GN4-3 WP5 T2 Trust and Identity Incubator

Using Distributed Identity for Managing Researcher Access

September 2021

Table of Contents

1. Definitions

Distributed Identity refers to a particular method of verifying one's identity and personal attributes to a relying party.

Distributed Identity is mainly used for accessing online services but there is no theoretical obstacle to presenting a digital identity in a non-virtual setting like a physical entry at a gate.

Distributed Identity is most useful at first-time access to a service, like at registration, because it allows for the user - the Holder - to share verifiable attestations and claims about their identity and attributes which may have been self-asserted or granted. These, in turn, should indeed be verified by the relying party. The relying party may then store some of the information gathered in a local account and issue local credentials, in which case in later interactions distributed identity plays a lesser role. However, it may also be the case that the claims are not stored, or that they need to be updated or maintained, and then Distributed Identity plays an equal role in subsequent interactions as in the first one.

The attestations and claims are granted to a person by a community or authority or by herself. ***These are stored by the user on their device, typically in a wallet application or browser extension*** and then presented to the relying party when accessing the service.

An advantage of distributed identity is that ***the user decides whether and what to release at any point***. Their ability to control attribute release improves privacy and data protection.

The verification ***could use a digital infrastructure like digital signature verification on a signed piece of data or consultation with a registry or a distributed ledger.***

2. Motivation

There are several advantages to be gained from a Distributed Identity system in Research and Education (DI4R) setting:

- **Better attribute aggregation**: in a DI4R setting, attribute aggregation happens within the user wallet. This enables attributes from more sources with the user in perfect control of the release.
- **Easier integration for the identity and service providers:** Providers need not federate - they can decide to provide or consume user information or stop doing that at any time and it is only up to the user whether they want to provide attributes or not.
- **No tracking by IdP:** In a SAML or OIDC setting, the Identity Provider can track in real-time where its users are logging in. In DI4R, the issuer cannot track any subsequent usage of the issued information and thus learn about the user's behaviour.
- Easier compliance with **GDPR**:
    - The user holds control over cards and can easily delete them.

    - For the IdP there is not much difference, except in terms of less control - the IdP cannot know/limit what happens with the credentials once issued - they can only track their inclusion into the user's wallet (including after a claim has expired or is revoked).
    - The IdP's ability to control attribute release improves privacy and data protection.
    - Not having a proxy (in the long run -proxies are temporarily very useful) is also a big advantage.
    - The authorization is decoupled from providing attributes.
    - The service is responsible for asking for only what it needs and whom it trusts and is responsible for claims regarding verification, authorization and GDPR-complied handling of released information.
- Easier in the ecosystem to exchange information without top-level trust route approval - basically a mesh-like federation.
    - Explanation: we came up with tagging in eduGAIN so that we don't break the trust model, while entities can still express additional content.

3. Work Done

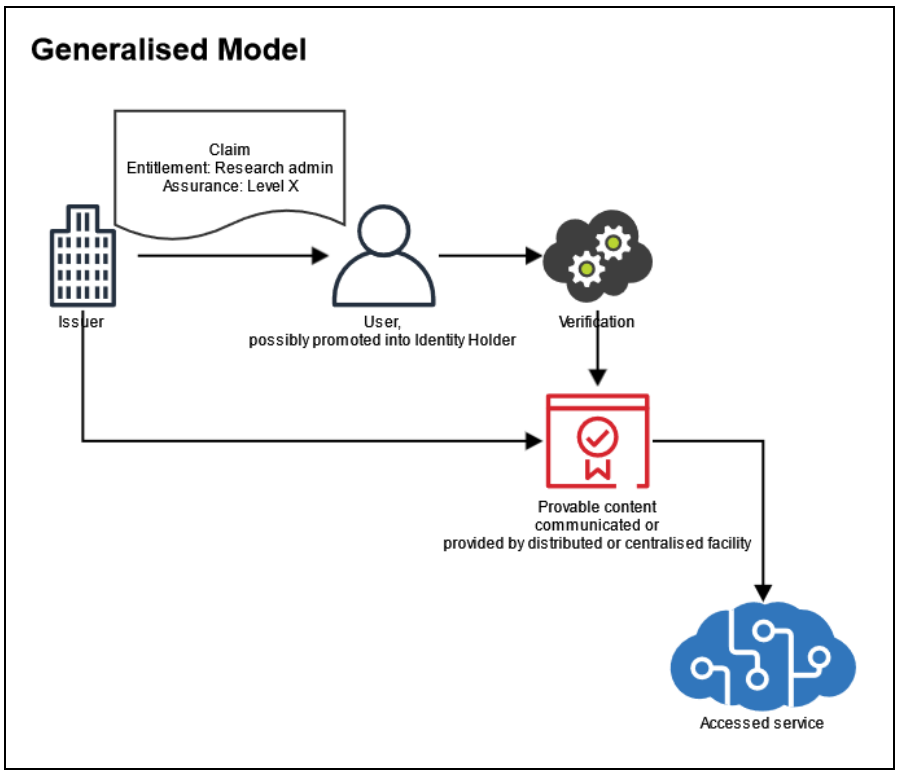
From Sprint Demo 4.6 - 21/22 September 2021:

- Implement and improve IRMA issuer in simpleSAMLphp
- Test verification of claims from multiple schemes
- Explore the best way to describe the scheme
- Discuss IRMA 'metadata' distribution risks
- Investigate assurance
- Device assurance
- Expressing assurance from the source
- Investigate revocation
- Multi-valued attributes

4. Functional Model

Here provided comparative overviews illustrate the transition toward distributed identities.

4.1. Sourcing of Claims



5. IRMA and Privacy-by-Design Federation

There is a Distributed Identity solution provider already in use in the EU, mainly in the Netherlands: the I Release My Attributes (IRMA) by Privacy By Design foundation.

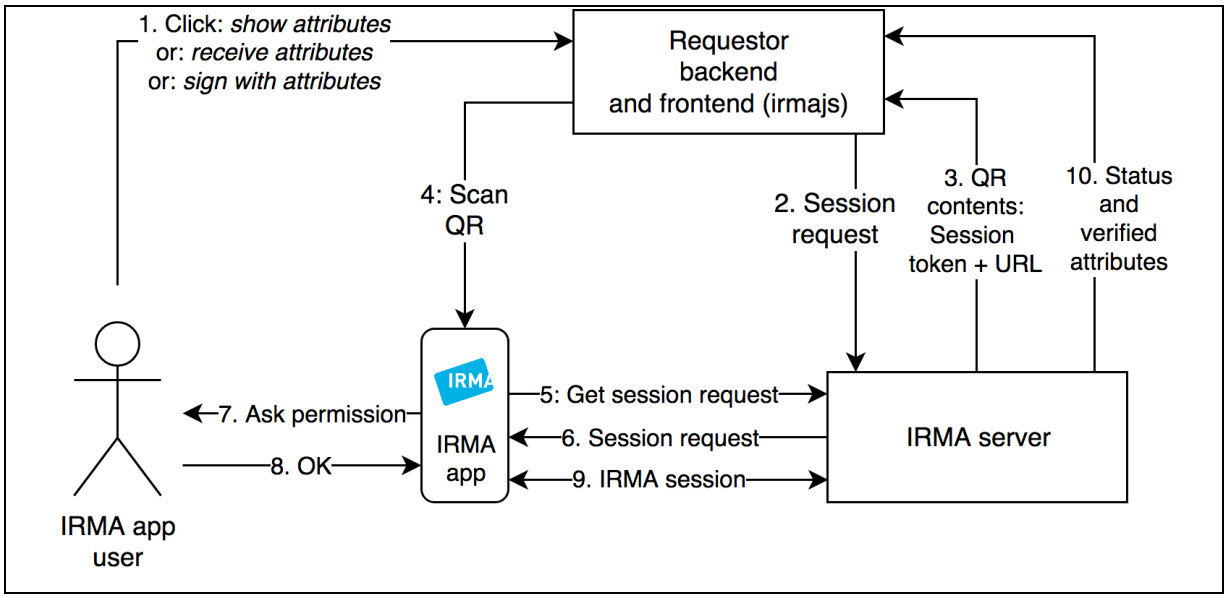
Since the components of that system are available on GitHub, and several key components are already Licensed with Apache 2.0, it is natural that we opted to experiment with that.

Therefore, from this point on, we specifically refer to IRMA as our prospective DI4R solution.

The source code of the system is available at https://github.com/privacybydesign.

6. Technical Model

How does verification work in IRMA?



Source: IRMA documentation: https://irma.app/docs/what-is-irma/#irma-session-flow

Software components [https://irma.app/docs/overview/]:

- *Requestor back-end and front-end*: Generally the requestor runs a website with a (JavaScript) front-end in the user's browser and a back-end server. During an IRMA session, the front end displays the IRMA QR that the IRMA app scans. All front-end tasks depicted in the diagram are supported by irma-frontend.
- *IRMA server*: Handles IRMA protocol with the IRMA app for the requestor.
- *IRMA mobile app*: Android, iOS.

Explanation of the steps:

1. Usually, the session starts with the user performing some action on the website (e.g. clicking on "Log in with IRMA").
2. The requestor sends its session request (containing the attributes to be disclosed or issued, or message to be signed) to the IRMA server. Depending on its configuration, the IRMA server accepts the session request only if the session request is authentic (e.g. a validly signed session request JWT) from an authorised requestor.
3. The IRMA server accepts the request and assigns a session token (a random string) to it. It returns the contents of the QR code that the front-end must display: the URL to itself and the session token.
4. The front-end (irma-frontend) receives and displays the QR code, which is scanned by the IRMA app.
5. The IRMA app requests the session request from step 1, receiving the attributes to be disclosed or issued, or the message to be signed.

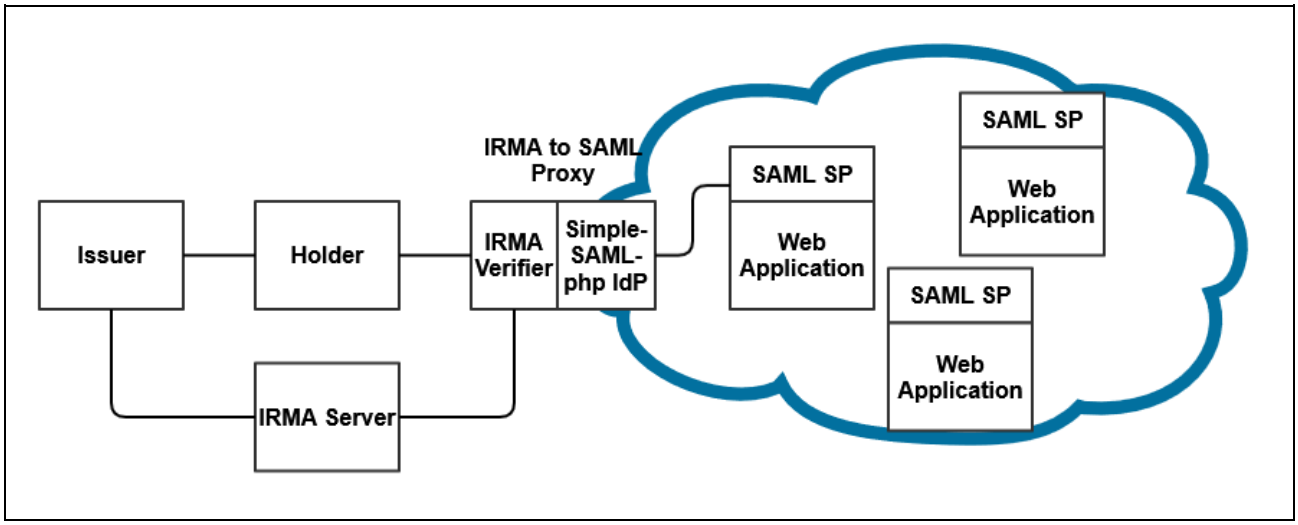
6. The IRMA server returns the session request.
7. The IRMA app displays the attributes to be disclosed or issued, or the message to be signed, and asks the user if she wants to proceed.
8. The user accepts.
9. The IRMA server performs the IRMA protocol with the IRMA app, issuing new attributes to the user, receiving and verifying attributes from the user's IRMA app, or receiving and verifying an attribute-based signature made by the user's app.
10. The session status (DONE, CANCELLED, TIMEOUT), along with disclosed and verified attributes or signatures depending on the session type, are returned to the requestor.

6.1. Use of Proxies

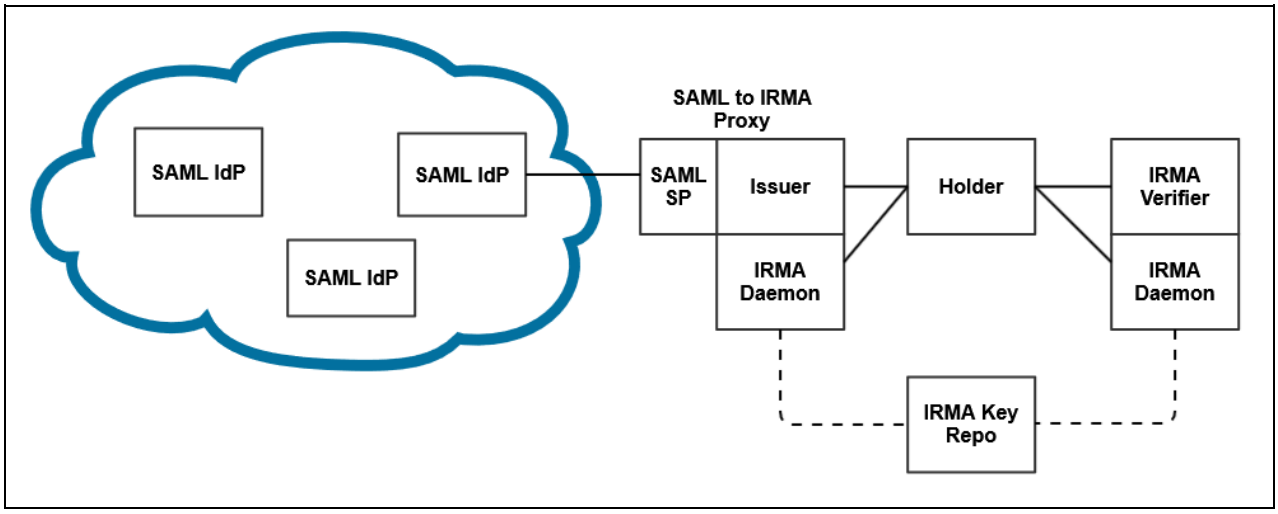
DI4R proxy approach is a logical extension of the available data sources for a service, for which the multi-protocol proxy was created in the first place.

In this arrangement, the sole source of all information is the Proxy from the SP's point of view.

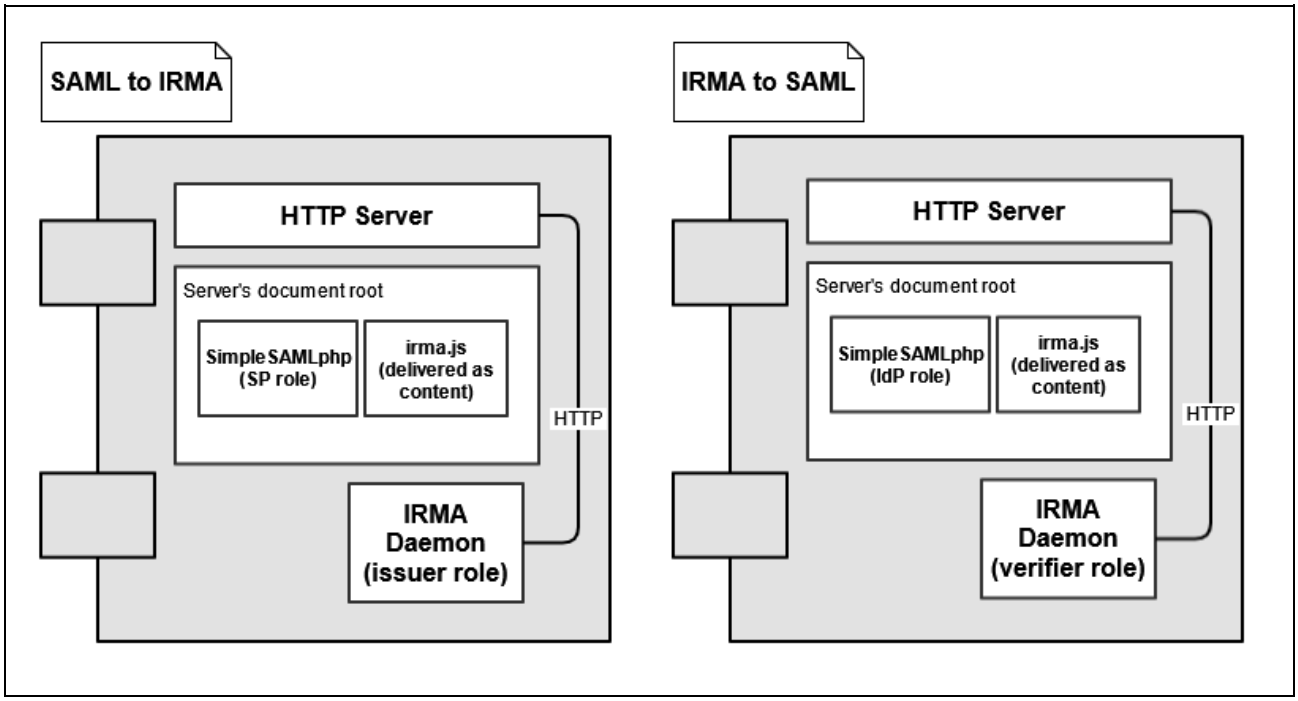
The IRMA-to-SAML proxy allows for logging on to SAML SPs with IRMA cards.



The arrangement works the other way around too: SAML-to-IRMA proxy provides the possibility of using a SAML federated account to get IRMA cards.



The next two figures illustrate the internal structure of a deployment.



6.2. Idemix

IRMA implements the Idemix Protocol to handle pseudonymous attribute handling. The Idemix protocol provides a way for users to use verifiable pseudonymous identifiers, coupled with certain attributes at services, without revealing their identity. With Idemix, this includes non-traceability across services, by the virtue of the user having credentials targeted to each service.

The credentials are always issued by an Issuer Organization. The User contacts the Issuer and registers an account and establishes a pseudonym. If the user is eligible for certain attributes, a credential will be issued containing the pseudonym and the attributes to the user.

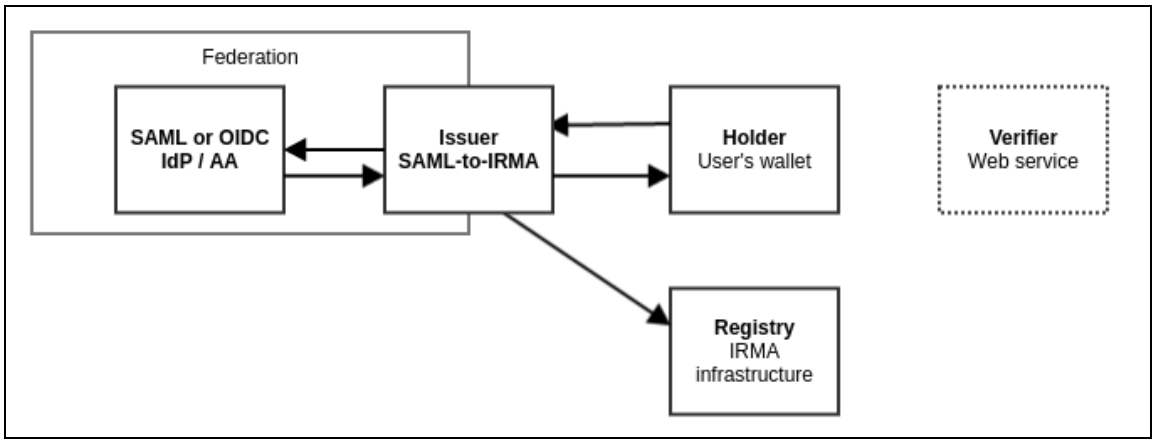
Then, the user can prove the possession of such credentials without actually revealing them to a verifier organization.

7. Use Cases (Generalised)

The following use case descriptions present some ideas of how the system may be used in an academic setting.

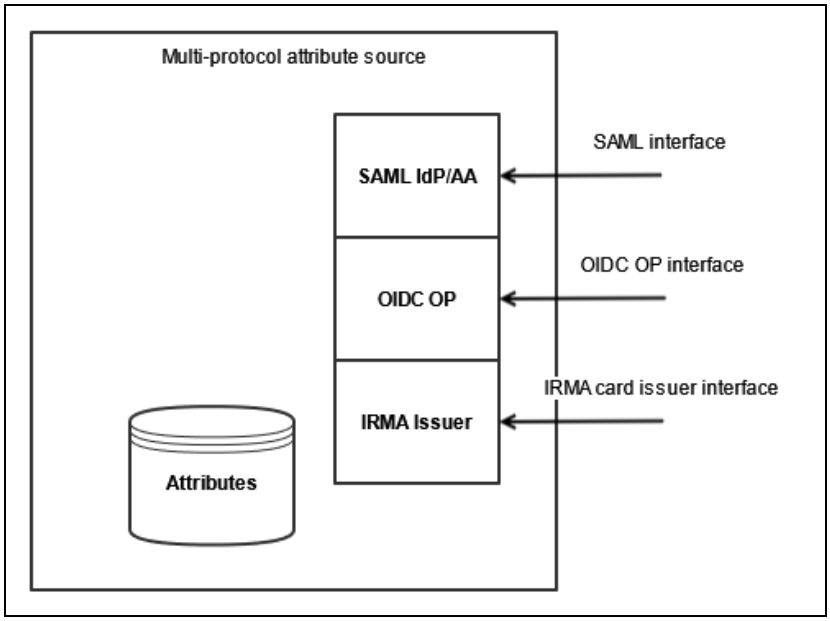
7.1. Issuer: SAML Attributes into IRMA Tokens

An obvious source of "cards" is a SAML federation. For a SAML attribute of a user to be converted to a card, the user needs to visit an entity that acts as a proxy. This proxy needs to behave as a SAML SP towards the user and the SAML federation. The user needs to visit the site with the intent of adding a card to their IRMA app so that the IRMA infrastructure can store the data as a card. The user will be logged in to this SAML SP which will consume the attributes from an IdP / AA and store them in the IRMA infrastructure.



7.2. Issuer: 'Native' Triple Stack IdP Issuing SAML, OIDC and IRMA

An authentication source may already have to support multiple protocols, (for instance, SAML and OIDC) in order to cater for the modern web environment. A logical extension of this idea is to support an additional protocol, the Card Issuer.

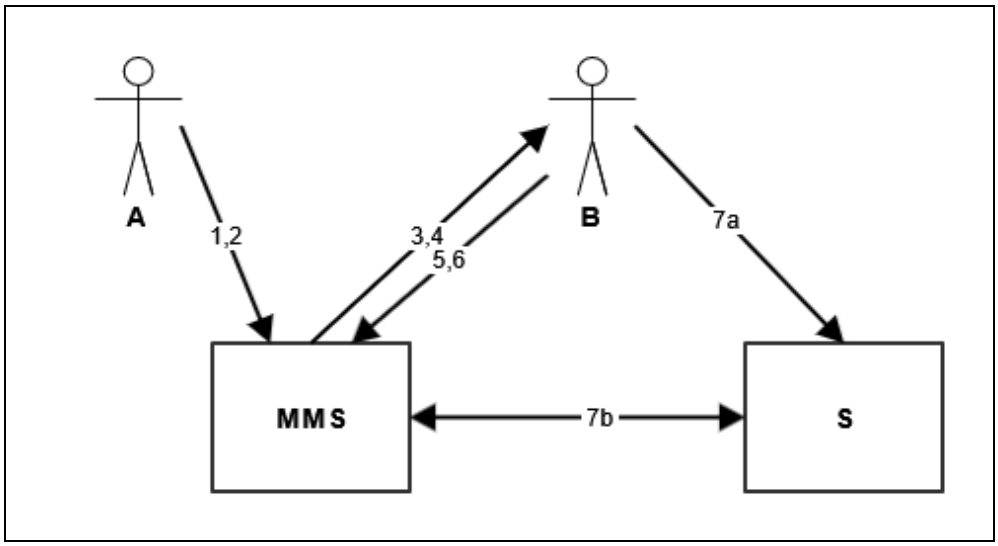


7.3. Issuer: Attribute Aggregation from Research AAI/MMS

In a traditional SAML flow, the following happens. The goal is to enable user **A**ladár (**A**) to manage the authorisation of user **B**éla (**B**) authorization to service **S**, but in a way that this information is not maintained in S but in an external source, the **M**embership **M**anagement **S**ervice (**MMS**).

1. **A** logs in on the web interface of the MMS, a SAML SP and an account are created.
2. **A** creates a Virtual Organization / Community / Group - terminology depends on the actual tool but let us call it (**VO)**
3. **A** wants to invite **B** to his **VO.** In order to do this, he needs an email address to **B.** This email address serves as a trust anchor for the moment, therefore it needs to really belong to **B** and not be compromised.
4. **A** sends an email invitation to B with a link containing a token. The email is sent by the MMS system.
5. **B** follows the link to the web interface of the MMS, prompted for login. **B** may already have a login (for previous participation in other **VO**s) or needs to create a new one. **B** may log in with a federated account but it could be the case that there is none, and a local account is created or a VHO account is used. This scenario is made possible by the fact that really the access to the email inbox is what provides the trust for the **VO** membership.

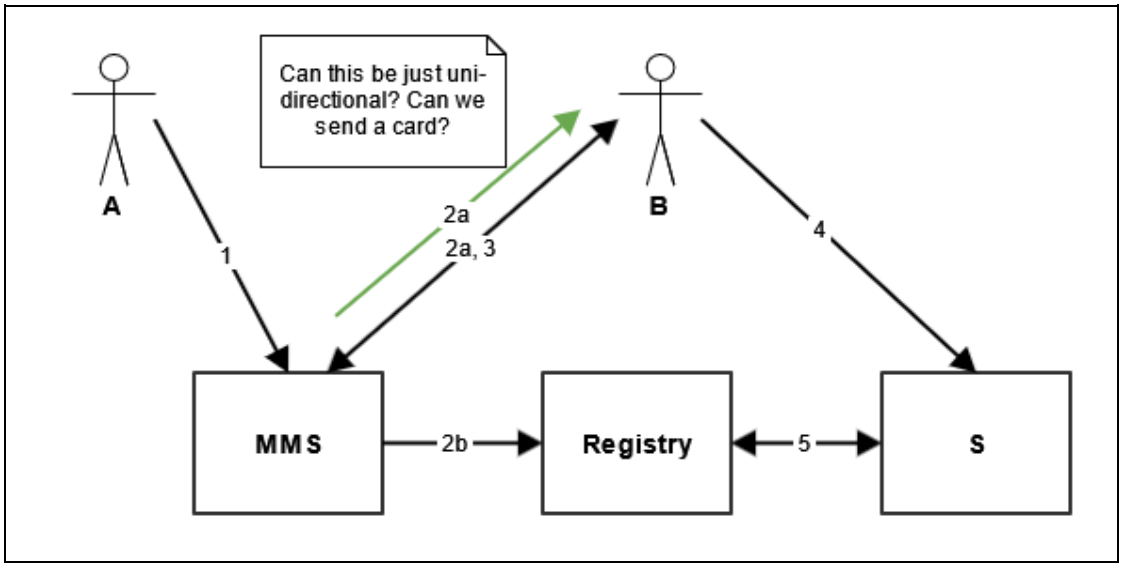
6. After creating/accessing a local account, the token sent in the link is processed and **B**'s account is now associated with the **VO**.
7. **B** will eventually access a service that needs this membership information, commonly called entitlement.
    1. The service will perform a login flow.
    2. With **B**'s user identifier queries the MMS back-end, for instance, a **SAML AA** or an integration. This requires the usage of the same user identifier that was used at the MMS, typically a common OIDC/SAML source.
8. **A** may revoke the entitlement at any time, which will take effect at the next session: the service accessed will query the MMS and will not get the entitlement.



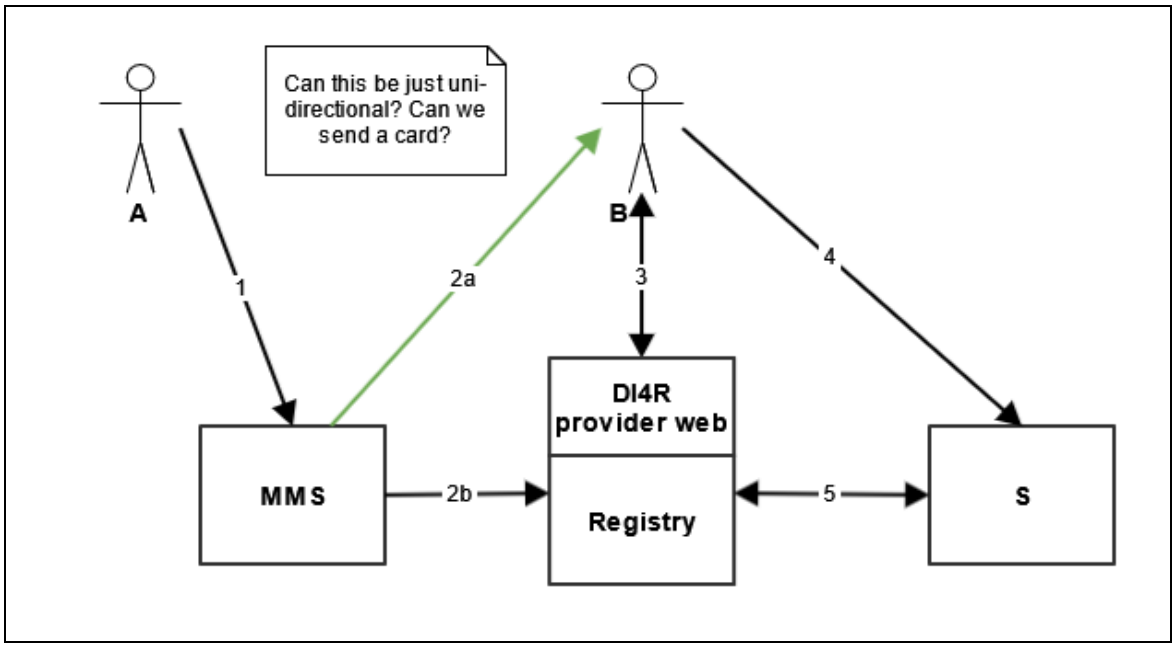
With the introduction of DI4R, the flow may be significantly simplified.

1. **A** creates a **VO** at the MMS service.
2. **A** sends an invitation to **B** to the **VO.**
    1. An email is sent to B from the MMS.
    2. A card is registered to the registry?
3. **B** visits the link and receives the card, which is added to the wallet.
4. **B** visits a service that needs the entitlement and presents the card.
5. (The card is verified in the common registry, therefore revocation is possible.)

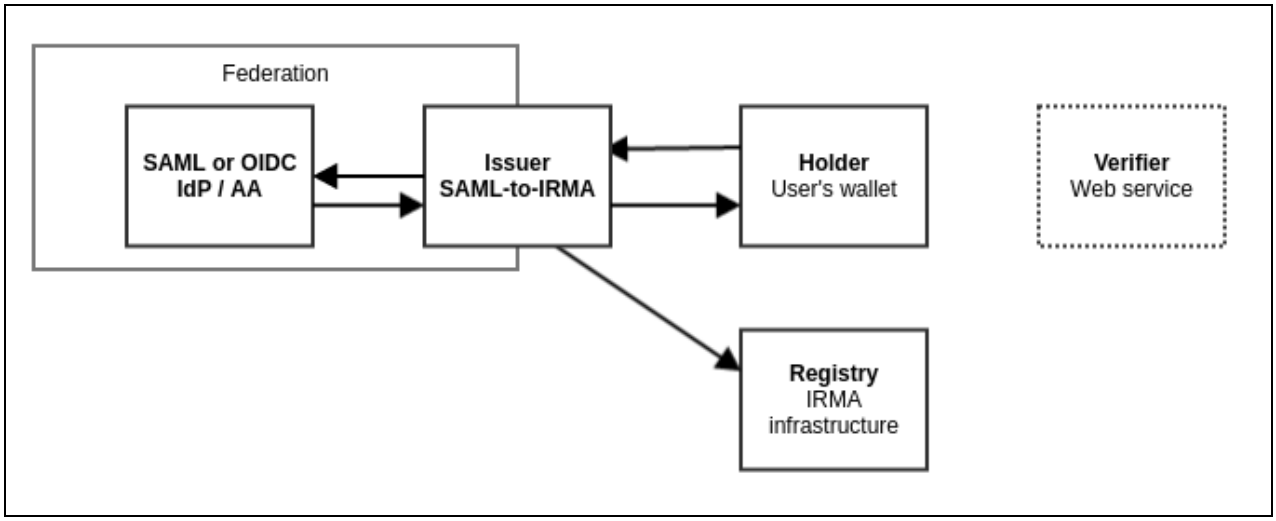
With this solution, **B** does not have to use the same login (i.e. the **MMS** and the target **S** do not need to be in the same federation). Possibly, B can receive the card at a page maintained by the DI4R provider.



Or, perhaps the DI4R provider's web interface serves as a landing page for the invitation:



7.4. IRMA Proxy as attribute aggregator



7.5. Issuer: Journal Use Cases

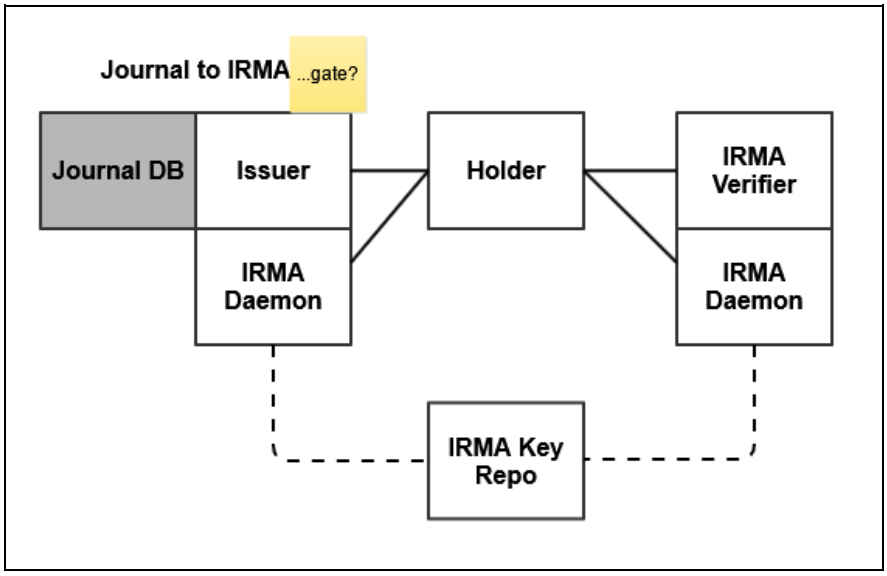
In the academic peer review process, honest opinions from an expert in the field are crucial. There is an inevitable tendency for specialization in science because any modern problems can only be tackled in focused, career-long efforts, so in most sub-disciplines, the researchers will tend to know each other. This, however, presents a challenge for the review process. In order to overcome the challenge, in the most widely used review processes, a degree of anonymity is introduced.

- The "Single Blind" process is considered to be a minimum requirement - in this case, the author does not learn the identity of the reviewer. For most journals, this is considered insufficient, since the reviewers still know the identity of the author and they may be biased in one way or the other. Yet, in some cases, especially in less common languages there is no true alternative as the content of the article drastically narrows down the set of possible authors, sometimes to one. In these cases the more anonymous methods are disingenuous.
- The "Double Blind" process means that neither the authors learn the identity of the reviewers nor the reviewers of the authors. This is the most common type of peer review process. But it still leaves significant control in the hands of the editor, who knows the identity of both, plus, due to the structure of the fields of science, she may personally know all parties and have their own interest. The editor may also know the review styles of particular reviewers based on previous engagements. Therefore it is possible to pick a lenient or a strict reviewer for a given paper for instance.
- The Triple Blind method prevents this problem as the identities of the author, editor and reviewer are unknown to each other. However, this is the hardest to implement, as the editor still needs to be sure about the expertise of the reviewer, moreover, she should also know that the author does not temper the process by being its own reviewer or bringing in friendly reviewers. At this point, the scientific process becomes somewhat analogous to e-voting systems.

- Furthermore, all three types of blind reviews have a common problem, which is that the work of the reviewer cannot be easily credited to them. This disincentivises the reviewers from participating and therefore is a drawback for the entire scientific process.

In order to overcome these challenges, an editorial system could issue certificates for editing, reviewing and acceptance.

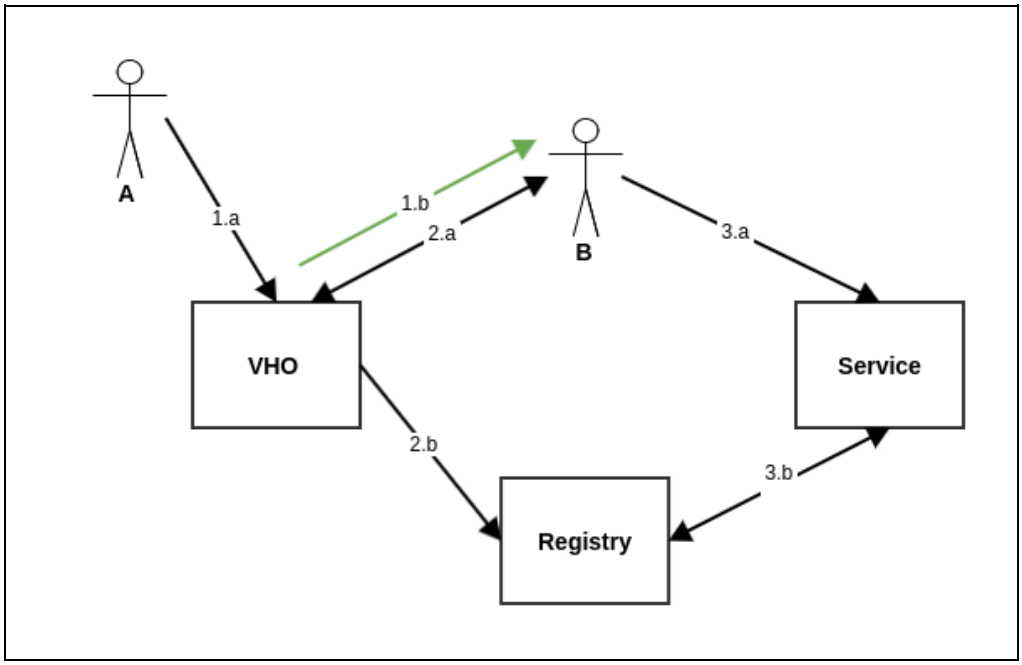
- The certificate of acceptance will contain the name of the author and the metadata of the article, therefore this can be handled as a simple card.
- The certificate of editing will also contain the name of the author and the edited issue, making it very similar to the certificate of acceptance.
- The certificate of review should also be connected to the person who did the review but it should not reveal what the review entailed. The way for doing that is to be in a large enough set of people so that the k-anonymity is sufficiently high. Otherwise, based on the exact timing and the fields of interest of a reviewer an author might be able to guess who did their review. Therefore, only larger time ranges (e.g., a year range) should be revealed. Smaller journals may want to pool themselves together and issue a certificate that only says that the review was done in one of the journals in question.



7.6. Virtual Home Organization

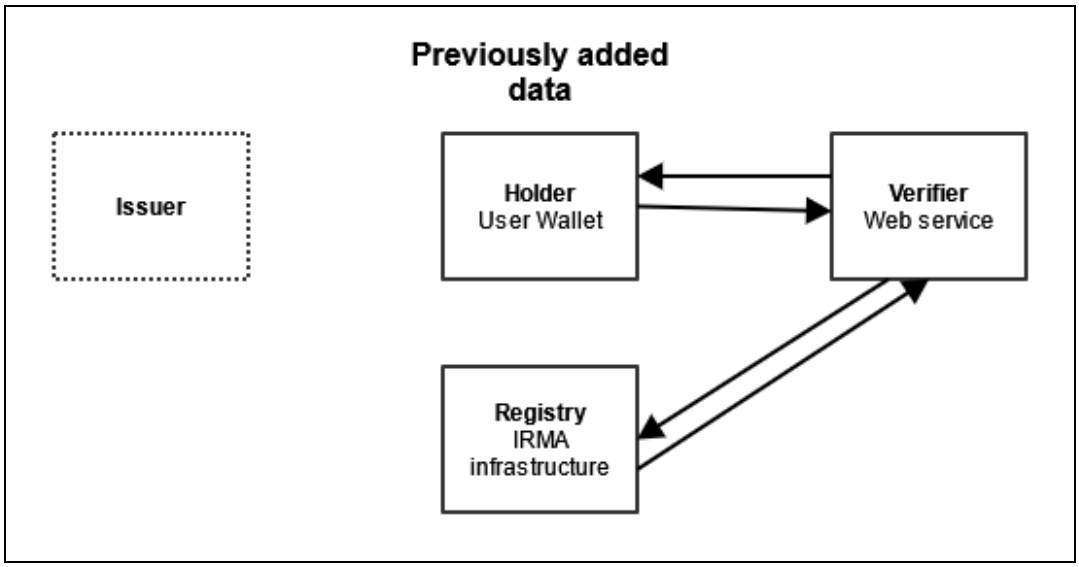
The Virtual Home Organization use case helps users wanting to access research & education infrastructure without having a home organization that is technically enabled with the accessed services on a technical level. While the technical integration is missing, the user may have a completely valid claim on access. In these cases, a virtual home organization (VHO) is used. In this description, we present the sponsored VHO use case, in which one user (within the technical collaboration) sponsors another (outside the collaboration) by an invitation.

1. User **A** (the sponsor already in collaboration)sends an email invitation to user **B** (outside the technical collaboration).
    1. A describes B on a form at the VHO and inputs the email address of B.
    2. The email is sent out.
2. User **B** visits the VHO service and receives a card that describes their identity as stated by **A.**
    1. **B** visits the service, reviews the data stated about her by A, and receives a card.
    2. The card gets registered to the registry.
3. User **B** can now access services within the collaboration.
    1. **B** attempts to access the service.
    2. The service verifies the card in the registry and allows access.



7.7. Verifier: Consume Holder's Credentials

Any entity that normally relies on an authentication flow that also aggregates attributes may use IRMA or another service for login. In this process, the user is challenged with a QR Code to brandish attributes with the help of the wallet app. The wallet app reads the QR code and engages in user interaction: it shows what is requested by the service and which "cards" - previously-stored attributes accommodate the request if any. Alternatively, in this flow, the user may acquire new cards to fulfil the request. The wallet then sends the attributes to the service, which can verify them with a background call.



With this method, the Verifier no longer trusts an IdP (something that is exposed on the public internet) but trusts the authentication and the possession of the wallet. Arguably, this provides the opportunity for a stronger level of assurance (i.e. two factors to the wallet+possession of the device).

8. Issues to Address and Discussion

8.1. Assurance

Since many sources can provide IRMA attributes, the IRMA platform does not standardise levels of assurance beyond individual profiles. Assurance levels are provided by using the corresponding schema-defined credential attributes, that is, IRMA passes on the level of assurance provided by sources only if these levels are incorporated into the used schema and implemented by the IRMA issuer.

For example, the attribute “assurancelevel” is used in schemas that provide data from passports or ID cards, and it conveys the levels set by the document issuer or an intermediate entity that collected and verified the information provided with the credential. This level is in line with eIDAS. Some other schemes use “digidlevel” to provide the level from the Dutch Digital ID (digid.nl), which is the assurance with which identity is verified in the Dutch population register.

The user may select what credentials from available they want to present to the verifier. The verifier can determine which attributes it does or does not accept from which sources. It can also state the required attribute bundles by using IRMA "Condiscons" (CONjunction of DISjunctions of CONjunctions), which allows verifiers to specify attribute sets coming from a single credential instance. With this, a service can require a composition of alternative bundles of attributes, even if they are using different schemes to provide the relevant data and corresponding LoAs. However, the use of a consistent attribute schema and semantics

of levels may greatly simplify this selection, along with a mechanism informing verifiers about trustworthy issuers participating in such a schema.

In support of assurance, the IRMA platform allows defining the optional validity period of credential at its issuance; if skipped, a default value of 6 months is assigned. The validity is always rounded down to the nearest week.

Another important supported mechanism is revocation which is described in more detail in the corresponding section.

8.2. Multi-Factor Authentication

When it comes to traditional authentication sessions, the need for separate authentication factors for high-stakes sessions has been long acknowledged.

Login names and passwords may fall prey to a hacked browser or operating system, in which case an independent channel - a one-time password challenge, a push notification on a secondary device, paper-based factors or even something as weak as an interceptable SMS can provide a crucial last line of defence.

Most of the currently trending second-factor options assume the primary channel to be a desktop or laptop computer and the mobile as a secondary device.

However, IRMA is a mobile-only application, so the user is less likely to have a second mobile device that may act as an independent source of authentication.

This is a drawback as no independent channel is provided for the user's access to their information. The information is stored on the user device, defended by a PIN, which, should the mobile operating system get exposed, will also be compromised, and the private information can be stolen.

At the same time, many factors alleviate this concern. For one, platforms with walled gardens are more controlled software/package management are likely to be less exposed to malware. This of course deteriorates once the operating system is no longer supported.

Another important factor to consider is the very nature of the distributed identity system - there are no huge amounts of sensitive data on any one device, instead, all the devices store their owner’s data only, making them much less valuable targets.

Moreover, the client devices normally don’t provide server functionality at all, that is, they are usually not addressable under a well-known DNS name, and there are no ports exposed.

Besides all this, the mobile device is much more personal than the desktop or laptop device and is much better tied to the user, also, by the virtue of providing the primary means of personal communication, a mobile device is much sooner reported and their connection remotely deactivated when stolen or lost.

This shows that the lack of a second-factor option in the case of a mobile wallet may be compensated by other factors.

8.3. Alternative Wallets

With any popular system that relies on a particular type of device there comes a point where the question arises: how to cater for those potential users who do not own the right kind of device?

An alternative wallet in this context means a non-smartphone implementation of a wallet. While having the ability to use alternative wallets seems a necessity as the user base grows, a non-smartphone implementation comes with several challenges.

One such challenge is the QR-Code reading for which the smartphone is especially well suited, but is not impossible in another architecture either, i.e. a browser extension. Another challenge is the safe storage of the 'cards', but that also can be done.

8.4. Attribute Revocation

Revocation is enabled per credential type in the IRMA scheme. If so, the properly configured issuer's IRMA will issue revocation-enabled credentials of that type. If the user has a revocation-enabled credential then proving non-revocation is not required; instead, they can just disclose attributes from the credential, which is much cheaper. Non-revocation is still ensured by using revocation update messages which are created whenever an issuer performs a revocation, which also distributes issuer-related information that is updated at the time of revocation and is necessary to disclose attributes.

During attribute disclosures, IRMA can prove non-revocation, but only if explicitly asked for by the requestor. The reason for this is that computing a non-revocation proof for a credential is much more expensive than just computing a disclosure proof out of that credential. For this, IRMA will only prove non-revocation for a credential type if the requestor explicitly requests it. Requestors should only request non-revocation proofs when it is really necessary for them to establish that they received non-revoked attributes.

Additional source: https://irma.app/docs/revocation

9. Development Work and Demos

9.1. IRMA Issuer Setup

IRMA issuer consists of a small PHP server that relies on simpleSAMLphp for authentication. In the case of success, this call results in a populated attributes array that is then fed into the IRMA daemon session request API for an issuance session and the result is handed over to the JavaScript handlers. The JavaScript then requests the IRMA daemon using the result of the issuance session request and shows the result.

9.2. IRMA Verifier Setup

The IRMA verifier is based on the simpleSAMLphp framework and implemented as an "authsource". It shows a web form and creates a disclosure session request using the IRMA daemon API. The result of this request is then handed over to the JavaScript handler and on receiving the successful disclosure response, the form is POSTed back to the simpleSAMLphp authIRMA handler and further processed as a valid authentication.

10. Future Work

- Multi-valued attributes
- Alternative wallets
- **Scalable schema definition for a size of a federation like eduGAIN (of issuers)**
    - 5k or more entities
- Peer-to-peer claims (cards)
- Pixie dusting - claiming that someone is your co-worker, club member, etc.
- Conventions on prefixes for wildcards used on attribute names
- Use of multiple schemas and schema selector
    - UX is not hard but needs to be done well
    - Allows for different universes of DI4R
- Enhanced presentation of cards
    - Since the user is in charge of exposing cards in their wallet to the service, it is important to present these cards, their content and their source in a clear but informative way. This requires further establishing of a standardised and scalable way to specify their presentation and access to supporting information which is also interoperable with existing identity infrastructures and trust frameworks.
- Usability testing/evaluation
    - DI4R is a new concept, so it is a reasonable question whether the users understand the flow at all and the benefits that justify the adoption of changes. Should be done with appropriate early adopters such as researchers involved in Open Science.