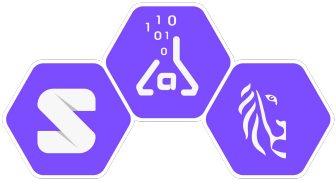
Shapes, Forms & Footprints

Towards Web generation of RDF data without coding

Patrick Hochstenbach (UGent)
SWIB 2022



Thanks to

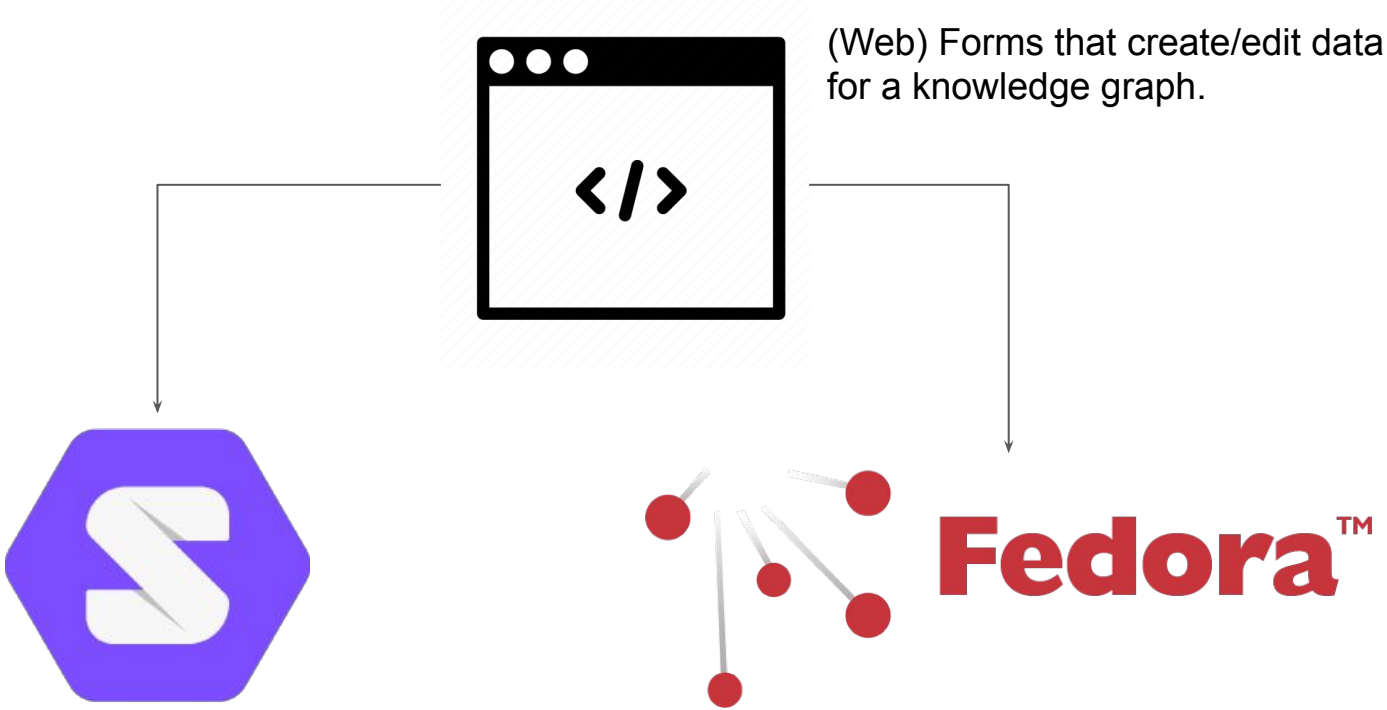
It all started at the start of COVID-19 crisis

- Library IT had to quickly invent teleworking tasks for staff working at home
- Crowd sourcing of metadata production
  - Card card catalog
  - Images
  - Digitized materials (with personal information)
- Could we make something quick to generate metadata?
- Couldn't use cloud services because of GDPR reasons
- First app took a weekend to create, the second weeks, months with many variations
- Could this been done easier? Google Forms but in our environment?
- Google Forms but with structured data (not Excel like output)?

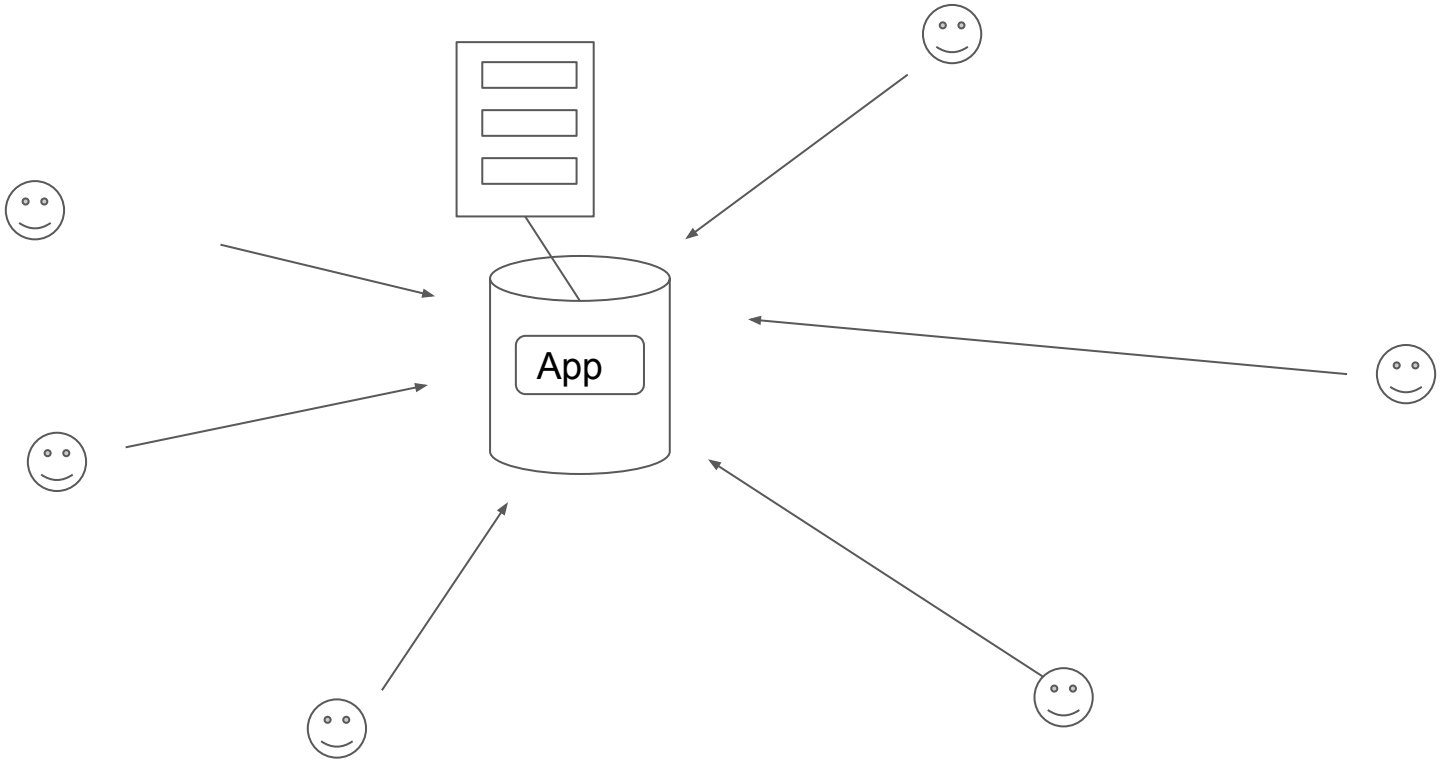
Mellon Research Pod project

- https://knows.idlab.ugent.be/projects/mellon/
- Scholarly Communication in a Decentralized Web
- Very decentralized production of (RDF) data
- Every researcher has her own Researcher Pod (mini institutional repository)
- … on which she stores her publications
- … and described these publications with RDF data
- How to produce RDF data in a very decentralized environment?
- … with many local variations in metadata requirements

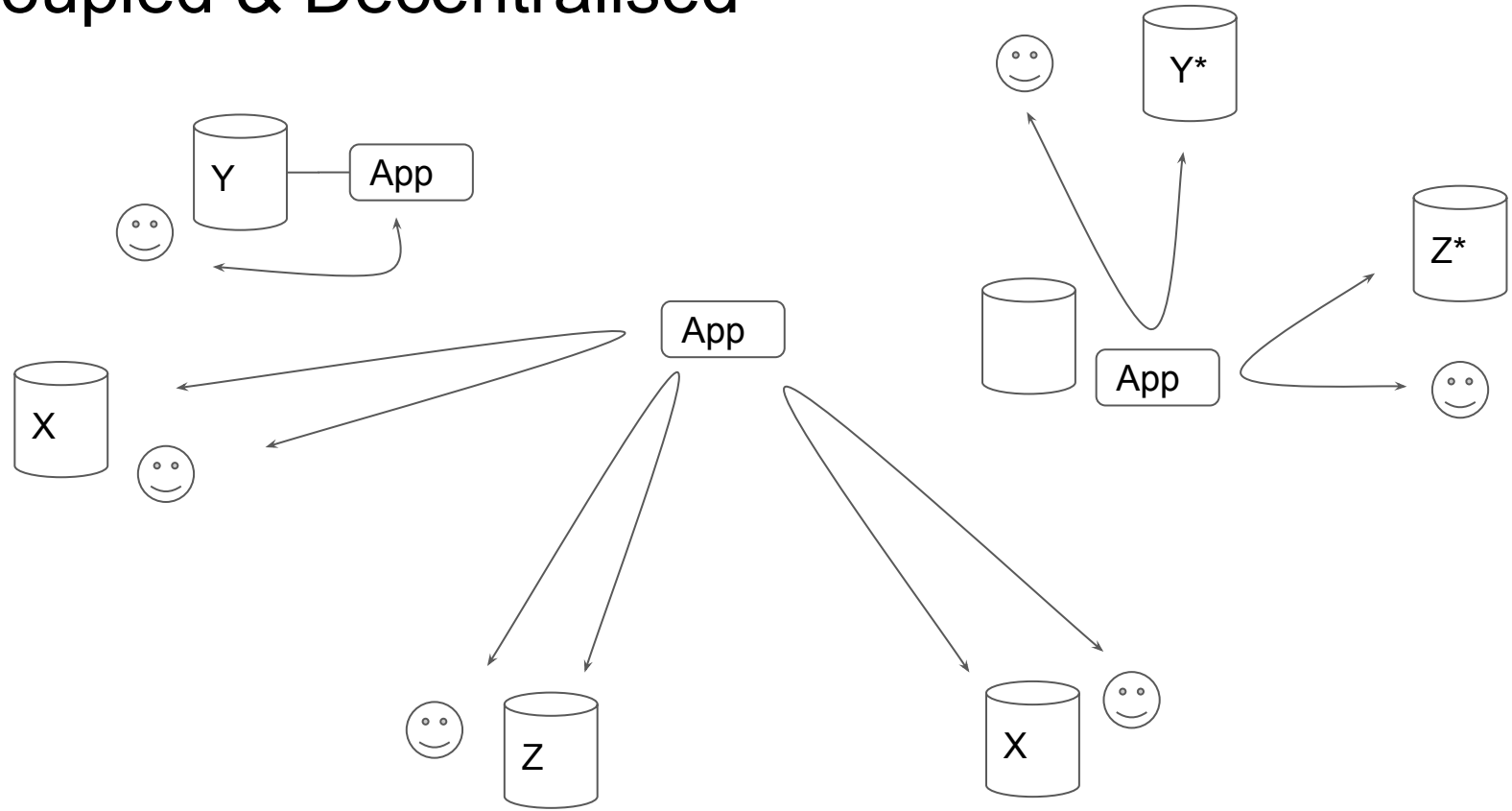
How to produce RDF data?



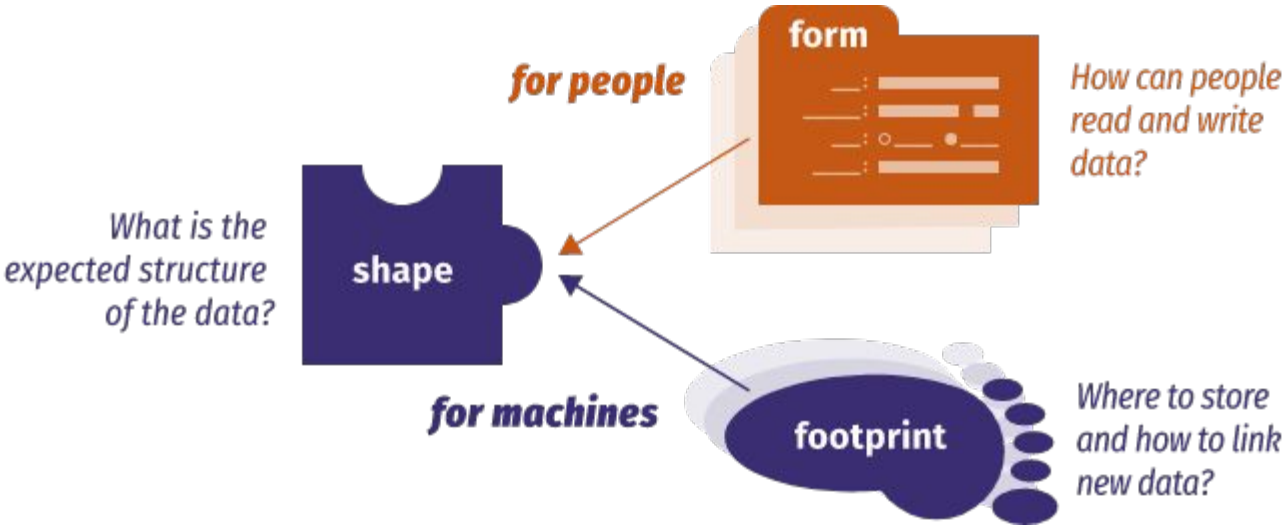
~~De~~coupled & ~~De~~centralised



Decoupled & Decentralised

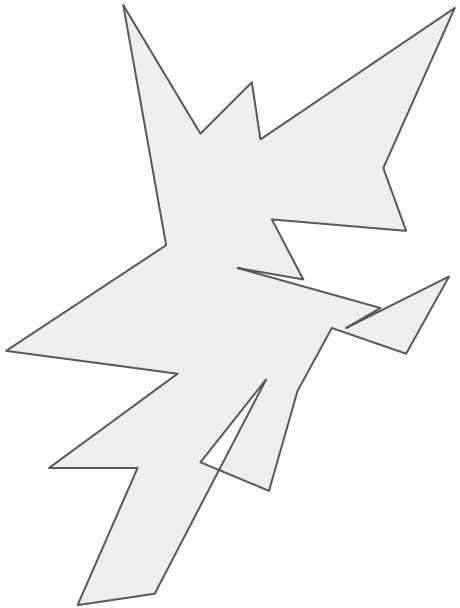


What components are needed for such apps?

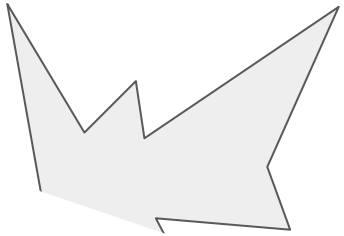


https://ruben.verborgh.org/blog/2019/06/17/shaping-linked-data-apps/

Shapes & Forms



Book shape with all
its complexities



Book form you want
users to fill in

Specifications + technologies

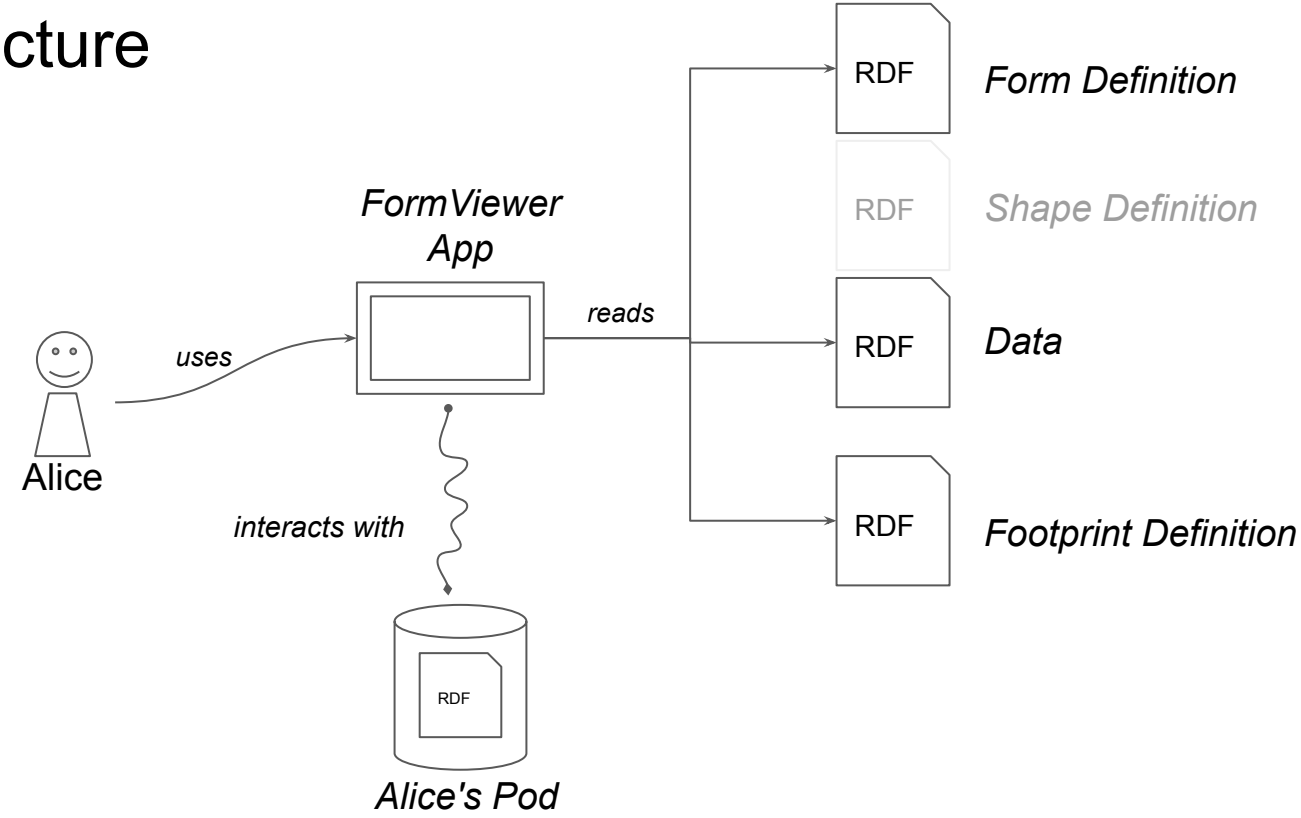
Shapes	**Forms**	**Footprints**
RDF • SHACL • ShEx XML • XForms	RDF (with coupling to validation) • SHACL-Form-React • Shaperone • SHACL Forms • ShExC forms • CLEF (without coupling to validation) • Solid-UI Forms • http://rdf-form.danielbeeke.nl • https://rdforms.org/	RDF • Hydra Core ? • Shape Trees ? • N3Logic ?

Towards a pragmatic solution

Focus:

- Decentralised Web Application
  - No server components needed (except for the final location where to store the data)
- Lazy User
- Lazy Developer
- Intuitive user interactions > complex data models
  - What are the features we need to solve 80% of the use cases
  - Data models should bridge the gap from **no data** -> **structured RDF data** -> **linked data** -> your desired data model
- Declarative App (tell what it does, not how and where)
- Should work against authenticated Solid API, possible other LDP endpoints
- Provide inspiration for standards, industry, etc

Architecture



Form Viewer App

- https://github.com/phochste/FormViewer
  - Form = RDF resource that defines what to show in an HTML form plus the RDF bindings
    - Currently using Daniël Beeke's rdf-form
  - Data = some RDF resource on the web (possibly on an (authenticated) Pod)
  - Shape = should be defined in the Pod
  - Footprint = RDF resource that defined what to do when the submit button is pressed
    - Currently using Hydra vocabulary
  - Pass all Form, Data, Footprint to the WebApp by reference
    - Currently using OpenURL (ANSI/NISO Z39.88-2004)

Form (rdf-form)

Create a ex:Book instance

Create a name/value pair as dc:title

Creata a dropdown as ex:rating

```
:form a form:Form ;
   hydra:endpoint <https://httpbin.org/post> ;
   hydra:supportedClass [
      a hydra:Class ;
      hydra:method "POST"
   ] ;
   form:binding ex:Book .

:help
   a form:Field ;
   form:binding ex:brol;
   form:widget "textarea" ;
   form:label "Instructions"@en ;
   form:placeholder "Please create a book review" ;
   form:readonly true;
   form:order 0 .

:title
   a form:Field ;
   form:binding dc:title;
   form:widget "string" ;
   form:label "Title"@en ;
   form:order 1 ;
   form:required true .

:author
   a form:Field ;
   form:binding dc:creator ;
   form:widget "string" ;
   form:label "Author"@en ;
   form:order 2 ;
   form:required true .

:rating
   a form:Field ;
   form:binding ex:rating ;
   form:widget "dropdown" ;
   form:option (
      [
        form:value    ex:NotLikeIt ;
        form:label    "★ - I don't like it"@en ;
      ]
      [
        form:value    ex:ItWasOk ;
        form:label    "★★ - It was ok"@en ;
      ]
```

https://formviewer.patrickhochstenbach.net/book-review.form.ttl

Data

```
@prefix ex: <https://example.org/> .
@prefix dc: <http://purl.org/dc/terms/> .

[] a ex:Book ;
  dc:title "Winne the Pooh" ;
  dc:creator "A.A. Milne" ;
  ex:rating ex:LikedIt ;
  ex:other "Test" ;
  dc:description "Not enough cats".
```

Data the form can create/update

Footprint (hydra)

```
@prefix hydra: <http://www.w3.org/ns/hydra/core#> .
@prefix dc: <http://purl.org/dc/terms/> .

[] hydra:endpoint <https://hochstenbach.inrupt.net/inbox> ;
  ## Optional define where to go after submitting the form
  # hydra:next <http://some.page.on.the.web>
  ## Optional header to show on form
  # dc:title "The title of my form"
  # dc:description "The description of my form"
  hydra:supportedClass [
      a hydra:Class ;
      hydra:method "POST"
  ] .
```

Send to this resource after submitting the form

Using HTTP POST

Acme Form Viewer

Login | Show details | New Form

Instructions

Please create a book review

Title

Winne the Pooh

Author

A.A. Milne

Rate this book

- Select a value -

Review

Not enough cats

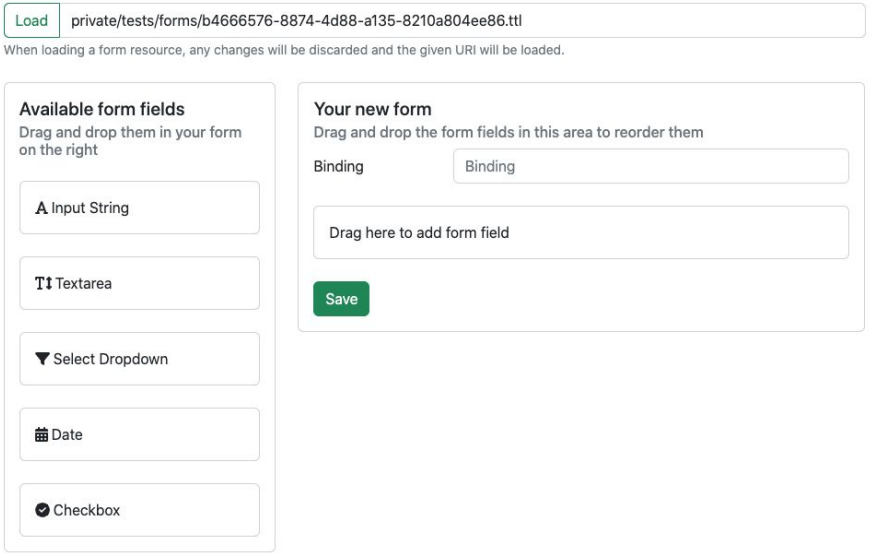
Save

RDF-Form supports

- Name value pairs
- Textarea
- Checkboxes
- Selection
- Autocomplete with SPARQL query
- Groups of combination of above
- Repeated fields,
- … but do we want / need all that complexity for Google Forms like use cases?
- What are the minimum requirements?

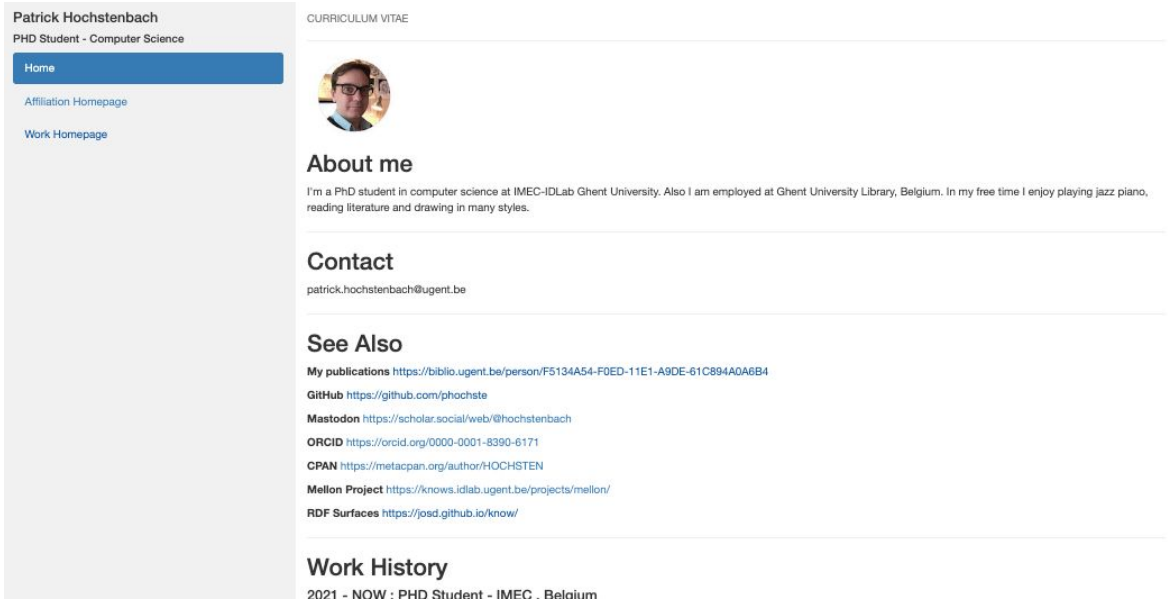
Form Generator

- Form Generator
  - Drag & Drop Web App by smessie to create the Shape of a Form
  - Based on Google Forms data model
    - Name/Value pairs
    - TextArea
    - Select Dropdown
    - Checkbox
    - Date
  - RDF Bindings
  - Labels
  - Generates shape in format
    - SHACL
    - Solid-UI
    - rdf-form



Create RDF linked data and what next?

- Start a workflow to create better linked data
- Create apps that use known shapes in a nice way



https://github.com/phochste/CVViewer

Next steps

- Adding reasoning to Solid Apps
  - Schema alignment
    - My FormViewer app uses Daniël Beeke's rdf-form vocabulary, but what if I get a form definition in SHACL or Solid-UI
    - What if the data source I want to edit has a slightly different shape then the form definition?
  - Footprints
    - The FormViewer uses Hydra and can only send data to hardcoded locations.
    - What I want to use multiple locations?
    - What I want to decide only storage location dynamically (based on the data)?
- Currently investigating N3Logic and RDF Surfaces with smessie

https://w3c.github.io/N3/spec/
https://josd.github.io/surface/

Questions?

- Patrick.Hochstenbach@UGent.be
- @hochstenbach@scholar.social