Importing National eID attributes into a decentralized IdM System

Concept of the Qualified eID Attribute Derivation into a Self-Sovereign Identity System

Version 1.0

Andreas Abraham – andreas.abraham@egiz.gv.at

Abstract: Blockchain and Distributed Ledger Technology (DLT) are increasingly in focus for security-relevant applications. Additionally, Self-Sovereign Identity (SSI) is a new and promising eID concept based on the DLT. This work describes a concept of connecting an eIDAS node to a decentralized identity management (IdM) system. In particular, the concept defines a way to derive qualified eID attributes from an eIDAS identity assertion.
Table of Content
1 Introduction ................................................................................................................. 3
2 Background ................................................................................................................. 5
   2.1 eIDAS ......................................................................................................................... 5
   2.2 Decentralized Identifier (DID) ................................................................................. 6
   2.3 Verifiable Claims ....................................................................................................... 6
   2.4 Redundant Byzantine Fault Tolerance Protocol ...................................................... 7
   2.5 Self-Sovereign Identity ............................................................................................ 8
3 Requirements .............................................................................................................. 10
4 eID Attribute Derivation Concept ............................................................................... 11
   4.1 High-Level Architecture Overview .......................................................................... 11
   4.2 Identity Data Transformation .................................................................................. 14
   4.3 Revocation Approach .............................................................................................. 15
5 Discussion ................................................................................................................... 18
   5.1 Comparison ............................................................................................................... 18
6 Conclusion .................................................................................................................. 20
7 References .................................................................................................................. 21

List of Figures
Figure 1: eIDAS Network High-Level Architecture ......................................................... 5
Figure 2: Verifiable Claim Data Model [9] ...................................................................... 6
Figure 3: Redundant Byzantine Fault Tolerance Protocol with f=1 ................................ 8
Figure 4: High-Level Architecture of a Self-Sovereign IdM System ............................... 9
Figure 5: High-Level Architecture of the Derivation Concept ........................................ 12
Figure 6: Transformation Process within the Extended RBFT Protocol [15] ................. 15
Figure 7: Revocation Process Flow .................................................................................. 16
Figure 8: Revocation Approach Sequence Diagram [15] .............................................. 17
1 Introduction

Every day we are using online services for different purposes. These services require electronic identities (eIDs). The increasing usage of online services leads to an increasing number of eIDs. Bertino and Takahashi [1] define an eID as a digital representation of known information, such as identity attributes, of a specific identity. A particular kind of eID is required for instance when using it for e-Government services since those services require identity data which can be trustworthy and where the originator is trusted. Qualified eIDs are eIDs that are trustworthily recognized and used in for instance e-Government services. A conventional approach is that a central trusted authority vouches for the correctness of eID attributes related to a specific person. Additionally, qualified eIDs require a strong binding of eID attribute to the corresponding person. Strong authentication provides a mechanism that guarantees the correlation of eID data and the related person.

The World Wide Web was not designed considering eIDs. Because of the missing identity layer, many companies or organizations create their own eIDs. This results in the creation of various identity silos. Those silos contain personal data of the users and require security mechanisms and safety measures that ensure the safety of the user’s data. The past has more than once shown that identity data silos can be attacked and data get stolen or stored identity data can be misused. An example was the Equifax data breach [2], where identity data of approximately 143 million Americans were exposed. These data contained sensitive personal information about US citizens. Another example is the major data breach of Facebook [3] where the personal data of over 50 million users were misused because the ownership of the idnetity data does not belong to the related user. The common problems here are first, that the user is not the sovereign owner of her data and second, that the user has to trust a central party.

In 2008, Nakamoto [4] introduced Bitcoin with its underlying blockchain technology. A blockchain is a special case of a distributed ledger (DL). Basically, a DL is an append-only replicated log of transactions that ensure data integrity. This DL technology (DTL) can be public accessible so that anybody can read or write to the ledger. To find a consensus of what should be written to the ledger, public ledger often use mechanisms where participants try to solve a cryptographic puzzles, the so-called mining, or using a smart reward and penalization system, the so-called proof-of-stake. In contrast, a private DL can only be accessed through special parties that can be for instance, trusted nodes that have read and write permissions. Private DLs use other methods like consensus protocols to reach consensus. The Byzantine fault tolerance protocol (RBFT) [5] is a common consensus protocol, which uses cryptographic mechanisms to ensure security. The DLT can solve fundamental trust issues in a central party when applying.
Since the number of eIDs is continuously increasing, the need of identity management (IdM) is increasing. IdM models describe the way the identities are being managed. IdM models evolved over time from the isolated IdM model to other models such as the user-centric identity model or the federated identity model. Using the DLT for IdM system was early recognized by Underwood [6] et al. The DLT could help to solve fundamental trust issues in a central authority of traditional IdM systems.

This work aims to connect a traditional IdM system with a decentralized IdM solution. In particular, this work describes a derivation process of qualified eID attributes into an SSI system. By applying this process, a user can use her qualified eID attributes after derivation in the SSI system. The purpose of the derivation is twofold. First, applying a DL based IdM system for managing eIDs could resolve certain trust issues. Second, applying SSIs give the users their data sovereignty. This work also aims to leave the traditional IdM as it is with as fewer changes as possible and tries to solve issues and problems itself. In our concept was as traditional IdM system the eIDAS network chosen. Section 2.1 describes the eIDAS network in more detail. This concept is part of a scientific paper by Abraham et al. [15].

Outline

This work is structured as follows. Section 2 describes and details the background to thoroughly understand this work together with the related work. The requirements to derive qualified eID attributes are identified in Section 3. Section 4 shows the identity derivation concept including the identity transformation and revocation approach. A discussion about this work as well as a comparison with other solutions are discussed in Section 5 followed by the conclusion in Section 6.
2 Background

This section provides the background information to understand the eID attribute derivation concept thoroughly. Additionally, this section also lists the related work.

2.1 eIDAS

In 2014, the member states of the European Union (EU) established the regulation for electronic identification, authentication and trust services (eIDAS) [7]. This regulation is the legal basis for the member states of the EU to define a common understanding and acceptance of eIDs among others.

An eIDAS node describes a gateway between other eIDAS nodes and on top of the national IdP infrastructure. Figure 1 gives a high-level view of the architecture of the eIDAS network. Basically, the eID infrastructure of a member state is connected to other member states via the eIDAS nodes.

![Figure 1: eIDAS Network High-Level Architecture](image-url)
2.2 Decentralized Identifier (DID)

Every day, we are using identifiers on the internet to identify entities such as identities, things, or systems. Decentralized identifiers (DIDs) [8] describe a new approach of identities with the primary purpose that the user is in control of the identifier. In particular, DIDs were designed to create SSIs. In traditional IdM systems, identifier depends on a central place that requires trust. In contrast, DIDs are independent of central trusted places and under full control of the related subject. From a technical perspective, DIDs are uniform resource locators (URLs) that point to DID Documents. This document contains cryptographic material such as the public key, authentication suites, and service endpoints. The service endpoints consist of valid uniform resource identifiers (URIs) representing any type of service for managing decentralized identities. To authenticate a subject related to a DID, the cryptographic material will be combined with the authentication suites. Besides an improvement through decentralization and self-sovereignty for users, it is possible to enhance the privacy using multiple DIDs for one person, to respect desired separations of identities and contexts.

2.3 Verifiable Claims

Claims are statements about a subject made by an entity that can be used to identify the subject. Verifiable claims [9] is a newly established data format with the purpose to easily exchange and verify statements about a subject. The authenticity is guaranteed by using this data format including a signature of the issuer. These claims are tamper-proof, and systems can cryptographically verify them by validation of the attached digital signature. Additionally, verifiable claims consist further components such as a subject identifier, claims about the subject and metadata about the claim or the set of claims depicted in Figure 2. Entity credentials comprise a set of one or more claims about a subject. Entity profiles summarize various entity credentials, whereby entities may have multiple profiles. Both models consist of a collection of name-value pairs considered as properties. They serve as links to the respective claims. Instances of the entity profile model are expressed as JSON objects [9].

![Figure 2: Verifiable Claim Data Model](#)
2.4 Redundant Byzantine Fault Tolerance Protocol

The redundant byzantine fault tolerance (RBFT) protocol [10] is a consensus protocol and extension of the Byzantine fault tolerance (BFT) protocol [11] based on the problem of Byzantine generals [12]. This consensus protocol is used for instance in DLT based systems in order to reach consensus.

A Byzantine failure represents a real-world environment or system in which computers and networks may behave unexpectedly due to various possible problems. The objective of the Byzantine fault tolerance (BFT) is to defend a system against failures, in which a computer/node behave faulty by stopping or crashing, sending and processing incorrect requests, producing incorrect outputs, or corrupting their locale state. As described in [10], the service models as a state machine that is replicated across various distributed nodes. The whole system consists of $N$ Nodes that maintain the service state and implement the same operations. The maximum number of faulty nodes/replicas, which can behave arbitrarily, is $f = \lfloor \frac{N-1}{3} \rfloor$. If any single process of a node is compromised, the whole replica is considered as compromised. Every node processes a specific sequence of processes and configurations called views. Every view consists of one primary node and more than one backup nodes. The general algorithm works as follows:

- First, a client sends a request to the primary in order to invoke a service operation.
- Second, the primary broadcast the request to all backup nodes.
- Third, the backup nodes, respectively the replicas, execute the request and send a reply to the client.
- Finally, the client waits for $f + 1$ replies coming from the various different replicas with the same result, which is the result of the service operation.

The RBFT protocol is an extension of the BFT protocol and depicted in Figure 3. The BFT protocol runs only one master instance with typically only one primary. The primary is receiving the request and is responsible for sending it to the replicas. This could be a problem when the primary, in this case, is behaving faulty or maliciously. In contrast, the RBFT protocol run backup protocol instances which also have primaries. This way, the malicious primary can be detected, and the protocol is still performing correctly.
2.5 Self-Sovereign Identity

Self-Sovereign identity (SSI), describes a new form of eIDs where the user is the sovereign owner of her identity data. An SSI system describes an IdM system based on the DL technology that deals with SSIs [13]. Those SSIs consists of decentralized identifiers (DIDs) [8], a new form of identifiers, together with the verifiable claim [9] format. A mechanism to find consensus, such as a consensus protocol, ensures that only when consensus is given, data are written to or read from the ledger. Figure 4 depicts the high-level architecture of a decentralized IdM system, in particular, an SSI system.
In this system, a user runs the software called client on her personal computer. This client consists of two main parts namely, the identity wallet as well as the client agent. The client wallet stores the identity data of the user as well as the keys. The client agent software handles the connection to the SSI network. The trust anchors in figure 4 represent trusted parties and organizations such as for instance a University or a Bank. These trust anchors issue verifiable claims for users and sign them. Additionally, the trust anchors are also responsible for holding a revocation list for their issued claims. When a user shares a claim with another user or an organization, the claim receiving party can then verify the issuer and trust this one or not. Moreover, the receiver can make a lookup in the revocation list to verify if the data are still valid. The trusted validator network consists some independent nodes where each node holds a copy of the ledger. Only these nodes can access the ledger. The decisions about what actions are performed on the ledger are based on a consensus protocol. The RBFT protocol could be a possible protocol to realize this. Utilizing an SSI system gives the user the sovereignty of the related identity data. This would include, that the user can decide, who can access what kind of data. Additionally, the user could also revoke before given access at any point.
3 Requirements

This section identifies the requirements for the identity derivation in order to maintain, after the derivation procedure, qualified identity attributes.

Qualified eID attributes depend at least on two parts. First, a trusted authority that vouches for the correctness of the data. Second, the binding between identity data and the identity itself, which can be ensured through strong authentication mechanisms. This work focuses on distributing trust, which a central authority would require. The strong authentication part of the user is out of scope for this work.

Furthermore, qualified eID attributes also have to be up to date. Typically, an eID assertion is only valid for the period of a session, which can vary depending on the eID data issuing IdP, on the level of assurance (LoA) of the eID data or the requirements of the service provider (SP). In this concept, a user should be able to use her identity data at a later point. Therefore, the validity of the data have to be ensured at a later point.

The goal of this work is to derive qualified eID attributes into a DL based IdM system and that a user can use her qualified identity data out of this DL based IdM system. Summarizing the before mentioned requirements on qualified eIDs, we can define three main requirements.

1. Trust in an issuing party, which vouches for the correctness of the data.
2. A revocation mechanism to ensure data validity.
3. Strong authentication to ensure a strong binding between the user and actual eID attributes.
4 eID Attribute Derivation Concept

This section describes and details the concept of the eID derivation into a DL based IdM based on the requirements identified in Section 2. This concept focuses on the eID attribute derivation only, therefore, the authentication to the user’s wallet is out of scope.

4.1 High-Level Architecture Overview

In this work, we want to derive qualified eID attributes issued by the eIDAS network into an SSI system. The two systems are using different protocols, for instance eIDAS is based on SAML 2! whereas the SSI system uses verifiable claims. An eIDAS node issues a signed SAML identity assertion which is valid for the time of a session, specified in the assertion. Therefore, this work defines an eID attribute transformation from a SAML identity assertion to the verifiable claim format that ensures the correctness of the transformation and provides distributed trust in this system. The transformation is detailed in Section 4.2. For the transformation, we have extended the RBFT protocol. Performing the transformation within this protocol provides the required trust in the system. At the end of the transformation is a multi-signature created, which ensures that the process was successful because a multi-signature can only be created when the participants successfully finish the transformation and had created a valid signature.

A user can use her eID attributes after successfully deriving it into the SSI system. When a user provides her identity data to for instance a SP, this SP wants the possibility to verify if the eID attributes are still being valid. For this reason, this work introduces a revocation approach that ensures the validity of the eID attributes. This revocation approach is detailed in Section 4.3.

To recognize the verifiable claim format, we have specified our own claim schema. The defined claim schema is then pushed to the leger. The schema is defined as illustrated in Listing 1.

```json
"seqNo": 42,
"identifier": <issuer_did>,
"data": {
"name": "eidas",
"version": "1.0",
"attr_names": ["id",
"eidasloa",
"givenName",
"familyName",
"dateOfBirth",
"personIdentifier"],
```


1 http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html
The attributes in this schema are described as follows.

- **id**: This identifier represents the DID of the eID data owner.
- **eidasloa**: This parameter describes the level of assurance (LoA) of the source data.
- **givenName, familyName, dateOfBirth, personalIdentifier**: These four attributes describe the minimum data set (MDS) of an eIDAS identity assertion.
- **multi_signature**: The multi-signature contains the multi-signature of the nodes, which have performed the transformation.
Figure 5: High-Level Architecture of the Derivation Concept shows a high-level architectural overview of the eID derivation concept. The actors of this diagram are detailed as follows.

**User Client**

The user client describes the software running on the user’s personal computer. This software contains of two components, first, the identity wallet and second the client agent. The identity wallet stores all the user related eID data in form of verifiable claims. The client agent is responsible for the communication to the SSI network and also to the eIDAS agent. The user who runs this software on her PC has to authenticate herself using a strong authentication mechanism.

**Trusted Validator Network**

The trusted validator network consists of various validator nodes. These nodes are trusted nodes that run the consensus protocol. Each of these nodes has a copy of the ledger.

**eIDAS Nodes and National IdP**

The eIDAS nodes represent the different eIDAS nodes of the EU member states together with the related IdP. The figure should point out that the eIDAS nodes act identity gateway between EU member states.

**eIDAS Agent**

The eIDAS agent describes the interface between the traditional IdM system, the eIDAS network, and the DL based IdM, the SSI system. The eIDAS agent is a newly introduced component. The primary responsibilities are (1) act as verifiable claim issuer, (2) trigger the eID attribute transformation and (3) performs the revocation approach. A trusted party such as a public authority can host this eIDAS agent.

The agent consists of three components.

1. The eIDAS SP represents the counterpart of the eIDAS IdP. The eIDAS SP is one of the components of the eIDAS agent and responsible for the communication to the eIDAS network.
2. The eIDAS agents implements a trust anchor of the SSI system. This component is responsible for the communication with the user and the SSI network, issuing the signed claims, holding the revocation list as well as triggering the eID attribute transformation operation.
3. Finally, the third component is responsible for performing the revocation approach. This includes performing the attribute queries as well as creating and adding revocation entries to the revocation list.

4.2 Identity Data Transformation

This subsection details the identity transformation process.

1. A user authenticates herself towards the user clients and presses the “import eID attributes from eIDAS” button. The user client triggers the eID attribute derivation process at the eIDAS agent.
2. A standard SAML authentication flow follows where the user is being redirected to her national IdP where the authentication process is performed.
3. When the SAML identity assertion arrives at the eIDAS agent, the agent sends an eID transformation request to the validator nodes. This request includes the original SAML assertion.
4. The validator nodes run the extended RBFT protocol that includes the transformation process depicted in Figure 6: Transformation Process within the Extended RBFT Protocol.
   a. The validator nodes broadcast the request to all other validator nodes in order to ensure that every node got the same request.
   b. All validator nodes verify the signature of the eIDAS identity assertion. This includes a cryptographic signature verification together with a certificate verification.
   c. Next, on success these nodes perform the SAML to verifiable claim (JSON) transformation.
   d. A signature is created and added to the created claim.
   e. The nodes exchange the resulting claim with each other.
   f. After sufficient created claims arrived at the nodes, these nodes create a multi signature containing all involved nodes. The multi signature is only successfully created, when all used signatures were valid. This way, the trust is a central place is split up and distributed into the whole system.
5. The eIDAS agent receives the verifiable claim and verifies the multi signature to ensure the correctness of the transformation.
6. Finally, the agent creates a signature over the whole claim. This signature is later uses to verify the claim issuer as well as data integrity.
4.3 Revocation Approach

After deriving the eID attributes into the SSI system, the revocation mechanism ensures at a later point that the identity data are still valid. Qualified eID attributes are normally only valid for the period of a session. Therefore, this approach provides a mechanism that guarantees that the derived eID attributes did not change and are still valid under the assumption that the eIDAS node provides the required functionality.

The eIDAS network uses the SAML identity protocol. The SAML protocol offers attribute queries [14] that allows the request of specific values of SAML attributes. This approach utilizes attribute queries to verify if the eID attributes have been changed or revoked. The full approach is described as follows and illustrated in the sequence diagram shown in Figure 7: Revocation Process Flow.

Figure 6: Transformation Process within the Extended RBFT Protocol [15]
The eIDAS nodes do not support SAML 2 attribute queries; therefore, in order to support these queries, the eIDAS node requires an extension. Since the eIDAS nodes serve as gateway, the national IdPs will also have to support the revocation mechanism. This concept assumes, that the eIDAS network including national IdPs already support attribute queries.

The revocation mechanism runs the attribute queries in a loop. The time between the queries depend on the source eID attributes. The validity time of the SAML attributes is specified in the SAML assertion. The time interval between the queries is a consequence of this validity time. The eIDAS agent runs the revocation mechanism in a loop and holds the revocation list. The loop includes the following steps.

1. The eIDAS agent sends in a predefined interval an attribute query to the eIDAS node. The attribute query includes the minimal data set of the user.
2. Next, the eIDAS node forwards the request to the corresponding eIDAS node if the request comes from another country and then to the national IdP.
3. National IdP verifies the request and creates response.
4. The response is forwarded by the eIDAS node to the eIDAS agent.
5. Finally, the eIDAS agent receives the response and verifies it. If the data have changed or have been revoked, the eIDAS agent creates a signed revocation entry and adds this to the revocation list.
In case that the eIDAS agent revokes the derived eID attributes, the related user has to perform the eID attribute import again.
5 Discussion

This work aims to change the existing eID infrastructure not or only as little as possible. The result might seem complex, but other solutions that seem not as complex might create trouble in other places. This section discusses the proposed solution in this work, and additionally, this section provides a comparison with other solutions.

The proposed derivation mechanism can help to overcome fundamental trust issues of a central authority. This is realized by distributing the trust from a central place to a network of validator nodes. If many independent validator nodes reach consensus and come to the same transformation result, followed by the creation of a multi-signature, the trust in the system can be high. Additionally, an SSI system gives the user the data sovereignty over the own identity data. Nevertheless, a user centric IdM can also be designed this way.

5.1 Comparison

In this subsection, the proposed derivation mechanism is compared and discussed with other solutions.

Other ways to connect a decentralized IdM to eIDAS

1. The eIDAS agent issues verifiable claims

A solution to connect a decentralized IdM to the eIDAS network would be to issue the identity data in the required format. This way, all eIDAS nodes will have to be changed, which might have to be enforced by a specification created by the related workgroup of the EU. Creating and changing this is a long process as well as reaching consensus on what data formats will be supported.

The proposed solution does not depend on laws and was created considering to not have to change the original IdM.

2. Store the original identity assertion as a whole in the verifiable claim

If the original identity assertion issued from the eIDAS node is stored in the verifiable claim as well, then a single party could also perform the transformation. This single party, nevertheless, requires high trust.

In our proposed eID attribute derivation, we try to solve the trust issue in a single place.
3. Using extra ledger for the transformation

When using its own private ledger for transforming the eID attributes, the trust is distributed as well, and the RBFT protocol does not have to be extended. Nevertheless, this solution makes the system much more complex without adding benefits.

Other ways for revocation

1. Perform the attribute query only when the SP wants to verify the data validity

The revocation procedure described in this work performs a frequent attribute query request to the eIDAS node which issued the identity assertion. Performing this attribute query request only when the identity data are being used would save network traffic. Nevertheless, this would also create a dependency on the eIDAS network and the national IdPs.

2. IdP should send a push notification when the user’s data have been changed or revoked

In contrast, to attribute queries, an IdP could send push notifications to the eIDAS agent, when the user’s identity data have been changed or revoked. This solution would have to be enforced by defining a specification so that all IdP within the EU will do it that way. Moreover, if there are various target systems, the IdPs would have to know all those systems and send many notifications.
6 Conclusion

In this work, we have proposed a solution to derive qualified eIDs, issued by an eIDAS node, into a DL based IdM system namely an SSI system. Utilizing this derivation approach can solve fundamental trust issues in a central place. Additionally, using the DL based IdM gives the user the data sovereignty of the related identity data.

The eID attributes derivation concept described in this work consists of two main parts, first the eID transformation and second the revocation approach. In the eID attribute transformation are SAML identity assertions transformed into a verifiable claim by using an extended consensus protocol together with a multi-signature of all nodes who performed the transformation. The revocation approach utilizes SAML attribute queries in a loop in order to verify if the eID attributes are still valid. If the data have changed or revoked, the eIDAS agent creates and adds a revocation entry to the revocation list.

Finally, we have discussed our proposed solution and compared it with other possible solutions. One of the primary purposes of this work was that the traditional IdM system, used as identity data source, does not has to be changed or only as little as possible.

Future work in this field could be to improve the deviation by utilizing eID attribute encryption to protect the user’s identity data and preserve the privacy.
References


