

# Designing an educational identity VC

Niels van Dijk  
([niels.vandijk@surf.nl](mailto:niels.vandijk@surf.nl))

TIIME 2026 unConference



<https://kidspot.co.nz/activities/walking-on-eggs-science-experiment/>

## TL;DR

An education identity verifiable credential:

- MUST obey to minimal disclosure, by design\*
- MUST NOT contain identifiers in its claim values\*\*
- MUST only support selective disclosure capable credential formats
- SHOULD include claim lifetime considerations
- Leverages capabilities of Digital Credential Query Language (DCQL)
- Provides assurance information by default
- Has schema consistent over credential types and protocols

\* **ANY** verifiable credential should!

\*\* YMMV and I mean VC payload here (as all things in a VC are technically “claims”)

## Why an educational identity credential?

We need a wallet representation of the identities we use in institutions and federations to link to other edu credentials (diploma, badges, dislexia, exam permission, room access, etc, etc) as these live in our *existing* back-end systems

Why not use a governmental credential (PID)?

- has no edu context
- will require High Assurance
- subject to governmental governance
- users may not want to use gov id all the time
- Gov wallets may not be ready to support our use cases any time soon

# REFEDs Personalized Access Entity Category

## 5.1 Required Attributes

The *entity category attribute bundle* consists (abstractly) of the following data elements:

- *organization*
- *user identifier*
- *person name*
- *email address*
- *affiliation*
- *assurance*

These abstract elements are bound to protocol-specific definitions in the following subsection(s) and additional bindings may be added in the future.

<https://refeds.org/category/personalized>

## But with a twist

### REFEDs Personalized

organization

user identifier

person name

email address

affiliation

assurance

### Educational credential

schac\_home\_organization

*This line is intentionally left blank*

name

given\_name

family\_name

email

eduperson\_scoped\_affiliation

is\_student

is\_faculty

is\_member

is\_staff

is\_alum

is\_affiliate

is\_employee

is\_library-walk-in

eduperson\_assurance

## VC claim lifetime

Wallets (currently) do not have a means to dynamically update credentials

The only way to get new credentials into the wallet is revisiting the issuer (and/or revocation)

This implies:

- Claims with similar lifetime may go together in 1 VC
- Claims which may be updated more frequently need to be separate (entitlements)
- Claims which are not governed by the same issuer need to be separatable (entitlements, external identifiers – ORCID, MyAcademiaID, voPerson)
- Make sure there is an understandable reason for the user to update the VC

## Entitlements et.al.

Any additional information (entitlements, other identifiers like ORCID, MyAcademicID etc can be added on top of this, and requested together by using *Digital Credential Query Language*

DCQL is a JSON-encoded language used within the OpenID4VP specification to enable verifiers to request specific, complex verifiable presentations from a holder's digital wallet. It allows for precise, privacy-preserving queries, such as requesting multiple credential types or specific attributes, supporting W3C VCs, SD-JWTs, and ISO 18013-5 mDLs

edu entitlement

Format: dc+sd-jwt

vct\_values

Add Reset

Used in `meta.vct_values` for `dc+sd-jwt`.

☐ multiple

☒ require\_cryptographic\_holder\_binding

Select all Clear

**entitlement**  
entitlement

Filter...



## **Selective disclosure**

DCQL and SD-JWT can also facilitate use of Selective Disclosure to allow subsets of VCs to be requested.

Side note: Selective Disclosure is hyped as one of the best features of the wallet ecosystem

Do note however the implementation of Selective Disclosure and SD-JWT format does not automatically warrant arbitrary user choice wrt release of claims

## Mid term recap: mvVC + DCQL

Splitting the credentials so we get 'minimum viable credentials' and combining these upon request using DCQL will:

- Improve credential re-usability
- Preserve privacy and data protection
- Properly distribute the responsibilities wrt personal data
- Allow for fine grained governance on the use of credentials (as we can also use DCQL to express per verifier claim usage permission in trust frameworks :)

## So, where is my identifier?

Any identifier we put in a VC claim

- will be baked in 'forever', hence lose most (all?) privacy preserving features
- may be issued to arbitrary verifiers, and decided upon by end users
- eppn
- ~~eduperson\_unique\_id~~
- ~~Any pairwise or transient id from OIDC/SAML~~
- ~~email, well eh thats a bit of a challenge~~

# Holder key

A cryptographic key pair (private/public) controlled by the user's digital wallet, used to prove ownership of a Verifiable Credential (VC) and bind that credential to the holder

During subsequent presentations (using OpenID4VP), the wallet uses this private key to sign the Verifiable Presentation (VP), allowing the verifier to check the signature against the public key embedded in the credential.

Pro:

- wallets can implement persistent, pairwise holder keys for VP
- which mimics identifier behavior we know from SAML and OIDC

Setbacks:

- It is *bound to the wallet instance* → But as we will see later, recovery is a requirement anyway
- Wallet behavior wrt holder keys is inconsistent → So it needs to be explicitly described in a profile

## Linking Holder key to Existing (Federated) Identity?

All of our existing data is already linked to some identifier

How do I correlate the wallet identity to this if it has no (known) identifiers?

**Ask for all possible data from the wallet**

Yeah right!

## Reconciliation on the fly – “Connect your wallet”

When receiving an unknown holder key, the verifier will use some other means to reconcile the identity, e.g. federated login

By default provides recovery mechanism

Supports multi wallet scenarios

Could be implemented as passwordless authN

## Reconciliation on the fly – “Ask a friend”

When receiving an unknown holder key, the verifier will *ask the issuer* for an identifier they know about, which matches the holder key

By default provides recovery mechanism

Supports multi wallet scenarios

While this seemingly breaks the prime directive that the issuer should never learn about what the user is doing, it might be a “lesser evil” as compared to identifiers ending up everywhere, or verifiers overasking.



## Assurance

With RAF 2.0\* we have a well defined framework, but do all of the statements there still hold in a VC & Wallet context?

\* <https://refeds.org/assurance>

## Schema

The semantic meaning of our SAML attributes & OIDC claims & VC claims *must* be the same for interop and usability.

However, that was never a requirement when VCs were designed

Many schema definition requirements for VC credential types contain mandatory “technical” elements and structure, which make it almost impossible to just use 1 schema across all VC types and protocols

## TL;DR

An education identity verifiable credential:

- MUST obey to minimal disclosure, by design\*
- MUST NOT contain identifiers in its claim values\*\*
- MUST only support selective disclosure capable credential formats
- SHOULD include claim lifetime considerations
- Leverages capabilities of Digital Credential Query Language (DCQL)
- Provides assurance information by default
- Has schema consistent over credential types and protocols

\* **ANY** verifiable credential should!

\*\* YMMV and I mean VC payload here (as all things in a VC are technically “claims”)