Evaluation Report

Proof of Concept Interoperabilität E-ID

Dr. Jan Camenisch
Principal Research Staff Member
IBM Research – Zurich
ibm.biz/jancamenisch
jca@zurich.ibm.com

Dr. Maria Dubovitskaya
Research Staff Member
IBM Research – Zurich
zurich.ibm.com/~mdu
mdu@zurich.ibm.com

IBM Research – Zurich January 31, 2018

Table of Contents

1 Introduction

The amount of transactions that citizens (users) can conduct on-line continues to grow. To secure such transactions and to counter frauds such as identity theft and impersonation, strong authentication of users is required. Such authentication includes (cryptographic) means to 1) establish the ownership of an identity and 2) to reveal identity attributes to the party the user means to authenticate to (relying party RP), where the identity attributes need to be vouched for by a dedicated authority (identity provider IDP). Such a system is depicted in Figure 1.

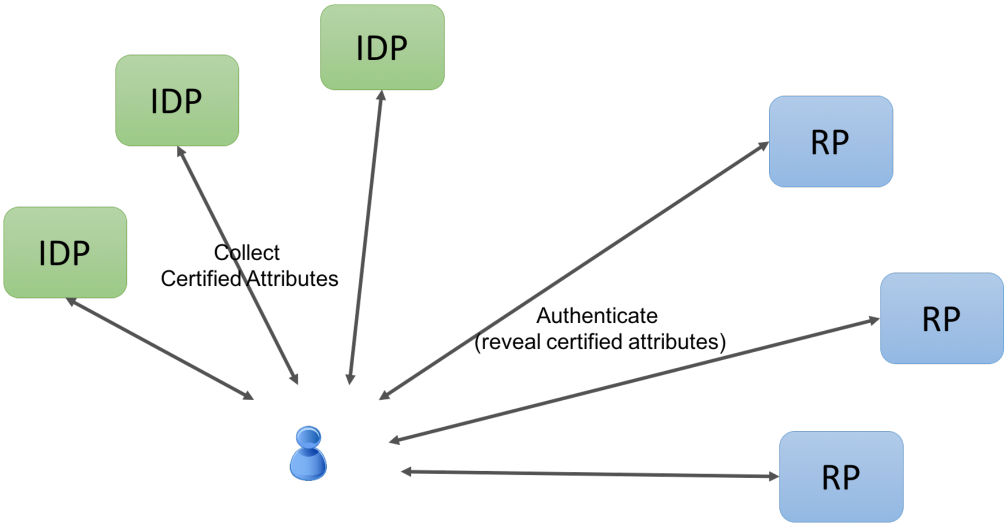

Fig. 1: System Overview. A user can register with one of more identity provider, each of them vouching for attributes of the user. Identity providers can be government organizations, private businesses, or social network providers. Having registered with an IDP, the user can authenticate to a relying party and, possibly, in addition, transmitting to this party identity attributes vouched for by an IDP.

To enable an identity infrastructure that will fulfil this need for strong authentication, the Swiss Federal council aims to introduce an officially recognized electronic identity (eID) that can be provided by different (identity) providers. Naturally, eIDs from different providers must be interoperable, i.e., if a user has obtained an eID credential from a specific IDP, this credential shall allow the user to authenticate with any service (relying party RP) that accepts an eID according the specification. There are a number of ways such an interoperability can be achieved.

In the request for "Proof of Concept Interoperabilität E-ID," the Federal Department of Justice and Police suggests two particular means so achieve such interoperability [EJPD17]. This report evaluates and compares these two solutions and puts the solutions into perspective with the state of the art and recent trends. To this end, the report first discusses requirements for an electronic identity infrastructure that have been established internationally, summarizes the most important standards for electronic identities, and reviews recent trends in the area including blockchain-based and self-sovereign identities. It then recalls both proposed solutions and provides an evaluation of both with respect to the defined requirements.

2 Requirements and Evaluation Criteria

2.1 Cameron's Seven Laws of Identity

Requirements for an electronic identity can be derived from those of a physical identity card. However, that will not be enough: one must also consider the different nature of electronic media, in which an electronic ID will be used. This media is radically different from the environment where a traditional identity card is used. For instance, a seller checking someone's identity card to see whether they are of age will likely forget name, address, and birthdays quite quickly, whereas in case such a check is conducted electronically, these data will be likely be stored. Electronic data is much more volatile and much harder to protect. It can easily be copied, shared, transmitted, etc. It also tends to be lost, stolen, and misused by cyber criminals that are much harder to tracks as they might conduct their fraught remotely, even from far abroad. Thus, an electronic identity system that does not properly consider the properties of electronic media, puts users of electronic ID at considerable risks.

Taking all such considerations into account, Kim Cameron has established "7 laws of identity" that are widely considered to be the guiding principle of any electronic identity solution [Cam05]. We list these requirements here and refer to Cameron [Cam05] for detailed explanations on the reasons behind them.

1. *User Control and Consent.* Users shall be aware of and what information about them is revealed to whom in and how this information is treated and consent to the release and use of this information.
2. *Minimal Disclosure for a Constrained Use*. User shall only reveal the information that is needed for the purpose of a transaction. This includes for example also that a user shall reveal that they are of age rather than their birthdate or that certain information that is required only in case of dispute shall be encrypted and only be available if a dispute arises.
3. *Justifiable Parties*. Digital identity systems must be designed in such a way that the disclosure of identifying information is limited to parties having a necessary and justifiable place in a given identity relationship. This for instance means that third parties mediating transactions shall not learn information about the user.
4. *Directed Identity*. This states that users can have different identities with different parties and that the user can control linkage between different identities and the exchange of attributes between these identities.
5. *Pluralism of Operators and Technologies.* A universal identity system must channel and enable the inter-working of multiple identity technologies run by multiple identity providers. This states that while standardized transport protocols could be used, the system must support different kinds of identities with different levels of assurances such as government issued attributes or self-asserted ones.
6. *Human Integration*. This requirement is related to the first one but speaks to mechanisms to involve the users in the process such as entering passwords or inserting a smart card.
7. *Consistent Experience Across Contexts.* An identity system shall be easy for a user to use and a user shall understand what information she reveals in a given context. Essentially, the user experience with different identities and credentials shall be similarly unified as we are used to from the different plastic cards we carry in our wallets.

While these laws apply to the two solutions under evaluation in principle, some will come in effect only once a solution is fully defined and all components are build. This includes in particular the design of user interfaces and devices as well as the specification of policies stating which information is requested by whom and for what purpose. Thus, the scope of this report is limited to the extent these laws are applicable to the basic properties of the two suggested solutions.

2.2 Requirements for an Identity System

In this section, we refine and extend Kim Cameron's "laws of identity" into a list of requirements. This list includes the requirements exhibited in the request "Proof of Concept Interoperabilität E-ID" [EDJP17] as well as further requirements that are relevant for this report.

1. **Simplicity of implementation.** It should be easy for both, RPs, and IDPs to implement and integrate the authentication solution into their products/services.
2. **Robustness of operation.** The system should effectively and robustly perform under different conditions. For instance, a user should be able to authenticate to an RP at any time, even if (some of) the IDPs she has registered with are not available at that moment.
3. **Ease of use for eID owners (users) and RP.** The system should be easy to use by the end-users, reducing the number of logins and providing consistent experience with different IDPs

and RPs for an end-user. The solution should be preferably, browser-based, without requiring a user to install additional software/plugins/mobile apps. This requirement relates to Cameron's Laws #6 (Human Integration) and #7 (Consistent Experience Across Contexts).

4. **Maintainability.** The system should be easy to maintain and propagate fixes and/or updates whenever necessary.
5. **Interoperability**. The system should support different IDPs and RPs running on different platforms. To authenticate with any RP subscribing to the system, it should be sufficient for a user to register with a single IDP (provided that this IDP can vouch for the attributes requested by an RP). This relates to Cameron's Law #5 (Pluralism of Operators and Technologies).
6. **Scalability.** The system should be easily scalable and should be able to support new IDPs/RPs dynamically joining the system.
7. **Cost.** The costs to roll out, operate, and maintain the system should be minimized.
8. **Market acceptance.** The solution should be compatible with what is widely used on the market at the moment and, therefore, should be easily adopted by the market. It also should have a potential to be prevalent on the market for the future.
9. **Security.** The system should follow the security-by-design principle. The system should comply to the existing security standards and follow best security practices. It should be extensively tested to mitigate potential vulnerabilities and prevent user impersonation and other security-related attacks.
10. **Privacy (data protection).** The system should follow the privacy-by-design principle [Cav12]. Privacy of both the end-users and the RPs needs to be preserved. Only the minimally required information about users shall be revealed to RPs. RPs should design their applications such that only the minimally necessary information is needed. This requirement relates to Cameron's Laws #2 (Minimal Disclosure for a Constrained Use), #3 (Justifiable Parties), and #4 (Directed Identity).
11. **Attribute exchange.** The users should be able to provide authentic attributes about themselves, certified by the trustworthy IDPs.
12. **User Consent.** The user should be in control of what data about herself she discloses and to whom. The user should be able to review and consent to any authentication and/or information disclosure. This requirement captures Kim Cameron's Law #1 (User Control and Consent).

3 Solution 1: Direct Interoperability through Protocols (Direkte Interoperabilität auf Protokollbasis)

3.1 Solution Overview

The first proposed solution is based on open protocols, in particular, OpenID Connect. Interoperability is ensured by using standard open protocols with defined message formats and APIs used by all parties in the system. A high-level architecture is presented in Figure 2. A user can register with one or more IDPs and then authenticate to different RPs that support OpenID Connect.

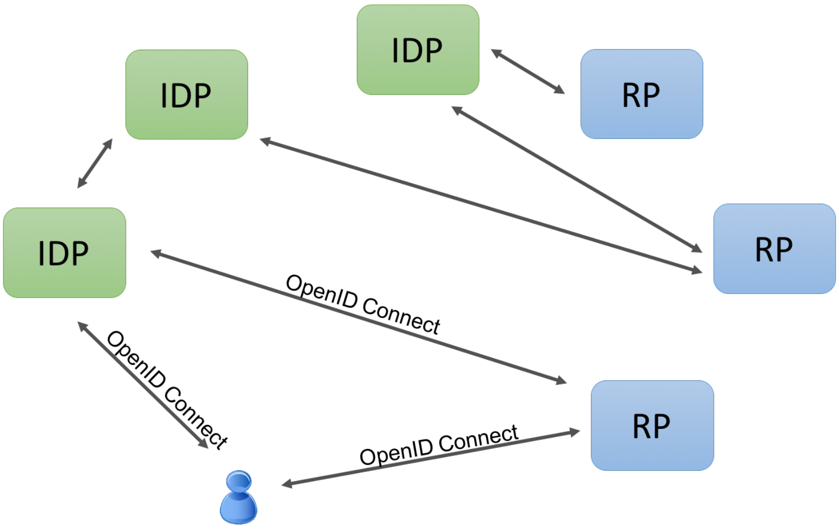

Fig.2: Solution 1 system overview.

A sketch of the message flow is provided in Figure 3. We note that this is one of the few possible message flows supported by the OpenID Connect specification. When a user requests a resource that requires authentication from an RP, the RP first checks if it is registered with the referenced OpenID Connect provider (IDP). If not, it will do so by running the OpenID Connect Discovery and Dynamic Client Registration protocols. Then the RP will redirect the user to the required IDP for authorization. The user performs login with the IDP and consents to disclosing the required information. Once the user authorization with IDP is successful, the user is redirected back to the RP and provides it with the authorization token received from the IDP. The RP can then use this token to call the IDP endpoint and to retrieve information about the user. Adding so-called "federation hubs" and enforcing naming conventions can potentially reduce the amount of RPs registrations and required connectivity between parties, however such solutions might be costly.

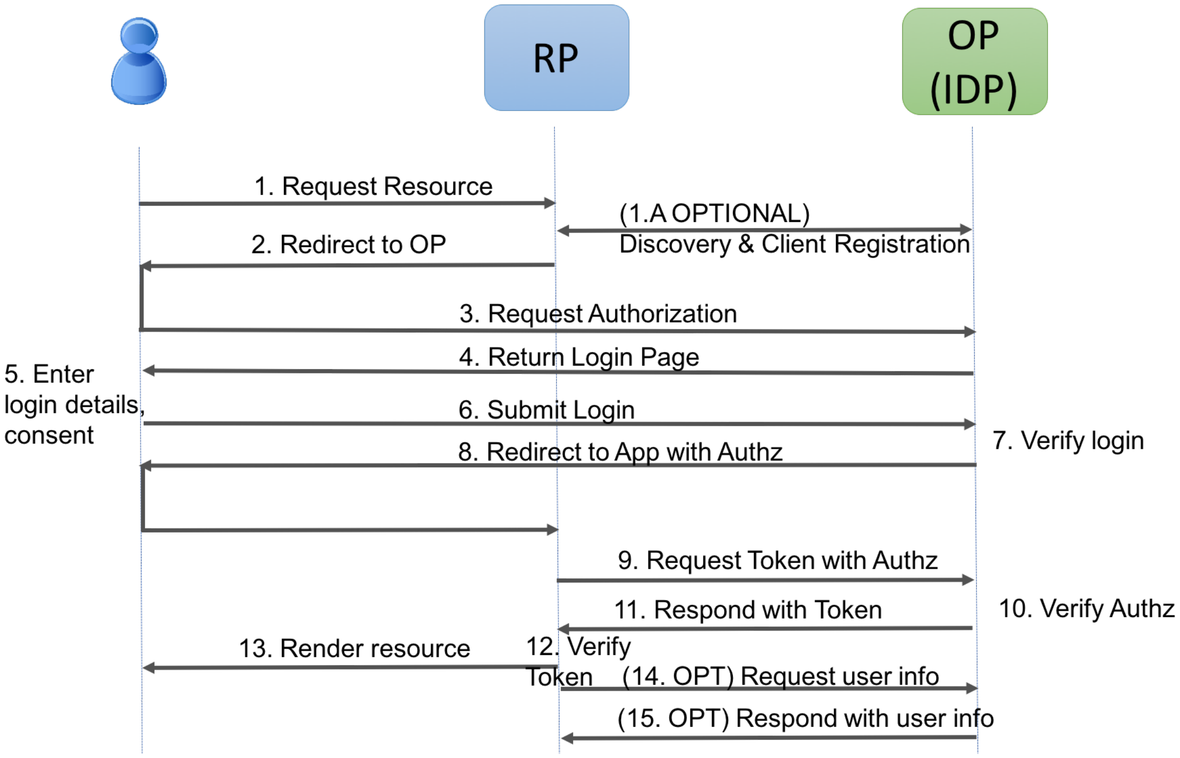

Fig.3: Solution 1 high-level message flow.

3.2 Hauptfrage 1: Solution 1 Assessment

As we analyse both solutions in detail in Section 5 (with respect to the requirements listed in Section 2), here we provide only a short evaluation of Solution 1.

The advantages of Solution 1 are the following. As the solution provides interoperability on the protocol level and the protocols are open and widely used, Solution 1 is easy to implement and deploy, it is scalable and is already supported by many IDPs and RPs. It also provides a certain level of privacy and security since the parties communicate directly and a user has full control of what information is disclosed to which party.

The main drawbacks of Solution 1 are the following. The IDPs need to be online for authentication to proceed. Furthermore, as it is clear from the message flow, an IDP knows what information the user discloses and to whom, since it is the IDP that provides the information every time to the RPs, so the level of privacy might not be as high as desired. Finally, the IDP-RP registrations, obligations, and financial relations between the parties need to be resolved in a distributed fashion.

4 Solution 2: Interoperability through a Mediating Party (IDV) (Vermittelte Interoperabilität mittels dem IDV)

4.1 Solution Overview

The second solution uses a broker such as IDV [IDV], so the user authentication protocols are routed via a trusted central party. IDV connects parties and ensures interoperability and translation between different authentication methods (cf. Figure 4).

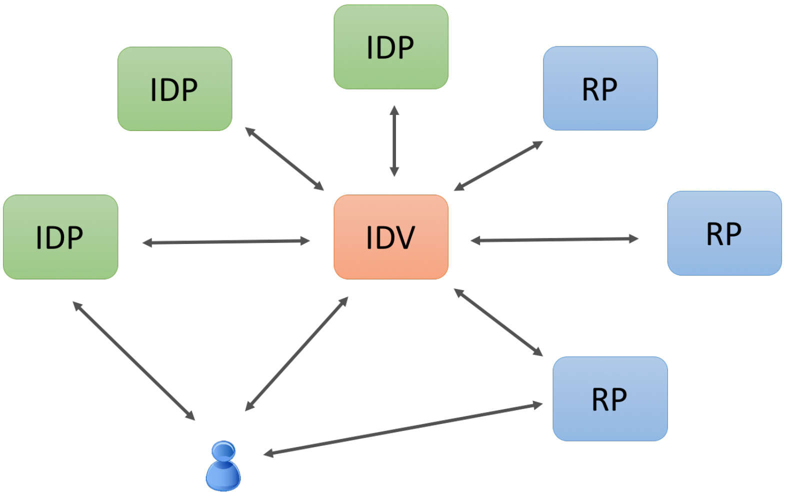

Fig.4: Solution 2 Overview.

The message flow presented in Figure 5 was derived from the IDV specification [IDV] and might vary for different brokers. On a high level, however, such solution would work as follows. After a user requested authentication with an RP, the user is redirected to IDV (or a broker) instead of the IDP directly. IDV offers the user to select an IDP from a list of possible IDPs that the user has registered with. IDV then redirects the user to the chosen IDP for authorization. After the user receives the authentication token from the IDP, the token is forwarded to IDV, who generates the IDV assertion from the received token. This assertion is then forwarded to the RP for authentication.

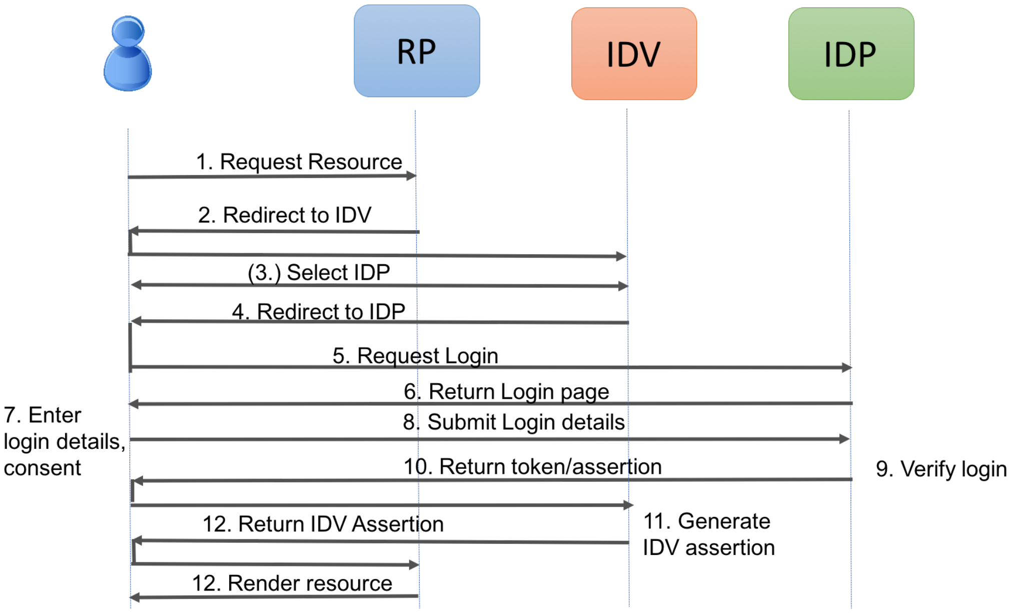

Fig.5: Solution 2 high-level message flow (derived from the [IDV] specification).

4.2 Hauptfrage 2: Solution 2 Assessment

Again, we refer to Section 5 for a detailed comparison of Solution 2 with Solution 1 and provide only a short evaluation of Solution 2 here. The main advantage of Solution 2 is the following. Since all communication is routed through the broker (IDV) rather than peer-to-peer connections, the relations (contractual and financial) between the parties would be easier to establish and maintain.

This advantage, however, comes at the price of the following drawbacks. Similar to Solution 1, the IDPs need to be online for authentication to proceed. However, now it is also IDV, which also needs to be online at all time. Furthermore, IDV is aware of all information exchange between parties, so the privacy of users, IDPs, and RPs is not preserved with respect to IDV. Even if protocols for communicating with IDV are standard and/or can be easily supported the RPs and IDPs might not be willing to join the system for privacy reasons. The cost of such solution is higher since the IDV infrastructure and its high availability needs to be maintained (IDV is a bottleneck in the system). Moreover, IDV has to be fully trusted since it is not privy to the information flow between the parties. The communication complexity of such system is also higher.

5 Comparison and Evaluation of the two Suggested Solutions

Below we compare the two solutions [EDJP17] with respect to the requirements in Section 2.2 and answer the additional questions raised in the request "Proof of Concept Interoperabilität E-ID" [EDJP17].

5.1 Hauptfrage 3: Which Solution Fares Better?

In Table 1, we compare the two solutions w.r.t. the criteria listed in the request [EDJP17]. A summary and overall evaluation follows.

Criteria	Solution 1 (Protocol-Based)	Solution 2 (IDV)
Simplicity of implementation	Based on open standard protocols that are easy to implement and integrate. Suitable open source implementation exist. Given its	IDPs and RPs would need to implement a protocol supported by the IDV. In case of the published IDV specification [IDV], the protocol of choice is SAML,

	popularity, the OpenID protocol likely is already supported by IDPs and RPs. To build a full, interoperable identity system, the OpenID Discovery and Dynamic client registration need to be carefully implemented.	also an open standard for which open source software components exists. However, as this solution is non-standard, an implementation will be more complex and thus more error prone.
Robustness of operation	The system is distributed, does not rely on any single party (no single point of failure or a bottleneck), and therefore, robust. However, if an IDP that is required by the RP is offline, the user would not be able to authenticate (IDPs required to be online) as protocols such as OpenID require the IDP to be online when a user want to authenticate to an RP.	The system relies on single party that mediates between RPs and IDP. Similar to Solution 1, if the IDP with which a user is registered if offline at the time a user wants to authenticate, authenticate will fail. In conclusion, Solution 2 is less robust as it introduces an additional party that might fail.
Ease of use for E-ID owners and RP	The user experience is easy and well-known (social network logins and many services already implement this flow).	Depending on the concrete implementation, the user experience should be similarly easy. User experience is consistent across all IDPs/RPs as the solution can be made such that users are interfacing only with IDV (or also IDP but without disclosing the information to which RP they authenticate) for authentication.
Maintainability	As the protocols are open, standardized, and extensively tested already, the system is easy to maintain. Does not require any additional infrastructure to maintain.	IDV infrastructure and services need to be maintained in addition.
Interoperability	Interoperability is easy to achieve as the OpenID protocols are well known, certified reference implementation and API descriptions are public, and many of the existing IDPs and RPs already support them.	Any IDP/RP joining the system needs to support protocols for communicating with IDV. In case the protocols used are open standards, this should not be an issue.
Scalability	Easily scalable. Discovery and Dynamic client registration protocols need to be carefully implemented, however.	Scalability depends on the IDV capabilities.
Cost	Low (no additional infrastructure required). Open source components available, integration with OpenID easy.	Higher (IDV infrastructure implementation, operating, and maintenance cost and support). As all authentication transactions are mediated by the IDV and is a single point of failure, the system must be highly available. As IDV collects lots of information about users' transaction. Both will make IDV a likely target of attack, protecting against which will raise the cost of operating IDV substantially.

Market acceptance	OpenID Connect is already widely used on the market. Thus, acceptance should be high.	Potentially not all IDPs/RPs would be willing to join the system as all communication will be routed through the central party (IDV) and the system might require implementing/supporting additional protocols or protocol extensions. Introduces a trusted third party that is able to collect lots of information. Users and other stakeholders might not be willing to assert this trust.
Security	Protocols are open, standard, and extensively tested. The risk mitigation and deployment recommendations are also available. Any discovered vulnerabilities and the corresponding patches would be propagated quickly since the protocols are widely used by the market.	The security of IDV (as a single point of failure) needs to be ensured. Adding an additional party to the identity system introduces additional vulnerabilities. Of course, being involved in all transactions might allow one to better detect attacks. However, this benefit could be achieved independently (e.g., by collecting and evaluating real-time reports from IPDs) without the extra complexity of adding an IDV.
Privacy	The communication of user's attributes is done solely between IDP and RP and does not involve other parties. However, the IDP learns which RPs a user visits and which information the different RPs request.	All communications between user, RP, and IDP are routed through IDV. Therefore, all communication patterns/details are known to IDV. The IDV could be used to hide the RP from the IDP and hence the latter would no longer be privy of which RP a user visits. However, the IDV now sees that for all users and all RPs.
Attribute exchange	Supported	Supported
User Consent	Supported	Supported

Table 1: Analysis of Solutions 1 and 2 with respect to the requirements from Section 2.2.

Overall analysis. In summary, both solutions achieve interoperability. Solution 1 is simpler, more cost effective, and also preferable from a privacy and security point of view because the involved parties operate directly with each other. Using OpenID Connect has the additional benefit that user consents to disclosing certain information as this functionality is embedded in the OpenID Connect protocol. A single trusted mediator, as in the second solution, can become a bottleneck for such a distributed system not only for performance reasons but also because becomes a main target of attacks (for a number of reasons) which will require increased protection. There are few advantages, however, that Solution 2 could offer, once implemented. For instance, with such a centralized solution, it is easier to implement and enforce system wide change, in particular w.r.t. user offer better user experience, user data management, or consent management. Such benefits, however, do not outweigh drawbacks of Solution 2. Furthermore, distributed solutions have in general fared better in the past. An example is the success of OpenID compared to the failure of Microsoft's Passport (which was conceptually the same as Solution 2). Also, solutions with such a central party failed because of lack of trust in this party and of financial incentive of operating such a party.

5.2 Zusatzfragen - Additional Questions

We turn our attention to the eight additional questions stated in the request [EJPD17].

Zusatzfrage 1: Can you confirm that interoperability can be achieved with the OIDC protocol suite (Solution 1)?
Yes, by implementing OpenID Connect protocols, agreeing on the identifiers format, and by using OpenID Connect Discovery and Dynamic Client Registration protocols, interoperability can be achieved.

Zusatzfrage 2: What is the current and expected over the next few years dissemination/use of the OIDC protocol suite on the market?
As of March 2016, there are over a billion OpenID-enabled accounts on the internet, and organizations such as Google, WordPress, Yahoo, and PayPal use OpenID to authenticate users. The amount of accounts continues to grow. Until any of the blockchain-based and/or privacy-preserving authentication methods start to be widely used and appear in standards, OIDC is expected to be a prevalent authentication protocol on the consumer authentication market.

Zusatzfrage 3: Is adding a naming convention regarding the identifiers (URI) to the OIDC protocol suite compatible with the basic specification of the protocol?
Yes, this is the case, however, there are some recommendations regarding the identifier formats specified in the OIDC standards.

Zusatzfrage 4: Can the defined protocols of the OIDC protocol suite be identified and integrated by IDPs and RPs with a reasonable effort into their IT solutions?
Yes, there exist detailed specifications and OIDC certified implementations [OIDC-CI] for both RPs and IDPs, some of which are open source libraries. In fact, many of the IDPs and RPs already support OIDC.

Zusatzfrage 5: How is the OIDC protocol suite compared to other protocol suites in the market (e.g., SAML 2.0, but not purely academic solutions)?
OpenID Connect protocol has a few advantages over SAML 2.0: First, one can implement all SAML use cases with a simpler and lighter JSON/REST based OIDC protocol; second, OIDC supports both Web browser and mobile apps, while SAML is not as mobile-friendly; and, third, SAML does not provide a discovery service and dynamic client registration capabilities, while OIDC does. See also Section 7 (Appendix) for the detailed description and comparison of the protocols.

Zusatzfrage 6: Can the interoperability between the eID systems be achieved in a different way and, if so, how and with what consequences (for example, costs, operational costs, risks, etc.)?
There are many different ways to achieve interoperability. If different protocols and format are to be used, a translation between protocols and formats will be necessary. This can either happen at every RP, every IPD, or via a trusted central translation intermediary. From an operational and cost point of view, the third solution is hardly an option. Thus, to enable interoperability is seems best to agree on a container protocol that is flexible enough to support different attribute format and tokens. OpenID connect seems to be a good choice for such a protocol. It is mature and is widely accepted.
However, concerning the formats of attributes and certificates, the choice is less obvious and an infrastructure should be built such that the use of different format will be possible in the future. Nevertheless, it is easier to support different formats than different protocols: the verification of attributes and certificates typically happens by calling a local algorithm that will return the value of the attributes if the verification succeeds. Thus, the support an additional format would only require the installation of a different algorithms for such verification. Because such verification algorithms rely on cryptography, any implementation needs to support the switching between different verification algorithms already for reasons of security.

Zusatzfrage 7: Is it possible to establish an interoperable eID ecosystem without every RP having a contractual relation with each IDP based solely on the convention on the eID and RP identifiers?

This question has three aspects: a technical one, a legal one, and a financial one. These should not be mixed and can be addressed by separate means. The technical aspect involves the verification by an RP whether the user the RP communicated to is registered at some IDP and what are the attribute values of the user the IDP vouches for. This can be achieved by the OPID protocols (using the OpenID Connect Discovery and Dynamic Client Registration protocols). The legal and financial aspects for the eID ecosystems are similar to the SSL certificates, i.e., who pays for the certificate/assertion/claim and who is held liable if a certificate/assertion/claim is issued falsely (because, for instance, the certificate authority made a mistake or because it got hacked and the attacker managed obtain certificate with false contents). Thus, when participating in an eID solution, IDP will probably need to be accredited and state what kind of guarantees they provide to RPs. Likely, an IDP will want to get insured against risks and liabilities. The financial aspect can also be dealt with by a number of ways. Either users will have to cover the costs, i.e., pay for the services their IDP offers them (and could potentially be reimbursed by an RP who offers its services cheaper to users that have registered with an IDP). Alternative solution for payments where the RP pays the IDP directly are also possible and could use existing financial infrastructures. Outlining such options is beyond the scope of this report, however.

Zusatzfrage 8: Are there any further hints, reservations or recommendations?

Both solutions require that the IDP that vouches for a user's attributes be on-line at the time of the authentication. This is problematic for a number of reasons.

1. Robustness: if the IDP is not available, the user cannot authenticate to an RP. Thus, an IDP must be online 24/7. Furthermore, if an IDP is compromised by an attacker, the attacker can impersonate any user at that IDP or register any user and then impersonate that user. Thus, the IDP must be well protected on the one hand, but highly available on the other hand. Achieving both at the same time is costly. (Notice that this is independent of Solution 1 or 2.)

2. Privacy: the IDP learn with RP a user authenticates to and what identity information the RP requests about the user. This might not be necessary. Off-line solution such as X.509 attribute certificates or privacy-enhancing attribute-based credentials (cf. Section 7 in the appendix) can be used to avoid this.

From a privacy as well as a security and operational perspective, the use of attribute certificates is preferable as it removes the IDPs from the authentication interactions. A naïve implementation of attribute certificates, however, has the drawback that the user would reveal to the RP all attributes contained in her certificates. To address this, one would either have to issue certificates on individual attributes or, better, resort to cryptographic techniques such as privacy-preserving attribute-based credentials (cf. Sections 7.2.1 and 7.2.2 in the appendix). OpenID Connect as a protocol is flexible enough to deal with such technologies and they can be incorporated in its message flow (instead of calling an IDP, a user's wallet with credentials will be called to provide a token).

Finally, it must be mentioned that with the raise of the blockchain technologies, the space of on-line identity has become very active again. New ways to authenticate using blockchain in one form or the other will emerge in the next couple of years.

6 References

[ABC4T]	Attribute-based Credentials for Trust. EU FP7 project, https://abc4trust.eu/
[Cam05]	**Kim Cameron "The Laws of Identity", 2005. http://www.identityblog.com/stories/2005/05/13/TheLawsOfIdentity.pdf**
[Cav12]	Ann Cavoukian "Privacy by Design and the Emerging Personal Data Ecosystem," 2012. http://www.ontla.on.ca/library/repository/mon/26010/319933.pdf
[EDJP17]	Eidgenössisches Justiz- and Polizei Departement, Bundesamt für Polizei fedpol, Auftrag "Proof of Concept Interoperabilität E-ID". November 23, 2017.
[IDV]	IDV Interface Specification (suitable for IDV Integration Pilot) https://www.idv-fsi.ch/de/idv-schweiz/spezifikationen/
[IDMX]	IBM Identity Mixer, https://www.zurich.ibm.com/identity_mixer/
[INDY]	Project Indy, https://www.hyperledger.org/blog/2017/05/02/hyperledger-welcomes-project-indy
[HL]	Hyperledger Project, https://www.hyperledger.org/
[DP18]	Paul Dunphy and Fabien A. P. Petitcolas: A First Look at Identity Management Schemes on the Blockchain http://arxiv.org/pdf/1801.03294.pdf
[OIDC]	OpenID Connect Protocol Specification, http://openid.net/connect/
[OIDC-CI]	OpenID Connect Protocol Specification, http://openid.net/developers/certified/
[OIDC-DCR]	OpenID Connect Dynamic Client Registration https://openid.net/specs/openid-connect-registration-1_0.html
[OIDC-D]	OpenID Connect Discovery https://openid.net/specs/openid-connect-discovery-1_0.html
[OAUTH]	OAuth 2.0 Protocol Specification: https://oauth.net/, RFC 6749: https://tools.ietf.org/html/rfc6749
[SAML]	Security Assertion Markup Language, http://saml.xml.org/saml-specifications
[SOV]	Sovrin Foundation, https://sovrin.org/
[UPR]	U-Prove, https://www.microsoft.com/en-us/research/project/u-prove/
[W3C]	Verifiable Claims Working Group, https://www.w3.org/2017/vc/WG/

7 Appendix - State of the Art for Authentication Protocols

We now provide an overview of the identity management systems and authentication protocols: the existing open standards, the future trends, and the current standardization efforts.

7.1 Existing Standards and Open Protocols

7.1.1 PKI Infrastructure (X.509 Certificates)

A Public Key Infrastructure is a set of roles, policies, and protocols needed to create, manage, distribute, use, store, and revoke digital certificates. In the context of authentication, digital certificates in a standard X.509 format are used. A certificate is a list of attributes and a public key of the entity that are digitally signed by a trustworthy Certificate Authority (CA). Parties get their certificates issued, store them locally, and can show them to the RPs. The validity of certificate (digital signature verification) is done with respect to the public key(s) of the CA, which an RP trusts (cf. Figure 6).

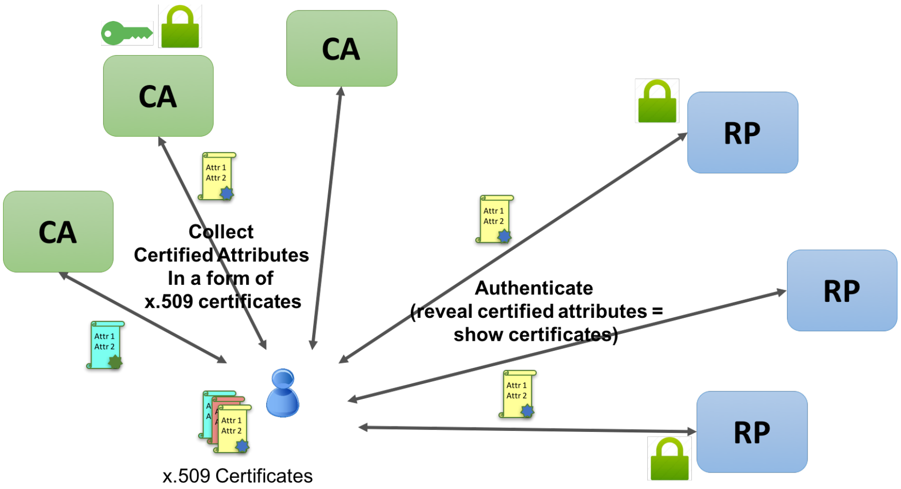

Fig.6: An example of X.509 based PKI.

The purpose of a PKI is to establish trust and facilitate the authentication and secure information exchange in different applications such as e-commerce, internet banking. It provides strong authentication for the servers and also for the users in cases where passwords-based authentication method is not enough.

7.1.2 SAML 2.0

Security Assertion Markup Language 2.0 (SAML 2.0) is a version of the SAML standard created in 2001 for exchanging authentication and authorization data between security domains. SAML 2.0 is an XML-based protocol that uses security tokens containing assertions to pass information about a user between a SAML authority (IDP), and a SAML consumer (Service Provider). SAML 2.0 is used for web-based, cross-domain single sign-on (SSO), which helps reduce the administrative overhead of distributing multiple authentication tokens to the user.

The SAML 2.0 specification defines assertions protocols, which are assertion requests and responses; bindings, or how these requests and responses happen between the service provider and identity provider, using standard communication methods (e.g. HTTP POST); and profiles, which are combinations of assertions, protocols and bindings for various use cases, like SSO. A high-level message flow for SAML 2.0 is presented in Figure 7 below.

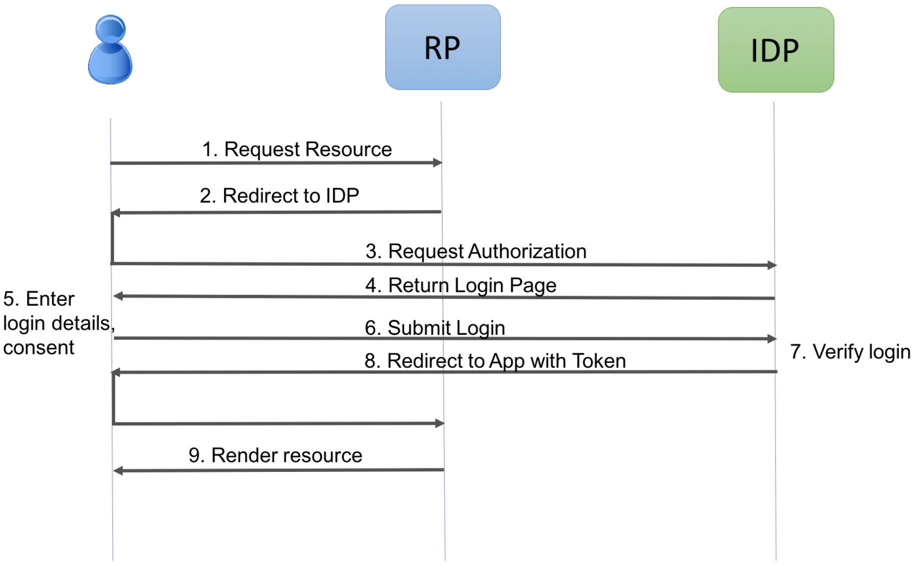

Fig. 7: SAML 2.0 message flow.

7.1.3 OAuth 2.0

OAuth 2.0 [OAUTH] is an authorization protocol that gives an API client limited access to user data on a web server. OAuth relies on authentication scenarios called flows, which allow the resource owner (user) to share the protected content from the resource server (client) without sharing their credentials. OAuth 2.0 does this by allowing tokens to be issued by an identity provider to these third-party applications, with the approval of the user. The client then uses the token to access the resource server on behalf of the user.
OAuth 2.0 is also the basis for OpenID Connect, which provides OpenID (authentication) on top of OAuth 2.0 (authorization).

7.1.4 OpenID and OpenID Connect

OpenID is an open standard for authentication, promoted by the non-profit OpenID Foundation. A user must obtain an OpenID account through an OpenID identity provider. The user will then use that account to login to any website (the relying party) that accepts OpenID authentication. The OpenID standard provides a framework for the communication that must take place between the identity provider (IDP) and the relying party (RP).

OpenID Connect (OIDC) is an identity layer built on top of the OAuth 2.0. It enables an RP application to authenticate a user, and to obtain information (attributes or so-called “claims”) about that user, such as the user name, email, etc. from an OpenID provider[1] in an interoperable and REST-like manner (see Figure 8 for the message flow). The information about the user is encoded in a secure JSON Web Token (JWT), called an ID token.

[1] Called OP in the context of OpenID Connect, but referred to as IDP in the rest of the document for notation consistency.

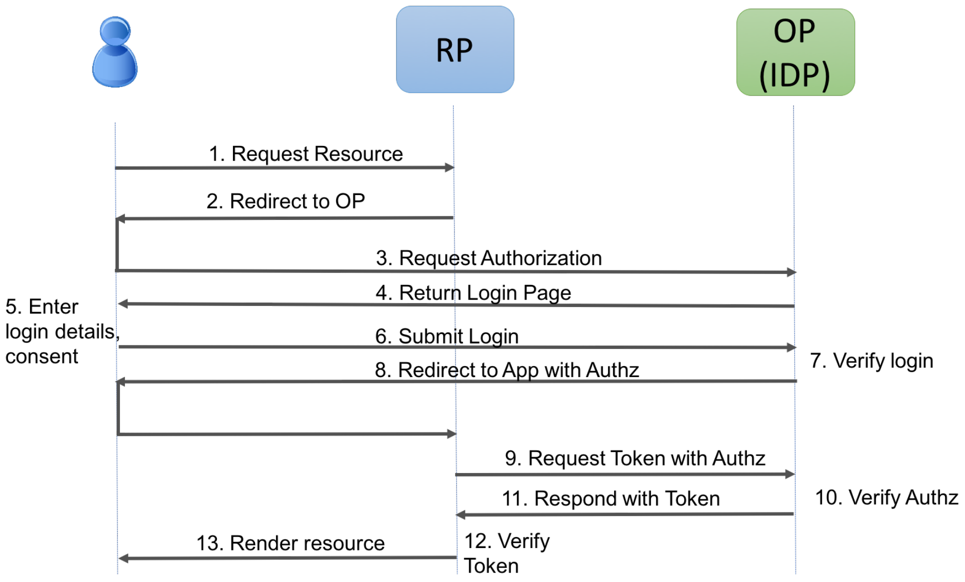

Fig.8: OpenID Connect high-level message flow.

To support interoperability and reduce the number of static registrations between RPs and IDPs the following extension protocols were proposed.

OpenID Connect Discovery. OpenID Connect defines a discovery mechanism, called OpenID Connect Discovery [OIDC-D], where an OpenID server publishes its metadata at a well-known URL, typically *https://server.com/.well-known/openid-configuration*. This URL returns a JSON listing of the OpenID/OAuth endpoints, supported scopes and claims, public keys used to sign the tokens, and other details. The clients can use this information to construct a request to the OpenID server.
OpenID Dynamic Client Registration. This specification [OIDC-DCR] defines how an OpenID Connect Relying Party can dynamically register with the End-User's OpenID Provider, providing information about itself to the OpenID Provider, and obtaining information needed to use it, including the OAuth 2.0 Client ID for this Relying Party.

In Table 2 we compare OAuth 2.0 with OpenID Connect and SAML 2.0 for authentication.

Comparison Criteria	**OAuth 2.0**	**OpenID Connect**	**SAML 2.0**
Token format	JSON or XML (SAML 2.0)	JSON	XML
Authentication	Authorization only	Yes	Yes
Transport methods	HTTP	HTTP GET and POST	HTTP Redirect (GET) binding, SAML SOAP binding, HTTP POST binding, and others
Comments	Not suited for user authentication	Can implement all SAML use cases with a simpler, JSON/REST based protocol Supports both Web browser and mobile apps Discovery service	SAML Web profile for Web browser only. No discovery service

Best suited for	API authorization	Authentication/SSO for consumer apps	Authentication and Authorization for enterprise

Table 2: Comparison of OAuth 2.0, OpenID Connect, and SAML 2.0 for authentication.

7.2 Academic/PoC Projects and Future trends

7.2.1 Privacy-preserving Attribute-Based Credential Systems

All the technologies described above do not provide advanced privacy features (selective attribute disclosure, unlinkability between different authentications performed by the same user). Fortunately, privacy-preserving attribute-based credential systems exist that provide those advanced privacy properties. We describe the two most well-known ones below.

7.2.1.1 IBM Identity Mixer

IBM Identity Mixer [IDMX] is a cryptographic protocol suite for strong privacy-preserving authentication, signatures, and transfer of certified attributes. Its trust model and security guarantees are similar to what is ensured by standard X.509 certificates, but the underlying cryptographic algorithms provide more advanced privacy features, such as unlinkability and minimal attribute disclosure, efficiently. As in X.509 case, users receive their attributes certified by a CA in a form of a credential. The difference is that IBM Identity Mixer cryptographic primitives (special signature schemes and so called zero-knowledge proofs) allow users to transform their credential into a fresh unlinkable token that only selectively reveals attributes or even proves predicates about their attributes. The verification of such tokens, similar to X.509, is also performed with the public key of the CA who initially issued the credential to the user (cf. Figure 9).

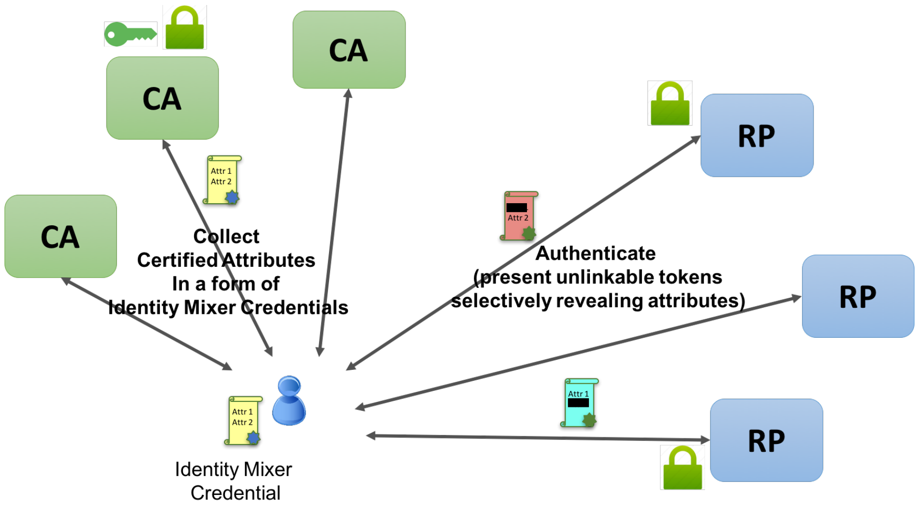

Fig.9: Using Identity Mixer Credentials.

Although the Identity Mixer algorithms and protocols are not standardized yet, they are used in different PoCs and provide the basis to implement privacy-preserving verifiable claims and blockchain-based privacy-preserving identity management solutions (see also Sections 7.2.2 and 7.2.3).

7.2.1.2 Microsoft U-Prove

U-Prove [UPR] is an attribute-based credential system which core specifications were released under Microsoft's Open Specification Promise. U-Prove is based on similar cryptographic primitives as Identity Mixer and can also provide selective disclosure and unlinkability properties. However, U-Prove does not provide the advanced functionalities as does Identity Mixer (predicates over attributes, inspection, revocation). Furthermore, U-Prove credentials are one-time use credentials (provide unlinkability only if used once). During the ABC4Trust project [ABC4T], however, the IBM Identity Mixer and U-Prove technologies were conceptually unified and the specifications and policy languages were developed to ensure interoperability of the two solutions.

7.2.2 Blockchain-based solutions

Blockchain is considered by many as the ideal tool to also solve digital identity. The common idea behind most blockchain-based solutions is to make use of a blockchain as a public directory for identity. Some identity systems propagate the idea that users' public keys and their (identity) attributes are stored on this public directory, possibly together with certificates from identity providers. Users can then refer to their entry on the blockchain to prover their identity. Such approaches have severe consequences from a privacy and security point of view, as identity related information is very sensitive and should not be made publicly available. However, there are other proposals for block-chain based identity systems that are very reasonable and promising. The idea here is to essentially use a blockchain in place of a public key infrastructure. Namely, a blockchain is used to store the public keys of IDPs (but not of users!), specifications of credentials and supported attribute formats, etc. The Sovrin Foundation [SOV] is an example of this approach that has gotten quite some traction. The foundation runs a blockchain for exactly this purpose. More precisely, the blockchain nodes are run by so-called stewards, which are trusted stakeholders from all over the world. These stakeholders include companies, university, and government organizations. The technology used by the Sovrin Foundation is open sourced as the Hyperledger Indy project [INDY]. Sovrin allow users to authenticate to RP in a privacy friendly way: by default, users have a different identity with each RP (a pairwise digital identity (DID)) and can then, using verifiable claims (see next subsection), selectively disclose attributes that are vouched for by one of the users' identity providers (which typically is a steward). To achieve this, Sovrin uses privacy-preserving attribute credentials (see previous section). We refer to Dunphy and Petitcolas [DP18] for a recent overview on blockchain-based identity systems.

7.2.3 Future Standardization for Identity and Verifiable Claims

Since privacy-preserving user authentication protocols are not standardized, a few standardization activities are carried out to fill in this gap. One of them is called Verifiable Claims and is driven by a W3C working group [W3C] to standardize protocols and languages for authentication and identity management. It defines the formats and message flows and supports different levels of privacy preservation. On a high level the solution works as follows (see also Figure 10). A holder (user) collects credentials from different issuers. A credential reveals multiple claims about the holder to service providers. A claim can reveal different attributes (e.g., email address) or just facts (e.g., OlderThan18) about the holder. Revocation and Inspection (Auditing) functionalities are also supported. To bridge these high-level protocols and specifications with the cryptographic formats and implementation, a lower-level specification is also being developed. It is called ZKLang and translates a high-level claims specification to a low-level zero-knowledge proof specification. Cryptographic algorithms that are used to implement verifiable claims are the same as the algorithms underlying the Identity Mixer technology.

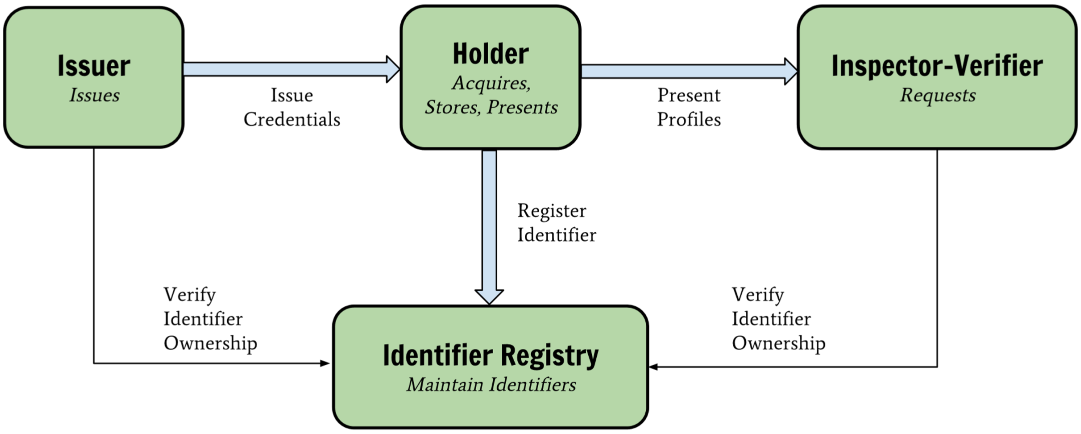

Fig.10: The roles and information flows that form the basis for the Verifiable Claims specification (source: https://w3c.github.io/vc-data-model/)