Driving innovation together

SSI wallet for education and research

A technical exploration

Many parties, including government and public and private organisations, have their eyes on wallets, or digital personal data wallets. Not only nationally, but also internationally. This solution offers advantages for education and research. The technology of self-sovereign identity (SSI), which gives users more control over their own personal data, holds great promise here.

SURF's Trust & Identity team has been researching SSI for several years. We are now applying that knowledge to the concept of wallets. We use this knowledge to support institutions in their IAM development. This report explores the possibilities of an SSI wallet in education and research settings via a proof of concept SSI wallet.

Table of contents

1 Introduction to SSI

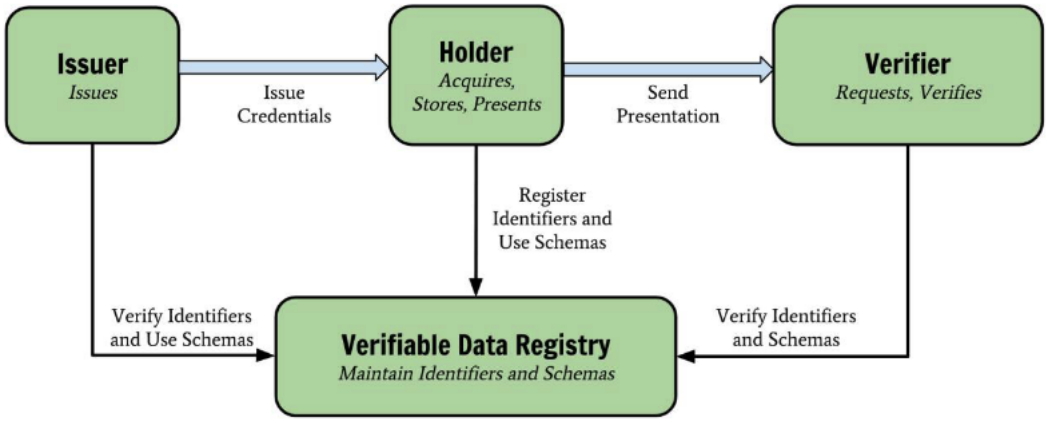
This brief introduction to self-sovereign identity (SSI) sets out the background against which we developed our proof of concept (PoC) wallet app, namely SSI technology.

What is SSI?
SSI is a new paradigm in identity and access (IAM) management. A key difference from existing identity management ecosystems, such as federated identity management, e.g. SURFconext, is that users have more control over the use of their personal data. However, SSI is also relevant in the broader playing field of data exchange, or data governance. This is visible at the national and European levels, including in the new eIDAS regulation: a login option intended for European citizens and companies who want to log in to Dutch services, for instance, using their national authentication means from their home country, i.e. without DigiD or eHerkenning.

How does SSI work?
In the SSI model, the data source, for example an educational institution, gives data or attributes ('credentials') to the user, who can store them in a wallet. This wallet usually comes in the form of an application on the user's mobile phone. When a user wants to purchase a service, the service asks the user for a certain proof, for example, that the user is a student. The user can then choose to release this to the service – via mobile phone.

Roles in an SSI ecosystem
The main roles in the SSI ecosystem are as follows:

- Issuers: parties who can issue attributes (data) about a person.
- Verifiers: parties who want to know something about a person; they request attributes.
- Holders: individuals who own a wallet and hence have control over attributes.
- Verifiable Data Registries: parties or places that keep track of which metadata and schemas can be used to share attributes and identifiers.



Exploratory technical report on SSI
In 2021, SURF carried out a technical exploration of SSI. The report of this study sets out what

the technical features, opportunities and challenges of this concept are. You can read about it in that report.

For more information on the basics of SSI, see: www.surf.nl/ssi

2 Developing an SSI wallet proof of concept

In this project, we take the opportunity to experiment with SSI, contributing to the development of identity & access management (IAM) within the education and research setting. We connect with existing projects and services and we identify relevant projects outside the sector. The focus in this proof of concept (PoC) is on the technology, functionalities and use cases of an SSI wallet, and we have opted in the implementation to develop a prototype of a working application.

2.1 Background and objective

Within SURF, we have been conducting research for some time into ways in which the education and research sector can respond to developments concerning SSI and wallets. We use this knowledge to support educational institutions in their IAM development and to improve SURF services. To provide a framework for the research into SSI, SURF launched an SSI lab to bring together and share knowledge and experience of SSI and SSI technologies within SURF. The technical exploration into an SSI wallet for education and research was carried out within the lab.

SURF and the sector wish to experiment with the technology behind SSI: how does it work and how does it build on current SURF services?

We describe below the main projects and services where we see common ground for our SSI Wallet PoC:

- eduID, a single identity that allows students to access any educational institution, could play several roles within the SSI ecosystem. Developing eduID into an identity wallet is on the radar, but its impact and desirability are not yet fully understood. We are investigating whether the current functionality of the eduID app (login and authentication) could be combined with SSI wallet features.
- We provide input for the PoC 'eduWallet', as formulated in Npuls, the eight-year programme to improve education in secondary vocational education and training (mbo), higher professional education (hbo) and research-oriented higher education (wo) by better exploiting opportunities of digitisation. We see the issuing of microcredentials, such as edubadges, as an important use case for a future eduWallet and want to develop this in an experiment.
- We test ideas from the HOSA IAM (in Dutch), related to SSI. HOSA is the Higher Education Sector Architecture. The HOSA domain architectures provide (architecture) frameworks for common sector facilities within higher education. The Identity & Access domain architecture defines frameworks for identity & access management (IAM).
- In the programme ‘Doorpakken op digitalisering voor het mbo', we test concepts and user patterns from our own file (in Dutch), as outlined in the programme ‘Doorpakken op digitalisering’.
- We are testing concepts and user patterns from our own file (in Dutch), as outlined in the national programme for mbo ‘Doorpakken op digitalisering’.

- We provide input for the European large-scale pilot DC4EU, in which SURF participates in 2023, and which contributes to the implementation of eIDAS2.0. For example, in DC4EU, we are working on issuing educational credentials and exchanging them in Europe via the eIDAS system.

We also see that, to achieve the goals in the digitisation impulse for education (Npuls), APIs and middleware are needed to access services in the education ecosystem. Simple authentication and authorisation of all users is a prerequisite for this. SSI offers opportunities to set this up under end-user control, but the introduction of this technology will require a long-term path. A key issue here is how an SSI-based infrastructure relates to the current federated infrastructure: the most likely scenario is that the two will coexist for the time being.

Objective of the technical proof of concept for the SSI wallet
Given the above requirements and demands, a technical proof of concept was conducted within SURF's Trust & Identity department to:

- develop a wallet app based on open-source components;
- experience the impact of developing a wallet in an education and research setting;
- identify and - where possible - answer questions that arise during development.

The goal is not to arrive at a production-ready environment. Besides testing out SSI-related use cases, the wallet was combined with components for federated authentication (SURFconext, eduID). We did this explicitly to explore what a possible growth path for the institutions might look like if an ecosystem of wallets become a reality.

Use cases from education and research, student target group
The use cases we examine are related to education and research, with students as the primary users. However, teachers and researchers may of course also use a wallet. An important aspect of SSI is that users themselves determine which (education) data they exchange and with whom. This data may be needed to follow a programme of study or to use certain services. For instance, providing proof of student status to access student accommodation. Or presenting a degree or microcredential to a future employer. A wallet - a digital wallet containing personal data - helps students do this.

What data to exchange
What data should we be thinking about in education and research? Below are some of the main attributes exchanged in the SSI ecosystem (see also chapter 1 Introduction to SSI):

- Affiliation (are you a student, lecturer, pre-student etc.)
- Home institution
- Host institutions
- Diplomas
- Microcredentials
- Identifiers
- Student number
- Access rights
- Student name

- E-mail address

This list is not exhaustive, but these attributes are the basis for many of the transactions that take place in processes in the education and research ecosystem. We explore the use cases in more detail in section 2.1.

Context exploration
In the field of wallets, development continues apace, and insight is advancing. The exploratory study which this report describes took place from November 2022 to February 2023. The study should be seen in the context of this period.

Many SSI wallet concepts have now been developed in the Netherlands and internationally, by private and (semi-)public parties. For instance, Yivi (formerly IRMA), Ockto, SOLID and EDI wallet (BZK). In practice, the places where these kinds of wallets are used are still limited. Many questions therefore remain unanswered:

- Functionality: what can you do with it? What should the wallet be able to do?
- Users: do people understand that they have control of data and what this means?
- Impact on institutions: how does SSI change the IAM landscape of institutions?
- Impact on service providers: how does SSI change access to service providers?
- Technology: how does it work? What technical standards are there? How is interoperability arranged?
- Policy: who decides which personal data can be shared with which party and when? And how? Is this really only up to end users?
- Maturity: to what extent can it already be deployed? What solutions are already in place?
- Business model: who will pay for the transition to SSI and does the added value of SSI make it worthwhile?
- Use cases: what are the practical examples? In which education cases is it relevant?
- Privacy and legal aspects: what is the division of roles in an SSI ecosystem? How do we implement privacy and data minimisation in this ecosystem?

These questions need input from various domains and areas of expertise. We cannot answer everything in this phase. In this PoC, the focus is on technology, functionalities and use cases. We also look at user-friendly designs. However, no user studies have yet been carried out.

Outside the sector, there are also several relevant developments, including:

- Development of European legislation around EU Identity Wallets (eIDAS 2.0).
- Further development of open standards, such as Open Badges 3.0, various W3C standards and at OIDC4VC.

2.2 Methodology: prototyping

The methodology for this technical study is prototyping. However, prototyping goes beyond paper; we produce a working application. To achieve concrete results, we develop the application using an agile methodology.

Setting up a wallet app with a corresponding ecosystem of connected entities requires several components. The starting point here is the use of open-source components, preferably components in line with the European Commission's specification for the eIDAS reference wallets (ARF - Architecture Reference Framework) and SURF's already available components.

In addition to the technical development of a PoC wallet, this project expressly serves as a means of identifying further questions about SSI and wallets and finding answers. The functionality produced can serve as a basis for pilots and user testing and be further developed through further iterations.

3 Use cases, design and application development

To arrive at the prototype, we initially created a design based on a number of use cases and design principles. The principles are in line with SSI, fit the context of the sector and comply with SURF's open development methods. The named use cases were elaborated into functional requirements. We then determined which technical components were desirable to use as a basis for the prototype.

3.1 Notes on the design

There is a great diversity of use cases that we expect wallets to be able to support. We described a number of cases and worked out the technical components needed for them. In addition, we looked at designs of existing wallet apps for inspiration.

3.1.1 Use cases

Most use cases in which a wallet would be beneficial are mainly related to the supplies and services students use during their studies. Use cases in which the flexibility of education is central or where lifelong learning is facilitated are particularly interesting because SSI and wallets are useful at improving the exchange of data across institutional boundaries. This translates into use cases such as the following:

- Gaining access to student accommodation facilities.
- Gaining access to further studies or modules based on affiliation to an institution.
- Receiving awarded diplomas, microcredentials and skills as a professional or student.
- Showing proof of awarded diplomas, microcredentials and skills to an employer, potential employer or further education provider.
- Obtain discounts based on student status when buying products or admission tickets.

This list of use cases is not exhaustive. Many variations can be defined based on the required services (offered by verifiers) and attributes (offered by issuers) so that the person (holder) can do what they need. We think the basic functionalities of a wallet to support these use cases are similar despite the differences between the use cases.

3.1.2 Functionalities

The above use cases need attributes such as affiliation with an institution, student number and microcredentials that can be shared via a wallet app. We have defined some generic functionalities needed to perform the following action:

- Activating the wallet
- Adding attributes
- Sharing attributes

Besides these basic functionalities, some functions are needed to give the person control over the data:

- Viewing and editing content (profile data, attributes)
- Viewing activities (view history of exchanges)

Expiration and revocation of attributes by the issuer (revocation) are also basic functionalities that a wallet must provide. Mainly for the sake of the feasibility of the prototype within the set timeframe of the study, we deliberately excluded revocation of data at this stage of the study.

3.1.3 Design principles: privacy by design

We apply the principles of privacy by design in the design. This means creating privacy-friendly products by already thinking about this privacy in the design phase. We also want to do this in a way that allows the individual to retain as much control and self-determination as possible. We guarantee this by applying the principles of SSI in the design.

This leads to the following principles:

- The user must be able to make privacy-friendly choices as easily as possible and be helped to do so through the design of the application.
- As close as possible to the ten SSI principles:

1. The user exists and has an identity.
2. The user has control over the identity.
3. The user has access to data.
4. Systems and algorithms should be transparent.
5. Identity persists for the long term.
6. Portability of data and identity.
7. Interoperability: the identity is as widely deployable as possible.
8. Consent: the user must give consent for the use of their identity.
9. Minimisation of data and its release.
10. Protection: the user's rights are protected.

Design system
One of the intended objectives was to test a wallet specifically for education and research settings. This means that the application should be recognisable to target groups in education and research. This is reflected in the application's style by using the SURF Design System wherever possible. The design system is in designed for websites. The design of the application simultaneously provides input to a mobile version of this design system. This leads to the principle:

- Use the distinctive SURF style in the design of the wallet.

Link with eduID
The wallet should also be able to interact functionally with already available technical components in the sector. This is reflected in the identity initially linked to the wallet: eduID. This identity is intended and available for applications in both education and research. In addition, we use attributes relevant to the sector. This leads to the principles:
Use an identity that is deployable across the sector (eduID).

- Use attributes needed within the sector (such as edubadges and affiliation).

Discussion topic: where to start the flow in the app
A key topic for discussion in the design was where a flow (the steps a user follows in the app) should start in the application. Looking at other wallet applications, we see that different approaches are possible. What is common is a trigger where the person is invited to fill the wallet with attributes. However, this leads to the collection of information in a new place, while there is no good reason for this (purpose limitations).

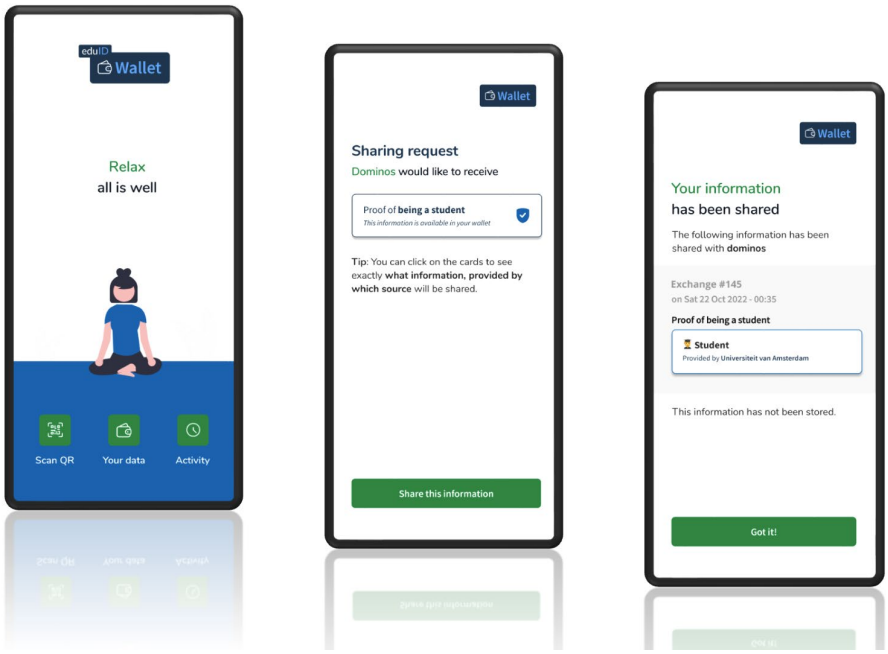
In this design, we prefer to start flows when the person starts using certain services. This leads us to start from the verifier (the service provider) and the information needed to enable the specific service at that moment. This means that a person will sometimes still need to retrieve attributes from sources although they may not yet be available in the wallet. This leads to the following principles:

- Always work from the perspective of the person who wants to perform a task and therefore initiates an action.
- The verifier's request and required attributes are authoritative in the representation.

The ambition is to use open-source components. In addition, where possible we want to align with proposed standards, such as the EDI wallet of the Ministry of the Interior and the European Commission's specification for the eIDAS reference wallets (ARF - Architecture Reference Framework). The reference wallet also takes mobile applications as being foundational. In addition, we expect the use of mobile apps to be commonplace among the target group of students. This leads to the principles:

- Interaction via mobile application is foundational.
- Align with wallet standards where possible.
- Use open-source components.

The designs show how the wallet can support the use cases. The designs produced are explained in more detail in ‘Appendix 1. Functional designs’.



3.2 Application components

We develop the functionalities specified as much as possible based on existing components. There are several reasons for this. Firstly, it creates a true representation of reality, because only then can we see how the various components connect to existing systems and services. Secondly, using existing components benefits the speed of the research, partly because we can build on existing expertise. Finally, it is important to give input to existing components within SURF on possible changes needed in a wallet-based future.

To achieve the desired functionalities, we used:

- Yivi (IRMA): an existing open-source SSI wallet that SIDN is developing in collaboration with Radboud University Nijmegen. Yivi provides the basis for the wallet component, the necessary schemas for attribute exchange and the structure of issuers and verifiers in this PoC.
- eduID: available and intended as an identity in education settings, developed by SURF. Here, the user activates and links the wallet. In addition, eduID makes it possible to obtain the affiliation attribute.
- SURFconext: existing identity federation which service providers and institutions can link to. It provides infrastructure for accessing education services and identification and is used in the PoC when obtaining the affiliation attribute from the institution if it is not already known within eduID.
- edubadges: the platform for digital certificates for Dutch education, developed and managed by SURF. edubadges is a system that allows users (holders) to exchange microcredentials and skills.

Based on these components, we expect to be able to make a good assessment of the challenges and the choices that need to be made to produce a workable wallet for education and research.

3.3 Products realised

The research and application development took place from November 2022 to March 2023. Several products were realised during the research period. All products are available via the SURF wiki.

Designs
Flows were designed for all functionalities stated, and corresponding screens were developed for the app. Designs were also made for the verifiers. An overview of the various screens is attached in ‘Appendix 1. Functional designs’.

App
All screens were implemented in an Android application, using Yivi as the back end for managing the wallet.

Source code app
The app went through several iterations during development. The app's source code is available under the Apache 2.0 licence (where possible).

Source code development suite
For testing and developing the various scenarios, extensive use was made of the Yivi CLI. This application makes it easy to set up verifiers and issuers and to set up and test scenarios for attribute release and retrieval. All interactions can be instantiated with simple command line statements and a JSON object. A sample library of the calls used in this pilot is available under the Apache 2.0 licence.

Source code issuer/verifier
To set up the issuers and verifiers, we used SimpleSAMLphp, for which purpose a specific Yivi module had previously been developed. The source code and configuration of these components are in SURF's GitHub repository.

Trust framework
Several attributes used in this study, e.g. edubadges, have not yet been standardised in the attribute frameworks used in Yivi. We have therefore created schema files for both eduID and edubadges, developed specifically for these use cases. These are available in a fork containing the Yivi demo schema.

4 Findings

In this section, we explain the main findings. The interim versions of the application were continuously evaluated by the development team throughout the study. At this stage, no testing among users was carried out. Where we refer to 'the user' in the findings, the findings are based on the researchers' assessments. We distinguish between findings related to:

- Design and presentation of the wallet (design findings)
- Findings specific to edubadges
- Findings specific to eduID
- Technical findings during development of the wallet (development findings)
- IRMA-specific findings

4.1 Design: design and presentation of the wallet

The sections below describe the challenges with the initial design of the wallet, as described in Chapter 3.

4.1.1 The wallet login process

The initial design did not consider actual repeated logins to the wallet. Logging into the wallet based on, for example, PIN or biometrics is a requirement. The method of implementation impacts both the user experience and the reliability of the wallet. This will have to be considered.

4.1.2 Presentation of query by verifier

The current design is based on what attributes the verifier requests. This requires explanation of why the verifier requires these specific attributes. This is necessary information for the user to make a decision on whether or not to accept the release of the attributes. It is also important because of the demonstration of purpose limitation. This is necessary to comply with the requirements of the GDPR.

The question is what level of detail attributes should be shown and explained to the user. Too deep a level of detail can lead to ambiguity, for example, in the case of technical data, such as identifiers. These may have little meaning for a user. The ideal presentation of a query requires further investigation.

4.1.3 Presentation of trusted verifiers

The wallet can be used to present trust in a verifier. It should be clear whether a verifier is trusted and known within the ecosystem or is unknown. In the design, a tick mark with the term trusted verifier was used for this purpose. It is currently unclear how such a classification should be established. It is likely that several trust frameworks will coexist, for example, one operated by government and one operated by commercial parties. Further elaboration of this is essential for a user to determine whether or not to share attributes.

4.1.4 Distinguishing the possible status of an attribute

When using a wallet, attributes pass through a number of stages indicated by statuses such as verified, expired, revoked and so on. In the current design, we have not yet detailed these statuses. The presentation should clearly inform the user about the different statuses an attribute can be in.

4.1.5 Accept and Deny

For every transaction involving data, either in or out of the wallet, the user should have the option to reject it. This is obvious at the moment of issuing data from the wallet to a verifier. However, also when accepting data, it is necessary to show what is being sent to the wallet and to explicitly ask the user for permission. After all, the data might not be correct, or the user may have second thoughts and decide not to trust the issuer. In the current wallet design, except in the flow to retrieve edubadges, this has not yet been applied consistently.

4.2 edubadges integration

The use of edubadges is important in the exchange of microcredentials as well as in demonstrating and verifying extracurricular skills or abilities. Testing whether a wallet can add and release edubadges is therefore important. Retrieving edubadges is technically possible, although the capabilities to share edubadges properly are limited. More functionality is needed than is available in the current variant of the wallet. In addition, being able to retrieve edubadges requires metadata about available badges. The solution to these issues does not always lie within SURF's remit.

4.2.1 Release of individual attributes from a single edubadge

In the wallet design, we chose to show individual attributes to the individual. This is technically possible and in line with the desire to allow the user to release only a minimal set of attributes. However, an edubadge consists of several attributes that have meaning in conjunction. Showing separate attributes to the user is not conducive to achieving a meaningful presentation. In addition, these separate attribute fields are not separately shareable, but must be released as a set. A possible solution is to implement *verifiable presentations*[1].

4.2.2 Retrieving edubadges

It is as yet unclear how verifiers can retrieve (a collection of) edubadges. This is because the verifier does not know exactly which types of edubadges exist, nor does it know which specific subjects and skills relate to specific edubadges. This makes it impossible to select edubadges that might meet the selection criteria.

One solution may be to make badge classes searchable, or search by edubadge content. Here, the question is whether that kind of complex logic should be part of the interaction between the

[1] A verifiable presentation of a collection of attributes. See also: https://www.w3.org/TR/vc-data-model/#dfn-verifiable-presentations

wallet and the verifier. It may be necessary to implement this in some other way, with the actual exchange of edubadges only taking place subsequently via the wallet principle.

4.2.3 Selecting of edubadges by the holder

A possible solution may be to support the user selecting options to share specific edubadges. This wallet functionality is highly desirable. However, it would require a more complex interface for selection of edubadges than provided for in the current design of the wallet. A subject-based preselection by the verifier could support this but is not currently possible (see 4.2.2). Follow-up research should further elaborate how badge selection by the holder can be made possible, both technically and in the presentation.

4.3 eduID integration

4.3.1 eduID as a basic identity

The premise of using eduID as a basic identity for further interactions in education and research settings seems perfectly viable in combination with a wallet. During the PoC, we successfully created a Yivi schema and used it to exchange attributes between eduID and the wallet. We also added attributes to the wallet that enrich the person's profile, for instance the affiliation as issued by the institution.

An important point to note here, however, is that eduID (via SURFconext) currently uses pseudonymised identifiers per affiliated service. This is done to prevent the linking of users between services. Services that want to serve their users both in the traditional federated way in a browser and via a wallet may not be able to recognise the user as a result.

4.3.2 Wallet initialisation

One of the design choices was that the user should always populate the wallet with a basic eduID identity. It seemed easiest to handle this on initialisation of the wallet (see also the Activate wallet flows in Appendix 1).

We mapped out the possible initialisation flows. As the user has several possible starting points (eduID website, eduID app, wallet app), the number of possible routes was very large and there are many scenarios where the user could hit a dead end. In the end, none of the proposed routes proved feasible, as they would all lead to mixing attributes with different trust frameworks (see 4.3.2).

For now, therefore, initialisation is limited to a flow that follows a 'regular' Yivi issuer flow.

4.3.3 Loading credentials via the wallet app

Particularly when first used, it is common to initialise the mobile app, for example by having the app communicate with a REST API shielded via OAuth. In the context of SSI, however, it is important to carefully consider what the source of authority is for the data being loaded. After all, that source must be recognisable as such within the trust framework of the wallet and by all verifiers. For this reason, it does not yet appear possible to load data in any other way than via a

regular Yivi issuer. A similar scenario may also come into play if, for example, we want to load data originating from a passport scanning app into the wallet.

It is probably technically possible through clever use of deep linking to largely hide the necessary complexity from the user. However, that would require adapting various provisioning flows in the eduID app and beyond the scope of this pilot.

4.4 Development

4.4.1 Complexity of user flow

The use of wallets soon leads to complex data exchange chains and a potentially complex flow for end users. This problem occurs when the wallet is initially created and when the attribute issue flow is complex. For instance, when additional attributes are needed during a query by a verifier, especially when the additional attributes need to be retrieved from multiple issuers. If retrieving an additional attribute takes a long time, this can lead to a timeout in the issue flow.

Some flows require users to use other apps on their mobile devices. For example, logging in to an issuer using the eduID or DigiD app. This flow can be confusing for users.

It takes (a lot of) cooperation and coordination, both technical and semantic, between stakeholders to make a chain work fully and correctly for users. The wallet concept was intended to reduce the dependency of parties on each other and the dependency of users on the parties. The high degree of decoupling in the various parts of the chain leads to flows with many exceptions and possible unfortunate flows. A lot of expertise is needed to achieve integration and a good and consistent user experience.

4.4.2 Development environment

Given the complexity, it is very important for successful testing and development of applications in this ecosystem to remove as many chain dependencies as possible by deploying (mock) interfaces. Yivi CLI was used for this purpose in our pilot project. This allows developers in different parts of the chain to work simultaneously and without interdependencies.

4.4.3 Trust framework

Yivi uses a metadata-based trust framework in which SIDN acts as trusted third party. This concept is very similar to SURFconext, Entree or eduGAIN federation arrangements. We therefore know that this model is essentially suitable for use with a (very) large number of participants. However, to scale to the level needed for national, pan-European or perhaps even global deployment, considerable changes are needed, both in the technology and in the management of the Yivi metadata. One possible solution is to introduce a delegation model where another party takes on this role for certain sectors. Within the Netherlands, SURF and Kennisnet would be the obvious choice for education and research. While in the past, these two federations have operated separately, it may be valuable to see whether a joint approach might make sense in this new ecosystem.

4.4.4 Categorisation

Users benefit if information is categorised in several places in the app. Examples include a presentation of all 'name'-related attributes in the wallet (regardless of issuer) or, for example, an indication that a particular verifier is 'trusted'. Given the dynamic nature of a wallet ecosystem, it is not practical to capture this kind of cataloguing in the app's code. Instead, list retrieval or a similar mechanism is the obvious choice. However, a registry would have an impact on the scalability and possibly also the reliability of the wallet.

4.4.5 Credential binding

When using a wallet, it is important for verifiers to be certain that the credentials available in the wallet actually belong to the owner of the wallet. In the case of edubadges, for example, the verifier needs to be able to establish that the edubadge was actually issued to a particular person. To do this, the verifier can rely on the wallet or its policies. However, it seems better to validate certain attributes in the edubadges directly against attributes that form part of the identity available in the wallet, e.g. via eduID. A verifier can then query both sets of attributes and compare them. The consequence, however, is that this does lead to a certain degree of traceability, as it requires the use of persistent identifiers across multiple source systems. On the other hand, issued badges could also end up in the wallet where these are not linked to an eduID (e.g. internationally issued badges). Another solution will have to be found for this.

4.4.6 Handling attributes

Dealing with and presenting attributes is complex. Choices in presentation have advantages and disadvantages. Below, we explain two findings and one challenge.

First, a recurring concept in other wallet applications is that of 'cards' containing various attributes - at least visually. How the specific attributes are displayed to a user is not obvious. As it stands, the data the user is shown, if they want to share it, may still be technical in nature. For example, the actual eduID identifier is an opaque set of characters. This identifier is visible in only a few places and could potentially confuse a user. The user may decide not to release this 'strange' attribute to the verifier because they do not understand its purpose.

Second, the verifier may not query all the data provided in conjunction. This can lead to inconsistent data or use of data without context. An example of this is an edubadge, where attributes within the badge may lose meaning when shared separately by the user. It is important to further investigate what level of granularity in displayed data is desirable.

Finally, complex datasets present a challenge. A user may want to share some data with a verifier, but the data does not lend itself easily to sharing via a wallet. For instance, audiovisual material created to support a CV, or a complex set of attributes that give a researcher access to research infrastructure, such as the GA4GH Passport. At the same time, the user does want to retain control over the access the verifier has to this data, in a similar way as can be done with wallet attributes. Several solutions to this problem are conceivable, such as combining a personal data vault for storage with a wallet for controlling access to the data in the vault. This should be investigated further.

4.5 Yivi-specific findings

4.5.1 Central key server

Yivi uses a central key server that stores half of the key needed to decrypt data from a user's wallet. This information is retrieved at wallet startup. The consequence is that this makes registration of the Yivi wallet with this key server mandatory. In addition, the key server poses a risk because users cannot use their wallet if the server is not available or reachable (single point of failure). If SURF itself were to run a wallet based on the Yivi model, we would have to make a decision on where to run this infrastructure.

4.5.2 Use of labels by the verifier

In the Yivi ecosystem, the verifier can add labels to the attributes they query. The user sees these in the wallet. However, there is no guarantee that these labels actually relate to the attributes an issuer has placed in the wallet. This could lead to confusion, but in the worst case also to abuse, as it is possible to request certain attributes from the user under false pretences. Labels may also be used in so-called disjunctions, where the verifier may ask a user to combine data from different source systems. However, use of labels is not mandatory in disjunctions, and this can lead to a confusing user experience.

4.5.3 SDK and open standards

The aim of this technical exploration was to test a wallet for use in an education and research setting, and also to assess the desire to use a recognisable SURF style in the design of the wallet. This turned out not to be directly possible with Yivi.

The current Yivi app was clearly not built as a software development kit (SDK). This makes combining Yivi in or with another app extremely difficult. This is evident not least in how the Yivi event model has been implemented. Furthermore, it has also been found that the back-end application (in GO) contains technical debt. As a result, actually combining the (already existing) eduID app with a Yivi wallet proved impossible within the timeframe of the pilot.
Yivi does not currently implement open standards for communication between the issuer, verifier and wallet. As a result, Yivi is currently a silo. In any case, Yivi is working on implementing the stated interoperability requirements in line with eIDAS 2.0, as described in the ARF.

5 Conclusion and next steps

Throughout the design and development of the prototype, we encountered many design and development issues and findings. What are the main conclusions we can draw? Not all questions and issues we encountered came within the scope of this PoC; we therefore also defined follow-up questions and directions for further research.

5.1 Conclusions

Complexity
Many parties are involved in a wallet ecosystem. The independence of those parties is, in theory, the strength of SSI. However, this independence also introduces complexity into development and testing throughout the chain.

Fine-grained attributes versus complex data
While it is desirable for a user to have fine-grained control over the release of attributes, there are several scenarios where a complex data object (e.g. an edubadge) needs to be provided to the verifier as a whole. It is also conceivable to simplify certain complex data objects to a yes/no question. For example, consider the question whether someone is a student. The user can answer this with a simple yes or no instead of sharing their affiliation with a specific institution.

A wallet ecosystem should be able to accommodate multiple scenarios. Here, clarity for the individual should be ensured: releasing attributes should be simple, but again not too simple, as the user would risk sharing too many attributes and/or that they are not usable by the verifier.

Linking the identity to attributes
It is very important to be able to prove that the data in the wallet belongs to the user. Several things are needed to establish this reliably, including connecting these attributes to a trusted identity. This is sometimes needed at the time of issue to a verifier and when retrieving attributes from an issuer. An example is the inclusion of the eduID identifier in an edubadge. In certain cases, the identity must be included as part of the data request. This increases the risk of linkability.

Dependence on central components
The SSI ecosystem aims to reduce dependency on central components. Various functional requirements, such as being able to group attributes under a common denominator (presenting all 'name' attributes in the wallet) or being able to show trustworthiness of verifiers by presenting the which trust frames they participate in promote ease of use, but probably require central components.

Attributes outside the trust frame and multiple trust frames
Combining attributes from different trust frames is particularly difficult, as there is currently no way to express trust between different ecosystems. The only practical route at present is to use proxies that can form a bridge between the two trust frames (e.g. SURFconext and Yivi).

Trust (metadata) distribution
The trust frame used in Yivi is based on a central infrastructure. This is not necessarily a problem, but it does come with scalability concerns.

5.2 Follow-up questions

5.2.1 User experience

Testing of the wallet application has shown that complex interaction patterns emerge. The data that users see during the data sharing process is still technical in nature. The level of intricacy in data presentation and its human interpretability is not an easy choice.

The aim of the designs in this PoC is that they should be user-friendly and understandable. However, this study did not have scope to conduct any user testing. Whether the designs and application meet our needs in this area is therefore still an open question.

It is advisable to have users test whether the interactions are feasible and understandable in follow-up research. Above all, whether users actually experience control over the data and understand what they have done.

5.2.2 Control over data

In the pilot, we developed a number of differing scenarios. For some scenarios, such as sharing an edubadge, it is obvious that the user has direct control over the issue, without the institution still being involved: analogous to the current use of a paper diploma. For other data, such as identity, for example, this is less clear. Suppose, for example, that the use of identity leads to costs at the institution because a licence is used. In that case, who is responsible for what? Where is the tension between control of data by the individual and the institution's legal duty? To what extent does the institution have the right and ability to decide where certain data is used? To what extent can or may the student be responsible for data exchange? What are the implications of this? Clearly, this raises questions at the functional and legal levels.

5.2.3 Technical aspects

During the pilot, we encountered several technical challenges that may impact on existing SURF services or their further development if we want to better connect them to a wallet ecosystem.

eduID integration
One of the intended goals of the pilot was an integration of the wallet with eduID and the eduID app. Loading eduID attributes into a wallet based on SURFconext and a Yivi issuer turned out to be relatively easy. In the design, however, we had described a flow where the loading of these credentials could already take place during initialisation of the wallet. However, it proved difficult to use the eduID screens and APIs to get this done. Since there is both a web-based and a mobile instance, this creates a very complex set of flows, with many possible edge cases. In addition, this way of filling the wallet creates a conflict with regard to mixing data from different authoritative sources.

eduID and SURFconext API security
The modifications in eduID and the extra complexity required in the wallet app for the desired integration were not within the scope of this pilot. If we aim for an integrated app where existing eduID app functionality and a wallet come together, we need to explore this further. As eduID uses SURFconext API security, this should also be considered. Currently, eduID provides a REST

API for querying the service. It would be a good idea to investigate whether the OpenId4VCI standard offers the capability to exchange data via SURFconext API security in a standardised way, in which case we would directly pre-empt one of the standards designated in EUID. This would also allow us to provide a basic facility around authentication and secure access on a large scale via SURFconext for various educational services, such as edubadges, for example.

Identifying users
Currently, eduID (and SURFconext, for that matter) follows the strategy of giving each service its own pseudonymous identifier to protect user privacy. As the wallet for eduID is a stand-alone service, it is issued with a pseudonymised identifier. The consequence of this is that the identity in the wallet has a different identifier than the identity linked to the eduID badges. This makes it difficult for a verifier to reliably determine whether an edubadge has been issued to the relevant person. This limits the possibilities for sharing badges via a wallet.

Lifetime of the attributes and revocation
Although we had provided functionality for removing and revoking attributes in a number of places in the PoC, this was not elaborated on in the technology. The reason being that it was not clear how this should actually be done and what its ultimate impact should be on previously issued credentials and underlying systems.

It is clear that both revocation and the lifetime of attributes in a wallet ecosystem can have a significant impact on the user experience as well as the reliability and security of the system. As this is likely to involve combining very conflicting interests, it would be wise to investigate this further.

5.2.4 Applicability

This PoC shows that existing solutions in the market can serve as a basis for the further development of an eduWallet. This does, however, present several challenges: not only in terms of technology, but also in terms of governance. The pilot shows that much is possible with a wallet ecosystem. The question is for which scenarios there really is a business case. It will only make sense to explore how wallets will change data exchange within our sector when there is more clarity on this.

Emerging public solutions, including the EDI wallet of the Ministry of the Interior and EU reference wallets, represent another challenge. These are currently early in the development phase and not yet mature for use within our sector. However, they are important and compatibility with these wallets is desirable.

How to proceed?
In conclusion, a wallet ecosystem will only be workable and scalable if a very high degree of technical and semantic standardisation is achieved. For the time being, we will have to live with having to provide credentials to a wide variety of wallets. SURF contributes to this through research and development in the field of SSI and wallets in the education sector in several ways:

SURF participates in the EU large-scale pilot DC4EU. This large-scale pilot contributes to the implementation of eIDAS2.0. At national level, SURF is experimenting in the Npuls programme by developing an 'eduWallet'. In both projects, SURF can build on the findings made in this PoC.

In the longer term, SURF will support institutions by providing integration with these environments when they are closer to production.

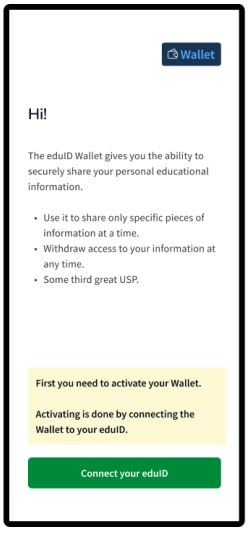
In the short term, SURF can carry out further experiments and pilots together with institutions based on specific use cases. Specific follow-up research within the SSI Lab of the Trust & Identity team includes taking stock of what the intended architecture will be when SURFconext is deployed as part of the service for issuing and verifying Verifiable Credentials based on the OpenID4VCI standard. We want to do this on the basis of realistic use cases in our sector, with the explicit involvement of the stakeholders of these use cases.

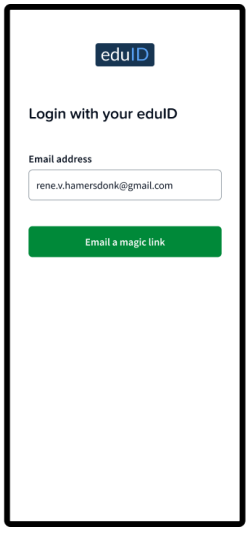
In the projects cited — but also beyond — we strongly invite institutions that want to participate in initiatives on this theme to come forward and share the knowledge gained.

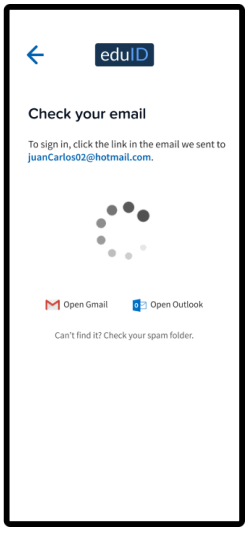
Appendix 1. Functional designs

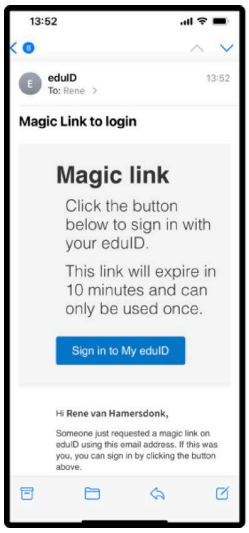
Wallet activation

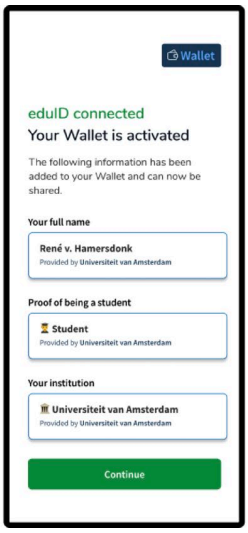
Wallets are activated based on an eduID. When a person downloads the wallet for the first time, they must log in in with their eduID. If they do not yet have an eduID, it must first be created. If they are already known to eduID or studying at an institution, this data can be retrieved while logging in to the wallet with an eduID. For opportunities and challenges around integration with eduID, see section 4.3 ‘eduID integration’.

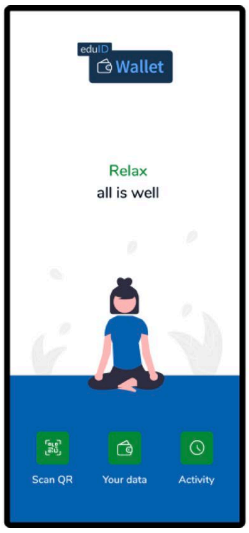


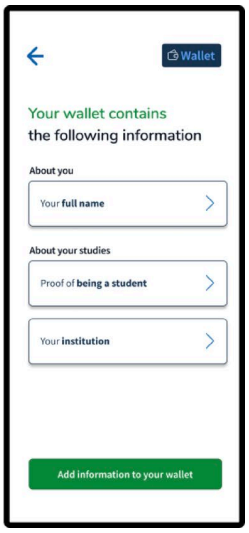






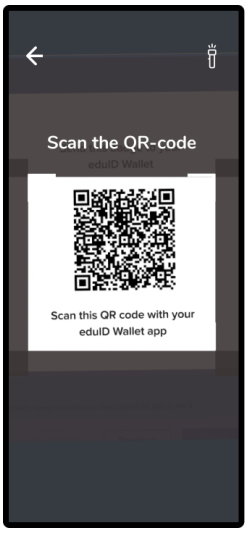


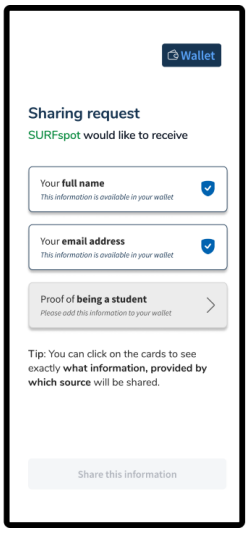


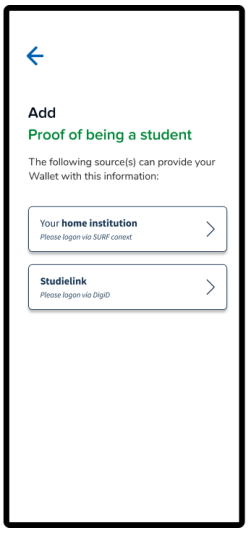


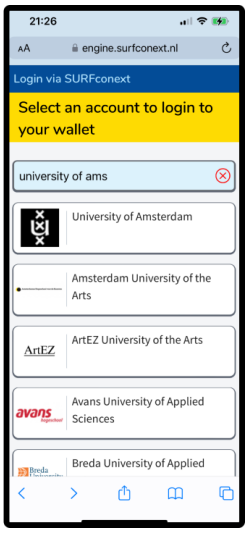
Adding attributes

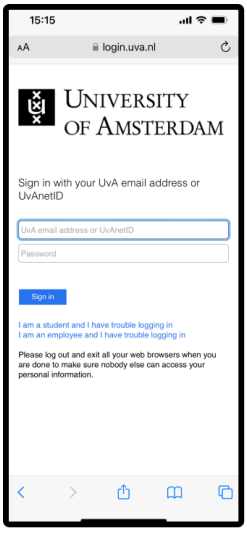
Adding attributes will generally start when the person takes action to use a service. In the example below, the user wants to order books from SURFspot at a discount. To do this, SURFspot wants to know whether the person is a student and presents a QR code to retrieve information from the wallet: the wallet does not yet have proof that this person is a student. The person retrieves this attribute via Studielink or an institution. The person can then share the required data with SURFspot.

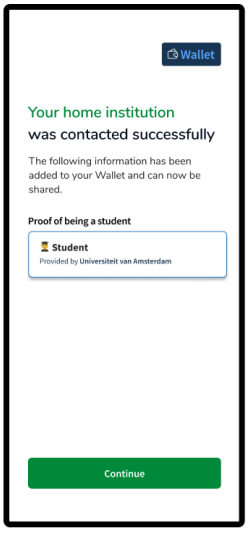


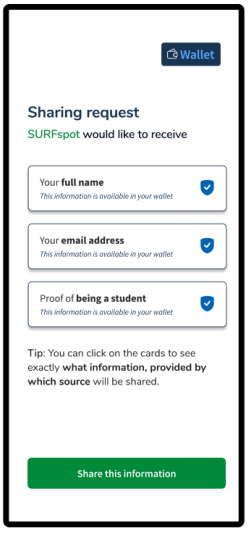


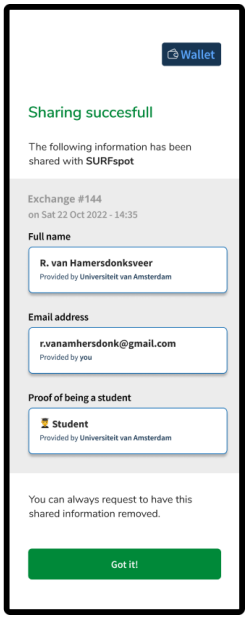






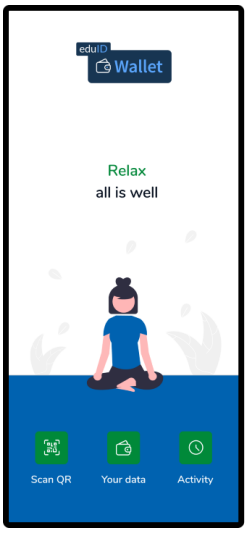


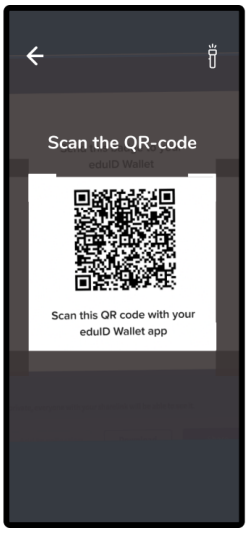


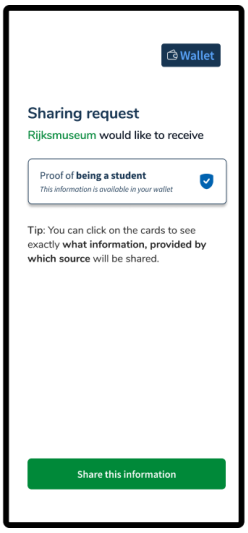


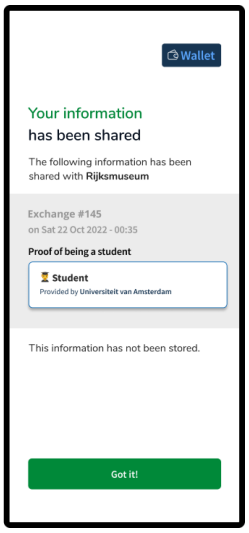
Sharing attributes

Once the wallet is activated and the data needed to use a service is known, the flow becomes easy. For example, in the case below, the person wants to buy a discounted ticket to the Rijksmuseum. They can scan the QR code displayed on the counter. At that point, the person can share the required attributes.



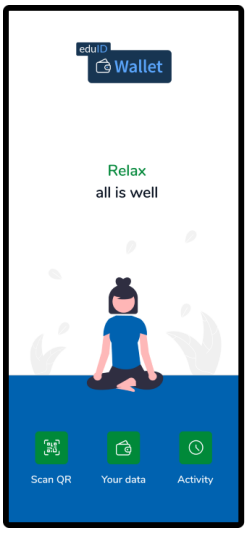


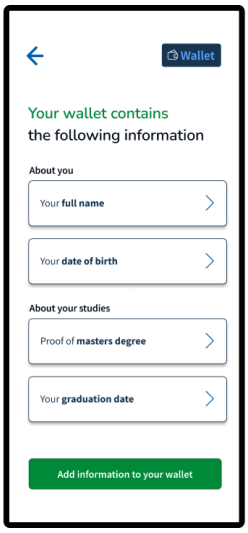


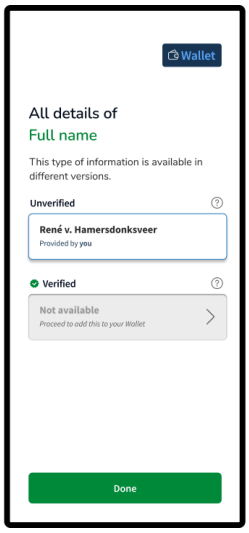


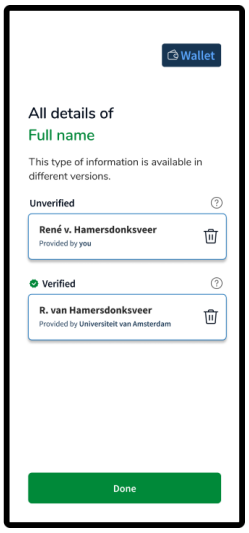
Viewing and deleting content

Having insight into data held in the wallet and being able to delete it are necessary basic functions. The user can view this via 'Your data'. If the person wants to, they can also add attributes here themselves.



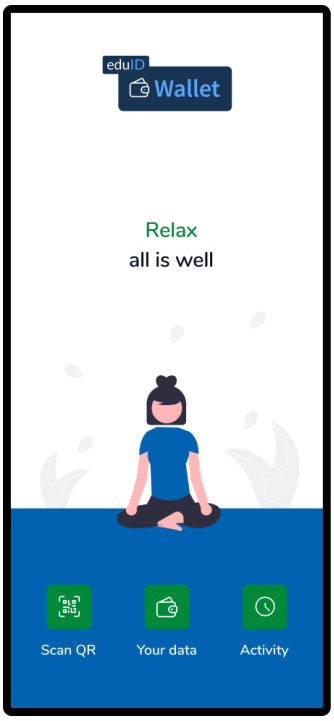


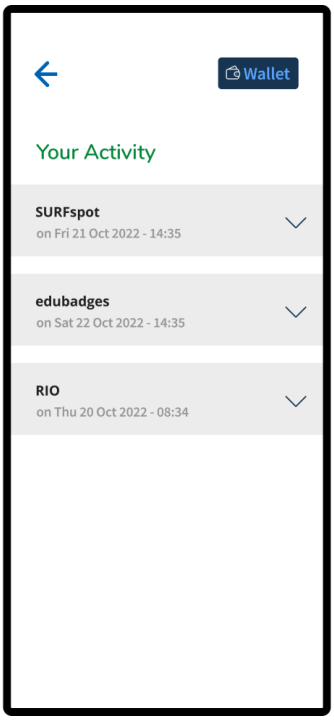


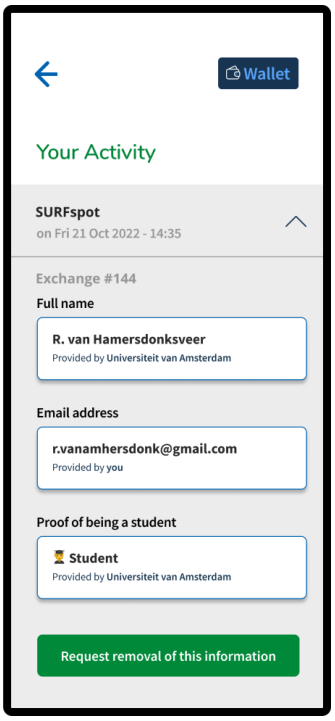


Viewing activity

Under 'Activity' is a summary of all transactions that have taken place. Ideally, the user can also initiate a request to remove information from the verifier here. To what extent this is applicable depends on the type of view request that has taken place.



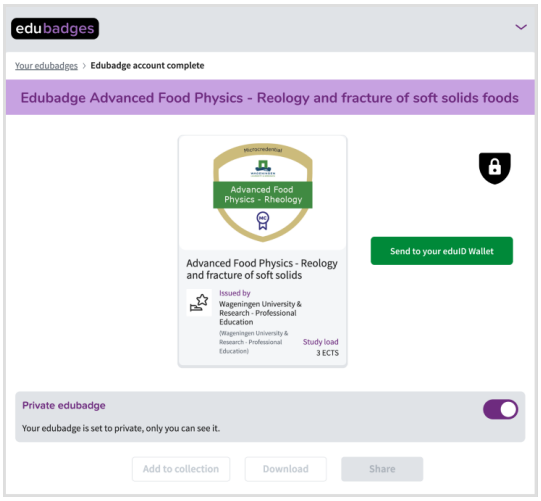


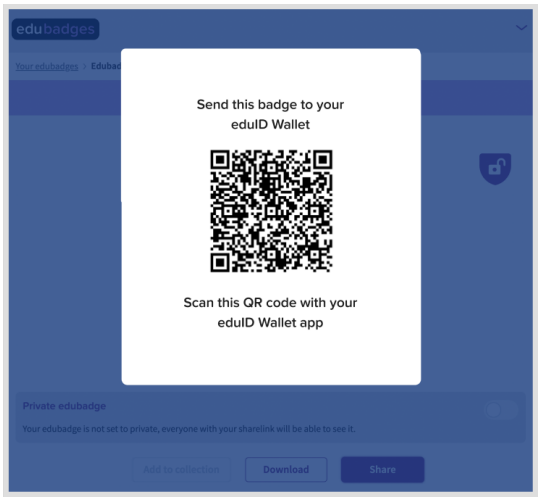


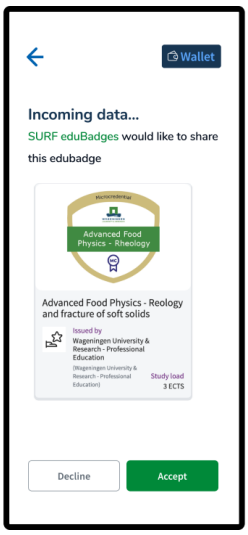
Retrieving edubadges

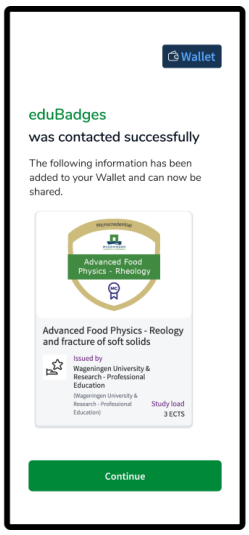
Adding edubadges is an activity that the person is most likely to perform of his or her own accord. For example, in preparation for a job application, further training or after obtaining a certificate.

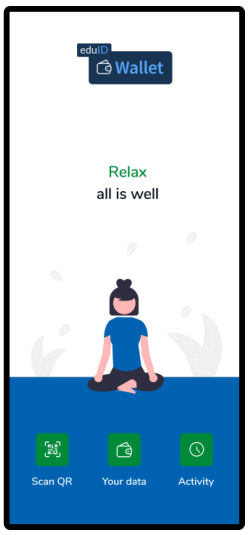
On the edubadges website, the person can scan a QR code to retrieve the information using the wallet. They can then retrieve the edubadges obtained with the wallet and add them to the wallet so the badges can be shared with a (potential) employer or institution. Challenges surrounding edubadge retrieval and queries are described in section 4.2 ‘edubadges integration’.











Authors
Niels van Dijk, technical product manager
Marlies Rikken, product manager

This report is a publication of SURF
May 2023