SOLID
Overview – Specifications – Use cases

Fabien Petitcolas, Kristof Verslype – April 2023

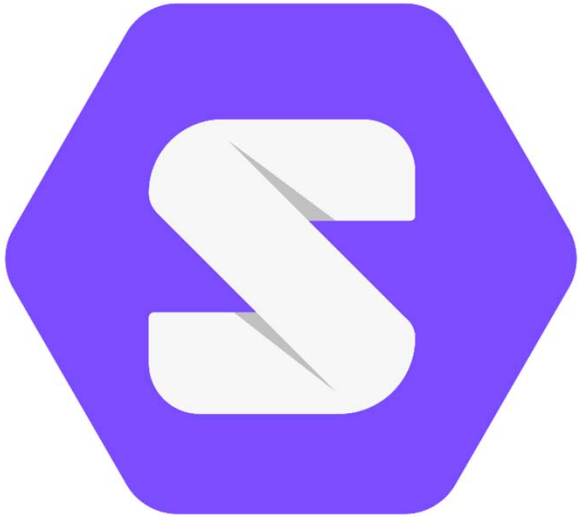
Smals
ICT for society

Content

- Concluding remarks on SOLID
- Background information:
  - About the SOLID project
  - Key SOLID draft specifications
  - Main data sharing patterns
  - Data sharing examples
    - with SOLID only (experiment at Smals Research)
    - with SOLID and verifiable credentials (Flanders PoC)

Concluding remarks on SOLID

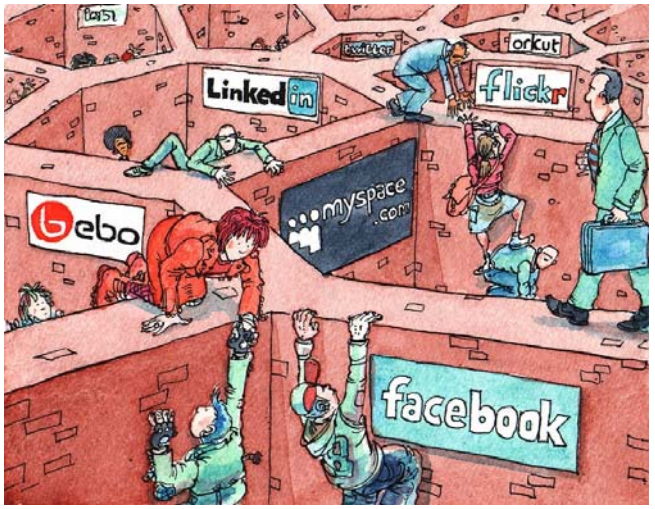
- Compelling vision for
  - Handling personal data
  - Giving back control to users/citizens
- Early stage project
  - Active area of research
  - Many different experimental open-source libraries and tools
  - One claimed “enterprise ready” offering
  - Enterprise offering tested with only a small number of customers
  - No evidence of existing applications in production
- Technically SOLID can support different data exchange patterns
  - Including a “sluis” model, but some GDPR-based legal arguments have been made against
- SOLID builds on top of the OpenID Connect standard for authentication
  - Compatibility with several identification mechanisms



About the SOLID project

SOLID's context and motivation

- **So**cial **Li**nked **D**ata
- Context for SOLID
  - Data is centralised in handful of Web platforms with far reaching privacy consequences
  - Data locked in siloes, not available for innovation
  - Web's universality is threatened
- SOLID is an ecosystem to "take back control"
- SOLID enables interoperable data storages through standardisation



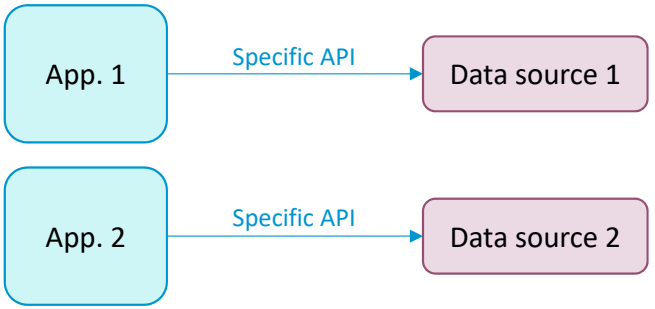
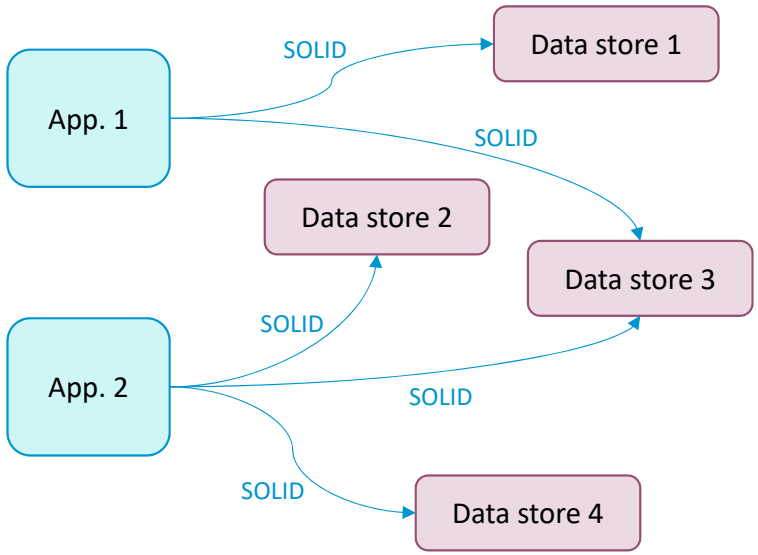


Image: Social Networking Sites as Walled Gardens by David Simonds, The Economist, 19 March 2008

Smals
ICT for society

SOLID

- Vision:
  Separate data from application to create independent choice



- Combination of existing W3C standards to define how agents should interact
- Strong use of the Resource Description Framework (RDF) to capture semantic information
- SOLID protocol specification covers:
  - Data storage servers
  - Applications or services
  - Identity
- A SOLID server is a regular Web server, with support for access control, that hosts one or more data stores
- A user can have several data stores on several servers

Smals
ICT for society

Current state of the SOLID specifications

- SOLID is set of draft specifications combining mature technologies

Technical Reports

Work Item	Repository	Current Stage	Expected Completion
Solid Protocol	https://github.com/solid/specification	Version 0.10.0	TBD
Solid WebID Profile	https://github.com/solid/webid-profile	Version 1.0.0, Editor's Draft	2023-06-30
Solid OIDC	https://github.com/solid/solid-oidc	Version 0.1.0	TBD
HTTPSig Authentication	https://github.com/solid/authentication-panel	Editor's Draft	TBD
Web Access Control	https://github.com/solid/web-access-control-spec/	Version 1.0.0-cr.1	2023-06-30
Access Control Policy	https://github.com/solid/authorization-panel	Editor's Draft	TBD
Solid Application Interoperability	https://github.com/solid/data-interoperability-panel	Editor's Draft	TBD
Shape Trees	https://github.com/shapetrees/specification	Editor's Draft	TBD
Solid DID Method	https://github.com/solid/did-method-solid	Unofficial Draft	TBD
Solid Notifications Protocol	https://github.com/solid/notifications	Version 0.2.0	2023-06-30

Source: https://solidproject.org/TR/

Solid project code base

- Relatively small project
- Several open-source projects for SOLID servers
  - CommunitySolidServer
    - https://github.com/CommunitySolidServer/CommunitySolidServer
    - Language: TypeScript
    - About 17 contributors with >1000 lines of code (esp. from U of Ghent)
  - node-solid-server
    - https://github.com/nodeSolidServer/node-solid-server
    - Language: JavaScript
    - Last update over a year ago
  - solid-nextcloud
    - https://github.com/pdsinterop/solid-nextcloud
    - Language: PHP
    - 4 contributors with > 1000 lines of code
  - php-solid-server
    - https://github.com/pdsinterop/php-solid-server
    - 2 contributors with > 1000 lines of code, last update about a year go
    - Language: PHP

Companies providing SOLID-based products

- We found only two companies providing SOLID products:

inrupt

- Boston-based company
- About 50 employees mostly in USA and UK
- Founded by Tim Berners-Lee
- https://www.inrupt.com/

Digita

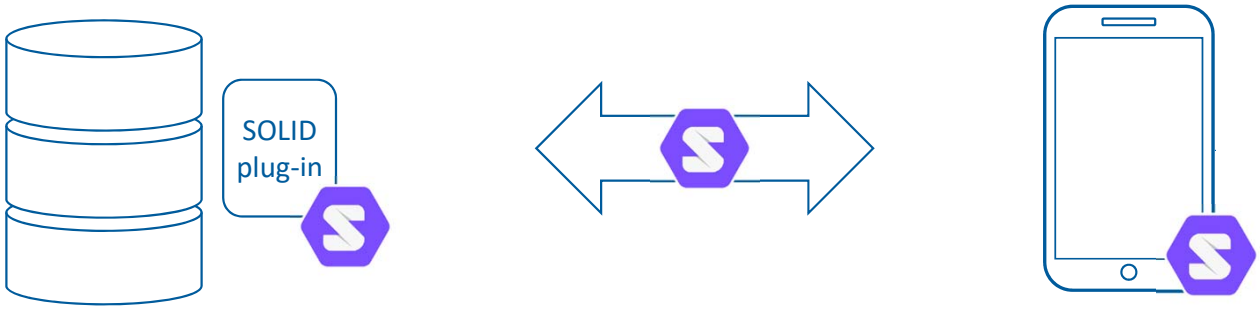
- Brussels-based start-up
- Founded by Wouter Janssens and Lauro Vanderborght
- https://www.digita.ai/

Smals
ICT for society

Key specifications

Personal online data-stores

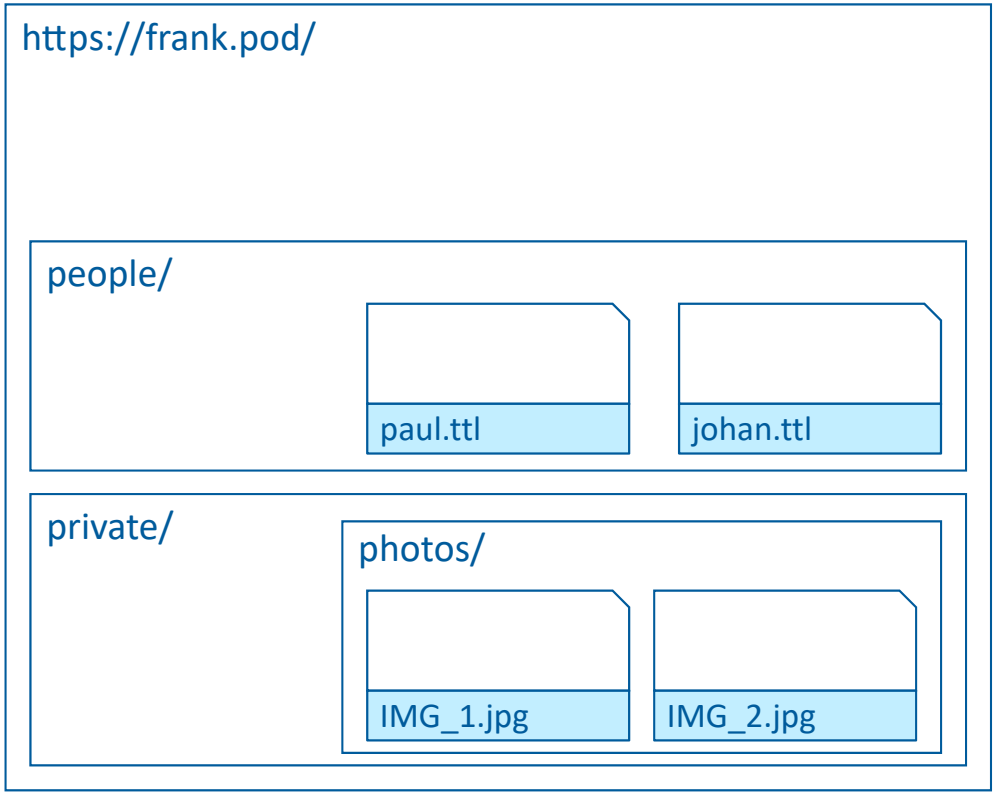
- Data containers are called “POD” in the SOLID context
  - Store data as web resources (e.g., identified by URI) in a hierarchical manner
    - Container could be a folder in file system
    - Resource could be a file
  - Store structured, semi-structured or unstructured data
  - End-user applications provide a *view* on the data
- Communication is governed by the “SOLID” protocol



Smals
ICT for society

POD example

- `https://frank.pod/` *(container)*
  - `people/` *(container)*
    - `paul.ttl` *(RDF document)*
    - `johan.ttl` *(RDF document)*
    - …
  - `private/` *(container)*
    - `photos/` *(container)*
      - `2023-01/` *(container)*
        - `IMG_1.jpg` *(non-RDF document)*
        - `IMG_2.jpg` *(non-RDF document)*
        - …
  - …

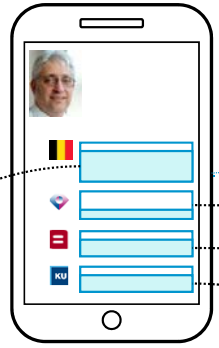


Source: Ruben Dedecker, et al. " What's in a Pod?, https://github.com/SolidLabResearch/WhatsInAPod

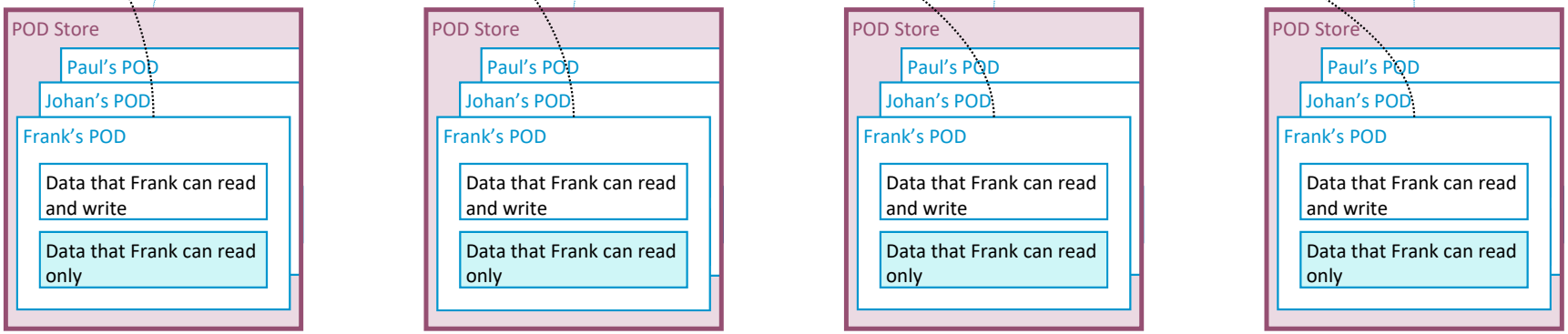
POD and data browsers

Frank's virtual data vault

The virtual data vault does not store data. It is an application that gathers information from different POD and combines them in a single interface. This allows the user to see all data about themselves in one place.



Directory of POD stores

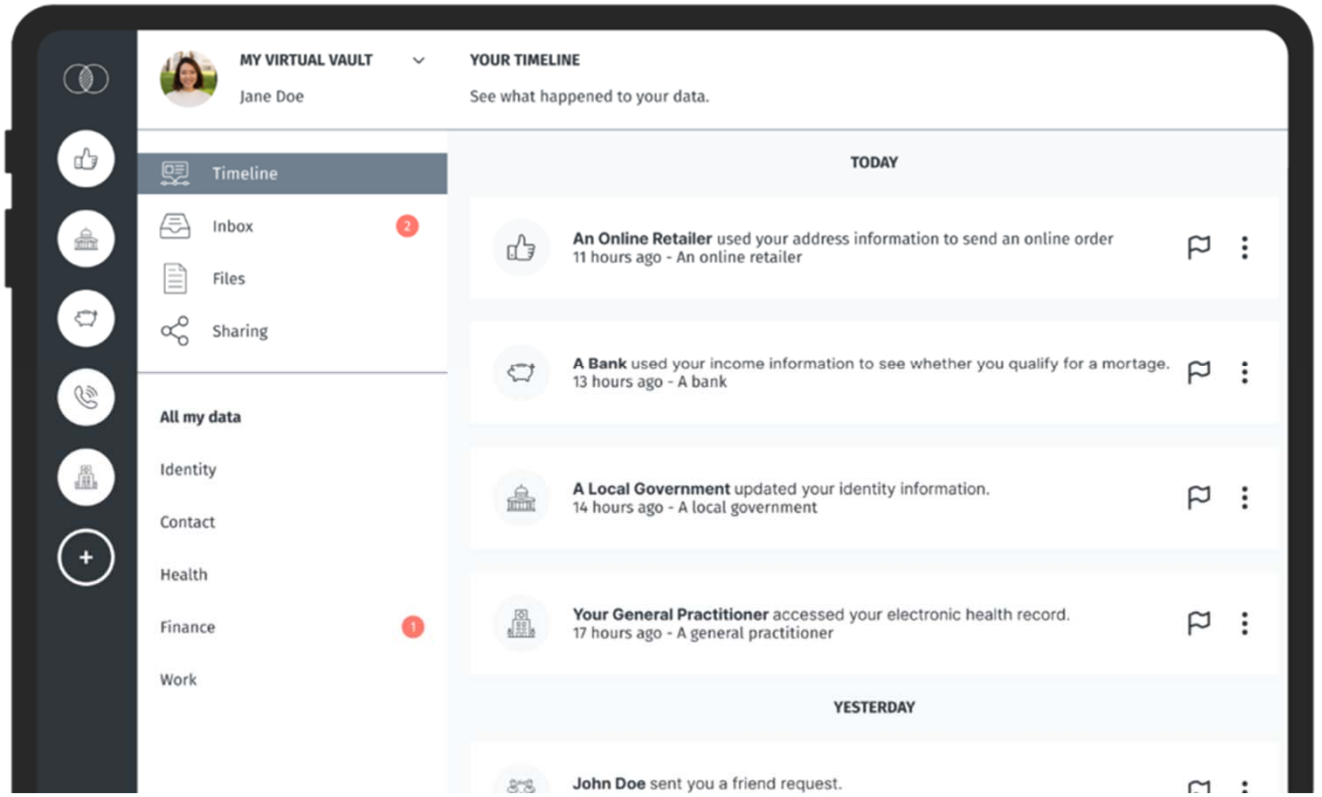


Ref: https://www.youtube.com/watch?v=hbqSyU6cqFQ

Smals ICT for society

Example of data browser

- Data browser in development by Digita.ai:



Smals
ICT for society

List of key SOLID related specifications

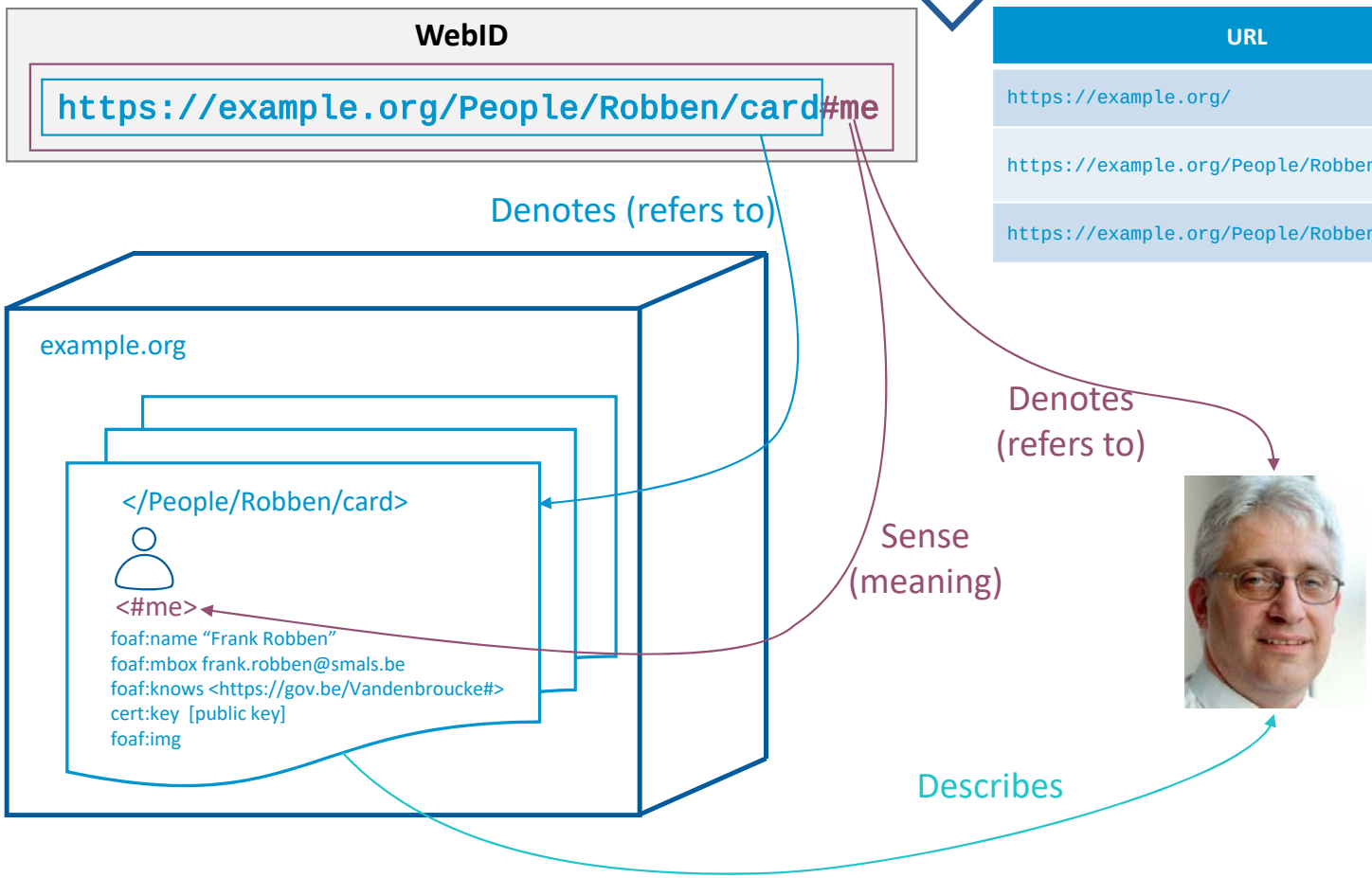
SOLID constrains HTTP with authentication and authorisation:

- Authentication:
  - **WebID**: URL that identifies an agent (person, organisation, app, device, etc.)
    - Dereferenced WebID: document describing various details of the agent
  - **Solid-OIDC**: OpenID-Connect based protocol to authenticate agent with given WebID
- Authorisation:
  - **Web Access Control (WAC)**: access control list (ACL) mechanism
  - **Access Control Policy (ACP)**: policy-mechanisms to facilitate use of WAC

Solid protocol requirements

- Support HTTP/1.1 protocol for client server interoperability based on existing:
  - RFC7230 “Message Syntax and Routing”
  - RFC7231 “Semantics and content”
  - RFC7235 “Authentication”
- SOLID server to provide storage:
  - Has storage for hierarchical “containers”
  - Container + metadata (type, modified, size)
- Auxiliary resources:
  - Web Access Control
  - Description resource
- SOLID actions (via HTTP/1.1) on resources:
  - Create: `PUT` or `PATCH`
  - Read: `GET`, `HEAD`, `OPTIONS`
  - Write: `PUT`, `POST`, `PATCH`
  - Modify: `PATCH`
  - Delete: `DELETE`
- Identity:
  - Use WebID as identifier for persons, organisations or software
- Authentication:
  - SOLID-OIDC standard (based on OpenID)

WebID example



URL	Meaning
https://example.org/	Identity provider
https://example.org/People/Robben/card	URI refers to document describing person
https://example.org/People/Robben/card#me	URI is identifier denoting the person

Source: https://www.w3.org/2005/Incubator/webid/spec/

Smals
ICT for society

Web access control (WAC)

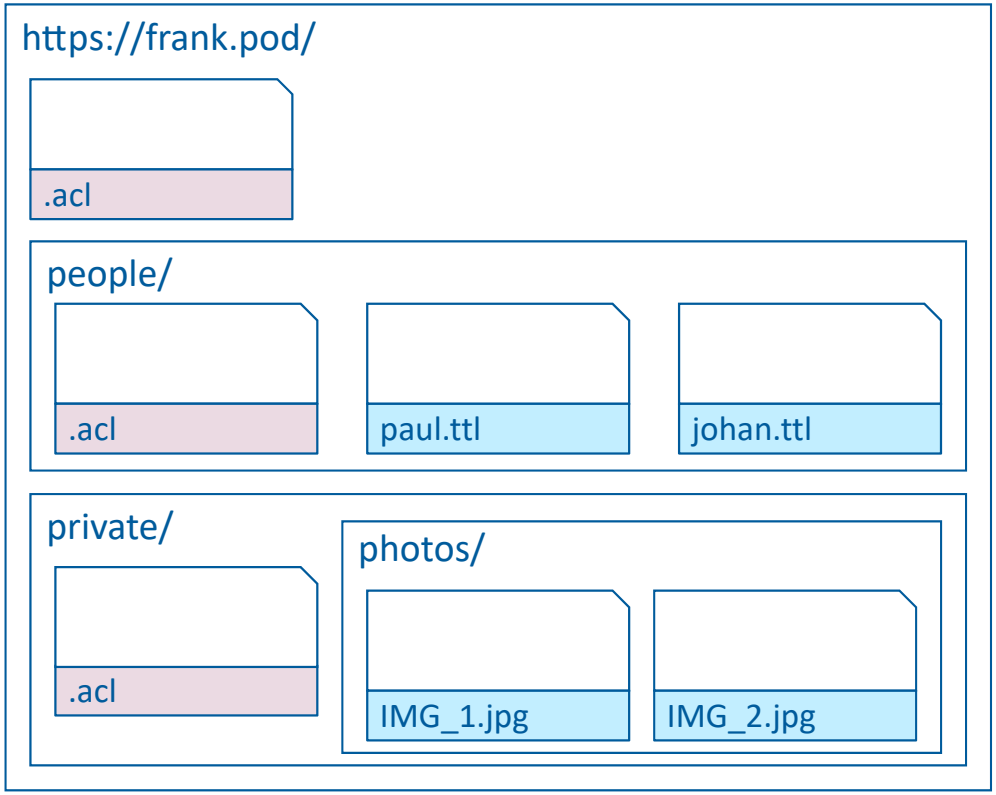
- Can enable applications to discover authorisations associated with a given resource
- Similar to Linux file access control, but with WebIDs
- `.acl` file per resource
  - Advertised in “link rel” response
  - Link: `<http://example.org/Frank/.acl>; rel="acl"`
  - If payslip is a file (resource), only the full payslip can be shared
- Authorisation
  - `acl:accessTo`
  - `acl:mode`
  - `acl:agent / acl:agentGroup / acl:agentClass`

- Access modes:
  - Read
  - Write
  - Append
  - Control
- Agent classes:
  - foaf:Agent
  - acl:AuthenticatedAgent

Smals
ICT for society

POD and access control example

- `https://frank.pod/` *(container)*
  - `.acl` *(RDF document for access control)*
  - `people/` *(container)*
    - `.acl` *(RDF document for access control)*
    - `paul.ttl` *(RDF document)*
    - `johan.ttl` *(RDF document)*
    - …
  - `private/` *(container)*
    - `.acl` *(RDF document for access control)*
    - `photos/` *(container)*
      - `2023-01/` *(container)*
        - `IMG_1.jpg` *(non-RDF document)*
        - `IMG_2.jpg` *(non-RDF document)*
        - …
  - …



Source: Ruben Dedecker, et al. " What's in a Pod?, https://github.com/SolidLabResearch/WhatsInAPod

Smals ICT for society

Solid-OIDC specification

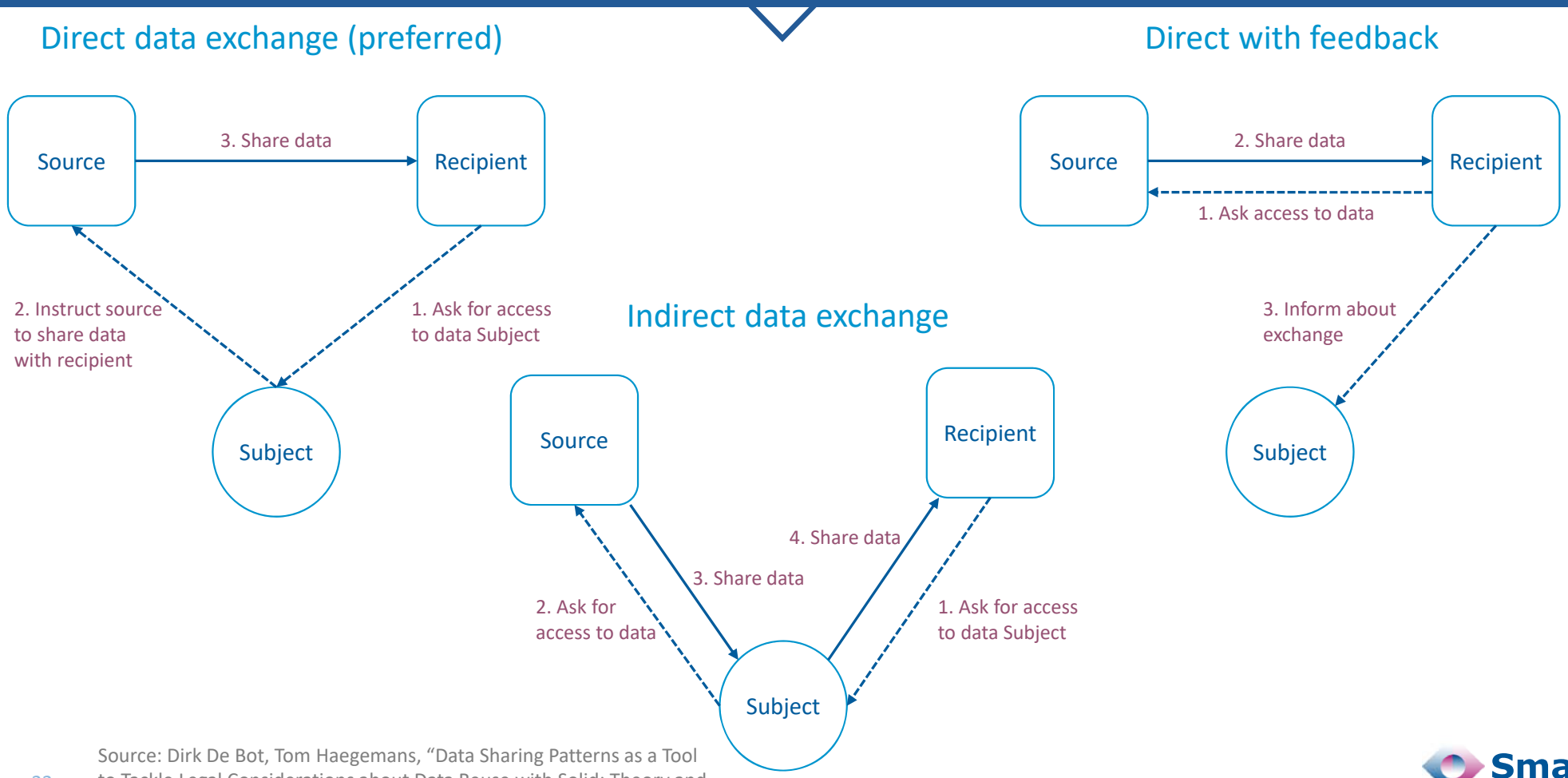
- Solid-OIDC defines how resource server verifies:
  - Identity of the relying party
  - Identity of end users
- Solid-OIDC uses authentication mechanism of an OpenID provider
- Solid-OIDC extends OpenID Connect with:
  - Resource and authorisation server that have no existing trust relationship with identity provider
  - Ephemeral clients
- WebID profile lists the OpenID Providers that can issue tokens on behalf of the agent who controls the WebID
- Access token for use within SOLID only

Source: https://solid.github.io/solid-oidc/

Sharing of personal data

Main patterns

Personal data exchange patterns compatible with SOLID



Source: Dirk De Bot, Tom Haegemans, "Data Sharing Patterns as a Tool to Tackle Legal Considerations about Data Reuse with Solid: Theory and Applications in Europe", http://go.digita.ai/reuse-patterns

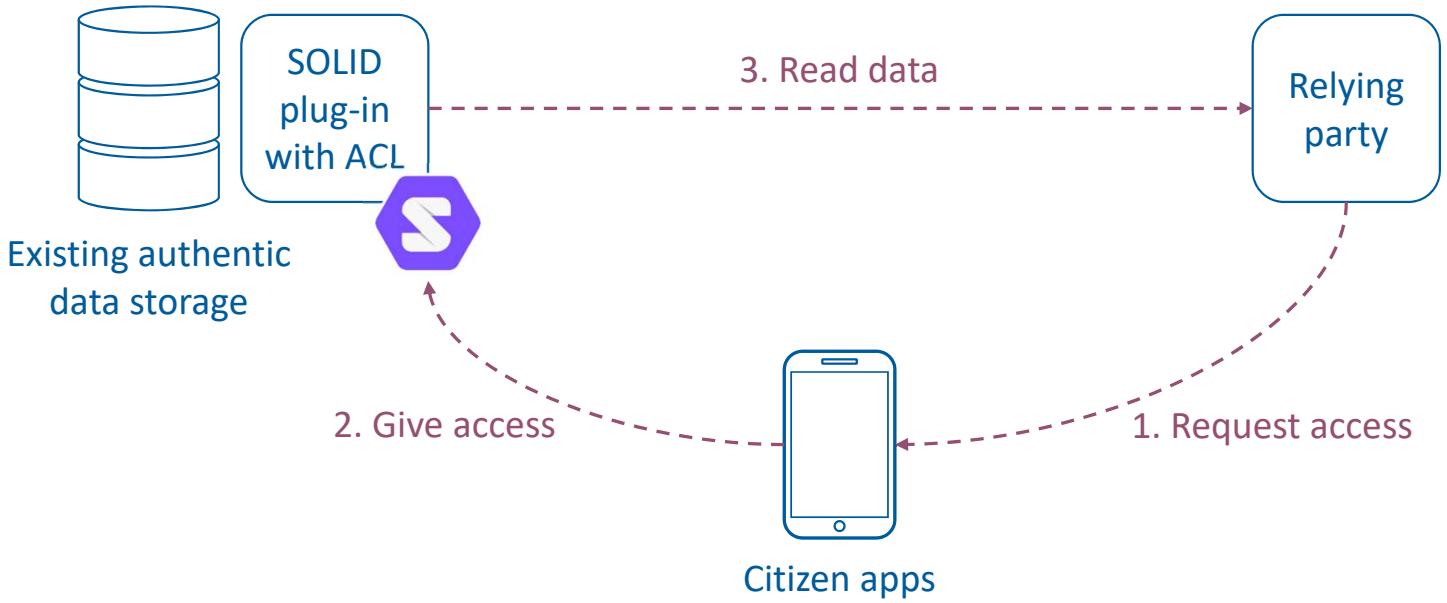
Smals ICT for society

Authentic sources

Direct data sharing with SOLID

Sharing authentic data

- Original data source adds a SOLID compatible layer
- Data subject can add viewers to the data but cannot modify the data



Smals
ICT for society

Sharing authentic data with SOLID only (1/3)

- Experiment done at Smals Research
- Use of CommunitySolidServer
  - https://github.com/CommunitySolidServer/CommunitySolidServer
  - 4 SOLID servers: http://solid1.smalsrech.be/, http://solid2.smalsrech.be/ etc.
- Created 3 agents authenticated on 3 different SOLID servers
  - First agent (e.g., KULeuven)
    - Has full control on a file (e.g., a diploma) on its SOLID server
    - Gives read access to a second agent
  - Second agent (e.g., Frank)
    - Can add other agents to the list of agents who can view the file
  - Third agent (e.g., Smals)
    - Can view file on first agent's data store

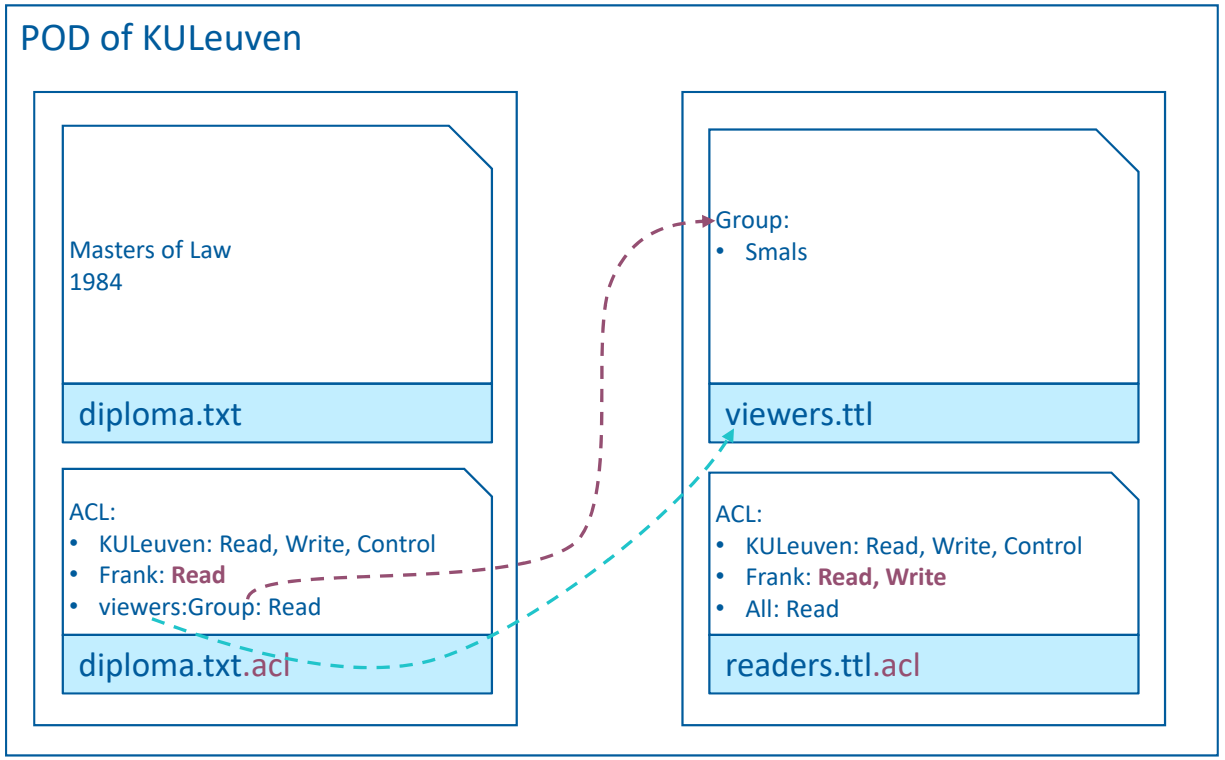
Sharing authentic data with SOLID only (2/3)

Resource content:

Resource name:

ACL resource content:

ACL resource name:



Limitation of *current* implementation of CommunitySolidServer: list of allowed readers has to be public.

Smals
ICT for society

Sharing authentic data with SOLID only (3/3)

- Experiment shows that it is technically possible to use solid as a “Sluis”:
  - A SOLID layer can be added to an existing server containing authentic data
  - A citizen can be granted permission to give read access to other agents
- According to discussion with Raf Buyle (Information Architect, Digitaal Vlaanderen)
  - Above-mentioned solution was the initial intention of Flanders
  - However a legal argument based on GDPR was made
  - This prompted Flanders to look at an alternative implementation using verifiable credential (see next slides)

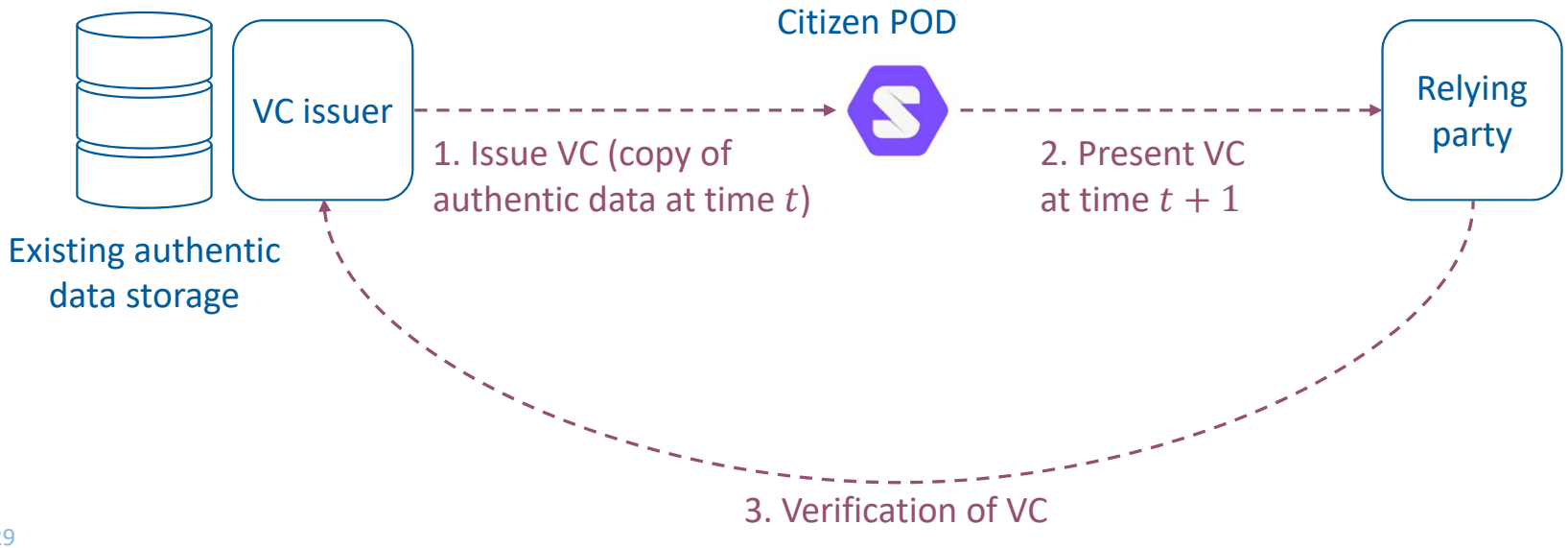
Authentic sources

Indirect data sharing with SOLID and verifiable credentials

PoC from Flanders

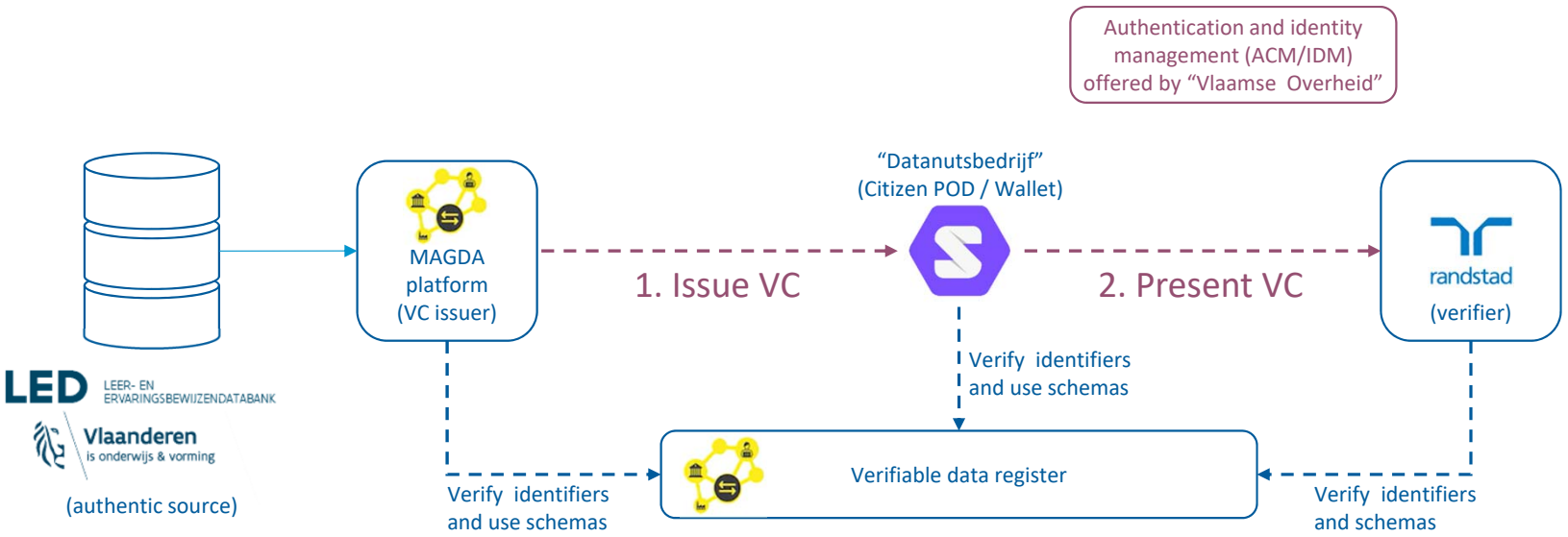
Sharing authentic data with verifiable credentials

- Original data source delivers verifiable credential (VC) to citizen
- Citizen keeps the VC in their SOLID POD or Digital Wallet
- Requires additional infrastructure for VC verification (e.g., revocation)



Smals ICT for society

Proof of concept in Flanders



Remarks

- POD offered by "*datanutsbedrijf*"
- Access via eIDAS authentication level "*substantial*"
- WebID managed by ACM/IDM
- ICON Project: "*Selective disclosure*" & "*ZK proofs*"
- Private parties could provide data to POD in the form of VC

MAGDA – Gegevensdelingsplatform:
https://overheid.vlaanderen.be/informatie-vlaanderen/producten-diensten/gegevensdelingsplatform-magda
ACM/IDM: Toegangs- (ACM) en Gebruikersbeheer (IDM) van de Vlaamse overheid
https://www.vlaanderen.be/acm-idm-standaard-aansluitingsproces
Vlaams Datanutsbedrijf
https://www.vlaanderen.be/digitaal-vlaanderen/het-vlaams-datanutsbedrijf

Source: https://www.youtube.com/watch?v=jZIzLmnBB98 (16:25)

Smals ICT for society

PoC Flanders – Diplomas

Om samen veilig data te delen...

Moeten we herkomst van die data kunnen valideren.

e.g. Mag een onderwijsinstelling een bepaald diploma ondertekenen?

Moeten
kunnen

g. Wat als ik straks zelf mijn looninformatie
npas?

Moeten

kunnen verzekeren.

Moeten we intrekking na aanpassing van de bron mogelijk maken.

Bijvoorbeeld wanneer data in authentieke bron gewijzigd (vb. intrekking diploma)

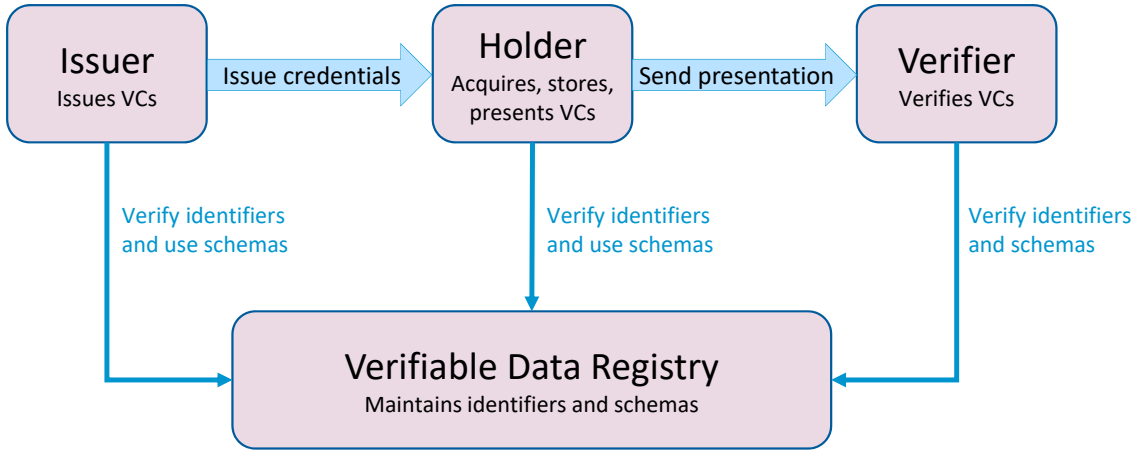
⇒ **Inconsistente data buiten authentieke bron = ongeldige data**

Source: https://www.youtube.com/watch?v=jZIzLmnBB98 (16:25)

Smals
ICT for society

Verifiable credentials

- Open standard for digital credentials
- Can represent:
  - Information found in physical credentials (e.g., passport, driving license)
  - Things with no physical equivalent (e.g., ownership of a bank account)



Source: https://www.w3.org/TR/vc-data-model/

Verifiable credential example

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1",
    "https://w3id.org/security/suites/ed25519-2020/v1"
  ],
  "id": "http://example.edu/credentials/3732",
  "type": [
    "VerifiableCredential",
    "UniversityDegreeCredential"
  ],
  "issuer": "https://example.edu/issuers/14",
  "issuanceDate": "2010-01-01T19:23:24Z",
  "credentialSubject": {
    "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
    "degree": {
      "type": "BachelorDegree",
      "name": "Bachelor of Science and Arts"
    }
  },
  "credentialStatus": {
    "id": "https://example.edu/status/24",
    "type": "CredentialStatusList2017"
  },
 "proof": {
    "type": "Ed25519Signature2020",
    "created": "2022-02-25T14:58:43Z",
    "verificationMethod": "https://example.edu/issuers/14#key-1",
    "proofPurpose": "assertionMethod",
    "proofValue": "z3BXsFfx1qJ5NsTkKqREjQ3AGh6RAmCwvgu1HcDSzK3P
                   5QEg2TAw8ufktJBw8QkAQRciMGyBf5T2AHyRg2w13Uvhp"
  }
}
```


Smals
ICT for society