J. Carretero, G. Izquierdo-Moreno, M. Vasile-Cabezas and J. Garcia-Blas, "Federated Identity Architecture of the European eID System," in *IEEE Access*, vol. 6, pp. 75302-75326.

DOI: 10.1109/ACCESS.2018.2882870

Federated Identity Architecture of the European eID System

Jesus Carretero[1], Guillermo Izquierdo-Moreno[1], Mario Vasile-Cabezas[1], and Javier Garcia-Blas[1]

Computer Science and Engineering Department
University Carlos III of Madrid
jesus.carretero@uc3m.es, gizquier@pa.uc3m.es, mvasile@pa.uc3m.es, fjblas@inf.uc3m.es

***Abstract*—Federated Identity Management is a method that facilitates management of identity processes and policies among the collaborating entities without a centralized control. Nowadays, there are many Federated Identity solutions, however most of them covers different aspects of the identification problem, solving in some cases specific problems. Thus, none of these initiatives has consolidated as a unique solution and surely it will remain like that in a near future. To assist users choosing a possible solution, we analyze different Federated Identify approaches, showing main features and making a comparative study among them. The former problem is even worst when multiple organizations or countries already have legacy eID systems, as it is the case of Europe. In this paper, we also present the European eID solution, a purely Federated Identity system that aims to serve almost 500 millions people and that could be extended in mid-term also to eID companies. The system is now being deployed at the EU level and we present the basic architecture and evaluate its performance and scalability, showing that the solution is feasible from the point of view of performance, while keeping security constrains in mind. The results show a good performance of the solution in local, organizational, and remote environments.**

***Index Terms*—User authentication, Single Sign-On, Identity Federation, Identity and Access Management (IAM), Authentication and Authorization Infrastructure (AAI), Federated Identity Architecture (FIA).**

I. Introduction

Federated Identification means that individuals (persons or entities) can use their "home" credentials (personal id, password, biometric, ...) to sign on to networks of different entities (governments, enterprises, etc.) without adhering to a centralized single-sign-on system. Federated Identity Management (FIM) aims to facilitate management of identity processes and policies among the collaborating entities with a decentralized control. A FIM framework consists of processes and all underlying technologies for the creation, management, and usage of digital identities shared among several organizations.

Nowadays, there is a so called "identity crisis" [1] caused by the substantial security, privacy, and usability shortcomings encountered in existing systems for identity management [2]. However, there are multiple factors, such as trust management and trust establishment techniques, preservation of user privacy, consistent access rights across Circles of Trust (CoTs), continuous monitoring of collaborating entities, and adaptation to unanticipated events, which are not favoring the adoption of FIM systems [3]. Thus, even if great efforts have been carried out to solve technical challenges related to identity federation technologies, FIM still has to face several challenges as a global solution [4].

This paper has four main contributions. First, the paper presents a structured survey of Federated Identity Architectures (FIA), going through the major solutions, including the most used open and proprietary solutions, guiding the reader through a global and comparative view of their major features. Second, a significant number of large-scale examples of identity federations are presented. Third, this paper introduces the European Identity Federation Initiative (eIDAS), an effort to identify European citizens in any EU country using the eID of their native country. This is a very large-scale eID scheme, as it could include almost 500 million citizens, thus scalability and performance constrains are a critical issue. Finally, to validate those metrics, as part of our contributions, we have carried out an evaluation of the eIDAS servers in multiple environments.

To cover all the contributions, while making an extensive presentation of related topics, the paper has been organized as follows. Section II shows common approaches of authentication and key distribution. Section III discusses about management of authentication of entities in the ICT world, how they are represented and which are their capabilities and their permissions. Section IV presents a study on Web based FIM systems and most representative examples. Section V covers FIM systems not based on Web services. Section VI introduces a number of tools and frameworks that implement at least one Identity Management protocol. Section VII shows representative examples of large scale identity federations. Section VIII presents the European Identity Federation Initiative, an identity federation solution proposed in Regulation (EU) 910/2014 by the Connecting Europe Facility (CEF) office. Section IX discuses about the experiments carried out to asses the feasibility and performance of the deployed solution. Finally, Section X summarizes major conclusions extracted from our work and introduces future work lines.

II. User Authentication

In a world where the services sector is increasingly established on the Internet, there is a necessity of providing mechanisms to authenticate users aiming to access personalized and confidential information. Recognition paradigms (i.e. the act of allocating the identity of a user) cover the identification

of an identity within the population and the authentication of an identity by confirming that it belongs to the user provided credentials. Authentication protocols are the basis of any identification system, as they enable the verification claims in every step of the subsequent communication among principals. Typical authentication problems include message integrity, data origin authentication, non-repudiation of the sender and the receiver, and impersonation of the identity of sender and receiver. There are many approaches to ensure user authentication depending of the applied security constrains. Authentication schemes may rely on *something the user knows* (Knowledge based. E.g password), *something the user is* (Biometric based. E.g. face or eye), or *something the user has* (Object based E.g. card) [5].

The study of the paradigms for exchanging **secrets** to guarantee the authentication of individuals and the confidentiality of messages has been a constant throughout the last decades. In the context of IT, passwords and keys may be exchanged in multiple ways: explicitly (e.g. Basic HTTP Authentication [6] or POST forms); encrypted (e.g. EKE [7]); Or through a challenge/response mechanism in order to avoid sending the secret. This last mechanism is becoming very popular in the last years. It consists in sending a challenge (e.g. pseudo-random string) that must be processed to get a response, which can only be obtained if the private key is known by the sender. Challenge/response mechanisms based on symmetric encryption require for both sender and receiver to know the private key in order to verify the response. Asymmetric encryption contemplates the use of a key pair, which consists in a private key and a public key, so that the sender can process the challenge successfully with its private key and the receiver is able to verify the response using the public key of the sender.

In any case, the key distribution mechanism is a major characteristic of an authentication infrastructure. Key distribution mechanisms may be symmetric or asymmetric, in addition of involving a trusted third party with capacity of distributing the keys. In terms of symmetric cryptography, one example of key distribution protocol is Needham-Schroeder Symmetric Key Authentication [8], which aims to establish a session key between two parties on a network without sharing their private key with the other party. There is the need of a trustable third party that owns all the secret keys of the participants. There are also protocols that do not rely on third parties, but they are less extended (e.g. ISO One-pass Symmetric Key Unilateral Authentication Protocol [9]).

On the other hand, the main expression of asymmetric cryptography key distribution mechanism is through *digital certificates* of identity, being X.509 [10] the most used format. A digital certificate includes at least an identifier of the subject, its public key, and the digital signature of a trusted entity called the Certification Authority (CA), over the certificate itself. A X.509 based certificate includes multiple fields, such as version, serial number, validity, issuer ID, cryptographic information to derive the public key such as the algorithm (e.g. RSA [11] or DSS [12]), extensions depending on the version, etc. The Public Key Infrastructure (PKI) [13] is the combination of components that guarantee the trust of a subject without needing to have knowledge about her in advance. It starts with a third party entity, a Certification Authority (CA), issuing digital certificates to end-users or other intermediate certificate authorities that also issues certificates, so that a chain of trust of the certificate is established. The chain of trust is an ordered list of certificates, containing an end-user subscriber certificate and certificates of intermediate CAs until the root CA. Trusting the root CA means to trust in all certificates issued by the root and intermediate CAs.

There are also asymmetric cryptography algorithms to negotiate a shared session key between two parties without sharing any secret or the need of a third party. The most widely used is Diffie-Hellman [14], where no compromising cryptographic material is sent between two parties in order to generate a shared secret. It is worth to mention that the choice of the encryption method and the distribution of keys depend on the problem faced, as there is no perfect combination. A good approach that is commonly adopted is to use authenticated Diffie-Hellman [15] to obtain a session key for encrypting the communication. With the objective of guaranteeing integrity, authentication, non-repudiation, and to prevent from impersonation, the cryptographic material exchanged between the participants is signed using a digital certificate, which is exchanged by the participants. If both participants trust the respective certificates, they both obtain a secret key shared by means of Diffie-Hellman and, from this, they derive a private key to encrypt the communication by means of symmetric cryptography (e.g. AES256 [16] and 3DES) since, by its nature, it can be made fast computationally by using hardware accelerators (FPGA, GPU, ...) and encryption of whole blocks.

A very different authentication approach that avoids key management is the use of biometry. Biometrics is, in the context of identification systems, the science of counting and measuring unmistakable and individual biological and/or behavioral patterns to recognize a person. Several biometric recognition modalities are used (e.g. speaking, face, iris, vascular, DNA, fingerprints, etc.), as each one adapts better than other to different use cases. For example, an organic sample like DNA is unique for each person, excepting identical twins, and its recognition has practically zero error rate when analyzed and compared, but the procedure is very time consuming and very expensive [17]. Human fingerprint is assumed to have a very high uniqueness and report low error rates when compared to other methods [18]. However, the reliability of the procedure partially depends on the skin conditions and its cost depends on the capture technology. Biometrics presents also issues related to the acquisition techniques, which may be categorized from invasive (e.g. blood sample, taken to collect a person's DNA), to minimally or non-invasive, (e.g. fingerprint or iris scan), or even collected without the subjects knowledge (e.g. face recognition). Each of them has different privacy implications [19].

There are many artifacts and devices, such as bar codes, OCRs, magnetic stripe cards, optical and laser cards, smart cards, RFIDs, etc., used for recognition of individuals. Two

TABLE I
COMPARING USER AUTHENTICATION METHODS.

Class	Authentication	Key Distribution	Drawback	Defense
Biometric based	Personalized.	Not needed.	More expensive.	User protection
	Device needed.	User permanent data.	Difficult to replace.	User protection
Knowledge based	Verification of secret	Symmetric		
		- Key exchange		Possession control.
		- Third party provider	May be guessed.	Distribution control.
		Asymmetric	Less secure with	Secret disclosure
		- Public Key. Digital certificates	time and use	
		- Diffie-Hellman		
Object based	Token possession.	Object distribution.	May be stolen or lost	Possession control.
	Device needed	Third party authority.		Avoiding remote accesses

of them are notable by the features the offer: smart cards and RFIDs. A smart card is an electronic data storage system with a minimum computing capacity due to its architecture and a minimum operating system combined with a file system with the aim to offer security services. They have been proved to be a trustworthy security provider, since they issue strong authentication for card holders, but they have to be in near contact with the card reader. They may be put into service as a Qualified/Secure Signature Creation Device (QSCD/SSCD) [20], may be used as a Secure Access Module (SAM), in PoS, and as a Hardware Security Module (HSM). Radio Frequency Identification (RFID) systems enable a contact-less transfer of data between the data-carrying device and its reader. They are closely related to smart cards, since data may be stored inside, but the power supply is achieved using magnetic or electromagnetic fields. RFIDs are not as cheap as bar codes, but support a larger set of unique IDs than bar codes and can incorporate additional data. Because of their added value, they are being large-scale adopted for managing services where smart cards are limited [21].

Table I shows a comparison of the three classes of user authentication mechanisms considering some relevant features

III. IDENTITY MANAGEMENT SYSTEMS

In the previous section, we have briefly covered common approaches of authentication and key distribution and introduced how trust infrastructures are built (e.g. PKI). In this section, we tackle authentication management, how entities are represented in the ICT world, and which are their capabilities and permissions. In this section, we focus on *identity management*, that refers to the process of administrating information about the identity of users and controlling their access to organization resources, and *identity management systems* (IdM), that are set of technologies that can be used in Information and Communication Technology (ICT) system to provide identity management.

In the scope of identity management there are many terms to refer to the parties involved and the identity information related to parties. A Digital Identity [22] is the information used to represent an entity in an ICT system. An entity may be a person, an organization, a device, an application, etc. The purpose of the ICT system is to determine which of the attributes describing an entity are used to build its identity. Within an ICT system, an identity shall be the set of those attributes related to an entity that are relevant to the particular domain of application served by that system. Depending on the specific requirements of this domain, this set of attributes related to the entity (the identity) may be, but does not have to be, uniquely distinguishable from other identities in the ICT system. Therefore, a digital identity is information of an entity used by ICT systems to represent an internal or external agent. This information usually consists of an identifier, which is an unique code used to refer the identity, and a set of claims or attributes that represent the entity in the specific domain, which may determine the specific capabilities of such entity in this domain [23]. The way for an entity to prove identity is by presenting its Credential in an authentication scheme, such as to send some secret known by the ICT system and the entity (e.g. login and password), to present an artifact, maybe issued by a third party (e.g. passport), or to to check some biometric in case of humans (e.g. fingerprint).

Three entities are usually involved in a typical IdM system: user, identity provider, and service provider. The user is an entity aiming to consume a service, it is usually registered in one or various identity providers. The Identity Provider (IdP) is an entity that stores identities and their associated credentials, provides authentication and authorization services and issue assertions. The Service Provider (SP) is an entity providing services to users. The SP relies on the IdP to handle authentication by consuming authentication assertions, for this reason, it is also referred as Relying Party (RP) [24].

IdM systems may be designed considering the control and consent over the digital identities to be used and the information that will be released. It is essential to inform users of the purpose for which information is collected and about the parties that will get involved while sharing their information. There are several regulations concerning privacy protection and identity disclosure. The "need to know" restriction is a good approach when collecting user information to minimize damage in case of a security breach [25]. The use of partial identities by enabling pseudonyms and links to the holder of the identity is also a good approach to fulfill privacy

requirements. However, it is necessary to enforce a good management of the pseudonyms to maintain private the linkage between an identity and its pseudonym, to achieve anonymity due to the use of a pseudonym in different contexts, to be able to transfer attributes from a pseudonym to another, and to perform blind digital signature as well [26].

According to [27], IdM systems can be classified by using Paradigms and Models. We refer to "paradigms" when we speak in terms of implementation and deployment of the system. "Models" refer to data storage and entity roles, it is to say, where identities are stored and which is the responsibility of each party.

A. Paradigms of Identity Management

The three major paradigms, shown below, are defined taking into account the entity leading the identification process.

The **Network Centric Paradigm** congregates all identities in a database located on a central IdM system, usually referred as Domain Controller, where authentication takes place. Each entity using a network device subscribed to the domain receives a unique and single identity from which authorization decision is taken. This was the earliest approach to the IdM technology development and it has many limitations. It does not support attributes extension and federation, and the semantics of the attributes are not taken into consideration. Examples of this approach are the Zentyal Directory Server [28] in Linux and the Windows Domain [29].

The **Service Centric Paradigm** consists of different service providers deployed at different domains. Service providers usually have registers of all identifiers and authorizations of user credentials linked to the authentication systems. Following this paradigm, users have to manage, at least, one identity per service provider, since an entity may have multiple identities and thus, every identity may have several identifiers or attributes. This reduces the usability from the user's point of view. Moreover, service providers that operate in a global environment, such as the Internet, need global identifiers. To achieve this uniqueness, several identifiers, such as company name, street address, domain name etc., are encoded within digital certificates subscribed to the PKI. This paradigm can achieve dynamic replacement of services [30], so that the service chosen might depend on user preferences or some other criteria. However, composition from different domains and delegation of users' access rights are still difficult to track and control.

The **User Centric Paradigm** supports identity management on the user side, so that identifiers and credentials from different service providers are handled in the user side of the communication, resulting in improved usability and authentication protocol flexibility [31]. Moreover, it enables users in any platform to access online services, enhancing mobility. User centric solutions are designed to be cost effective and scalable from the perspective of the user, while at the same time being compatible with traditional identity management systems [32].

A table comparing major features of the three previous paradigms is presented in [27]. Three major features are compared: Centralized; Trust Domain; and Identity Handling, including authentication, identity number scale, identity uniqueness, and credentials transmission. The table reflects that the user centric paradigm is more flexible, as it can be distributed, can include multiple trust domains, and can support large scale identity management with unique Id, but it may face problems of credentials disclosure and privacy protection, but still security is better than in network and service centric paradigm.

B. Models of Identity Management

The former paradigms can be used to provide identity management following several models. The most extended identity management models are the isolated, centralized, and the federated model, which are briefly detailed below.

In the **Isolated Model**, the Service Provider (SP) and Identity Provider (IdP) are combined in a single server. This implies that identity storage and user operations, which includes authentication and authorization, are carried out at the same point (Figure 1-left). This model is a very simple approach, which may cause many problems [31]. From the point of view of the user, increasingly adoption of online services make users to handle one identity for each service they are registered in, which means managing several credentials. If credentials are managed properly by users (different credentials for each service) there may be a lack of usability, and if credentials are poorly managed (same credentials for each service) there may be a lack of security.

The **Centralized Model** consists of centralizing the identity storage while separating the services. Multiple SPs have to authenticate their users against a central IdP (Figure 1-center). 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. Other implementations are the common identifier model and the meta-identifier model. 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: the IdP. Kerberos and CAS are examples of implementations of this model.

The **Federated Model** is a step forward on the Centralized Model. In order to support the deployment of identity management systems at heterogeneous topologies, 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 as, and the configuration parameters of the participating system parties. This model is a good approach for authenticating a user across multiple sites within a company or across independent and disparate domains [33]. 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 (or any Security Token Service) in another domain [34]. The result is that a group of SPs are able to

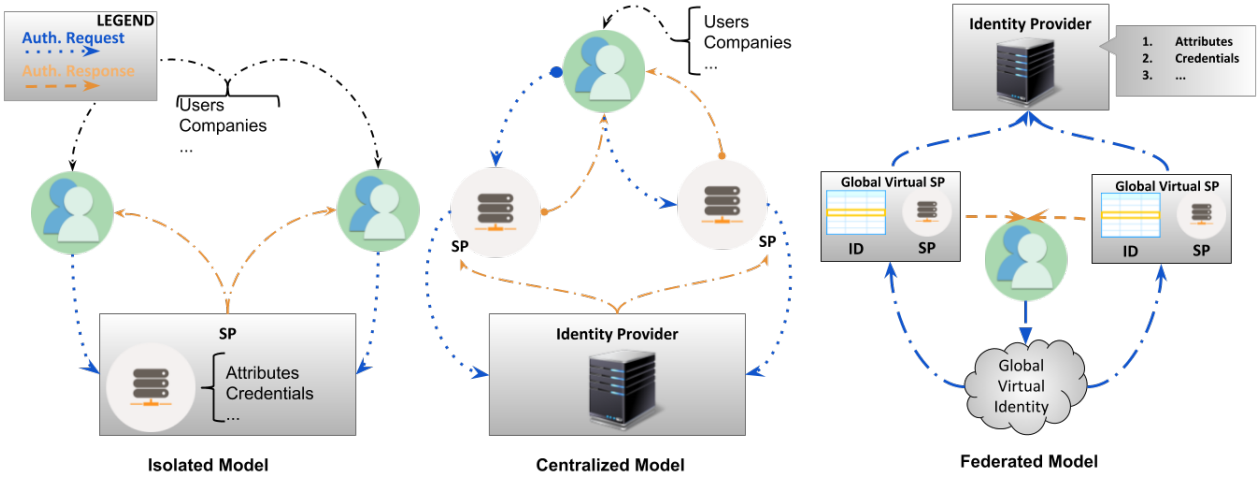

Fig. 1. Identity management models.

recognize a user identity from other SPs within a federated domain. This is called a *Federated Identity* [35].

Actors in the federated model define a Federated Identity Architecture (FIA), that groups IdPs and SPs that trust each other to exchange and share digital information preserving privacy of the user information. As shown in Figure 1, SP and IdP can be independent or they can be grouped in the same SP (similar to a cooperation of isolated models). This configuration is usually called metadata (at least at web-based specifications) and can be exchanged in an informal and unstructured way or normalized and structured way. Metadata is configuration data required to automatically negotiate agreements between system entities, comprising identifiers, binding support and endpoints, certificates, keys, cryptographic capabilities and security and privacy policies. Several specifications normalize metadata structures and contents, thus supporting efficient management processes for scalable deployments [36]. Discovery and exchanging federation metadata make easy to establish trust relationships and also to determine policies for obtaining services.

While the main goal in Identity Federation is to provide an efficient Single Sign On (SSO) service to identified individuals, identity providers and relying parties, the most common services provided by a Federation consist of: confirming that an identity belongs to the user presenting the credentials (Authentication); allowing or denying access to a protected resource (Authorization); transferring profile information or performing claim-based authorization (based on Attributes); and preserving the privacy of a principal representing her using anonymous identifiers establishing an association (steady or transient) with her local identity (pseudonymization).

Pseudonyms and claims-based authorization services enhance user privacy in a federation. A Resource Provider can request just a subset of attributes to access a resource and an Identity Provider can assert that a particular principal possesses these attributes without divulging the actual identity of the principal. This allows the users to cross the boundaries

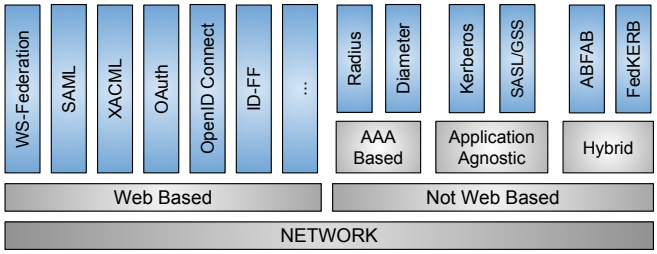

Fig. 2. Identity federation specifications classification.

of their organization and to be able to use partial identities. As an example, a user may have one or more partial identities for working, other for administrative procedures, etc. [37]. In addition, the federation approach enables the mapping of identities from providers of different organizations involved in the federation and considerably reduces the need to handle many identifiers and tokens from the user perspective [38].

Specifications and standards for identity federation can be widely classified, depending on the protocol used to exchange messages, as Web-based and Non-Web based [39], [40]. Figure 2 shows a possible classification of federation systems that will be developed in the next sections.

IV. Web-based Federated Identity Management

Currently, most the services are hosted in the web. Web-based systems rely on the exchange of files using the Hypertext Transfer Protocol (HTTP) [41]. This allows to improve the interoperability of the system, since high modularity is supported it also eliminates the need of a specific-client application for each service by making all of them accessible via web, being usually linked to the interaction of the end-user through a UI in a web browser. The unification of the services access with different purposes increases the usability due to user-friendly interactions. In addition, it is compatible with any device or platform. Service is delivered through a browser of the users choice, with no need of specific software to

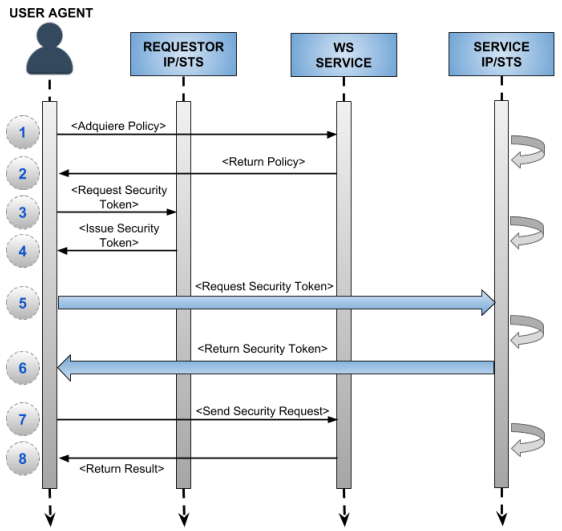

Fig. 3. SSO WS federation message flow.

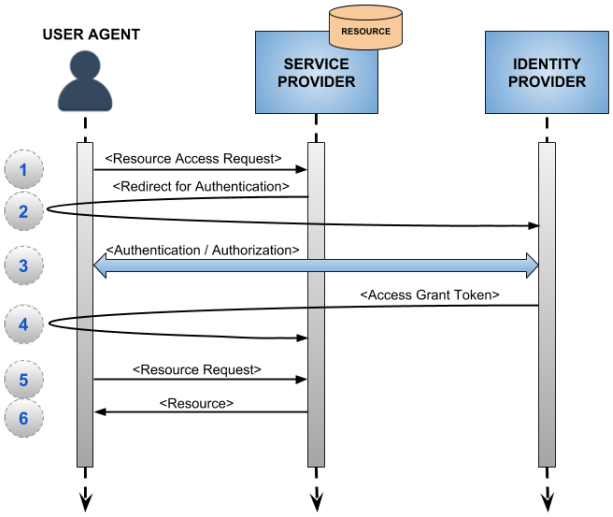

Fig. 4. SAML SSO message workflow.

download, install, update or manage. Service is in most of the cases up to date and down times are minimized with redundant systems. In addition, the security of the transactions is usually guaranteed by enabling HTTPS [42] (HTTP over SSL/TLS [43]), but other solutions could be used with this purpose as well. Unfortunately, HTTP is not perfect. It is a stateless protocol, which makes necessary to use client-based mechanisms (i.e., cookies) in order to store and retrieve client-side state information. These mechanisms significantly extend the capabilities of Web-based client/server applications, but they are more like a patch on the protocol rather than an extension.

Some of the actors that have prompted the adoption of web based identity management specifications include supporting multi-domain SSO (MDSSO). A strong limitation is that an Identity Provider has no standardized means to generate a cookie in a DNS domain that can be consumed by a Service Provider in other DNS domain in order to avoid re-authentication. In addition, proprietary protocols are sometimes impractical in heterogeneous environments.

For those reasons, many specifications consider means to establish agreement between organizations and to share or publish their metadata in order to build Identity Federations. The specifications most frequently used in this field are discussed below.

A. *WS-Federation*

WS-Federation [44] is a standard based on the WS-* family (WS-Security, WS-SecurityPolicy, WS-Trust, etc.) to establish a set of mechanisms for federation of security domains. It defines the communication flow among the IdP, the Security Token Services (STS) and the RP, which are the ultimate receivers of the issued token from the perspective of the STS. The standard also includes a method for discovering and exchanging federation metadata.

Each participant has its own policies that, combined, determine the security requirements to communicate. As shown in Figure 3, the Requestor initiates the communication flow by identifying a Resource Provider and querying it for its policies. A client obtains an identity security token from its IdP, which contains claims about a security principal that correspond to the Requestor, and then delivers then to the STS for the desired resource. If successful, that is if trust exists and authorization is approved, the STS returns an access token to the client. The client then uses the access token on requests to the resource or Web service. Note that it is assumed that there is a trust relation between the STS and the IdP. An IdP STS can also be used by a Resource Provider to validate tokens it has received from clients. A Relying Party is also referred as Resource Provider from the perspective of the client.

The mechanisms defined in this specification can be used by Web service (i.e., SOAP) clients, which are assumed to understand WS-Security and WS-Trust, as well as Web browser clients, which encode the communication using HTTP messages. WS-Federation extends WS-Trust and uses its Security Token Service to exchange security tokens, but supports other security token formats since protocol processing (RST/RSTR) is agnostic of the type of token being transmitted, enhancing the interoperability. In addition, it allows pseudonyms, attributes, and claims-based authorization services with generic STS [45].

B. *SAML (Security Assertion Markup Language)*

The Security Assertion Markup Language (SAML) [46] is an OASIS standard that defines a XML-based framework for exchanging security information between entities regarding authentication, authorization, and specific attributes. Authentication, attribute and authorization decision statements are expressed as SAML assertions, which may be trusted by all parties involved in the domain.

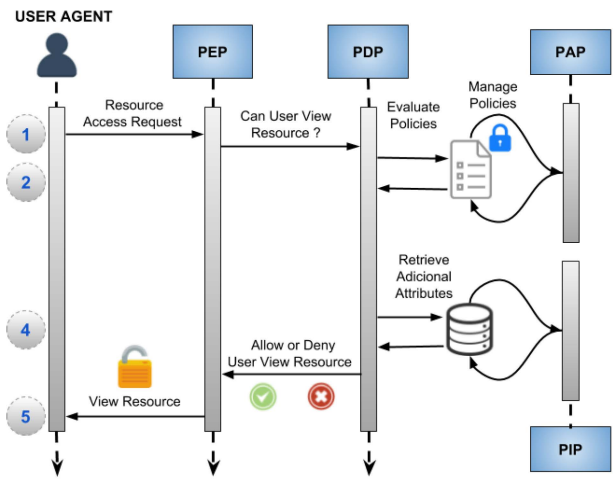

Fig. 5. XACML Message Flow.

The standard defines assertions, protocols, bindings, and profiles. An *assertion* allows for one party to assert security information in the form of statements about a subject. Generalized request/response *protocols*, such as Authentication Request or Single Logout among others, have been defined. The assertions are typically embedded in other structures for transport. The defined *bindings* for assertions and request-response messages are SOAP [47], Reverse SOAP (PAOS), HTTP Redirect, HTTP POST, HTTP Artifact, and SAML URI, with the possibility to specify additional ones. SAML profiles specify how assertions, protocols and bindings are combined to provide interoperability in particular scenarios. Examples of those profiles are Web Browser Single Sign-On, Identity Provider Discovery and Single Logout. An example of a simple SSO is shown in Figure 4.

In addition, the specification follows a modular approach that can be used in different protocol contexts, since assertions are defined as security tokens (i.e. WS-Security). Due to this, SAML has become the most consolidated standard protocol for identity federations. SAML has also been adopted to be used with several other standard frameworks (e.g. WS-Security, XACML, etc.) because its components are modular and highly extensible.

C. XACML

XACML stands for "eXtensible Access Control Markup Language", an international standard for describing security access control policies in a compositional way by promoting common terminology and interoperability between access control implementations by multiple vendors. It is primarily an attribute-based access control system, but it can be extended to a role-based access control system [48]. XACML was developed to standardize access control through XML and as a way to unify the access control policy across a variety of enterprise applications by externalizing access control decisions.

XACML defines an architecture with four major entities (see Figure 5): policy administration point (PAP), defining policies; Policy Decision Point (PDP), that evaluates applicable policy, match requests against policies, and renders an authorization decision; Policy Enforcement Point (PEP), performing access control; And Policy Information and Retrieval Point (PIP), to get and to store access authorization policies and attribute values. The user requests access to PEP, which ask to the PDP for the attributes. PDP applies the policy set by the PAP and returns the attributes, which are consulted with the PIP and combined with contextual information to create the obligations to be enforced by the PEP. The architecture is complemented with an attribute-based access control policy language and a processing model to execute policy rules.

D. OAuth (Open Authorization)

The OAuth 2.0 protocol (RFC6749) [49] is an authorization framework that enables a third-party consumer application to obtain access to some resource operated/hosted by a service provider with the resource owner consent and without sharing its credentials. This approach prevents issues associated to storing resource owner's credentials, password compromising, and limiting and revoking access to the resources.

OAuth addresses these issues by introducing an authorization layer and separating the role of the client from that of the resource owner. The client application requests access to resources controlled by the resource owner and hosted by a resource server. The authorization grant is then accomplished by redirecting the user agent from the client application to the authorization server, where it is asked for authentication as the resource owner. The user agent is returned with an authorization code and redirected to the client application, which, in turn, sends the code to the authorization server obtaining the Access Token as a response. This token is used to manage user consent against the resource server and to consume the resource, even though it may be considered with a kind of pseudo-authentication. The access token consists of a string denoting a specific scope, lifetime, and other access attributes. The message exchange flow is shown in Figure 6, but the request of user information exchange must be excluded from the OAuth protocol, since it only occurs if the OpenID Connect authentication layer is added.

This specification is designed to be used with HTTP. The client must use the HTTP "POST" method when making access token requests. It defines two types of clients, confidential and public, depending on the capability to maintain the confidentiality of their client. In addition, it supports TLS to secure communications.

E. OpenID Connect

OpenID Connect 1.0 [50] implements an authentication layer on top of the OAuth 2.0 protocol. 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 sessions and end-users in an interoperable and REST-like manner. OpenID Connect includes a new authentication request message, a new

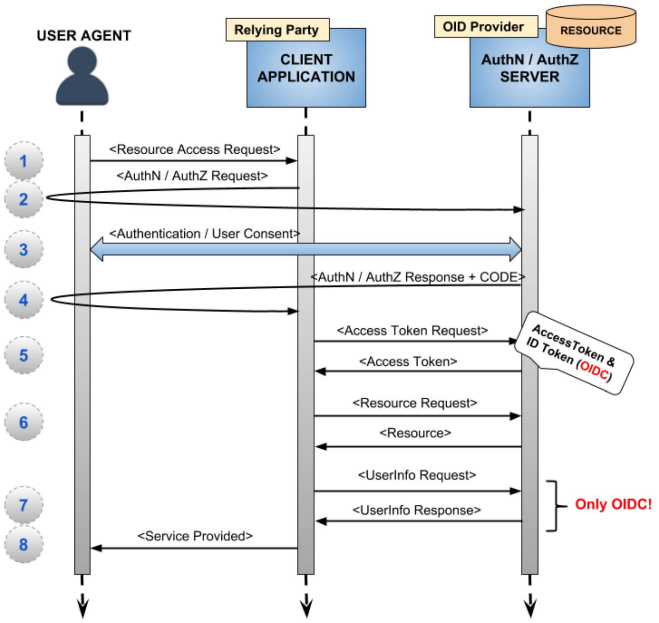

Fig. 6. OAuth/OIDC message workflow.

ID Token, which contains claims about the authentication and is represented as a JSON [51] Web Token (JWT), and new request/response messages to get additional user data. OAuth 2.0 Authentication Servers implementing OpenID Connect are also referred to as OpenID Providers (OPs). OAuth 2.0 Clients using OpenID Connect are also referred to as RPs. OpenID Connect provides the benefit of not exposing any tokens to the User Agent and possibly other malicious applications with access to the User Agent.

The specification suite is extensible, allowing participants to use optional features, such as encryption of identity data, discovery of OpenID Providers, and session management, when it makes sense for them.

A simplified message workflow is shown in Figure 6. The client application prepares an authentication request containing the desired parameters and redirects the user to the Authorization Server for authentication and authorization grant. The user agent is returned with an authorization code and redirected to the client application, which, in turn, sends the code to the token endpoint obtaining the ID Token and the Access Token as a response. Finally, the client validates the ID token and retrieves the end-user's Subject Identifier so that the user does not get the token.

Despite the OpenID Connect standard specifies how a RP can discover metadata about an OP, and then register to obtain client credentials, during discovery and registration there is no automated mechanism for the OP or the RP to verify the information exchanged during this process. All the information is self-asserted. Thus, in an identity federation context, the participants of the federation must be able to trust information provided about other participants in the federation. There is a project to extend Signed Metadata, as introduced by OAuth 2.0 Authorization Server Metadata, to create Metadata statements. Metadata statements, together with the use of a trusted third party that verifies and enforces some common policy, can be used to transfer verified data and trust in the data between clients and servers. This way it is possible to build an identity federation around a trusted third party: the federation operator [52].

F. Other Specifications and Protocols

In addition to the main Web-based identity systems described above, there are several frameworks providing identity services for specific tools or for open federations. Usually, frameworks for specific tools are offered to other tools to be used as centralized identification points. This is the case, for example, of Facebook Connect, that is nowadays used as identity server for many other tools located in the Web. A brief description of the most popular frameworks is shown below.

Liberty Identity Federation Framework (ID-FF) was one of the first approaches for identity federation. It implements single sign-on as well as federation of identities [53], [54]. Implementations of this protocol must perform authentication, identity federation, use of pseudonyms, support for anonymity and global logout. Its three main components are web services, as a protocol to directly communicate Service and Identity Providers, metadata and schemas with a specific format to notify the provider-specific and status information, and web redirection made from the user agent enabling the service provision. Despite it is a complete specification, it was not widely adopted by the community, but it have been included in the Kantara Initiative [55], which is a global and open initiative to provide a strategic vision and real world innovation for the digital identity transformation that includes identity relation management, user managed access, identities of things, and minimum viable consent receipt.

User-Managed Access to Web Resources (UMA) [56], [39] is an open specification under active development by the Kantara Initiative. It uses the OAuth resource registration and OAuth and OpenID Connect grants. UMA provides to the users with a mechanism to make the authorization decision over their resources hosted by a Web based resource owner. Users define their own policies: customer privacy, consent, and control data sharing.

Systems for Cross-domain Identity Management (SCIM) [57] is a protocol for cross-domain identity provisioning via HTTP. The SCIM specification is a standard designed for provisioning and managing identities, such as users and groups, in cloud-based applications and cross-domain services. Its two major components are a core schema based on attributes within JSON objects, and a protocol specification with a REST API for exchanging identity resources via JSON that allows massive bulk updates. SCIM does not provide authentication or authorization, instead it relies on the Transport Layer Security (TLS) or standard HTTP authentication and authorization schemes.

Facebook Connect is a single sign-on application that allows users to interact on other websites through the Web [58] using their Facebook identity. Third party Web sites can

access to Facebook users' data to make the authentication. Facebook Connect is proprietary, thus the protocol is not open. Facebook Connect is subject to replay attacks and masquerade attacks [59].

Google Sign-in [60] is a secure authentication system that enables the user to sign in with their Google Account to access all Google services in a secure manner. Google also provides a user-centric APIs and services to integrate Google Sign-in in the applications. It is compatible with authorization protocols OAuth2.0 and OpenID Connect.

Eclipse Higgins [61] is based on an active client (called a selector–a personal identity manager) based on information cards or i-cards. Those cards contain a set of data fields (claims) about the user and they can be shared with friends and businesses trusted by the user. They can be used to login to i-card-compatible websites, as well as to present other information about the user. The selector client can be integrated in browsers and it runs on a computer or mobile device.

Microsoft U-Prove [62] is a user-centric proprietary system based on device-protected tokens using anonymous credentials on smart cards. A U-Prove token is a new type of credential, similar to a PKI certificate, that can encode attributes of any type, that is unlinkable to users to avoid tracking, and that allows to disclose dynamically subsets of attributes following user specification to compare with clear data. It allows developers to create interoperable implementations of U-Prove protocol participants.

VOOT [63] is a protocol providing a data model to associate users in a group with a resource or service in read-only mode. As described by Kremers in [64], the specification defines the communication protocol between a client application and a VOOT provider, while the user group directory may be accessed by the provider by other mechanisms, such as LDAP or SQL. The VOOT API is based in REST, and the provider support OAuth and Basic HTTP Authentication. The proxy operation make possible to combine information from several group providers using one API service. In addition, VOOT does not provide federation mechanisms, but it plays well with SAML or OpenID Connect identifiers for federation purposes.

PAPI [65] is an identity service that allows a users to access to the web servers of the resource provider through access points in order to evaluate the trust of the user requests to the resource providers. The access in only available for a limited period of time set by the authentication server. It consists of two main components: the Authentication Server (AS), in the role of IdP, providing a single point of authentication, and the Point of Access (PoA) handling access decisions over resources. There are PAPI components implemented in Java, Perl, and PHP.

CAS Protocol [66] is a ticket-based protocol for CAS. It relies on a single server that is responsible for authenticating users and granting accesses to applications. Authentication requests come from a client application through embedded CAS services. The protocol uses two kind of tickets: TGT (Ticket Granting Ticket), valid for single sing on session of a user, and ST (Service Ticket), valid for the access granted by the CAS server to an application for a specific user.

TABLE II
WEB-BASED FIM SPECIFICATION COMPARISON

Specification	Authentication	Authorization	Access Delegation	Single Sign-On	Attribute/Claims	Pseudonym	Federation	Metadata	Discovery	Identity Provisioning	Extensibility
WS-Fed	•	•		•	•	•	•	•	•		•
SAML 2.0	•	•		•	•	•	•	•	•		•
XACML		•	•		•			•			
OAuth 2.0		•	•								•
OIDC 1.0	•	•	•	•		•	•	•	•		•
ID-FF 1.2	•	•		•	•	•	•	•	•		
SCIM 2.0					•				•	•	
UMA 2.0		•	•		•		•		•		•

Idemix [67] is a a prototype of an anonymous credential system. It is based on high-level primitives and interfaces providing an abstraction layer to handle security and privacy features allowing integration into access control systems. The credential system relies on protocols for pseudonym registration, credential issuing and credential verification. Parties involved in the system are users, obtaining and showing credentials, and organizations, issuing and verifying credentials. Users may maintain unlinkable pseudonyms with different organizations while perform claim based authentication. Idemix also enables de-anonymization in some scenarios.

SWIFT Identity [68] is a framework to manage access services to users thought their virtual identities. It allows performing SSO between services. The main goal is to make an aggregation of identities from different providers to find them in a single meeting service in a pseudonymous and unlinkable way. Parties involved are the End User, accessing services with their identities, the Service Provider, consuming authentication asserts to authorize access to a service, the Identity Aggregator, as the main point of trust and performing identity provisioning, the Authentication Provider, issuing authentication asserts about the End User, and the Attribute Provider that manages End User's attributes. The process contemplates the creation of the virtual identity, its aggregation and the consumption of the service.

G. Comparison

A schematic comparison of multiple features of some of the most relevant web specifications is shown in Table II. It takes into account literature about security analysis of federated IDM systems [69] and background on the field [70], [71].

V. NOT WEB-BASED FEDERATED IDENTITY MANAGEMENT

Apart from the proliferation of web-based mechanisms for authentication and single sign-on, there are protocols that man-

age these processes in environments where it is not feasible to base communications on HTTP. They are usually much older and studied protocols, which are more related to domain-level management and deployed at networks where devices are more controlled.

A. Authentication, Authorization and Accounting (AAA) Based Identity Federations

Its a term used to refer to a family of protocols that mediate network based access [72] managing authentication and authorization of users and the accounting of network resources information between a Network Access Server (NAS) and an Authentication Server.

Two protocols for AAA are worth to mention in terms of relevance: Remote Authentication Dial In User Service (RADIUS) [73], [74], that is the most widely deployed; Diameter [75], a more complete protocol fixing some RADIUS limitations, but less popular. The mode of operation of both protocols is similar. RADIUS operates in a pure stateless client-server paradigm, while Diameter operates more like a peer-to-peer protocol, since all nodes can initiate the communication and keep some state information. RADIUS runs over UDP, while Diameter can use reliable transport protocols, such as TCP or even SCTP. Both RADIUS and Diameter allow intermediate nodes in the communication between client and server. They are referred as "proxies" in RADIUS and "agents" in Diameter. In terms of authentication, both protocols support the use of NAIs, CHAP, EAP and PAP, but RADIUS implementation of these protocols has some security flaws and limitations [76], [77]. From the authentication point of view, both protocols offer some support for access rules, authorization restrictions, and filters, but not a complete support. In addition, Diameter Base protocol allows users to be periodically reauthenticated and/or reauthorized on-demand [78], and authorization without authentication, better than RADIUS. For compatibility and extensibility reasons, Diameter is also a better choice since its design fixes limitations in RADIUS mechanisms [79]. There is a proprietary solution developed by Cisco called TACACS+, which also complements the independent authentication, authorization, and accounting (AAA) architecture.

Protocols commonly used in combination with AAA protocols are EAP and LDAP. The Extensible Authentication Protocol (EAP) [80] is an authentication framework that supports multiple authentication methods (e.g. EAP-TLS, EAP-SIM, EAP-AKA, PEAP, LEAP y EAP-TTLS, etc.). This protocol was designed to manage authentication in accesses to the network, so it does not require support for IP network layer. However, EAP usually is encapsulated by other protocol, such as Point-to-Point Protocol (PPP) or IEEE 802. The Lightweight Directory Access Protocol (LDAP) [81], [82] is an application layer protocol that provides access to distributed directory services to manage domain-level information. The protocol stores authentication information, such as credentials, and it is used to authenticate, although it is possible to store other identity attributes (user contact data, location of various network resources, permissions, certificates, etc). In short, LDAP is a unified access protocol to a set of information about a network that implements the data and service models specified in the X.500 Directory Access Protocol (DAP).

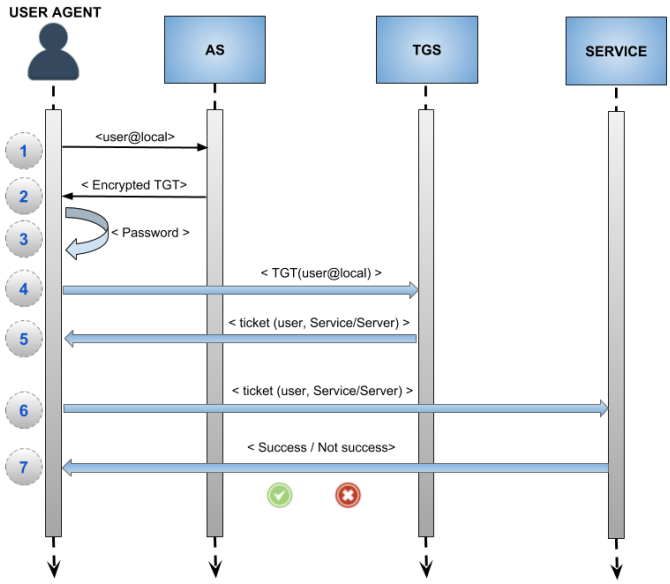

Fig. 7. Kerberos Message Flow.

B. Application-Agnostic Identity Federations

This section describes some protocols that allow adding an underlying authentication layer to applications that initially do not implement it on their own. They also provide mechanisms to reduce the degree of coupling of the applications that consume these protocols.

a) Kerberos Protocol: The Kerberos Network Authentication Service RFC4120 [83] is a distributed authentication service allowing a user to prove its identity to a service without exchanging any compromising material through the network, and thus, avoiding for an attacker to impersonate the user by intercepting the communication. Kerberos is based in part on Needham and Schroeder's trusted third-party authentication protocol [84]. It may include integrity check and confidentiality for messages sent between the client and server.

The parties involved in the system are the user, using a client software, the network service and the Key Distribution Center (KDC), where the credentials are securely stored. The KDC is composed by an Authentication Server (AS), which proves the identities of principals, and the Ticket Grant Server (TGS), which issues tickets to access the desired network service. The concept of the Kerberos system consists of the client proving its identity against the AS, which return a Ticket Granting Ticket as a proof of successful authentication. The client has to present it to the TGS together with the ID of the service it want to consume. The TGS return then a Service Ticket from which the Session Key between client and service can be derived. From now on, the communication is confidential.

Kerberos limitations are contemplated in its specification, since a custom client must be deployed and integrated with other parts of the system to be useful. Only coupled software will benefit from the protection of the messages. In order to consume network level security services, changes to the network software of the hosts involved may be performed. In addition, Kerberos lacks of support of authorization by itself, but V5 Kerberos supports passing authorization information generated by other services [7].

When two domains belong to a same organization, a Kerberos client or KDC can automatically establish a chain of trust between these two sub-domains by following the DNS domain hierarchy and assuming for a parent domain to share keys with its child sub-domains. However, it is not possible to accomplish this between domains from different organizations in the chain of trust. This problem is solved by *Cross-Realm Kerberos Authentication Operation*.

Kerberos was redesigned in the RFC6806 [85], [86] to operate across organizational boundaries, allowing inter-organization operation. A client in one organization can be authenticated to a server in another. The idea is that each organization wishing to run a Kerberos server establishes its own domain. The name of the domain in which a client is registered is part of the client's name, and can be used by the end-service to decide whether to honor a request. However, Sakane showed in [87] that Kerberos cross-realm operations might have problems in large-scale Kerberos deployments, like lack of scalability in the clients or exposure to man in the middle and denial of service attacks.

b) Generic Authentication Frameworks: GSS-API and SASL: Simple Authentication and Security Layer (SASL) and the Generic Security Service Application Program Interface (GSS-API) are application frameworks to generalize authentication. GSS-API [88] is a standard interface that provides a generic authentication and secure messaging interface. It does not actually provide security services itself, security mechanisms can be plugged in. It supports a range of underlying mechanisms and technologies and hence allows source-level portability of applications to different environments. For example, applications are able to consume directly from the Kerberos API, but due to interoperability reasons, it is a good approach to use the GSS-API mechanism to abstracts requests to the Kerberos API. Thus, if a new authentication mechanism is to be supported by the application, the implementation will be much simpler due to the low degree of coupling to the authentication mechanism. Thus, the application consuming the GSS-API interface can use Kerberos, as well as other security mechanisms supported by GSS-API [89].

SASL [90] is a framework for method for adding authentication support with an optional security layer to connection-oriented protocols, such as LDAPv3. It provides a structured interface between protocols and mechanisms. An application that uses more than one protocol, for example an email client, it is required to support authentication for each protocol (IMAP and POP), which may imply the implementation and support for various authentication mechanisms. The SASL framework provides a mechanism for securing subsequent protocol exchanges within a data security layer. Thus, old and new protocols can reuse existing mechanisms [89].

The **GS2** Mechanism Family (RFC6616) [91] approaches the convergence between GSS-API and SASL by adapting the GSS security context token exchanges to support SASL semantics and negotiation to implement channel binding negotiation. Thus, an implementation of a SASL component may provide the GSS-API interface as well. As an example of its use, a pure SASL mechanism for OpenID exploiting the channel binding between SASL and the GSS-API was standardized [92]. It defines the communication between SASL and OpenID to assert identity and other attributes to Relying Parties. In this way, clients would be able to choose the OpenID mechanism when SASL servers (as Relying Parties) notify their mechanisms.

C. Hybrid Specifications

In addition to web and non-web based specifications, there are some standards that are able of "tunneling" the assertions of one family of specifications through protocols of another family, since the channel does not support these assertions, but the receiver can interpret and decode them. These specifications are a hybrid between these two worlds. Two examples are Application Bridging for Federated Access Beyond the Web (ABFAB) [93] and Federated Kerberos (FedKERB) [94].

As described in [33], they are architectures that achieve federated authentication and authorization access control to applications beyond the web. They are able to transport SAML assertions of a network-centric IdP over RADIUS attributes by means of a deployed AAA infrastructure. Both of them manage user authentication against this IdP through EAP by using GSS-API and they allow to freely deploy access control policies without forcing to use a specific one. The main difference between them are that, for the authentication and single sign-on process, ABFAB relies on a component in the client side called Identity Manager that remembers user credentials provided to access a specific Resource Provider (RP) in case it has to use them in a new access to the same RP. On the other hand, FedKERB manages authentication using Kerberos, and thus, using a centralized KDC that provides the well-known features of this protocol.

VI. Frameworks and Tools for Identity Federations

There are many tools and frameworks that implement at least one Identity Management protocol in order to provide a solution that supports various use cases (e.g. authentication, authorization, identity provisioning, accountability, etc.). In addition, many of those tools are designed to facilitate integration into organizational environments in production, supporting in many cases scalability and modularity.

A. Open Frameworks and Toolkits

Open source solutions are developed and supported, either by the community of developers, or by organizations that offer maintenance of the software in production.

Shibboleth is an open source SAML implementation. The Shibboleth Consortium is responsible for the development of the Shibboleth SP and IdP as open source software, among others [95]. It allows authentication, authorization, content personalization, and enables single sign-on across a broad range of services from many different providers. Shibboleth consists of several individual components: IdP, SP, and discovery service (DS), which may be deployed separately depending on the specific needs of the organization. It currently supports most of the profiles defined by SAML 1.1 and SAML 2.0 and the possibility to enable WS-Federation with ADFS at the SP [96]. It could also be improved by installing software extensions into the Identity Provider, such as delegated CAS authentication metadata extension, discovery, two factor authentication (e.g. Google Authenticator) and support for a Kerberos Login Handler, and support for X.509 login handler, among others. They are also working on adding support for the OpenID Connect protocol on Shibboleth 3.

UNITY is a complete solution for identity federation and inter-federation management [97]. Management of groups and group hierarchies, Single Sign-On and federation services can be outsourced to Unity. It supports authentication of users from identity providers based in SAML, OIDC, or LDAP, as well as native usernamepassword and X.509 certificates. UNITY also acts as an OAuth authorization and resource server to issue access tokens, enables delegated access to user attributes and enables bridging of SAML identity federations [98].

simpleSAMLphp is an open source native PHP application that is mainly focused in dealing with SAML 2.0 as a Service Provider (SP) and SAML 2.0 as an Identity Provider (IdP). However, it also supports some other identity protocols and frameworks, such as Shibboleth 1.3, A-Select, CAS, OpenID, WS-Federation or OAuth, and it is easily extendable, so that new modules can be developed [99]. It offers user consent, discovery, validation, statistics, monitoring, metadata management. In addition, SimpleSAMLphp scales pretty well. Configuring a high availability SP or IdP is not difficult since there is very little state to share between the nodes of a cluster. It also offers support for Memcached, so that sessions between multiple instances may be shared.

Central Authentication Service (CAS) [100] is, in addition to an open and well-documented protocol, a client-server application providing a library of clients for Java, .Net, PHP, Perl, Apache, uPortal, and others that are able to interconnect with their open-source Java server component to offer enterprise Single Sign-On. It supports various methods of authentication (e.g. LDAP, X.509, RADIUS, etc.), multiple authentication and authorization protocols (e.g. CAS, SAML, OAuth, OpenID, etc.), multifactor authentication and many other features. It integrates with uPortal, BlueSocket, TikiWiki, Mule, Liferay, Moodle and others [101]. An open source implementation of a CAS framework is available through their GitHub repository [102].

OpenAM [103] is an open source centralized access management solution providing authentication, authorization decisions to access resources, and federation services in a single, integrated solution. The whole system is developed in Java and it is deployed as a WAR project, providing a high availability and a highly scalable and extensible multi-platform infrastructure. Features may be extended to the framework through its SDKs and several REST endpoints. OpenAM supports SAML, OAuth2, OpenID Connect and UMA. It offers a small SAML 2.0 application, which lets service providers quickly add SAML 2.0 support to their Java applications. OpenAM is part of the ForgeRock Identity Relationship Management platform, which features other products such as OpenIG, a SSO solution to enterprise, legacy, and custom applications that may work together with OpenAM to integrate Web applications, and OpenICF, an Enterprise and Cloud Identity Infrastructure Connector offering a consistent layer between target resources and applications.

B. Other Open Frameworks and Tools

The **Apache mod_auth_mellon** [104] is an authentication module for Apache. It authenticates the user against a SAML 2.0 IdP, and grants access to directories depending on attributes received from the IdP. mod_auth_mellon can easily be used as a proxy service to handle secure authentication between a client application and a SAML 2.0 compliant Identity Providers by indicating these applications are Service Providers for the IdP, this allows clients to avoid handling all SAML 2.0 requests and responses. It is able to validate all SAML assertions. Once assertions are validated, SAML claims attributes are made available in Apache and can easily be passed down to applications through Response Headers.

The **Globus Toolkit** is an open source software toolkit used for building grids [105]. The toolkit includes software services and libraries for resource monitoring, discovery, and management, plus security and file management [106]. The Globus Toolkit Authentication and Authorization Module relies on Globus Security Infrastructure using X.509 certificates, TLS and proxy certificates to authenticate and authorize grid users. It relies on MyProxy [107] to manage user credentials and to handle authentication, GridFTP [108], [109] to deploy the Globus service, and GRAM [110] for resource management. Identity federation and attribute-based authorization through the Globus toolkit is studied in [111].

Moonshot [112] is a technology to provide users SSO and for extending the benefits of federated identity to non-web services, which includes cloud infrastructures, high performance computing & grid infrastructures, and other commonly deployed services including mail, file store, remote access, and instant messaging. Moonshot is based on the IETFs Application Bridging for Federated Access Beyond web (ABFAB) open standards [113]. It applies strong network authentication by means of EAP/RADIUS, and strong authorization as used by many national federations, by using SAML assertions [114].

The **Virtual Organization Membership Service (VOMS)** [115] is an authorization management technology that allows virtual organizations to manage their users, roles, groups and attributes assigned to users. Authorization policies are

expressed as Access Control Lists (ACLs) linked to VOMS groups. It also includes an audit log if necessary. VOMS outputs can be used in a delegation workflow. VOMS mainly focuses on X.509, but the VO membership can be queried through a SAML attribute query as well, although this feature is not used in production [116].

The **LCMAPS** [117] framework is a Local Credential Mappping Service that allows to take various credentials as input (e.g. a certificate and/or VOMS credentials) and to map them to Unix credentials as output (POSIX credentials like User ID, Group ID, etc.). Just like LCAS, its predecessor, LCMAPS is mainly focused on Grid jobs. It provides an advanced and flexible plugin engine and a wide variety of plugins exist, including plugins to support the mapping onto a local Unix account and group, including VOMS attributes, process enforcement by setting the effective user and group ID for the current process, between others.

The **Argus Authorization Service** [118] provides a distributed authorization framework.As it is based on the XACML standard, it uses security policies to take the authorization decision to perform a certain action on a particular service. The authorization process relies on three components: the Policy Administration Point (PAP), in charge of the authorization policy management, the Policy Decision Point (PDP), implementing the authorization engine, and the Policy Enforcement Point Server (PEP Server), ensuring the integrity and consistency of the authorization requests received from the PEP clients, which are also provided by the framework. Argus supports SAML2-XACML2, X.509 and VOMS specifications.

LASSO [119] is a free software library used to define processes for federated identities, single sign-on and related functionality. Lasso is built on top of libxml2, XMLSec and OpenSSL and implements the Liberty Alliance ID-FF 1.2 protocols and supports SAML 2.0 and several features of ID-WSF. It works in GNU/Linux, MacOS X and Microsoft Windows distributions.

LemonLDAP::NG (LL::NG) [120] is a modular WebSSO (Single Sign On) based on Apache modules. It simplifies the building of a protected area with a few changes in the application. It manages both authentication and authorization and provides headers for accounting. So you can have a full AAA protection for your web space, as described below. LL::NG can easily interconnect to other authentication systems using SAML, OpenID, CAS, as in a federation. Its SOAP API can also be used to dialog directly with custom applications. It is composed by three main components: the Manager, for administrators to configure and explore sessions, the Portal, the core of the system where authentication of users take place, and the Identity Provider, with services provided via SAML, OpenID or CAS, and application lists. The Handler consists of Apache modules used to protect applications.

WSO2 Identity Server is a unified authentication server and rights management tool, developed since 2007, notably by Dr. Sanjiva Weerawarana, one of the fathers of the WS-* architectural vision. WSO2 Manager API is based on the Carbon platform and implements OSGi specifications common to all WSO2 products, which are modular and scalable. The solution allows users to load data from any external source, LDAP, Active Directory, JDBC, its own base or an integrated Apache Directory Server. It provides a unified authentication system via OAuth 1.0 & 2.0, OpenID, SAML2 and Kerberos KDC. Policy access control is done via the XACML 2.0 and 3.0 specifications.

TABLE III
TOOLKIT COMPARISON IN TERMS OF GOALS AND SUPPORTED PROTOCOLS.

Tool	Goal	Supported Protocols
Shibboleth	AuthN, AuthZ, SSO, Federation	SAML 1.1/2.0, X509, Kerberos,
UNITY	AuthN, AuthZ, SSO, Federation, Identity Provisioning	SAML 1.1/2.0, OIDC, OAuth, X509, LDAP
simpleSAMLphp	AuthN, AuthZ	SAML 1.1/2.0, X509, OpenID, OAuth 2.0, Kerberos, VOOT, LDAP, RADIUS
CAS	AuthN, AuthZ, SSO	CAS, SAML, OAuth, OpenID, X.509, LDAP, RADIUS
OpenAM	AuthN, AuthZ, SSO, Federation	SAML, OAuth2, OIDC, UMA
OKTA	AuthN, AuthZ, SSO, Federation, Identity Provisioning	SAML 1.1/2.0, WS-Fed, OIDC, OAuth, X509, AD/LDAP, SCIM
OneLogin	AuthN, AuthZ, SSO, Federation, Identity Provisioning	SAML 1.1/2.0, WS-Fed, OIDC, OAuth, AD/LDAP, SCIM, IWA
Auth0	AuthN, AuthZ, SSO, Federation, Identity Provisioning	SAML 2.0, WS-Fed, OIDC, OAuth, LDAP

LinOTP is an open source One-Time Password (OTP) solution maintained by the German company Leading Security Experts (LSE GmbH). It is a robust, professional solution that can be integrated with a heterogeneous infrastructure. OTPs are passwords generated at a given time, that are valid over a short period of time and used only once. These passwords are generated thanks to certain hardware, like tokens and even smartphones. LinOTP has interfaces with all types of tokens that support the HMAC-OTP protocol, as well as with hybrid solutions like MOTP devices. LinOTP is distributed under the AGPL v3. An Enterprise edition is also available. From a technical standpoint, LinOTP is a server written in Python providing communications via simple HTTP queries. This means it can be also administrated using tools different than those provided as part of the distribution. For example, a custom web interface could be developed and included in a special section of an Intranet site.

There are several other tools in the scope of identity management, such as HEXAA [121], COmanage [122], Grouper [123], Perun [124], Apache Syncope [125], Evolveum MidPoint [126] etc. also aimed to manage projects, virtual groups, resources, identity provisioning, etc. They are not commented in this paper, as we are addressing major initiatives.

C. Proprietary solutions

The need of managing identity of users in the different environments and controlling access to products, has generated a plethora of commercial solutions for identity management, being some of them open to any user or company and other proprietary solutions to generate identities valid only in a line of products or a commercial provider. The end goal is to avoid multiple sign-on and digital channels for the same user, which creates siloed and fragmented customer data, by providing the users with a single sign-on mechanism. The most representative examples of proprietary solutions are shown below.

OKTA Identity Cloud [127], [128] is a high-availability identification service providing single sign-on and multifactor authentication to customers. The service can be linked to any web and mobile application, providing two-factor authentication. Moreover, it can be integrated with AD/LDAP across multiple domains federation.

OneLogin [129] is a cloud software solution that integrates with web servers to provide secure access to company and commercial applications available through the web (SAP, ORACLE, G-Suite, etc.). It performs transparent multi-factor authentication and authorization using SAML or LDAP.

Auth0 [130] is an identity management platform that enables SSO, enterprise federation, and connection to Active Directories such as LDAP and ADFS. It also provide user management, mechanisms for multifactor authentication, protection against password leakages, and biometric access.

RSA SecurID Access [131] provides a cloud RSA Authentication Service using RSA SecurID tokens with on-premise multifactor authentication, SSO, centralized access, and authentication policies. It also accepts authentication requests from third-party SSO solutions configured to use RSA SecurID access.

Microsoft Azure Active Directory for Microsoft solutions [132]. It is a comprehensive identity and access management service that combines directory services, identity governance, application access management, and a standards-based platform for developers. Azure AD is also designed to work with on-premises Active Directory and other directories, allowing organizations to leverage existing on premises infrastructure for the cloud.It has a programmable API through the Azure AD Authentication Library (ADAL) and the Multi-Factor Authentication SDK.

Salesforce Identity Management (SIM) [133] provides Customer Identity and Access Management (CIAM) services to manage customer identities fully integrated into business processes. It is an integrated identity platform providing single sign-on, single user profile and consent management, and extension to IoT devices. SIM is compatible with several identity standards, including SAML, OAuth, OpenID Connect, SCIM and FIDO U2F.

Microsoft Active Directory Federation Services (AD FS) [134] is a standards-based service that allows the secure sharing of identity information between trusted business partners across the Internet. AD FS is Microsoft's implementation of the WS-Federation Passive Requestor Profile protocol. It provides single sign on, centralized federated partner management, and an extensible architecture.

A schematic comparison of some of the most relevant solutions for identity federation is shown in Table III.

VII. Large Scale Identity Federations

While most proprietary solutions promote centralized services with a unified identification scheme, in many situations the organizations already have their own identification systems that are not easy to modify. A typical example is the creation of worldwide thematic networks (universities, publishers, education, etc.) grouping institutions or companies with preexisting identification systems, which in some cases are even defined by the regulations of the countries.

Identity federation is the perfect solution for these situations. In a federation, when a user clicks on a Service Provider resource, the request is relayed to her Identity Provider, which authenticates the user and returns the result. They use federated single sign-on software, which releases only enough identity data to allow the Service Provider to make an access decision. Thus, nowadays several federations have started to be deployed, as shown below.

eduGAIN is an international federation service connecting research communities and higher education identity federations around the world, providing a single integration. It links over 2,500 identity providers accessing services from more than 1,700 service providers. It has been developed and operated by GEANT, under a series of projects financed by the European Commission [135], and it uses SAML protocol and the Interoperable SAML 2.0 Profile [136]. Figure 8 shows the architecture of the eduGAIN identity system. Currently, eduGAIN has been extended to provide interfederation services, making it the largest interfederation service in operation.

InCommon Federation is the U.S. education and research identity federation, providing a common framework for trusted shared management of access to online resources [137]. Through InCommon, Identity Providers can give their users single sign-on convenience and privacy protection, while online Service Providers control access to their protected resources. It is an initiative equivalent to eduGAIN, but in USA. It is also SAML-based federation operated by Internet2 that in 2016 served 10 million end-users.

Other examples of large scale identity federations can be found in other thematic networks. Examples are CLARIN Federated Identity [138] for language resources and technology, DARIAH Authentication and Authorization Infrastructure [139] for arts and humanities, and ELIXIR Authentication and Authorization services [140] for life sciences. A complete list of federations can be found at the REFEDS web page [141]

Several identity federations can be further connected to provide Interfederation Identity Services. This is the case of several federations around the world providing unified accesses to educational, research resources and publisher companies in different countries. Examples are: **GakuNin** [142], the Academic Access Management Federation in Japan; **AAF**

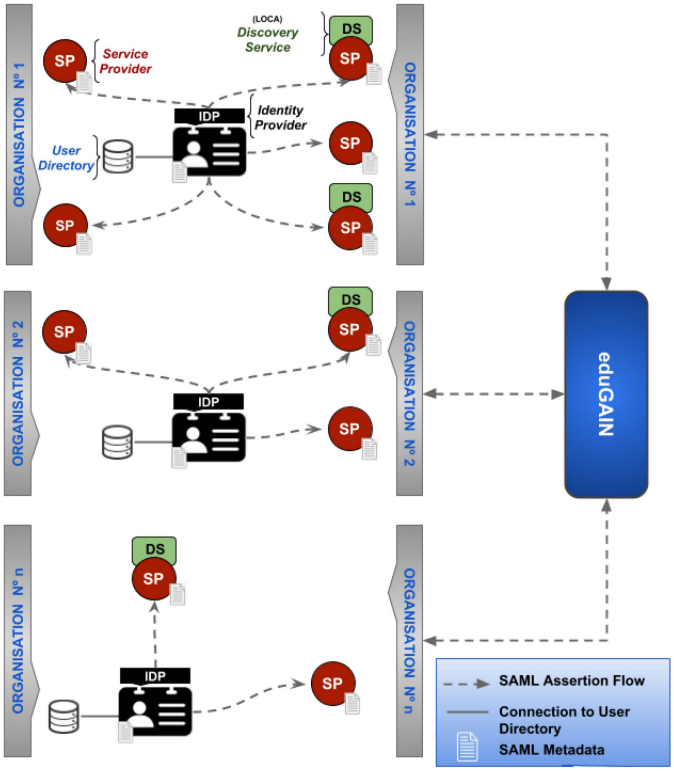

Fig. 8. eduGAIN architecture.

[143], the Australian Access Federation; or **CAF** [144], the Canadian Access Federation. All of them are connected to eduGAIN, that provides interfederation identity services, creating an identification network all around the world.

Large-scale FIM must solve two main problems: interoperability and scalability. Interoperability is needed to interconnect the participating identity federations and to allow Web Single Sign On (Web SSO) through the whole federation. Scalability is needed to allow the federation growing, without having performance losses, by adding more federations or institutions in the federation.

The goal of interoperability is that organizations using different FIMs are able to make cross identification without changing their FIMs. An optimal integration can also allow users to maintain their identities between federations and, at the same time, access to services of other federations. Thus, an interoperability solution has to offer mechanisms to solve different aspect related to protocol interoperability, identity management, trust and access control (authorization mechanisms), achieving bidirectional matching between identities. Two solutions are possible: Designing a new model and forcing the federations to change all their production systems to adapt them to a new scheme; Creating a new federation service to provide the equivalence between entities attributes and all the required mechanisms to translate and do the matching between systems in the federation. The last one is the only feasible to promote federated Id management for already existing large scale federations. Thus, to promote interoperability the federation has to define the set of attributes that should be exchanged and the protocols for allowing full communication between the different federated authentication systems. For example, interoperability proposals to connect PAPI and Shibolet may be seen in [145] [146]. Bidirectional filters to implement the solution in [145] can be found in the PAPI project Web page [147].

The complexity of interoperability services is manageable provided that all the members of the federation use the same authentication and authorization protocol (SAML, OAuth, OpenID Connect, ...). However, the complexity grows if the federation, or the interfederation, protocols are different. For this reason, the Interoperable SAML 2.0 Profile (SAML2Int) is the only SAML 2.0 profile allowed in eduGAIN, and other federations, like eIDAS, STORK or InCommon (described in next sections) are also SAML-based federations. In [148], the authors propose to use an intermediate entity to connect several federations like eduGAIN, STORK, and eIDAS.

Scalability is directly related to the architectural design of the federation network. Two main architectural approaches are currently used for identity federations [149]:

- Full mesh federations. In this structure everything is distributed and there is no need for a central component. Thus, every organization in the federation operates their own Identity Provider (IdP) connected to a local user database and an arbitrary number of Service Providers (SP). This structure is the most common due to its simplicity, distributed deployment, and availability features. Moreover, it allows easily incremental growing of the federation.
- Hub & Spoke with distributed or central login. This architecture relies on a central hub or proxy via which all security assertions are sent. The hub serves as a Service Provider versus the Identity Providers and as an Identity Provider versus the Service Providers in the federation. In distributed login, each organization still operates their own Identity Provider connected to a local user database, but, both, the Identity Provider and the Server Provider must contact the central hub for Identification. With centralized login, there is only one single Identity Provider in the federation, connecting to all user databases. This architecture, specially the central login version, may have availability problems due to the existence of a single point of failure. Scalability is also limited due to the capacity of the single elements.

Federated identity management in full mesh federations provides very good scalability properties, as Id management can be made at several levels (SP, local IdP, federated IdP, ...), which allows to distribute the load and avoiding single points of failure. The number of steps needed for authentication is however proportional to the federation levels. For example, in STORK it was usually 2, but in interfederation identification, like those created using eduGAIN would be at least 3. Thus, the latency of the operation will be directly proportional to the depth of the federation. However, this latency can be reduced

by using proxies in different places.

VIII. European Identity Federation Initiative

Since the beginning of the XXI century many European countries have developed eIdentification systems for their nationals, leading to a very diverse landscape. Examples are the German nPA, the Dutch DigiD or the Spanish eDNI. As a result, the idea of using Federated Identity Management across Europe become a reality. Thus, along the FP7 programe several projects were carried out to develop a pan-European eID interoperability infrastructure allowing cross-border identification using national Ids.

STORK [150] and STORK 2.0 [151] projects showed that it was possible to use national eIDs in cross border use cases by designing a system using a Pan European Proxy Service (PEPS), which acts as a single gateway and intermediary for foreign eIDs towards domestic Service providers, or relying on a middleware that allows a distributed implementation of the identification system. FutureID project [152] built a comprehensive, flexible, privacy-aware, and ubiquitously usable identity management infrastructure for Europe that is integrated with existing eID technology and trust infrastructures, emerging federated identity management services and modern credential technologies to provide a user-centric system. The feasibility and results of those projects were the base for a proposal of an identity federation solution in anticipation of the adoption of the Regulation (EU) 910/2014 [153], the so-called eIDAS Regulation, which should be in force in all Member States at the end of 2018. This regulation provides a common method to enable secure and seamless electronic interactions between businesses, citizens, and public authorities in Europe. The eIDAS Regulation ensures that people and businesses can use their own national electronic identification schemes (eIDs) to access public services in other EU countries where eIDs are available. As a result, it creates an European level trust network on electronic services (e.g. digital signatures) by ensuring that they will work across borders and have the same legal status as traditional paper based processes [154].

The federation solution consists of a network of Member States, each one subscribed to a federated operator, namely eIDAS Node. The approach that has been taken is that each Member State has to deploy an eIDAS Node, which has the role of Identity Provider for the national electronic identification scheme (eID) from any other country. All Service Providers participating in the network must be subscribed to the eIDAS Node of this country. This way, every citizen recognized by a member state is to be recognized within the trust network at European level, enabling the consumption of services in other member states that, until now were not allowed, or whose concession was tedious. This is a very ambitious approach, since it enables cross-border authentication of Member States citizens without the need to unify the authentication method (eID Scheme) of the member states participating in the Identity Federation.

The deployment of the nodes to create this trust network is in charge of a governmental institution of the Member State (e.g. a ministry), since it is going to be coupled to the national public eID Scheme and authentication assertions are going to be managed. A flaw in the system or a disclosure of confidential information could compromise the security of the infrastructure. Technically, the eIDAS node implementation is still evolving, as the deployment is going on and several works are evaluating the behavior of the system. For example, a security analysis of eIDAS has been published in [155] showing that 7 of the 15 European eID services deployed were still vulnerable to Denial-of-Service (DoS) and Server Side Request Forgery (SSRF). Another example is the work presented in [156] that proposes an extension of eIDAS that enables the protocol to authenticate further transaction data, such as phone numbers or PGP keys.

So far, the creation of this trust infrastructure has only addressed the identification issue from the point of view of the public sector. That is, only public services can be consumed by natural persons. The behavior of this infrastructure is not defined for legal entities (e.g. companies), which are organizations not linked to the public sector. But the technical specification of the eIDAS nodes already takes into account the attributes to be managed for the authentication of legal entities according to the eIDAS Regulation, to be applied in a near future. Examples of these attributes are the legal name, the legal identifier, the legal address, the tax reference number, etc.

In parallel to the deployment of the eIDAS infrastructure at European level, the authors are participating in the EU Project eID@Cloud[1], which focuses on the integration of eID in 5 private cloud platforms of different EU Member States to act as Service Providers for private entities. These platforms are in charge of the implementation of a connector prototype to interact with the 5 different Member State eIDAS nodes allowing the subscription from a public and a non-public Service Provider to the federation. The project is based on the technical specifications defined by the CEF and their test eIDAS Node, that provides an already conformant solution to create a trust network between organizations without the need to operate with the public eID Scheme. As already mentioned, one of the most notorious benefits of this initiative is that these specifications already meet the requirements on management of attributes and authentication established by the eIDAS regulation, both for natural and legal entities.

A. CEF eID Specification

The technical specification defines the interaction between two federated operators called eIDAS Nodes. Each eIDAS Node has two operation modes: the Connector assumes the role of requesting cross-border authentication, and the Proxy Service is in charge to provide cross-border authentication. The interconnection between nodes generates a trust network for cross-border authentication. Service and Identity Providers are subscribed to an eIDAS Node that, in its role of federated operator, is able to manage cross-border authentication with

[1] https://www.arcos.inf.uc3m.es/eidcloud

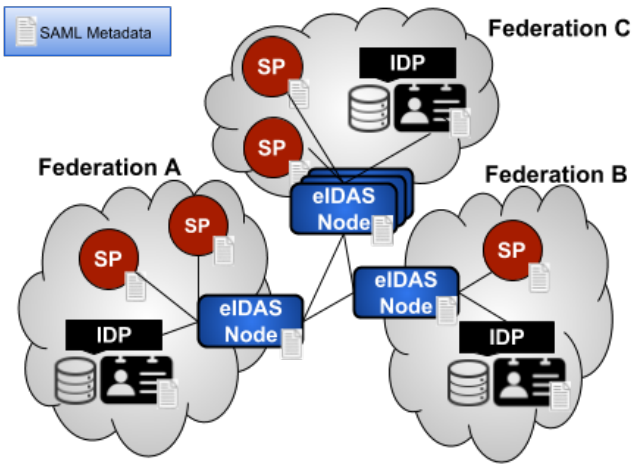

Fig. 9. CEF eIDAS architecture.

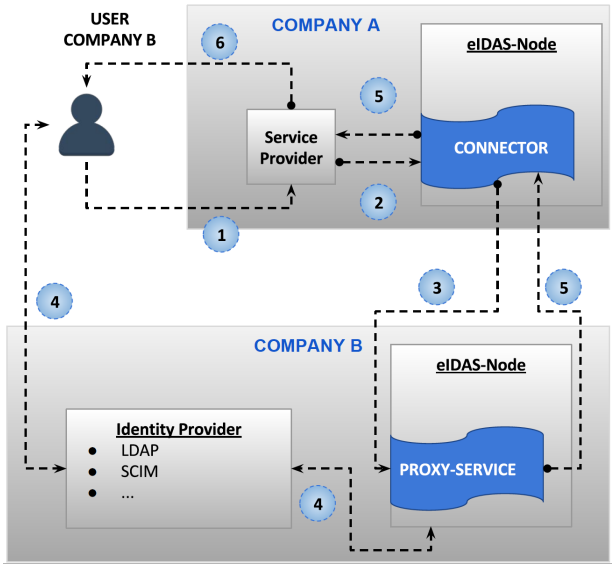

Fig. 10. Message exchange in the CEF eID framework.

other Service and Identity Providers subscribed to other eIDAS Nodes in the same network. The infrastructure is based on SAML 2.0 and defines a set of profiles that may be supported in order for the implementation to be conformant. The specification also includes a common understanding of the parties involved, which is an inherent feature of any federation.

Each SP is subscribed to an eIDAS Node Connector, requesting for authentication, and each IdP is subscribed to an eIDAS Node Proxy Service, issuing authentication assertions. The IdP must support the SAML profiles and bindings defined in the technical specification in order to interoperate with the rest of the parties involved, but the IdP may implement their own authentication and authorization mechanisms with the only need to make the translation between SAML assertions issued and these mechanisms. Each party involved in the system have their specific metadata published, but the IdP discovery service is not covered by the specification, so registration of new parties must be performed in a static and controlled way. In addition, load balancing is supported in the eIDAS Node by creating node replicas that are distributed, installing one master node that is configured and deployed as a load balancer. A scheme of the CEF eIDAS architecture is shown in Figure 9.

B. CEF eID Protocol

In this section, we briefly describe the eID protocol defined by CEF by showing in Figure 10 the flow of requests (messages) exchanged between different actors in the eIDAS architecture, together with a description of the messages in the text below.

1) The citizen requests access to a Service Provider, typically using HTTPS, in her host country using the home eID (Company A).
2) The Service Provider for Company A sends the request, using a HTTP Redirect Binding (or HTTP POST Binding) containing a SAML AuthnRequest, to its own eIDAS-Node Connector.
3) The SAML Request is forwarded by the eIDAS-Node Connector in Company A to the eIDAS-Node Proxy Service of the citizen's Member State (Company B).
4) The citizen authenticates with their country's IdP using their electronic identity and the confirmation is forwarded via the IdP using an HTTP POST Binding to the eIDAS-Node Proxy Service. Depending on the implementation there may be two additional steps within step 4:
   a) for the citizen to select the attributes to be provided (therefore giving consent);
   b) for the citizen to agree on the values of the attributes to be given.
5) The eIDAS-Node Proxy Service sends a SAML Response containing an encrypted SAML Assertion to the requesting eIDAS-Node Connector, which forwards the response to the Service Provider.
6) The Service Provider grants access to the citizen.

IX. eIDAS Performance Evaluation

We have made experiments to evaluate the performance of the eIDAS servers and the overhead of the point-to-point protocol. In this section, the experiments made and their results are shown. First, in order to understand the performance, we define the metrics used for the evaluation. Second, we define four test scenarios (local, organization, federated EU and federated Mexico). Finally, we have run the same set of experiments in the four scenarios, to measure the cost of each actor in the identification service.

A. Performance metrics

The time needed to exchange the different messages of the protocol between the different components of eIDAS Node, shown in Table IV, have been used as metrics to evaluate

TABLE IV
PROTOCOL TIME PARAMETERS.

Param	Description
TSR	Time for signing request reception by eIDAS node.
TVR	Time for validating the request generated by TSR.
TUA	Time for validating the client's signature to confirm their ID.
$TLGN$	Time associated with logging in to the IDP.
$TIDP$	Time associated with identity provider response against TLGN
TDA	Time for decoding the response generated by the IDP.
TRA	Time associated with the response of decryption attributes.

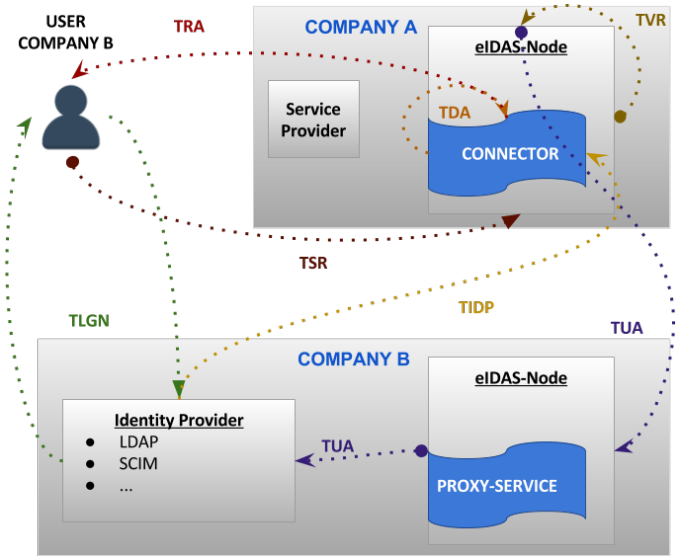

Fig. 11. Message Exchange with metrics associated.

eIDAS Node performance. Figure 11 compares the flow of messages and their relation to the associated metrics.

Based on our experience, we observe that, typically, the chain of inequalities reported in Equation 1 holds. As we can observe, we state that the TLGN process is the step that should require more time, due to the communication and the validations that must be performed.

$$TLGN > TSR > TRA > TIDP \sim TDA \sim TUA > TVR \quad (1)$$

The equation reflects that most of the execution time for a request will be used to process SAML claims (encryption and decryption) to ensure that the authentication and authorization process is valid. Time for login is predominant when passwords are used, however it could be reduced using other methods such as secure tokens or biometry.

Consequently, we state the full time to process a request to eIDAS as in Equation 2:

$$T_{eIDAS} = TSR + TVR + TUA + TLGN + TDA + TIDP + TRA \quad (2)$$

B. Experimental environment

The eIDAS identity federation can be used in heterogeneous environments due to the use of SAML and because of having

TABLE V
FEATURES OF THE SERVER NODES USED IN THE EXPERIMENTS.

Element	Local, organization, EU	Mexico
CPU	Xeon E5-2686v4, 12/24 Cores	Xeon E5645, 6/12 Cores
RAM	32 GiB DDR3	12.00 GB DDR3
HDD	4TB SATA 6Gb/s	2TB SATA 6Gb/s
Network	1 Gbps	1 Gbps

defined a set of solid attributes and standards that all organizations can use to validate different users. Thus, to evaluate the behavior in different scenarios, we have defined four different test scenarios, that are described below, where we performed the experiments:

- Local Environment. The first scenario we analyzed refers to an installation running the eIDAS node and the clients in the same node. This scenario is designed to study performance of the eIDAS node alone, avoiding the overhead of the network and the other servers. It can be used as a baseline, since there are no other external elements that may interfere with the operation of eIDAS.
- Organization Environment. The second experimental scenario was designed to evaluate intra-organization identification performance. It was aimed to analyze how the eIDAS Node behaves in a private cloud environment, where there are many users using different resources. This environment is representative of the organization of ICT services in many companies nowadays, therefore the experiment results should provide interesting information on the evolution of the eIDAS Node according to the variation of the number of clients and the network overhead.
- Federated EU Environment. In the third scenario, we created a federated environment joining connecting an eIDAS node in Frankfurt (Germany) to University Carlos III of Madrid (Spain). This scenario is designed to evaluate the behavior of eIDAS Nodes in more realistic conditions, as they will be initially deployed in EU countries. Average latency of the network is 20 ms.
- Federated Mexico Environment. In the fourth scenario, we created a federated environment using a remote organization in Cinvestav (Mexico), connected to University Carlos III of Madrid (Spain). We are employing a partner in Mexico aiming to get worst-case results due to network distance. Average network latency is 120 ms.

A single eIDAS Node was set up for each environment to serve client requests arriving from different organization subdomains. Table V summarizes the environments where we are performing the evaluation. The client is the same than in all environments. The server in the EU is the same than in local and organization environments. The server in Mexico is similar, but it has only 6 cores instead of 12. Again, both client and server nodes are based on Linux Ubuntu 16.04 LTS. The eIDAS node used was version 1.4.

C. Experiments

For each environment, we run two kind of experiments, scalability and performance, as described below. Every ex-

periment was executed 10 times and the average results and standard deviation were computed. Average results are shown in the evaluation results section. Certificates were used for the verification of the eIDAS nodes each time a connection is made. In our experiments, users are identified through user id and password.

Scalability with independent clients. To evaluate the scalability of the eIDAS Node to serve eID requests from new clients, a stress test was executed on the four scenarios by varying the number of clients from 1 to 64. For each experiment, we run a set of independent clients executing them in parallel (simultaneously). Each client sent a new identification request to the eIDAS node. For those tests, we studied the breakdown of the steps of an eID request, throughput of the servers, and the use of CPU and RAM memory.

We run two kind of experiments for the eIDAS nodes: a single process server and a server with 3 replicas, one of them acting also as load balancer using round-robin. In every experiment the eIDAS nodes were restarted to avoid warm cache effects.

Performance study for a client session. This experiment measures the time needed in the eIDAS node to serve an ID request for a client that has been already identified. As the eIDAS node has a temporal cache for some ID data, if the cache is warm, time should be reduced considerably, as compared with the values of the independent clients.

This experiment consisted on running a client that issued id requests at the maximum possible rate, which means that there were no delays between successive requests. The number of ID requests varied from 1 to 64 and for the results we have removed the ID number 1 to avoid the cold cache effect.

D. Scalability evaluation

We analyzed the overhead of the different actors participating in the identification process by making experiments increasing the number of simultaneous clients for one server in the four environments described and breaking down the eID request time considering the messages shown in Figure 10.

Figure 12(top) shows a breakdown indicating the average time per metric and number of clients for the *local environment*. In the figure, we can observe how the time of the request increases with the number of clients in a significant way for more than 8 clients. This is due to the saturation of the computing resources of the server. This effect is more important on the time corresponding to TSR, which represents the cost in terms of time for signing request reception by eIDAS node, TRA that represents the cost in terms of time for sending the answer to the client, and $TLGN$, which represents the time for login. The effect of TSR occurs because the eIDAS Node queues up the requests while the rest of the received jobs are being processed, increasing the waiting time of the requests. TRA is due also to a similar effect. $TLGN$ grows due to the saturation of the server CPU. As may be seen, TVR, the time for validating the request generated by TSR is much more smaller and remains almost stable.

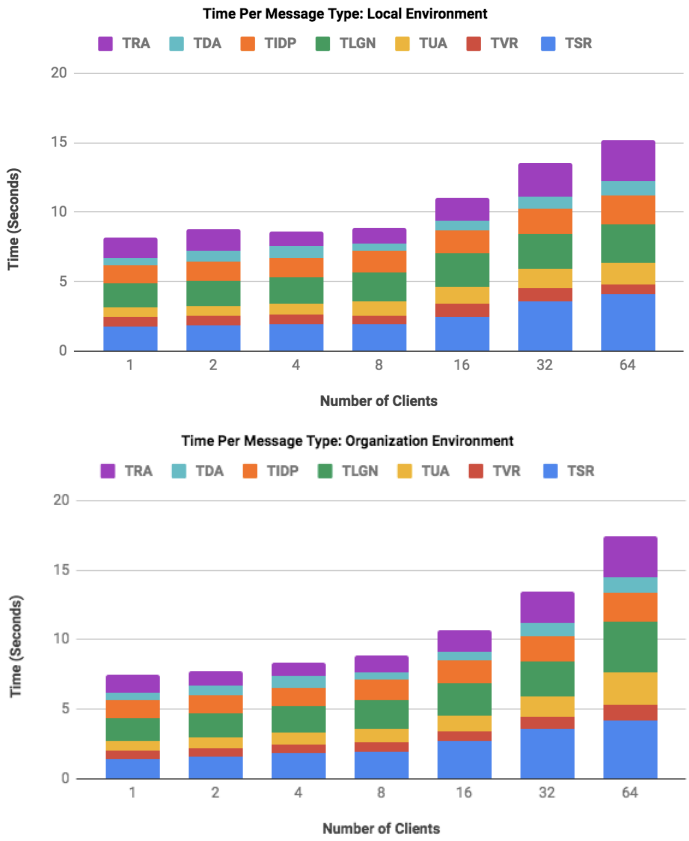

Fig. 12. Effect of increasing number of clients on the time per message type of each client request in the local (top) and organization (down) environments.

Figure 12(bottom) shows the time breakdown of an eID request, while varying the number of clients, for the *organization environment*. The behavior is similar to the local scenario, but consumed time grows slightly due to the network latency. The figure shows a similar behavior for the first 8 clients, but starts growing beyond that number, due to the saturation of the cores of the server. Again, the penalty is identified in the time consumed for signing and validation of requests in the server.

Figure 13(top) plots the breakdown of the cost of the steps needed to achieve the identification for an increasing number of clients in case of *Federated EU Environment*. In this environment, TSGN grows steadily compared to the other metrics due to the latency between clients and eIDAS Node (around 20 ms.). Same effect appears in TLGN and TRA. However, the server is still able to attend the simultaneous new eID requests with good scalability. Figure 13(bottom) shows the breakdown of the cost of the steps needed to achieve the identification for the *Federated Mexico Environment*. In this environment, TSGN significantly grows compared to the other metrics, due to the latency between clients and eIDAS Node This high latency creates a burst effect, as all clients start and send requests to the eIDAS node, but only the first client "pays" for the network transport, causing the rest of clients

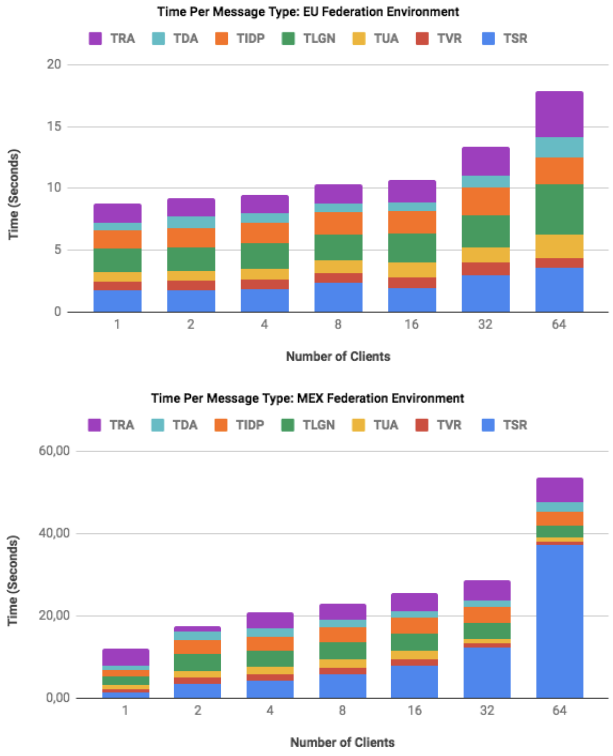

Fig. 13. Effect of increasing number of clients on the time per message type of each client request in federated environment in EU (top) and Mexico (bottom).

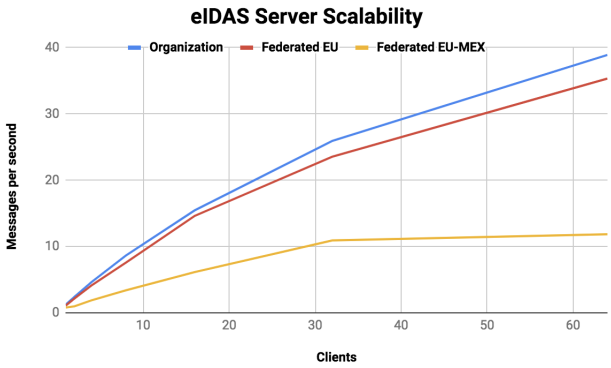

Fig. 14. Scalability of eIDAS nodes. Message throughput in different environments.

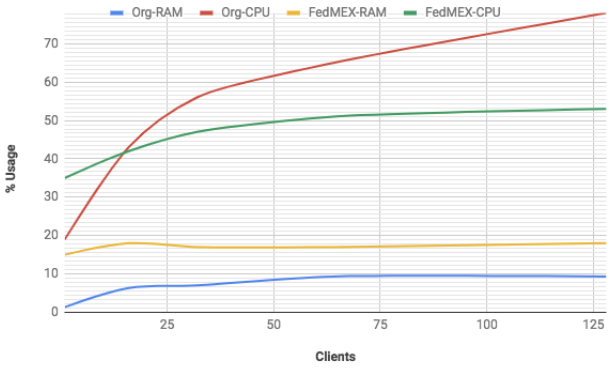

Fig. 15. CPU and RAM Evolution per Requests in organization and federated Mexico Environment

arriving to the eIDAS node as a burst almost at the same time, causing congestion in the server side, which has to queue the requests in order to process them.

The former results are consistent with those in Figure 14, which shows the scalability in messages per second processed by the servers of organization, federated EU and federated Mexico environments respectively. We are using messages as we can measure very well that metric and because they map directly to clients, as each eID client request has a fixed number of messages to the eIDAS nodes due to the protocol. As may be seen, the throughput of organization and federated EU are similar, but the server in Mexico provides a poor scalability, mostly due to network conditions that presents a bottleneck in the link from EU to America and the congestion created by the burst effect.

As far as CPU and RAM usage is concerned (see Figure 15), in our experiments we identified the same behavior than in the previous scenarios. The results show a consistent behavior for the evolution of the CPU and memory in the servers according to the number of clients that access the node and the conditions of the network. Basically, network latency in the organization environment is very low and it is not affecting performance, as clients arrive in sequence and the server is able to attend the requests. This effect can be seen in the incremental growing of CPU usage due to the eIDAS node execution time for the requests. The situation in Mexico is different due to bursting effect and network latency, which is preventing good usage of CPU as many threads in the server as blocked waiting for the network. RAM behavior has a growing evolution line as the server loads new requests data in memory, but it is not saturated. which is similar in both environments. Proportional usage is lower in EU server as the memory size is larger.

eIDAS Node Load Balancing

Figure 16 plots the comparative service time between a single node and a load balanced system composed by three nodes. In the figure, we observe that with a small number of clients (four clients), using a single eIDAS Node is similar to the load balancing approach. However, as the number of clients grows, the time needed to respond a request is always lower in the load balanced system. For the EU environment, we obtained up to $2\times$ performance in the load balanced server. However, in the Mexico environment, scalability was almost $3\times$, compared to the baseline. Thus, we can conclude that the load balance mechanisms provide good results with very low overhead.

E. Performance evaluation

Table VI shows the average time needed for a user identification after she has been identified at least once, which means that the cache of the eIDAS node is storing some data for the user during some time.

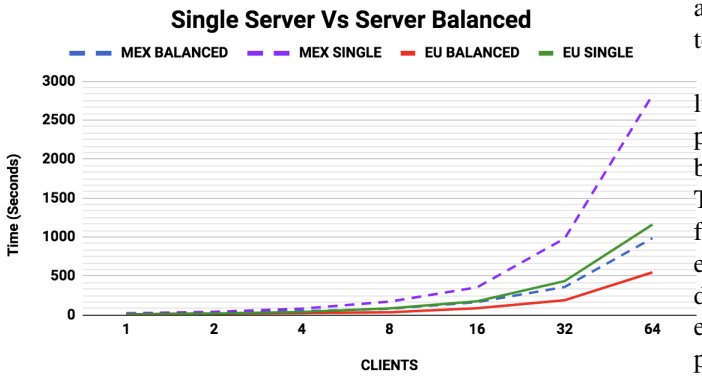

Fig. 16. Scalability of the eIDAS node activating three servers and load balance in a federated and local environment.

TABLE VI
PERFORMANCE RESULTS DURING A SESSION, WARM CACHE.

Messages	Local	Organizational	Federated EU	Federated Mex
TSR	0.07	0.09	0.3	1.04
TVR	0.07	0.09	0.16	0.74
TUA	0.07	0.08	0.16	0,71
TLN	0.13	0.16	0.41	1.6
TIDP	0.14	0.14	0.39	1.75
TDA	0.21	0.23	0.46	1.06
TRA	0.38	0.45	0.69	2.52
TOTAL	1.07	1.24	2.58	9.42
STD	5.61%	6.68%	5.30%	1.00%

As may been, times are considerably reduced for all environments, being around 1 second for local and organizational environment, 2.5 seconds for federated EU nodes and 9 seconds for federated environment in Mexico. Furthermore, time increases in the last environment are due to the network latency, which cannot be avoided. The results shown are logical as most of the CPU time is dedicated to cipher and decipher operations, which are simplified when there are already results cached from a valid previous identification.

In these evaluations, the number of CPU cores is irrelevant, as we are metering only one Id request and only one instance of the eIDAS server is running. This is the reason because local and organizacional are so similar. However, given the scalability results of Figure 16, we can predict that it would be strongly enhanced in real operation of the system with warm caches and a large number of clients.

X. CONCLUSION

Digital identity management is a relevant security subject to access ubiquitous on-line services, as organizations have to exchange personal information in a secure way for preserving integrity and confidentiality. This problem is even worst when the organizations do not share a centralized architecture for single sign-on, which may be due to the existence of previous eID systems or even to political reasons. The Federated Identity Architecture aims to tackle this problem, allowing each organization to use their own eID. However, a complex architecture must be built to rely the identification of any entity to their original eID server in a secure way.

As shown in this work, there are already many FIA solutions. Examples are eduGAIN and InCommon Federation, popular solutions arising in EU and USA respectively. Both are based on Shibboleth, an open source SAML implementation. The WS-Federation initiative, a standard based on the WS-* family, focuses on the Web Services environment and allows each participant to have its own policies, which, combined, determine the security requirements to communicate. However, most of them cover different aspects of the identification problem, solving in some cases specific problems. Thus, none of these initiatives has consolidated as a unique solution and surely it will remain like that in a near future. To assist users choosing a possible solution, we have presented a survey of FIA, showing main features and making a comparative analysis among them.

The former problem worsens when organizations or countries, already have legacy eID systems, as it is the case of Europe. In this situation, highly distributed FIAs is a good technical and secure approach to provide a feasible solution. In the paper, we have presented the European eID solution, a purely federated identity system that aims to serve almost 500 millions people and that could be extended in mid-term also to companies eID. The system is now being deployed at the EU level and we have presented the basic architecture and evaluated its performance and scalability, showing that the solution is feasible from the point of view of performance, while keeping security constrains in mind.

As shown in this paper, federated identity systems still have research lines open, going from the improvement in performance and scalability to the proposal of new models to make integration easier for large scale organizations. In the case of the European eID system, there are still a lot of open lines, such as managing the evolution of the eID authentication servers, shortening the latency and complexity of the deployment and integrating existing private eID systems.

ACKNOWLEDGEMENTS

This work has been partially supported by the EU under the INEA/CEF Project “Integrating the eIdentification in European cloud platforms according to the eIDAS Regulation (eID@Cloud)”. Action No: 2016-EU-IA-0064.

REFERENCES

[1] G. Alpár, J. Hoepman, and J. Siljee, “The identity crisis. security, privacy and usability issues in identity management,” *CoRR*, vol. abs/1101.0427, 2011. [Online]. Available: http://arxiv.org/abs/1101.0427

[2] J. Jensen, “Federated Identity Management Challenges,” in *2012 Seventh International Conference on Availability, Reliability and Security*, Aug 2012, pp. 230–235.

[3] A. A. Malik, H. Anwar, and M. A. Shibli, “Federated Identity Management (FIM): Challenges and opportunities,” in *2015 Conference on Information Assurance and Cyber Security (CIACS)*, Dec 2015, pp. 75–82.

[4] R. Dhamija and L. Dusseault, “The seven flaws of identity management: Usability and security challenges,” *IEEE Security Privacy*, vol. 6, no. 2, pp. 24–29, March 2008.

[5] L. O'Gorman, "Comparing passwords, tokens, and biometrics for user authentication," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2021–2040, Dec 2003.

[6] J. Reschke, "The 'Basic' HTTP Authentication Scheme," Internet Requests for Comments, RFC Editor, RFC 7617, September 2015.

[7] S. M. Bellovin and M. Merritt, "Limitations of the Kerberos authentication system," *ACM SIGCOMM Computer Communication Review*, vol. 20, no. 5, pp. 119–132, 1990.

[8] R. M. Needham and M. D. Schroeder, "Using Encryption for Authentication in Large Networks of Computers," *Commun. ACM*, vol. 21, no. 12, pp. 993–999, Dec. 1978. [Online]. Available: http://doi.acm.org/10.1145/359657.359659

[9] J. Clark and J. Lawrence Jacob, "A survey of authentication protocol literature: Version 1.0," 12 1997.

[10] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile," Internet Requests for Comments, RFC Editor, RFC 5280, May 2008, http://www.rfc-editor.org/rfc/rfc5280.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5280.txt

[11] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978. [Online]. Available: http://doi.acm.org/10.1145/359340.359342

[12] E. B. Barker, "Digital signature standard (DSS)," *Federal Inf. Process. Stds. (NIST FIPS)*, no. 186-4, 2013. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

[13] S. Chokhani, W. Ford, R. Sabett, C. Merrill, and S. Wu, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework," Internet Requests for Comments, RFC Editor, RFC 3647, November 2003.

[14] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, Nov 1976.

[15] S. Blake-Wilson and A. Menezes, "Authenticated diffie-hellman key agreement protocols." in *SAC: Conference on Selected Areas in Cryptography*. Springer, 1998.

[16] J. R. N. J. F. L. E. B. E. R. Morris J. Dworkin, Elaine B. Barker and J. F. D. Jr., "Advanced encryption standard (AES)," *Federal Inf. Process. Stds. (NIST FIPS)*, no. 197, 2001. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

[17] J. Lynch, "From fingerprints to DNA: Biometric data collection in US immigrant communities and beyond," 2012.

[18] R. Cappelli, M. Ferrara, A. Franco, and D. Maltoni, "Fingerprint verification competition 2006," *Biometric Technology Today*, vol. 15, no. 7, pp. 7 – 9, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0969476507701406

[19] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric recognition: security and privacy concerns," *IEEE Security Privacy*, vol. 1, no. 2, pp. 33–42, Mar 2003.

[20] European Parliament and The Council of 13 December 1999, "Directive 1999/93/EC on a community framework for electronic signatures," 2000.

[21] R. Want, "An introduction to RFID technology," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 25–33, Jan 2006.

[22] "Information technology — Security techniques — A framework for identity management," International Organization for Standardization, International-Standard, Dec. 2011.

[23] E. Bertino and K. Takahashi, *Identity Management: Concepts, Technologies, and Systems*. Artech House, 2011.

[24] S. C. Lee, "An introduction to identity management," SANS Institute InfoSec reading room, Tech. Rep., March 2003. [Online]. Available: https://www.sans.org/reading-room/whitepapers/authentication/an-introduction-to-identity-management-852

[25] K. Cameron, "The laws of identity. 2005," *Microsoft Corporation*, 2009.

[26] S. Clauß and M. Köhntopp, "Identity management and its support of multilateral security," *Computer Networks*, vol. 37, no. 2, pp. 205 – 219, 2001, electronic Business Systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128601002171

[27] Y. Cao and L. Yang, "A survey of identity management technology," in *2010 IEEE International Conference on Information Theory and Information Security*, Dec 2010, pp. 287–293.

[28] "Zentyal," accessed: 2017-12-04. [Online]. Available: http://www.zentyal.com/about-us/

[29] "Windows domain." [Online]. Available: https://technet.microsoft.com/en-us/library/cc977987.aspx

[30] L. Bussard, E. Di Nitto, A. Nano, O. Nano, and G. Ripa, "An approach to identity management for service centric systems." in *ServiceWave*. Springer, 2008, pp. 254–265.

[31] A. Jøsang and S. Pope, "User centric identity management," in *AusCERT Asia Pacific Information Technology Security Conference*. sn, 2005, p. 77.

[32] T. E. Maliki and J. M. Seigneur, "A survey of user-centric identity management technologies," in *The International Conference on Emerging Security Information, Systems, and Technologies (SECUREWARE 2007)*, Oct 2007, pp. 12–17.

[33] A. Pérez-Méndez, F. Pereñíguez-García, R. Marín-López, G. López-Millán, and J. Howlett, "Identity federations beyond the web: A survey," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 2125–2141, Fourthquarter 2014.

[34] D. Chadwick, "Federated identity management," *Foundations of security analysis and design V*, pp. 96–120, 2009.

[35] S. S. Shim, G. Bhalla, and V. Pendyala, "Federated identity management," *Computer*, vol. 38, no. 12, pp. 120–122, 2005.

[36] R. Hörbe, "SAML metadata guidance version 1.0," OASIS Security Services (SAML) TC, Working Draft, 2014.

[37] J. Torres, M. Nogueira, and G. Pujolle, "A survey on identity management for the future network," *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 787–802, Second 2013.

[38] M. Goodner, M. Hondo, A. Nadalin, M. McIntosh, and D. Schmidt, "Understanding ws-federation," *Microsoft and IBM*, 2007.

[39] M. P. Machulak, E. L. Maler, D. Catalano, and A. van Moorsel, "User-managed Access to Web Resources," in *Proceedings of the 6th ACM Workshop on Digital Identity Management*, ser. DIM '10. New York, NY, USA: ACM, 2010, pp. 35–44. [Online]. Available: http://doi.acm.org/10.1145/1866855.1866865

[40] H. Takabi, J. B. D. Joshi, and G. J. Ahn, "Security and privacy challenges in cloud computing environments," *IEEE Security Privacy*, vol. 8, no. 6, pp. 24–31, Nov 2010.

[41] R. T. Fielding, J. Gettys, J. C. Mogul, H. F. Nielsen, L. Masinter, P. J. Leach, and T. Berners-Lee, "Hypertext transfer protocol – HTTP/1.1," Internet Requests for Comments, RFC Editor, RFC 2616, June 1999. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2616.txt

[42] E. Rescorla, "HTTP over TLS," Internet Requests for Comments, RFC Editor, RFC 2818, May 2000, http://www.rfc-editor.org/rfc/rfc2818.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2818.txt

[43] T. Dierks and E. Rescorla, "The transport layer security (TLS) protocol version 1.2," Internet Requests for Comments, RFC Editor, RFC 5246, August 2008, http://www.rfc-editor.org/rfc/rfc5246.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5246.txt

[44] M. Goodner and T. Nadalin, "Web services federation language (ws-federation) version 1.2," *OASIS Standard (May 2009)*, 2009.

[45] M. Goodner, M. Hondo, A. Nadalin, M. McIntosh, and D. Schmidt, "Understanding ws-federation," *Microsoft and IBM*, 2007.

[46] S. Cantor, I. J. Kemp, N. R. Philpott, and E. Maler, "Assertions and protocols for the OASIS security assertion markup language," *OASIS Standard (March 2005)*, 2005.

[47] M. Gudgin, M. Hadley, N. Mendelsohn, J.-J. Moreau, H. F. Nielsen, A. Karmarkar, and Y. Lafon, "Simple object access protocol (SOAP) 1.2," *World Wide Web Consortium*, 2003.

[48] H. R. N. Carroline Ramli and F. Nielson, "The logic of XACML," *Science of Computer Programming*, vol. 83, no. 1, pp. 80–105, April 2014.

[49] D. Hardt, "The OAuth 2.0 authorization framework," Internet Requests for Comments, RFC Editor, RFC 6749, October 2012. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6749.txt

[50] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, and C. Mortimore, "OpenID connect core 1.0 incorporating errata set 1," *The OpenID Foundation, specification*, 2014.

[51] T. Bray, "The JavaScript Object Notation (JSON) Data Interchange Format," Internet Requests for Comments, RFC Editor, RFC 7159, March 2014, http://www.rfc-editor.org/rfc/rfc7159.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7159.txt

[52] "OpenID connect federation," accessed: 2018-01-21. [Online]. Available: http://openid.net/specs/openid-connect-federation-1_0.html

[53] "The Liberty Alliance," accessed: 2017-12-15. [Online]. Available: http://www.projectliberty.org/

[54] "Liberty Alliance ID-FF 1.2 specifications," accessed: 2017-12-15. [Online]. Available: http://www.projectliberty.org/resource_center/specifications/\\liberty_alliance_id_ff_1_2_specifications/

[55] Kantara, "Kantara Initiative," 2018. [Online]. Available: https://kantarainitiative.org/

[56] E. Maler, D. Catalano, M. Machulak, and T. Hardjono, "User-managed access (UMA) profile of OAuth 2.0," 2015.

[57] P. Hunt, K. Grizzle, M. Ansari, E. Wahlstroem, and C. Mortimore, "System for Cross-domain Identity Management: Protocol," Internet Requests for Comments, RFC Editor, RFC 7644, September 2015.

[58] M. Miculan and C. Urban, "Formal analysis of Facebook Connect single sign-on authentication protocol," in *Proc. Student Research Forum, 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2011)*, vol. 11, 2011, pp. 22–28.

[59] ——, "Formal analysis of Facebook Connect single sign-on authentication protocol," in *Proc. Student Research Forum of 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2011)*, vol. 11, 2011, pp. 22–28.

[60] Google, "Google Identity Platform," 2018. [Online]. Available: https://developers.google.com/identity/

[61] "Eclipse Higgins," accessed: 2018-01-04. [Online]. Available: https://projects.eclipse.org/projects/technology.higgins

[62] M. W. and V. P., "Efficient U-Prove implementation for anonymous credentials on smart cards," in *Security and Privacy in Communication Networks. SecureComm 2011. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 2012, pp. 243–260.

[63] "VOOT protocol," accessed: 2017-11-30. [Online]. Available: https://openvoot.org/

[64] M. Kremers, "VOOT specifications," 2014, accessed: 2018-01-04. [Online]. Available: https://wiki.geant.org/display/gn3pjra3/VOOT+specifications

[65] R. Castro-Rojo and D. R. López, "The PAPI system: point of access to providers of information," *Computer Networks*, vol. 37, no. 6, pp. 703–710, 2001.

[66] C. Project, "CAS Protocol," 2016, accessed: 2018-01-03. [Online]. Available: https://apereo.github.io/cas/4.2.x/protocol/CAS-Protocol.html

[67] J. Camenisch and E. Van Herreweghen, "Design and implementation of the Idemix Anonymous Credential System," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, ser. CCS '02. New York, NY, USA: ACM, 2002, pp. 21–30. [Online]. Available: http://doi.acm.org/10.1145/586110.586114

[68] A. Pérez, G. López, Ó. Cánovas, and A. F. Gómez-Skarmeta, "Formal description of the SWIFT identity management framework," *Future Generation Computer Systems*, vol. 27, no. 8, pp. 1113–1123, 2011.

[69] S. Simpson and T. Grossß, *A Survey of Security Analysis in Federated Identity Management*. Cham: Springer International Publishing, 2016, pp. 231–247. [Online]. Available: https://doi.org/10.1007/978-3-319-55783-0_16

[70] A. Jøsang, "Identity management and trusted interaction in internet and mobile computing," *IET Information Security*, vol. 8, pp. 67–79(12), March 2014. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/iet-ifs.2012.0133

[71] J. Werner, C. M. Westphall, and C. B. Westphall, "Cloud identity management: A survey on privacy strategies," *Computer Networks*, vol. 122, no. Supplement C, pp. 29 – 42, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128617301664

[72] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence, "AAA authorization framework," Internet Requests for Comments, RFC Editor, RFC 2904, August 2000. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2904.txt

[73] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote authentication dial in user service (RADIUS)," Internet Requests for Comments, RFC Editor, RFC 2865, June 2000. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2865.txt

[74] C. Rigney, "RADIUS accounting," Internet Requests for Comments, RFC Editor, RFC 2866, June 2000, http://www.rfc-editor.org/rfc/rfc2866.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2866.txt

[75] V. Fajardo, J. Arkko, J. Loughney, and G. Zorn, "Diameter base protocol," Internet Requests for Comments, RFC Editor, RFC 6733, October 2012, http://www.rfc-editor.org/rfc/rfc6733.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6733.txt

[76] B. Aboba and A. Palekar, "Ieee 802.1 x and radius security," *Submissions to IEEE*, vol. 802, 2001.

[77] W. A. Arbaugh *et al.*, *Real 802.11 security: Wi-Fi protected access and 802.11 i*. Addison-Wesley Longman Publishing Co., Inc., 2003.

[78] D. S. P. Calhoun, G. Zorn and D. Mitton, "Diameter network access server application. rfc 40005," The Internet Society, 2005. [Online]. Available: https://tools.ietf.org/html/rfc4005

[79] A. Hosia, "Comparison between RADIUS and Diameter," *T-110.551 Seminar on Internet working*, 2003.

[80] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, "Extensible authentication protocol (EAP)," Internet Requests for Comments, RFC Editor, RFC 3748, June 2004, http://www.rfc-editor.org/rfc/rfc3748.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3748.txt

[81] K. Zeilenga, "Lightweight directory access protocol (LDAP): Technical specification road map," Internet Requests for Comments, RFC Editor, RFC 4510, June 2006, http://www.rfc-editor.org/rfc/rfc4510.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4510.txt

[82] J. Sermersheim, "Lightweight directory access protocol (LDAP): The protocol," Internet Requests for Comments, RFC Editor, RFC 4511, June 2006, http://www.rfc-editor.org/rfc/rfc4511.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4511.txt

[83] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, "The Kerberos network authentication service (v5)," Internet Requests for Comments, RFC Editor, RFC 4120, July 2005, http://www.rfc-editor.org/rfc/rfc4120.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4120.txt

[84] B. C. Neuman and T. Ts'o, "Kerberos: an authentication service for computer networks," *IEEE Communications Magazine*, vol. 32, no. 9, pp. 33–38, Sept 1994.

[85] S. Hartman, K. Raeburn, and L. Zhu, "Kerberos principal name canonicalization and cross-realm referrals," Internet Requests for Comments, RFC Editor, RFC 6806, November 2012.

[86] I. Cervesato, A. D. Jaggard, A. Scedrov, and C. Walstad, "Specifying Kerberos 5 cross-realm authentication," in *Proceedings of the 2005 workshop on Issues in the theory of security*. ACM, 2005, pp. 12–26.

[87] S. Sakane, M. Ishiyama, and S. Zrelli, "Problem statement on the cross-realm operation of Kerberos," 2010.

[88] J. Linn, "Generic security service application program interface version 2, update 1," Internet Requests for Comments, RFC Editor, RFC 2743, January 2000.

[89] T. Bialaski and M. Haines, "Ldap in the Solaris operating environment: Deploying secure directory," 2003.

[90] A. Melnikov and K. Zeilenga, "Simple authentication and security layer (SASL)," Internet Requests for Comments, RFC Editor, RFC 4422, June 2006.

[91] S. Josefsson and N. Williams, "Using generic security service application program interface (GSS-API) mechanisms in simple authentication and security layer (SASL): The GS2 mechanism family," Internet Requests for Comments, RFC Editor, RFC 5801, July 2010.

[92] E. Lear, H. Tschofenig, H. Mauldin, and S. Josefsson, "A simple authentication and security layer (SASL) and generic security service application program interface (GSS-API) mechanism for OpenID," Internet Requests for Comments, RFC Editor, RFC 6616, May 2012.

[93] S. Winter and J. Salowey, "Update to the extensible authentication protocol (EAP) applicability statement for application bridging for federated access beyond web (ABFAB)," Internet Requests for Comments, RFC Editor, RFC 7057, December 2013.

[94] R. Marín-López, F. Pereñíguez, G. López, and A. Pérez-Méndez, "Providing EAP-based Kerberos pre-authentication and advanced authorization for network federations," *Computer Standards & Interfaces*, vol. 33, no. 5, pp. 494 – 504, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092054891100016X

[95] R. L. Morgan, S. Cantor, S. Carmody, W. Hoehn, and K. Klingenstein, "Federated security: The Shibboleth approach," *Educause Quarterly*, vol. 27, no. 4, pp. 12–17, 2004. [Online]. Available: https://www.learntechlib.org/p/103716

[96] "Shibboleth wiki." [Online]. Available: https://wiki.shibboleth.net/confluence/display/SHIB2/Home

[97] "Unity," accessed: 2017-11-29. [Online]. Available: http://www.unity-idm.eu/

[98] "Aarc wiki: Unity." [Online]. Available: https://wiki.geant.org/display/AARC/UNITY

[99] "SimpleSAMLphp," accessed: 2017-11-29. [Online]. Available: https://simplesamlphp.org/

[100] P. Aubry, V. Mathieu, and J. Marchal, "Open-source single sign-on with CAS (central authentication service)," *Actes of EUNIS*, pp. 1318–1882, 2004.

[101] "Central authentication service (cas)." [Online]. Available: https://www.apereo.org/projects/cas

[102] "Apereo cas - enterprise single sign on for all earthlings and beyond," accessed: 2018-1-8. [Online]. Available: https://apereo.github.io/cas/

[103] "Forgerock identity platform. access management (based on the openam project)," accessed: 2018-1-16. [Online]. Available: https://www.forgerock.com/platform/access-management/

[104] "mod_auth_mellon respository." [Online]. Available: https://github.com/UNINETT/mod_auth_mellon

[105] "Globus web site," accessed: 2017-11-29. [Online]. Available: http://toolkit.globus.org/toolkit/

[106] I. Foster, *Globus Toolkit Version 4: Software for Service-Oriented Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 2–13. [Online]. Available: https://doi.org/10.1007/11577188_2

[107] J. Basney, M. Humphrey, and V. Welch, "The MyProxy online credential repository," *Software: Practice and Experience*, vol. 35, no. 9, pp. 801–816, 2005.

[108] W. Allcock, J. Bresnahan, R. Kettimuthu, M. Link, C. Dumitrescu, I. Raicu, and I. Foster, "The Globus striped GridFTP framework and server," in *Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*, ser. SC '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 54–. [Online]. Available: https://doi.org/10.1109/SC.2005.72

[109] J. Bresnahan, M. Link, G. Khanna, Z. Imani, R. Kettimuthu, and I. Foster, "Globus GridFTP: what's new in 2007," in *Proceedings of the first international conference on Networks for grid applications*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007, p. 19.

[110] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, and S. Tuecke, "Security for grid services," in *High Performance Distributed Computing, 2003. Proceedings. 12th IEEE International Symposium on*, June 2003, pp. 48–57.

[111] T. Barton, J. Basney, T. Freeman, T. Scavo, F. Siebenlist, V. Welch, R. Ananthakrishnan, B. Baker, M. Goode, and K. Keahey, "Identity federation and attribute-based authorization through the Globus toolkit, shibboleth, gridshib, and myproxy," in *5th Annual PKI R&D Workshop*, vol. 4, 2006.

[112] "Moonshot." [Online]. Available: https://www.jisc.ac.uk/rd/projects/moonshot

[113] J. Howlett, S. Hartman, H. Tschofenig, and J. Schaad, "Application bridging for federated access beyond web (ABFAB) architecture," Internet Requests for Comments, RFC Editor, RFC 7831, May 2016.

[114] "Moonshot wiki," accessed: 2018-1-9. [Online]. Available: https://wiki.moonshot.ja.net/

[115] "Virtual organization membership service (voms)." [Online]. Available: http://italiangrid.github.io/voms/index.html

[116] "Aarc wiki: Voms." [Online]. Available: https://wiki.geant.org/display/AARC/VOMS

[117] "Local credential mapping service (lcmaps)." [Online]. Available: https://www.nikhef.nl/grid/lcaslcmaps/lcmaps

[118] "Argus documentation." [Online]. Available: http://argus-documentation.readthedocs.io

[119] "Lasso," accessed: 2017-12-04. [Online]. Available: http://lasso.entrouvert.org/

[120] "Lemonldap ng," accessed: 2018-1-10. [Online]. Available: https://lemonldap-ng.org/

[121] "Higher education external attribute authorities (hexaa)." [Online]. Available: https://geant3plus.archive.geant.net/opencall/Authentication/\\Pages/HEXAA.aspx

[122] "Comanage," accessed: 2017-12-05. [Online]. Available: https://www.internet2.edu/products-services/trust-identity/comanage/

[123] "Grouper," accessed: 2017-12-05. [Online]. Available: https://www.internet2.edu/products-services/trust-identity/grouper/

[124] "Perun." [Online]. Available: https://perun.cesnet.cz/web/index.shtml

[125] "Apache Syncope," accessed: 2018-02-28. [Online]. Available: https://syncope.apache.org/

[126] Evolveum, "midPoint: the identity governance and administration tool," https://evolveum.com/midpoint/, accessed: 2018-04-27.

[127] T. Ferrill, "Okta identity management," *PC Magazine*, May 2017.

[128] "OKTA Multi-Factor Authentication," https://www.okta.com/products/it/, accessed: 2017-12-14.

[129] Onelogin, "OneLogin Web Access Management (WAM)," https://www.onelogin.com/product/web-access-management.

[130] "Auth0," accessed: 2018-03-09. [Online]. Available: https://auth0.com/

[131] R. Link, "RSA SecurID access overview," https://community.rsa.com/docs/DOC-54266.

[132] Microsoft, "Azure Active Directory," https://azure.microsoft.com/en-us/services/active-directory/, accessed: 2018-01-06.

[133] Salesforce, "Salesforce Identity Management Solution," https://www.salesforce.com/products/platform/products/identity/, accessed: 2018-01-05.

[134] M. D. Network, "Active Directory Federation Services," https://msdn.microsoft.com/en-us/library/bb897402.aspx, accessed: 2018-01-06.

[135] "EduGAIN overview," https://www.geant.org/Services/Trust_identity_and_security/eduGAIN/Pages/About-eduGAIN.aspx, accessed: 2018-01-05.

[136] K. Initiative., "Saml v2.0 interoperability deployment profile v1.0 (draft)." 2018. [Online]. Available: https://kantarainitiative.github.io/SAMLprofiles/saml2int.html,

[137] "InCommon identity management federation," https://www.incommon.org/, accessed: 2018-01-05.

[138] "CLARIN federated identity," https://www.clarin.eu/content/federated-identity, accessed: 2018-01-05.

[139] "DARIAH authentication authorization infrastructure," https://wiki.de.dariah.eu/display/publicde/DARIAH+AAI+Documentation, accessed: 2018-01-05.

[140] "ELIXIR authentication and authorisation services," https://www.elixir-europe.org/news/developing-elixir-authentication-and-authorisation-services, accessed: 2018-01-05.

[141] REFEDS, 2018. [Online]. Available: https://refeds.org/federations

[142] N. I. of Informatics, "Academic Access Management Federation in Japan (GakuNin)," https://www.gakunin.jp/en-fed/, accessed: 2018-01-27.

[143] "Australian Access Federation (AAF)," https://aaf.edu.au/, accessed: 2018-01-25.

[144] C. N. Research and E. Network, "Canadian Access Federation (CAF)," https://www.canarie.ca/identity/caf/, accessed: 2018-01-20.

[145] C. Pérez, 2007. [Online]. Available: http://papi.rediris.es/rep/PAPIShib.pdf

[146] L. Catuogno and C. Galdi, "Achieving interoperability between federated identity management systems: A case of study," *Journal of High Speed Networks*, vol. 20, no. 4, pp. 209–221, 2014.

[147] Rediris., "ShibbolethFilter: An implementation that allows PAPI-based federations to interact with shibboleth-based resources." 2011. [Online]. Available: http://papi.rediris.es/java/

[148] E. M. Torroglosa-García and A. F. Skarmeta-Gómez, "Towards interoperabilty in identity federation systems," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 8, no. 2, 2017.

[149] S. Andrėj and L. Hämmerle, "Identity Federtion Architectures," eduGAIN, 2017. [Online]. Available: https://wiki.geant.org/display/eduGAIN/Federation+Architectures

[150] J. L. Hernandez-Ardieta, J. Heppe, and J. F. Carvajal-Vion, "Stork: The european electronic identity interoperability platform," *IEEE Latin America Transactions*, vol. 8, no. 2, pp. 190–193, 2010.

[151] H. Leitold, A. Lioy, and C. Ribeiro, "Stork 2.0: Breaking new grounds on eid and mandates," in *Proceedings of ID World International Congress*, 2014.

[152] H. Roenagel, "FutureID - Shaping the Future of Electronic Identity." 2011. [Online]. Available: http://futureid.eu/

[153] E. Regulation, "No 910/2014 of the European Parliament and of the Council of 23 july 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC (eIDAS Regulation)," *European Union*, pp. 44–59, 2014.

[154] C. Cuijpers and J. Schroers, "eidas as guideline for the development of a pan european eid framework in futureid," 2014.

[155] N. Engelbertz, N. Erinola, D. Herring, J. Somorovsky, V. Mladenov, and J. Schwenk, "Security analysis of eidas – the cross-country authentication scheme in europe," in *12th USENIX Workshop on Offensive Technologies (WOOT 18)*. Baltimore, MD: USENIX Association, 2018. [Online]. Available: https://www.usenix.org/conference/woot18/presentation/engelbertz

[156] F. Morgner, P. Bastian, and M. Fischlin, "Securing transactions with the eidas protocols," in *IFIP International Conference on Information Security Theory and Practice*. Springer, 2016, pp. 3–18.