidentiality, integrity and availability of personal data are the major challenges in this area. STORK aims at implementing an EU-wide interoperable system for recognition of

eID and authentication that will enable businesses, citizens and government employees to use their national electronic identities in any Member State. It pilot trans-border e-government identity services and learn from practice on how to roll out such services, and to experience what benefits and challenges an EU-wide interoperability system for recognition of eID will bring.

The STORK interoperable solution for electronic identity (eID) is based on a distributed architecture that will pave the way towards the full integration of EU e-services while taking account of specifications and infrastructures currently existing in EU Member States [STORK2010]. STORK 2.0 is a pilot based on the STORK project and carried out by 19 European Member States and 59 partners of different types, such as governmental institutions, banks or universities. The initiative was planned with the aim of being helpful in the preparation of the eIDAS regulation. It is based on SAMLSTORK, which uses SAML extension capabilities to introduce new attributes and custom information. These modifications and the security restrictions make STORK incompatible with other standard SAML federations.

epSOS / eHDSI

Smart Open Services for European Patients (epSOS) is the main European electronic Health (eHealth) interoperability project co-funded by the European Commission and the partners. It focuses on improving medical treatment of citizens while abroad by providing health professionals with the necessary patient data [EPSOS]. epSOS aims to change this by ensuring standards for the exchange of medical information, subject to patient consent. epSOS aims to design, build, and evaluate a service infrastructure that demonstrates cross-border interoperability between electronic health record systems in Europe.

In the epSOS project identity management is one of the essential tasks that are being addressed. The core principle of epSOS is to bridge existing national eHealth infrastructures instead of setting up a new, centralised European healthcare service network from scratch. epSOS is trying to find solutions which are compatible with the national regulations and concepts of the participating countries. In epSOS a patient is not an active user of the platform, the patient does not perform any authentication procedure and does not use any of the epSOS software (this is the main difference between epSOS and InteropEHRate), he simply presents his identity documents to the HCP. On the other hand, the attending epSOS health professional is authenticated within the health professional's home country [EPSOS] and uses epSOS Identification Service to discover a valid patient identifier from an ID assigning authority by providing identifiers and/or demographic data that are sufficient for patient identification. The implementation of the epSOS Identification Service is based on the HL7 V3 Identification Service standard (HL7 IS) and is an extension to the IHE profile XCPD "Cross-Community Patient Discovery" [IHE ITI TF-1].

Results of epSOS project have been used in its successor project called eHealth Digital Service Infrastructure (eHDSI or eHealth DSI). This project's objective is the initial deployment and operation of services for cross-border health data exchange (Patient Summary and Prescriptions) under the Connecting Europe Facility (CEF).

Despite InteropEHRate and epSOS/eHDSI projects are both focused on cross border health data exchange, the context in which they operate is very different, especially for what concerns authentication: in epSOS there is no authentication mechanism for the citizen, there is no app given to the citizen and the only user

that performs an electronic authentication is the HCP, but using proprietary authentication mechanism provided by his country.

e-SENS

e-SENS (Electronic Simple European Networked Services) aiming to consolidate and solidify the work done, to industrialize the solutions and to extend their potential to more and different domains [ESENS2017]. e-SENS focuses strongly on core building blocks such as eID, e-Documents, eDelivery, semantics and e-Signatures across the different LSP domains. The solutions will be based on already existing systems in Member States, and so no changes will be needed at national level.

e-SENS has been formed to consolidate and solidify the work done in previous LSP projects, and to extend these solutions to new domains. They will be tested for scalability and the ability to be reused in a number of domains. These building blocks aim to provide the foundation for the platform of “core services” for the eGovernment, cross-border, digital infrastructure foreseen in Regulation (EU) No 1316/ 2013 for establishing the Connecting Europe Facility (CEF).

3. INTEROPEHRATE IDENTITY MANAGEMENT

Identification in health domain refer to the necessity to establish the identity of patients and health professionals in the healthcare process. The purpose of this Chapter is the how InteropEHRate project will handle Identity Management and authentication mechanisms, regarding the two mechanisms : remote and D2D. As already described, the remote R2D protocol includes Internet connection, while in the D2D protocol, Internet is not available.

InteropEHRate D2D security protocol for IDM supports two variants. The first variant leverages existing ID-Card identification mechanisms for the citizen, QR code and digital signatures and the second utilises qualified digital signatures for eIDAS regulation compliance for both parties. In addition, both variants needs to be valid without the usage of Internet. However, when the S-EHR app has Internet connectivity, an extra verification for the validity of HCP and Health Organization certificates will be done by obtaining the revocation status per certificate using the Online Certificate Status Protocol (OCSP). Regarding authentication, self-registration and offline login is more suitable (for both variants) for InteropEHRate since in the D2D scenario no internet is assumed.

InteropEHRate remote R2D security protocol for IDM leverage existing standards established by the EPSOS project and related infrastructure eHDSI, and regulations like eIDAS and related EU services like CEF eID. Healthcare Organization Information System and Research Centre Information System uses the Service Providers interface (SPsI) exposed by the eIDAS node (for cross border identification of the citizens). Through this interface, the Healthcare Organization Information System and the Research Centre Information System send authentication requests to the eIDAS-Node and receives the authentication responses. However, the procedure of user authentication takes place between the user and the Identity Provider, is outside of eIDAS Network and InteropEHRate system.

We provide an overview of how the different actors and organizations involved in the InteropEHRate architecture interacts with each other in the context of the two security protocols. The security protocols, aim to guarantee the cross-border identification of the citizens and the privacy, integrity, and trustability of data exchange and set of functional APIs for performing specific health data exchange operations.

3.1. IDM in D2D Protocol

As analytically described in D4.1 [D4.1], D2D protocol allows the exchange of health data between a S-EHR and a near HCP App without the usage of Internet. Regarding the IDM in D2D protocol we decided to provide two variants:

1. In the first variant, the identification will be done with the **ID-Card** of the citizen and a **QR code** generated by the hospital that includes software signatures of the HCP and the Health Organization. This variant is more feasible, to be used immediately after the end of the project.
2. In the second variant, the identification will be used in the future and will be done with hardware signatures (**Qualified**) of the citizen, the HCP and the Health Organization. This variant will be also used for experimentation reasons, during the duration of the project.

This section will list and explain through Sequence Diagrams all the interactions between the S-EHR app and the HCP app (i.e. abstract diagram) regarding IDM for Health Organization, HCP and Citizen Identification in the two aforementioned variants.

3.1.1. Conceptual D2D IDM 1st Variant

The following sequence diagram provides a high-level overview of the first variant of the conceptual D2D Identity Management. As already aforementioned, citizens' ID-card is necessary for the identification, since the lack of the Internet restricts us to leverage existing EU services like CEF eID. In the D2D protocol the two main actors are the S-EHR mobile app and the HCP Terminal app. A CA is also necessary only the first time, in order the HCP and the Health Organization to acquire the necessary credentials.

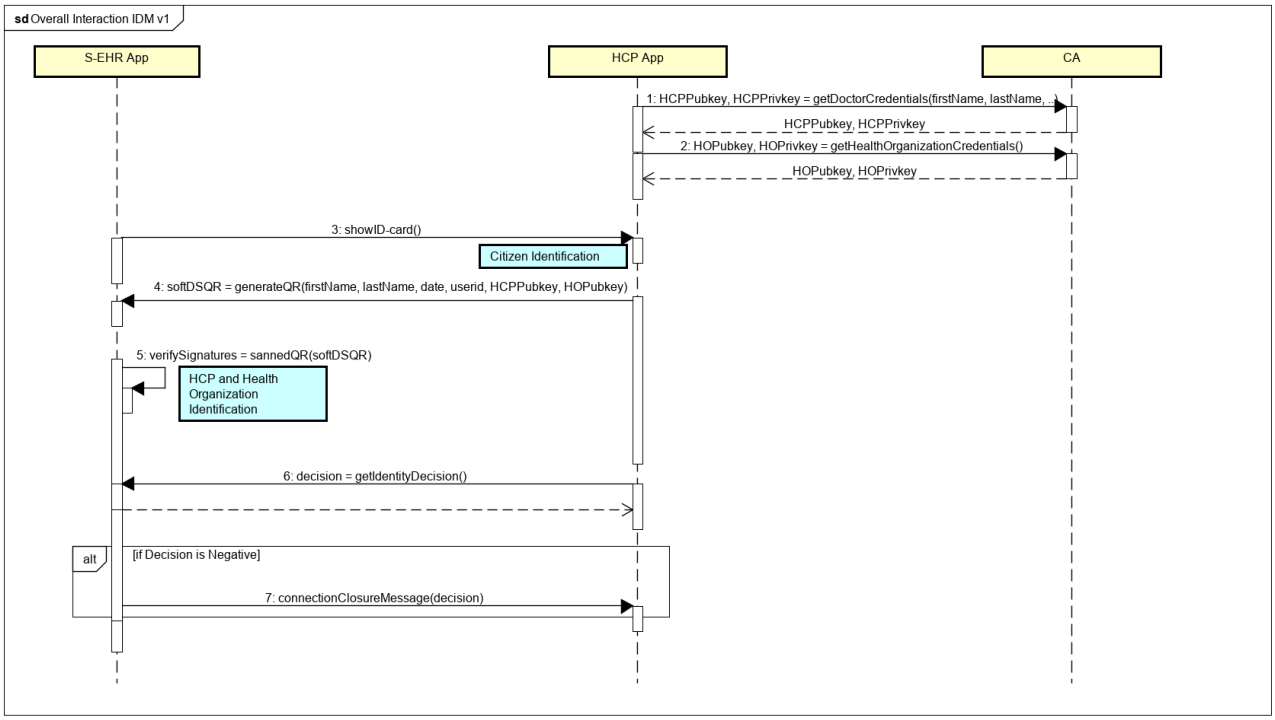


Figure 13 - D2D Identification and authentication (1st variant)

Following a detailed description of the sequence diagram:

- **Step 1** - HCP acquires the first time his credential (public and private key) from a CA. HCP provides his first name, last name, address and other optional information and receives his certificate that contains his public key (HCPPub) and his private key (HCPPriv). In this variant we will not examine qualified certificates and there is no need to use an extra HSM device to access these credentials for digital signing purposes.
- **Step 2** - Health Organization obtains the first time his credential (public and private key) from a CA. Health Organization provides the name of the organization and other optional information of the organization and receives organization's certificate that contains his public key (HOPub) and his private key (HOPriv). In this variant we will not examine qualified certificates and there is no need to use an extra HSM device to access these credentials for digital signing purposes.
- **Step 3** - According to the D2D scenario, the citizen show his ID-card to the HCP for his identification, since there is no Internet and he is face to face with the HCP. The HCP is able to identify the citizen based on the citizen's personal photograph on the ID-card.

- **Step 4** - The HCP generates a QR code, which contains HCP's first name, HCP's last name, the date, the user id (HCP id), the public keys of the HCP (HCPPub) and the public key of the Health Organization (HOPub). The QR code is double digital signed, first from the HCP and second from the Health Organization, in order the citizen to be able to verify both identities. In this variant, the digital signatures are created by software.
- **Step 5** - The citizen with S-EHR App, scans the QR code and receives all the information that contains included the public keys of the HCP and Health Organization and the corresponding double digital signature. The S-EHR app will be able to validate the double digital signature based on the acquired public keys and identify the HCP and the Health Organization. The S-EHR, in the future and when will be connected to the Internet, will have the ability (optionally) to validate the acquired certificates obtaining the revocation status by utilizing the OCSP protocol.
- **Step 6** - The HCP App, request for the identification decision, in order to proceed further with the next protocol steps.
- **Step 7** - If the validation of the double signature fails and one of the identities is not valid, the citizen send through his S-EHR app negative decision regarding the identification.

3.1.2. Conceptual D2D IDM 2nd Variant

In this variant we will examine the usage of Qualified Certificates (QC) in order to be bound to legal constraints. The most important feature of qualified certificates is they are subject to direct regulatory oversight. A Qualified Signature Creation Device (QSCD) is cryptographic hardware, such as a hardware security module (HSM), that passed the certification process under the eIDAS regulation. HSMs are recognised as QSCD under Article 51 (Transitional Measures) of the eIDAS Regulation. Each EU Member State is responsible for publishing a list of TSPs that its national supervisory scheme has recognised as Qualified, either under the eIDAS Regulation or the earlier Directive.

The eIDAS Regulation defines two types of electronic signatures: a) advanced electronic signatures and b) qualified electronic signatures. Both advanced and qualified signatures are based on digital signature technology, however an advanced electronic signature is less stringent than a qualified electronic signatures.
An advanced electronic signature, at a minimum, must be:

- Uniquely linked to the signatory.
- Capable of identifying the signatory.
- Created in a way that ensures the signatory can maintain sole control.
- Linked to the data it relates to in such a manner that any subsequent change to the data is detectable.

A qualified electronic signature must meet all the requirements for an advanced signature, and must also be supported by the following components:

- A qualified signature creation device, such as a smart card or HSM, which is certified by Common Criteria and meets the requirements of the eIDAS Regulation.

A qualified electronic signature is assumed to have at least the legal equivalence of a handwritten signature. Under the eIDAS Regulation, several kinds of trusted services requires HSMs in order for the TSP to become qualified. An HSM is a major purchase, so choosing the right HSM will pay off many times over.

There are important features to consider when selecting an HSM solution, such as certification, flexibility and future-proofing. A list of certified devices is published by the commission at [QUALIFIED DEVICES]. It should be possible to use the same HSM for signing, time-stamping, sealing as well as other security needs such as data privacy. Flexibility also means the ability to extend the use of HSM to support additional functions such as user authentication.

This variant selection with Qualified Signature is made to be our solution eIDAS compliant. However, the usage of Qualified Certificates, as aforementioned, implies two things that we summarize below for completeness reasons.

- Every natural person that uses the S-EHR and HCP Apps must issue a certificate from a Trusted Service Provider (TSP) which is accepted by the European Union Qualified TSPs. The list of the accepted providers are maintained in a repository called European Trust Lists (ETL). ETL is centrally maintained and each application can store this list locally in order to perform verification of digital certificates offline. This can be done during the installation of the S-EHR App.
- Certificates should meet a minimum quality characteristics during their generation (hashing algorithm, key size etc.) and storage/usage. This means that Qualified Certificates must be stored in Hardware Security Modules (HSMs) which means that the certificate per se cannot exist with the mobile phone or desktop computer. This means that the certificate must exist in Smart Card or a USB Token which complies with the PKCS11 standard [PKCS11]. The most state-of-the-art solution for that is the HSM to be in the form factor of a smart card and users to have a bluetooth reader that interconnects the smart card with the mobile (using PKCS11 abstraction).

The following sequence diagram provides a high-level overview of the second variant of the conceptual D2D Identity Management.

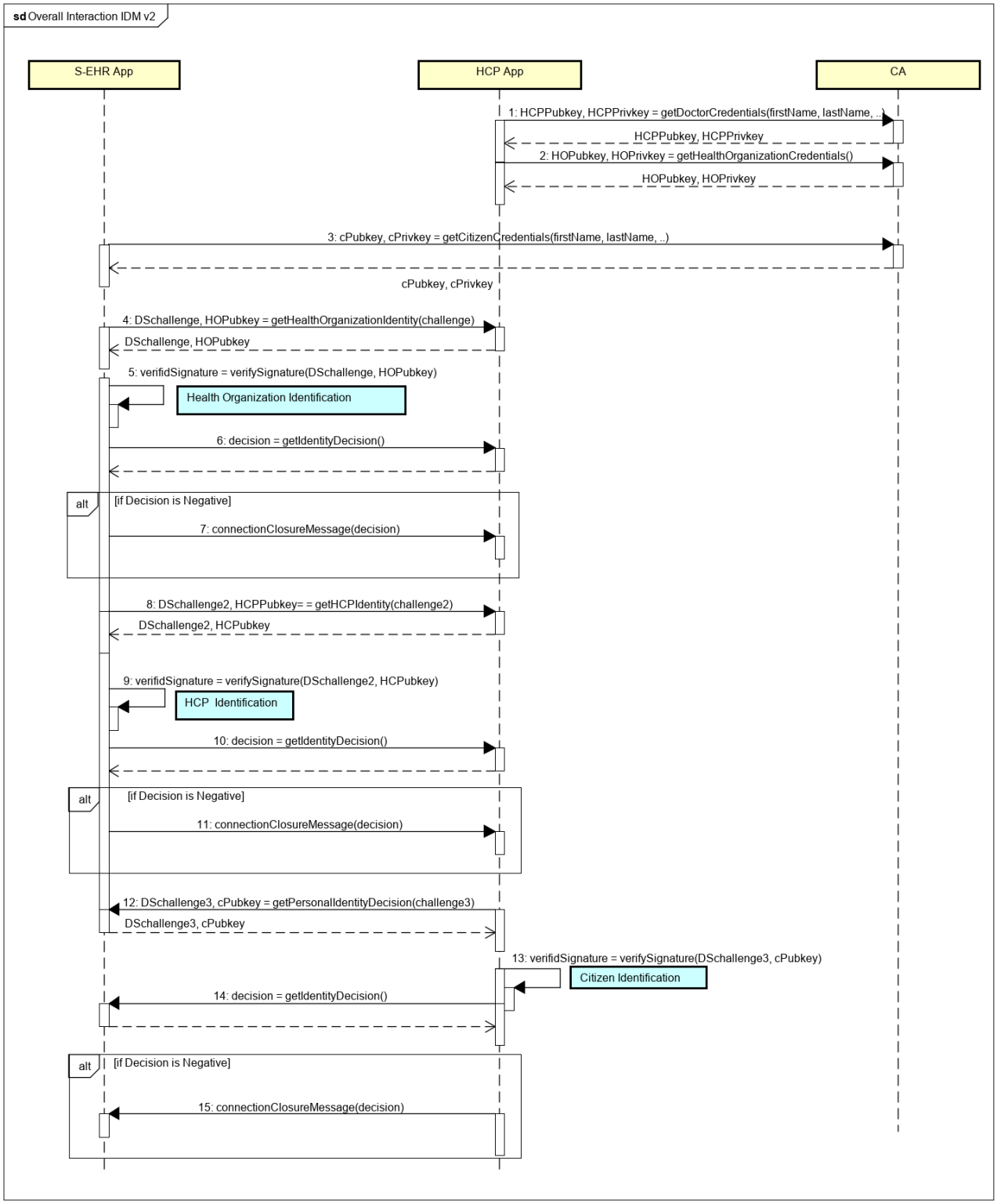


Figure 14 - D2D Identification and authentication (2nd variant)

Following a detailed description of the sequence diagram:

- **Step 1** - HCP acquires the first time his credential (public and private key) from a CA. HCP provides his first name, last name, address and other optional information etc. and receives his certificate that contains his public key (HCPPubkey), his private key (HCPPrivkey) and is digitally signed by the CA.
- **Step 2** - Health Organization obtains the first time his credential (public and private key) from a CA. Health Organization provides the name of the organization and other optional information and receives organization's certificate that contains his public key (HOPubkey), his private key (HOPrivkey) and is digitally signed by the CA.
- **Step 3** - The citizen acquires the first time his credential (public and private key) from a CA. Citizen provides his first name, last name, address and other optional information etc. and receives his certificate that contains his public key (cPubkey), his private key (cPrivkey) and is digitally signed by the CA.
- **Step 4** - The citizen request for identification of the Health Organization, by challenging with a random value (e.g. challenge) and receives the public key of the Health Organization (HOPubkey) and the challenge digitally signed (DSchallenge) by the Health Organization. The digital signature is the hash of the challenge, encrypted with HO's private key.
- **Step 5** - The citizen validate the signature of the Health Organization by decrypting the signature with HO's public key to retrieve the calculated hash. The citizen store the ETL list locally in order to perform verification of HO's certificate offline. Then he recalculate the hash based on the challenge he provided and compares the two hashes. If are the same signature is valid, otherwise signature is no valid.
- **Step 6** - The HCP receives the decision on HO identity.
- **Step 7** - If the decision is negative, a connection closure message is created and the protocol stops here.
- **Step 8** - The citizen request for identification of the HCP, by challenging with a random value (e.g. challenge2) and receives the public key of the HCP (HCPPubkey) and the challenge digitally signed (DSchallenge2) by the HCP.
- **Step 9** - The citizen validate the signature of the HCP by decrypting the signature with HCP 's public key to retrieve the calculated hash. The citizen store the ETL list locally in order to perform verification of HCPs certificate offline. Then he recalculate the hash based on the challenge he provided and compares the two hashes. If are the same signature is valid, otherwise signature is no valid.
- **Step 10** - The HCP receives the decision on his identity.
- **Step 11** - If the decision is negative, a connection closure message is created and the protocol stops here.
- **Step 12** - The HCP request for identification of the citizen, by challenging with a random value (e.g. challenge3) and receives the public key of the citizen (cPubkey) and the challenge digitally signed (DSchallenge3) by the citizen.
- **Step 13** - The HCP validate the signature of the citizen by decrypting the signature with citizen 's public key to retrieve the calculated hash. The HCP store the ETL list locally in order to perform verification of citizen's certificate offline. Then he recalculate the hash based on the challenge he provided and compares the two hashes. If are the same signature is valid, otherwise signature is no valid.
- **Step 14** - The citizen receives the decision on his identity.

- **Step 15** - If the decision is negative, a connection closure message is created and the protocol stops here.

3.2. IDM in R2D Protocol

The R2D protocol, defines the set of operations and structure of data used for enabling (in a standard way) the exchange of health data between an EMR or a National EHR and the S-EHR App with the usage of the internet [D4.1]. In accordance with the InteropEHRate objectives of Year 1, the protocol defines only operations for reading medical data from a remote EHR source. In the next version of this deliverable, we will analyse the security mechanisms needed to upload/download citizen's medical data from his mobile to S-EHR Cloud for personal backup.

In order to acquire the medical data the first time from national EHR, security operations regarding identification and authentication of the citizen are required. As already described in section 2.2, the epSOS project proposed a standard for exchange medical information for cross-border interoperability between electronic health record systems in Europe, which rolled out of the eHDSI under the Connecting Europe Facility (CEF). We utilize the architecture of eHDSI and its trusted federation of National Contact Points (NCP) as well as eIDAS eID for identification, authentication and finally acquiring patient medical data.

However, eHDSI has been designed considering as primary actors only other NCPs and HCPs, and not the citizen which instead is the primary actor of InteropEHRate. In order to enable a citizen to acquire his data, eHDSI authentication and security model needs to be extended, allowing a citizen to access only the NCP of his country and at least in read-only mode.

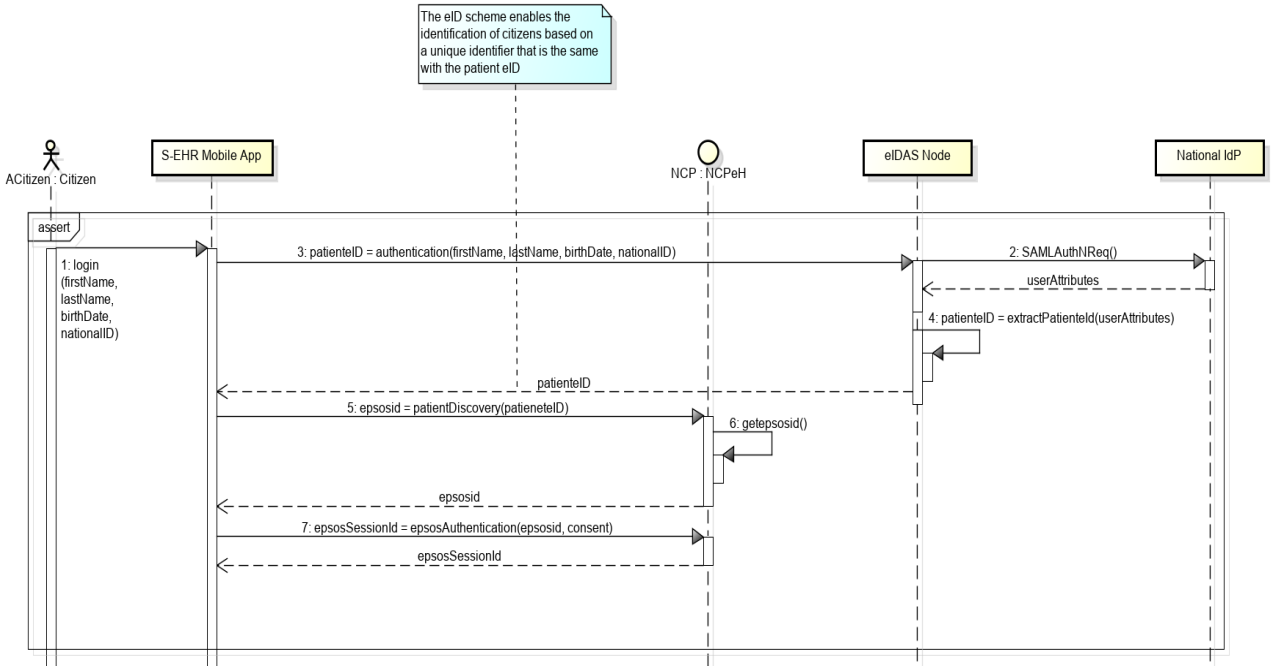


Figure 15 - R2D Identification and authentication to import data from NCP

Following a detailed description of the sequence diagram:

- **Step 1** - Citizen login to the S-EHR app, by providing his firstName, lastName, birthDate and his nationalID. This step is necessary to login to InteropEHRate S-EHR app and to authenticate to eIDAS Node.
- **Step 2** - S-EHR app requests for authentication leveraging the eIDAS Node and forwards the necessary information of the user, that already added in the first step.
- **Step 3** - eIDAS Node creates and send an SAMLAuthNReq request to the Nation Identity Provider (IDP), in order to retrieve the user attributes.
- **Step 4** - eIDAS Node extracts the patient eID from the attributes. Patient eID should be the same with CEF eID.
- **Step 5** - Based on the patient eID the S-EHR request from the NCP his medical records in order to be imported in the S-EHR app for the first time. These medical records include Patient Summary and ePrescription. This is a necessary step in the first scenario, that includes Internet connection once.
- **Step 6** - NCP extracts the epSOSid, which is necessary for authentication.
- **Step 7** - S-EHR request for authentication to the NCP, based on the epSOSid and his consent. In addition, there are two main states regarding identification and authentication. The citizen is AUTHENTICATED or NOT_AUTHENTICATED.
    - AUTHENTICATED: is the state that allows the use of the operations for the exchange of medical data. In this state we assume the citizen has been successfully identified and authenticated.
    - NOT_AUTHENTICATED: is the state that does not allow the use of the operations for the exchange of medical data, but allows only the use of the operations for the authentication of a citizen. In this state we assume the citizen has not been successfully identified or has not been authorized.

4. CONCLUSIONS AND NEXT STEPS

In this report, we defined the first version of the specification of remote and D2D identity management mechanisms for HRs interoperability focus on the objectives of the first year. A technical background with state-of-the-art protocols and standards is also provided. IDM and authentication in the D2D protocol will support two variants: a) identification with the ID-Card of the citizen and a QR code generated by the hospital including digital signatures from the HCP and the HO and b) identification with hardware-based digital signatures (e.g. qualified digital signatures) from both sides. In R2D protocol we utilize the architecture of eHDSI and eIDAS eID to acquire the medical data for the first time from national EHR.

Similarly to other reports of the InteropEHRate project, this document presents a first draft of the IDM including authentication mechanisms, reflecting the current understanding by the project consortium. A second updated version (final version) of this report is planned on March 2021.

The following version will include a clearer view on the IDM and authentication mechanisms based on the new knowledge acquired from the first two years. Major changes are expected in particular with respect to IDM and authentication mechanisms needed in the remote R2D and research protocols that are involved in usage scenarios [D2.1] that still not analysed in detail.

REFERENCES

- **[JOSANG2005]** A. Jøsang and S. Pope, ‘‘User centric identity management,’’ in Proc. *AusCERT Asia Pacific Inf. Technol. Security Conference*, p. 77, 2005.
- **[CARRETERO2018]** J. Carretero, G. I.-Moreno, M. V. Cabezas and J. G. Blas, “Federated Identity Architecture of the European eID System”, Digital Object Identifier, 2018
- **[EMTG2017]** E. M. Torroglosa-García and A. F. Skarmeta-Gómez, ‘‘Towards interoperability in identity federation systems,’’ J. Wireless Mobile Netw., Ubiquitous Comput., Dependable Appl., vol. 8, no. 2, pp. 1–25, 2017.
- **[SCUDDER2010]** Scudder, J., and Josang, A. “Personal federation control with the identity dashboard”. In IDMAN, Springer, pp. 85–99, 2010.
- **[IT2011]** Information Technology—Security Techniques—”A Framework for Identity Management”, Standard ISO/IEC 24760-1, Dec. 2011
- **[EU2009]** Digital Single Market, “Person Identification and Authentication – Key to eHealth and eGovernment Service”, 2019.
- **[FITZPATRICK2005]** B. Fitzpatrick, “Distributed Identity: Yadis,” May 2005.
- **[OPENID2011]** A. S. G. S. Gilbertson, “OpenID: The Webs Most Successful Failure,” Jan. 2011.
- **[SAML2005]** OASIS Security Services, “SAML Specifications | SAML XML.org,” Mar. 2005.
- **[XACML]** OASIS, XACML Oasis Specification. Website: https://www.oasisopen.org/committees/tc_home.php?wg_abbrev=xacml
- **[OAUTH2010]** E. Hammer-Lahav, “The OAuth 1.0 Protocol,” Apr. 2010. Website: https://tools.ietf.org/html/rfc5849
- **[STORK2010]** Vasilis Koulolias, “Secure idenTity acrOss boRders linKed (STORK)”, 2010, Website: https://joinup.ec.europa.eu/document/secure-identity-across-borders-linked-stork
- **[EPSOS]** epSOS, “epSOS Technical Aspects”, Website: http://www.promisalute.it/upload/mattone/gestionedocumentale/epSOS_Technical_Aspects_784_2462.pdf
- **[IHE ITI TF-1]** IHE IT Infrastructure (ITI) Technical Framework, “Integrating the Healthcare Enterprise”, Volume 1: Integration Profiles. Revision 11.0 – Final Text, September 23, 2014. Website: http://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_TF_Vol1.pdf
- **[ALTICELABS2014]** ALTICE LABS WHITEPAPER, “Identity and Access Management”, 2014. Website: https://www.alticelabs.com/content/WP-Information-Access-Control-Models.pdf
- **[ESENS2017]** Masi , M., Bittins, S. Cunha, J. and Atzeni, A., “e-SENS 5.2 eHealth eIDAS eID Pilot: Technical Feasibility Report”, 2017.
- **[EE2017]** European Commission, “Trust Services and eID (eIDAS regulation),” 2017, Website: https://ec.europa.eu/digital-single-market/en/trust-services-and-eid
- **[PKCS11]** OASIS, OASIS PKCS 11 TC, 2018 Website: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=pkcs11
- **[KAI2009]** R. Kai, Denis Royer and André Deuker, “The future of identity in the information society: Challenges and opportunities”, Springer Science & Business Media, 2009.
- **[NISTDSS]** FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, “Digital Signature Standard (DSS)”, 2013, Website: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
- **[NGUYEN2018]** Nguyen, K., “Certification of eIDAS trust services and new global transparency trends”, Datenschutz und Datensicherheit, 2018

- **[KENNEDY2016]** Kennedy, E. and Millard, C., "Data security and multi-factor authentication: Analysis of requirements under EU law and in selected EU Member States", Computer Law & Security Review, 32/1, 91-110, 2016.
- **[KATEHAKIS2017]** Katehakis, D.G, Gonçalves, J., Masi, M., and Bittins, S., "Interoperability Infrastructure Services to Enable Operational Secure Cross-Border eHealth Services in Europe", 17th International HL7 Interoperability Conference IHIC, 2017.
- **[UAF2017]** FIDO Alliance, "FIDO UAF Architectural Overview", 2017 Website: https://fidoalliance.org/specs/fido-uaf-v1.1-ps-20170202/FIDO-UAF-COMPLETE-v1.1-ps-20170202.pdf
- **[U2F2017]** FIDO Alliance, "Universal 2nd Factor (U2F) Overview", 2017 Website: https://fidoalliance.org/specs/fido-u2f-v1.2-ps-20170411/fido-u2f-overview-v1.2-ps-20170411.pdf
- **[SAML2]** OASIS, "Security Assertion Markup Language (SAML) V2.0 Technical Overview", Committee Draft 02, 2008. Website: https://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf
- **[RC6749]** FInternet Engineering Task Force, "The OAuth 2.0 Authorization Framework", 2012. Website: https://tools.ietf.org/html/rfc6749
- **[MOBILECONNECT]** GSMA, "Mobile Connect for Cross-Border Digital Services Lessons Learned from the eIDAS Pilot", 2019. Website: https://mobileconnect.io/wp-content/uploads/2019/02/MC-for-cross-border-digital-services_eIDAS_Feb2018-FINAL-web-2.pdf
- **[WSFEDERATION]** OASIS, "Web Services Federation Language (WS-Federation) Version 1.2", 2009. Website: http://docs.oasis-open.org/wsfed/federation/v1.2/ws-federation.html
- **[BERBECARU2019]** Diana Berbecaru, Antonio Lioy and Cesare Cameroni, "Electronic Identification for Universities: Building Cross-Border Services Based on the eIDAS Infrastructure", *MDPI Information*, 2019.
- **[RCF6749]** Internet Engineering Task Force, "The OAuth 2.0 Authorization Framework", Request for Comments: 6749, 2012.
- **[D4.1]** InteropEHRate consortium. *D4.1: Specification of remote and D2D protocol and APIs for HR exchange - V1,* 2019. www.interopehrate.eu/resources
- **[QUALIFIED DEVICES]** European Commission, "Compilation of Member States notification on SSCDs and QSCDs", 2019. Website: https://ec.europa.eu/futurium/en/content/compilation-member-states-notification-sscds-and-qscds