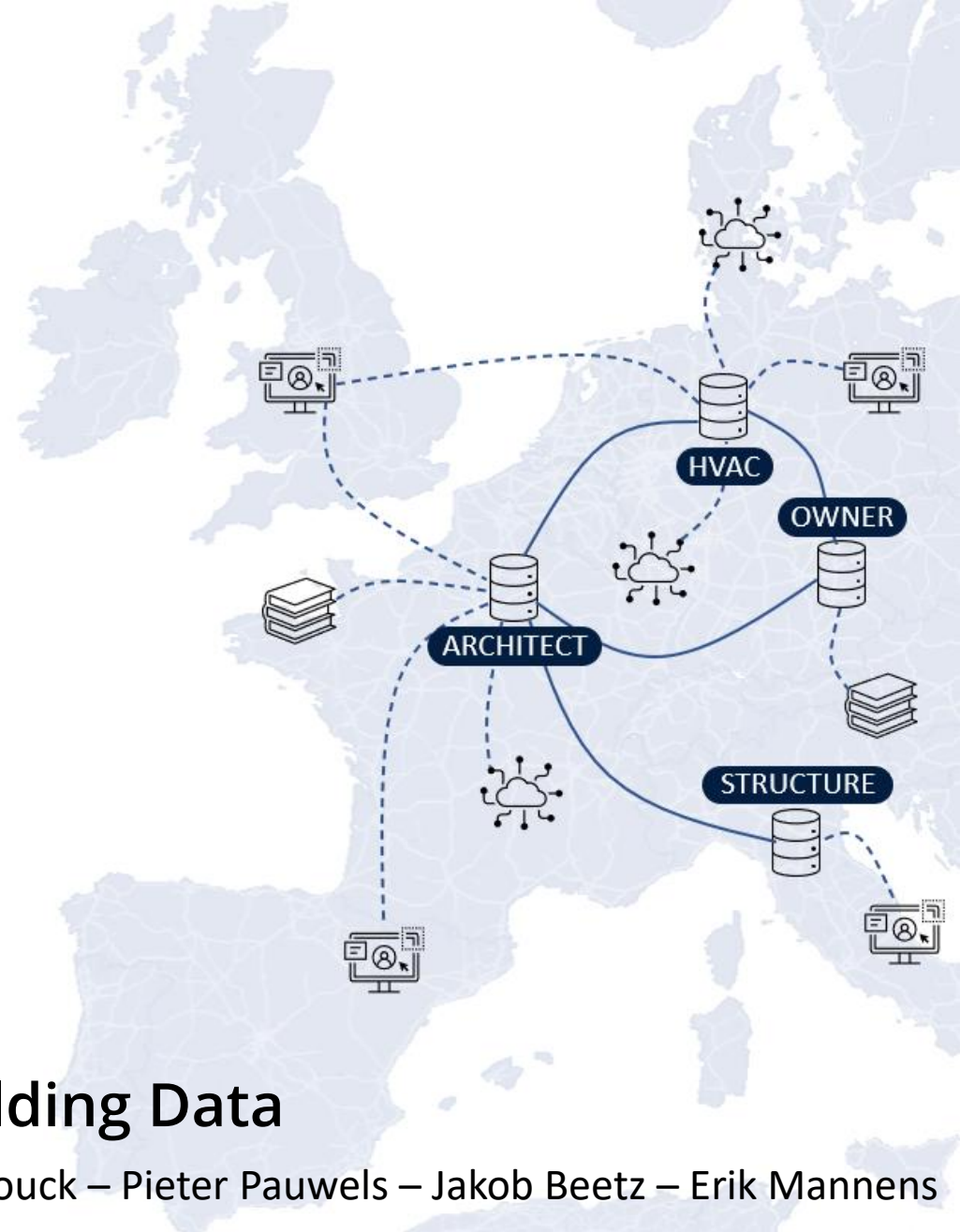


LBDserver

A Federated Ecosystem for Heterogeneous Building Data

Jeroen Werbrouck – Pieter Pauwels – Jakob Beetz – Erik Mannens



PROJECT BACKGROUND

- ❖ M. Sc. in Engineering: Architecture (UGent, 2013-2018)
- ❖ PhD research grant FWO Flanders (2019-...)
- ❖ Web- and data-based AECO projects
- ❖ Data federation
- ❖ Semantic data models (RDF) & sub-document identifiers (no RDF)
- ❖ Modular, “plugin-based” service infrastructure



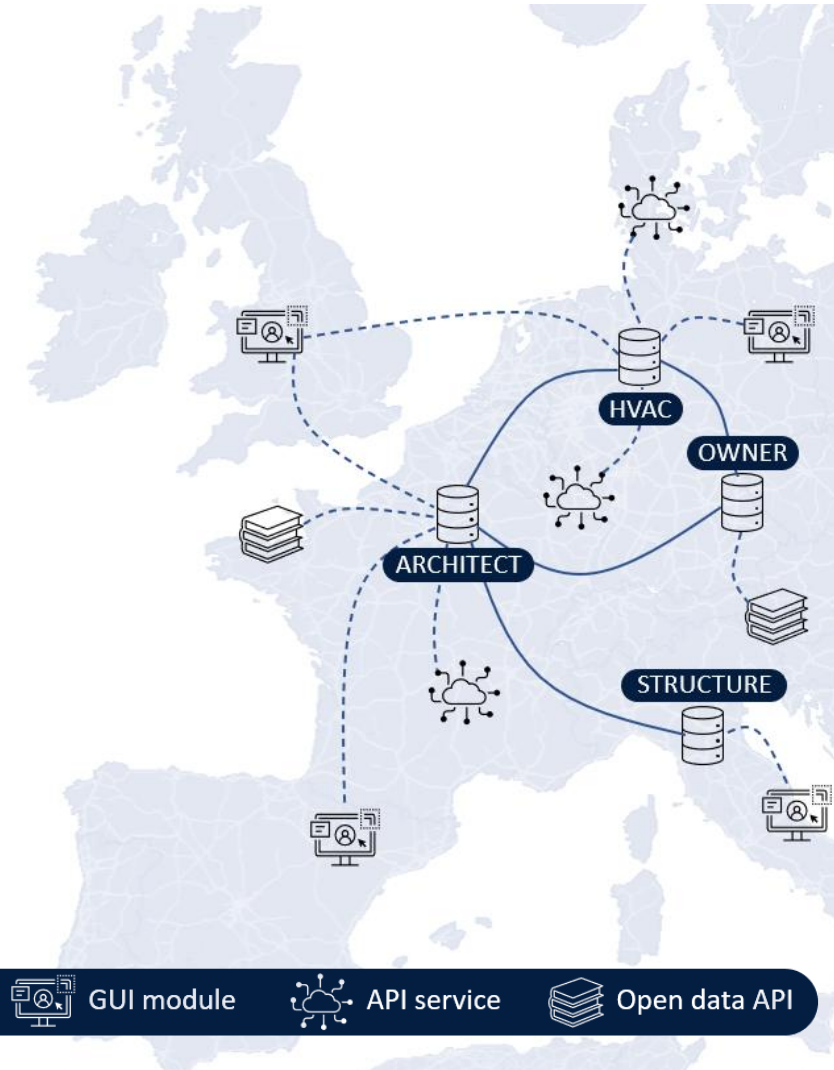
INTRODUCTION

CONTEXT: DATA FEDERATION IN AEC

- ❖ Project-specific data
- ❖ Contextual data
- ❖ Services
- ❖ Data heterogeneity

Decentralised access control
Sub-document information linking

“Using a 3rd party Web BIM service, the **Asset Owner** localises a damage pattern on a picture of the **Facility Manager**, linking it to an existing building element via the as-built 3D model provided by the **Architect**, and referring to external regulation datasets provided by the **government**.”



SEMANTIC ENRICHMENT OF HETEROGENEOUS DATA

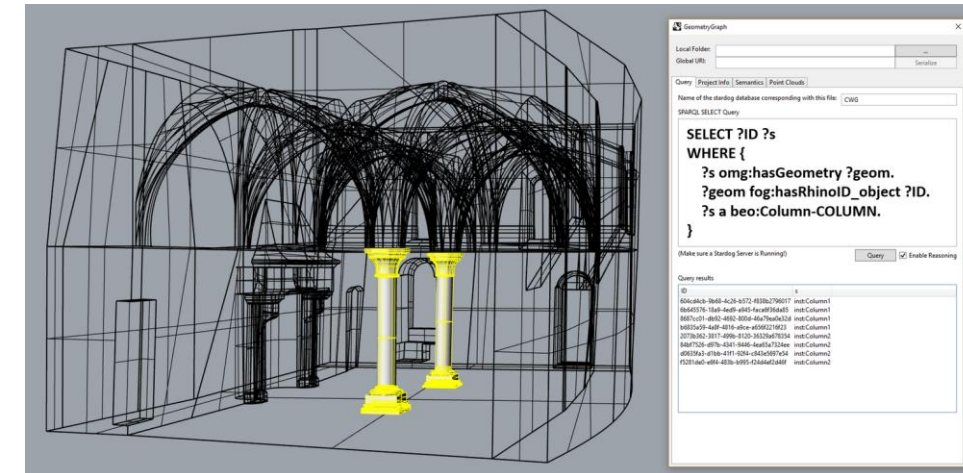
❖ Scan-to-Graph → semantic enrichment of existing building geometry (LBD CG 01/12/2020)

❖ BUT: limited scope

- ❖ Central data storage (triple store)
- ❖ Use geometry to access/enrich semantics

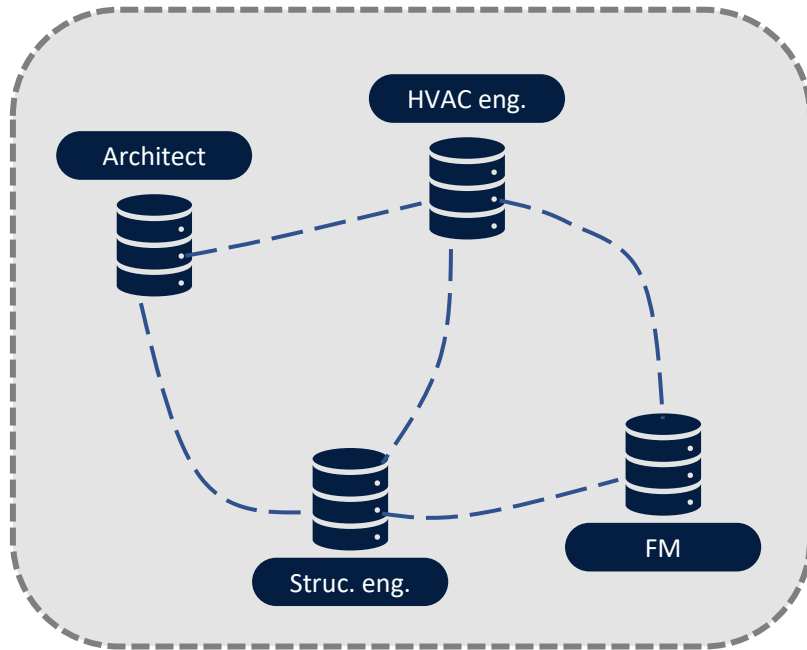
❖ EXTENSION:

- ❖ Federated data storage
- ❖ Use *any* resource to access *any* other resource in the project



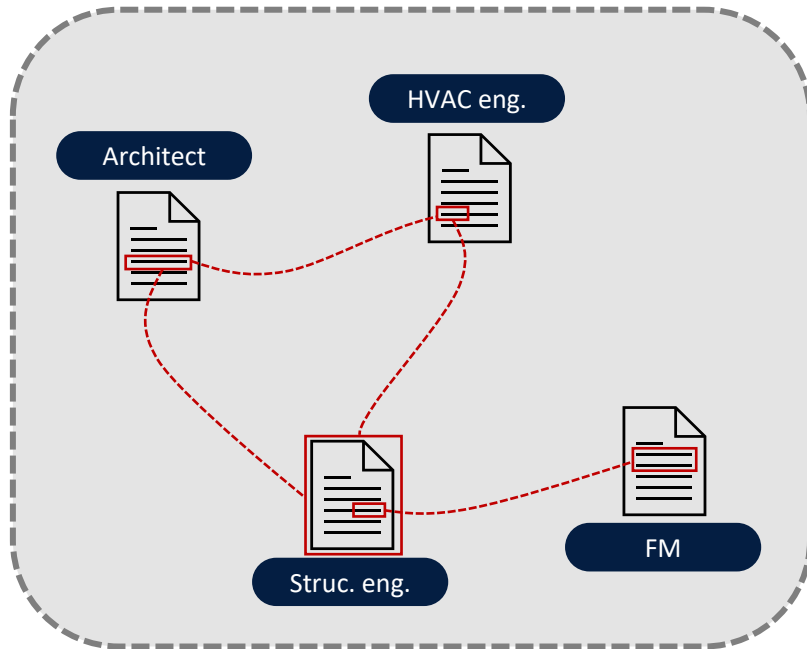
<https://doi.org/10.1016/j.autcon.2020.103286>

PRESENTATION OVERVIEW



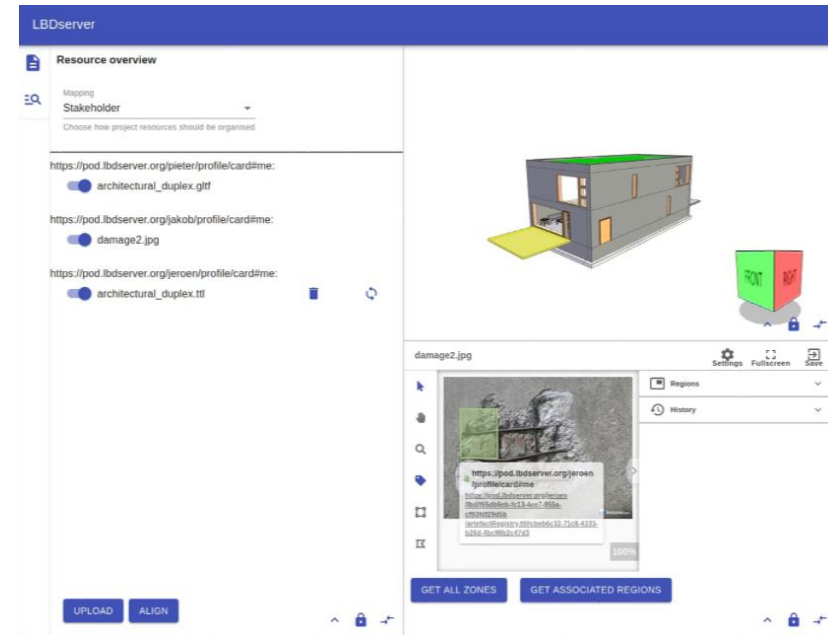
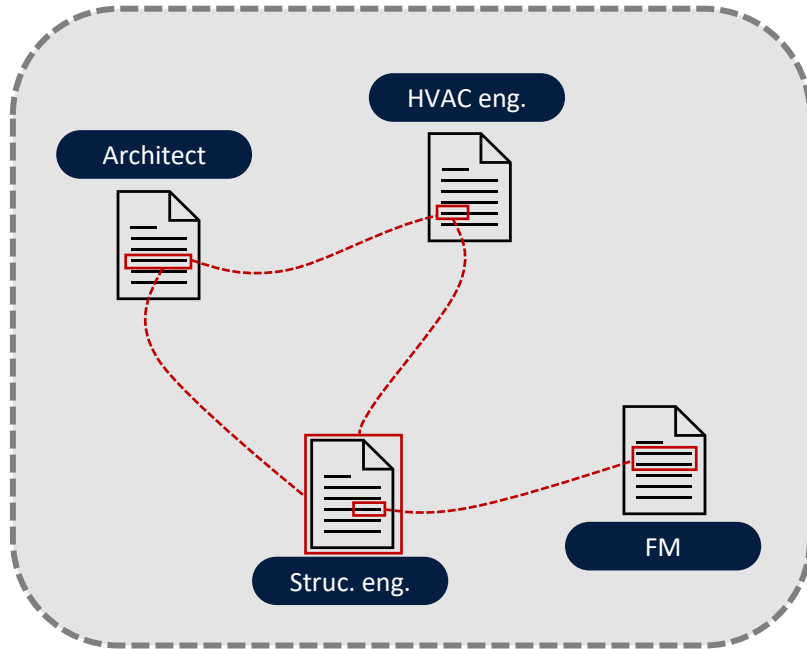
1. Project Discovery and storage patterns

PRESENTATION OVERVIEW



2. Federated sub-document linking

PRESENTATION OVERVIEW

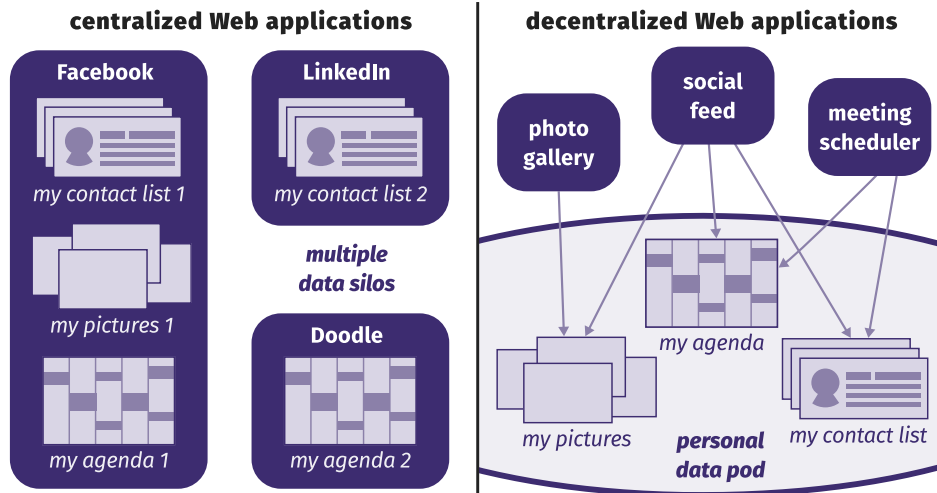


3. Concepts for flexible end-user interaction

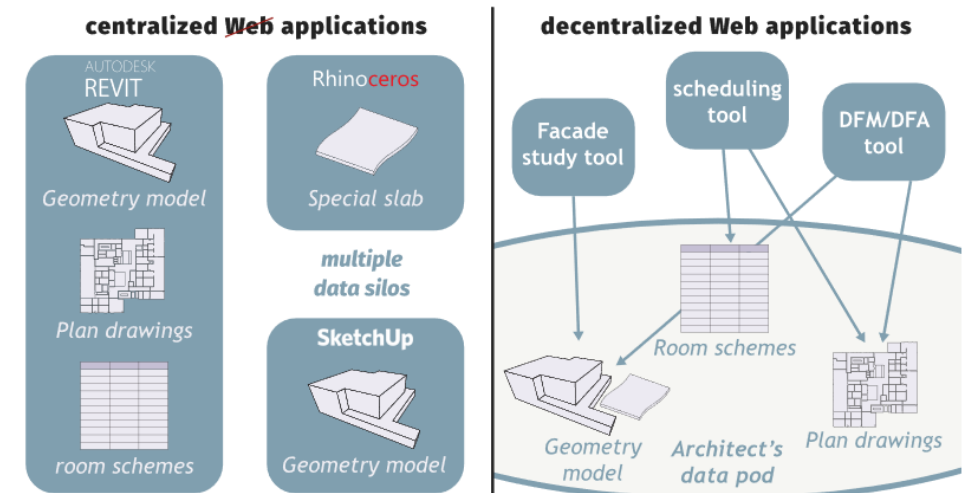
1. DATA PATTERNS

CONTEXT: SOLID

- ❖ Web ID: A URL that represents an actor on the Web ➡ “Web username”
- ❖ Data Pod: Personal data storage linked to a Web ID ➡ Linked Data Platform (LDP) + AUTH
- ❖ Identity Provider: Instance hosting your WebID (and Pod) ➡ Can be self-hosted



© Ruben Verborgh, 2018



© Mads Holten Rasmussen, 2018

SOLID FOR AEC?

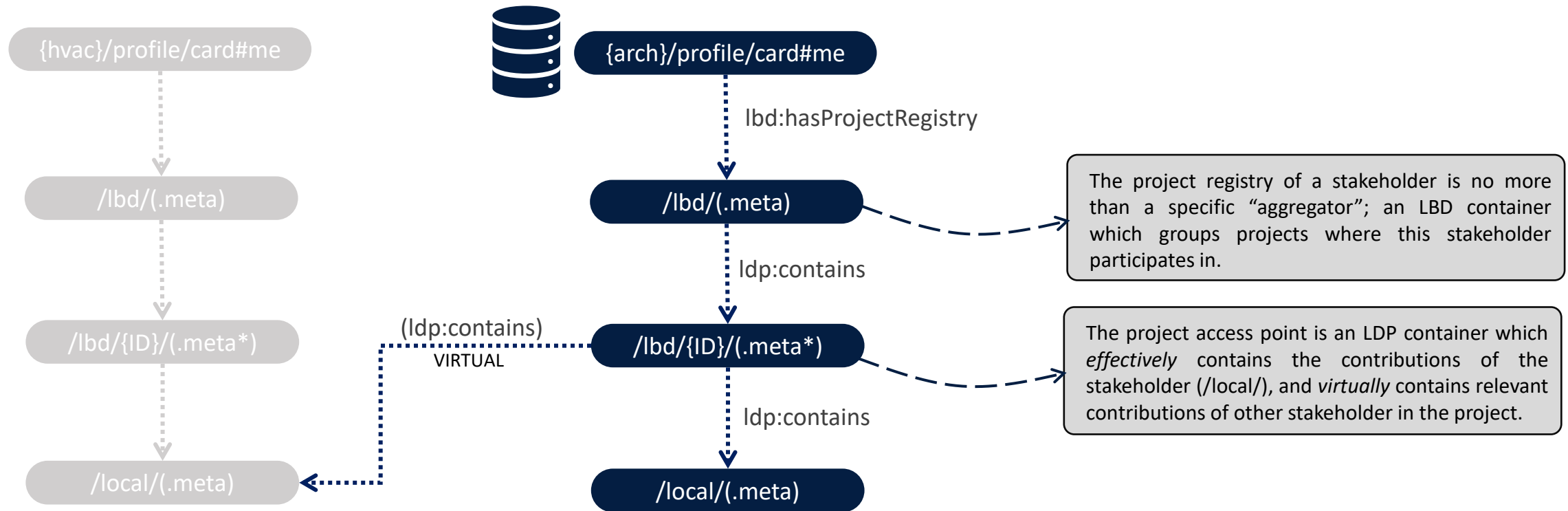
- ❖ Professional use ➡ self-hosting of Pods and WebIDs is possible
- ❖ WebIDs are public ➡ project discovery starts from a WebID
- ❖ Although projects will be federated, their boundaries can still be well-defined (i.e. limited to the contributions of consortium members)
- ❖ Boundaries can be defined by every stakeholder separately, as everyone can have their own access point
- ❖ It should be possible to include information that is not Solid-based.

AGGREGATORS

- ❖ LDP containers with 'virtual' subcontainers
- ❖ Point to project **access point URLs** ➡ LDP containers that aggregate local project contributions of offices
- ❖ Group projects (public or private) based on specific parameters:
 - ❖ Participation / ownership
 - ❖ the project registry of an office POD
 - ❖ All projects owned by the government
 - ❖ Typology
 - ❖ All libraries in Flanders
 - ❖ All bridges in Germany
 - ❖ ...

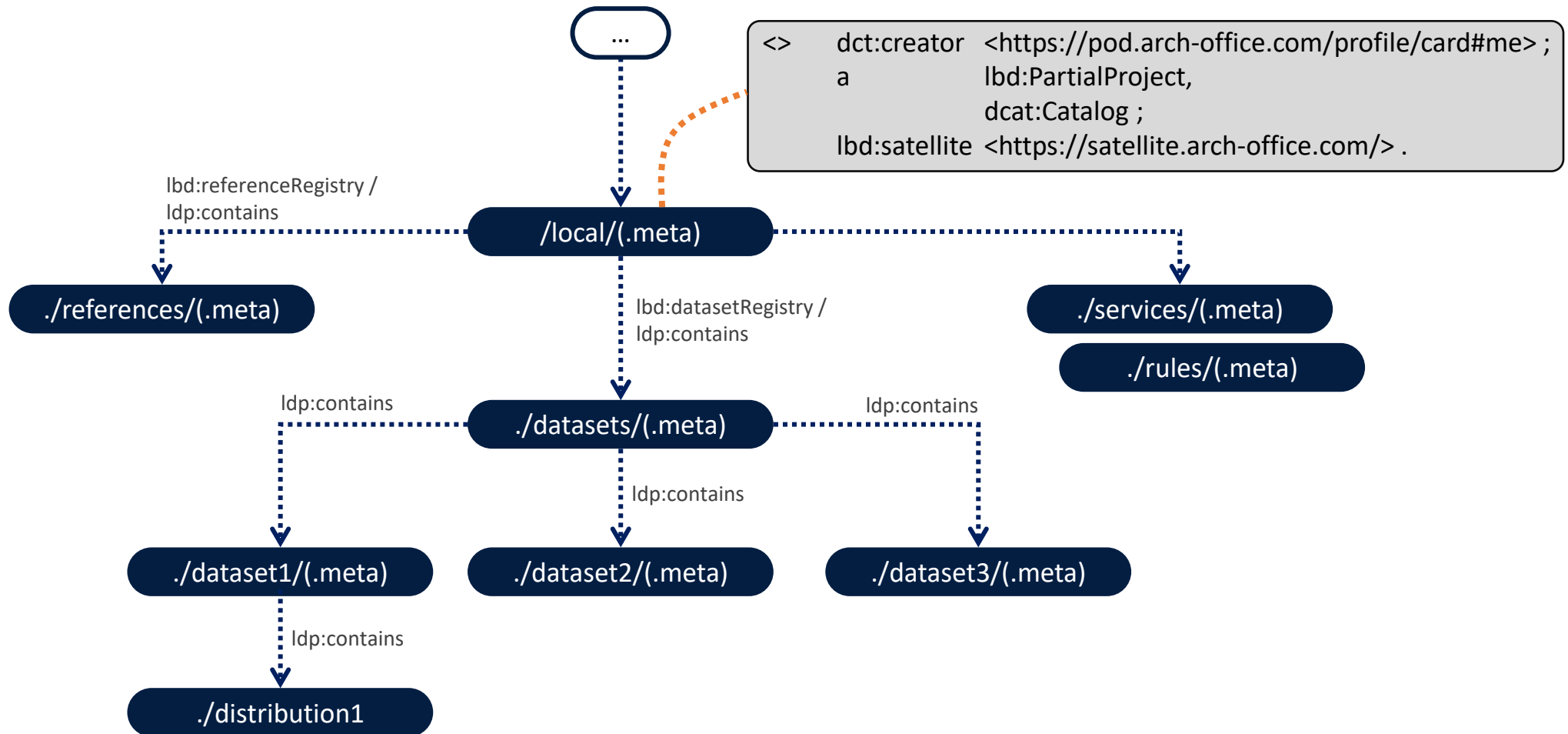


PROJECT DISCOVERY

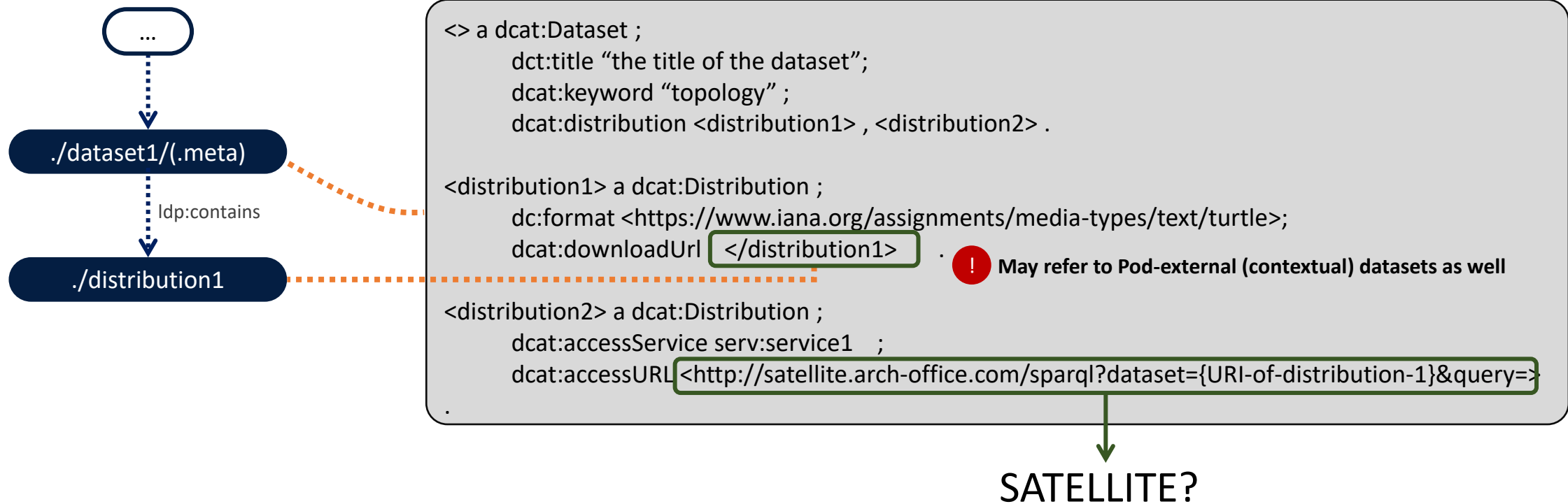


* In Solid, “.meta” files are mapped to specific LDP containers. Dereferencing the container yields the .meta resource.
** By default, “.meta” files in Solid cannot be edited directly. In this case, this restriction was overruled in favor of project organisation

PROJECT ORGANISATION



DATASETS

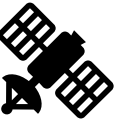


SATELLITES

- ❖ Solid Pod is the main access point
- ❖ Sometimes, external services are needed...
 - ❖ APIs / orchestration services:
 - ❖ CRUD project data
 - ❖ Maintain consistency
 - ❖ Provide virtual views on top of datasets
 - ❖ Querying (SPARQL service, time series)
 - ❖ Document store (MongoDB, ...)



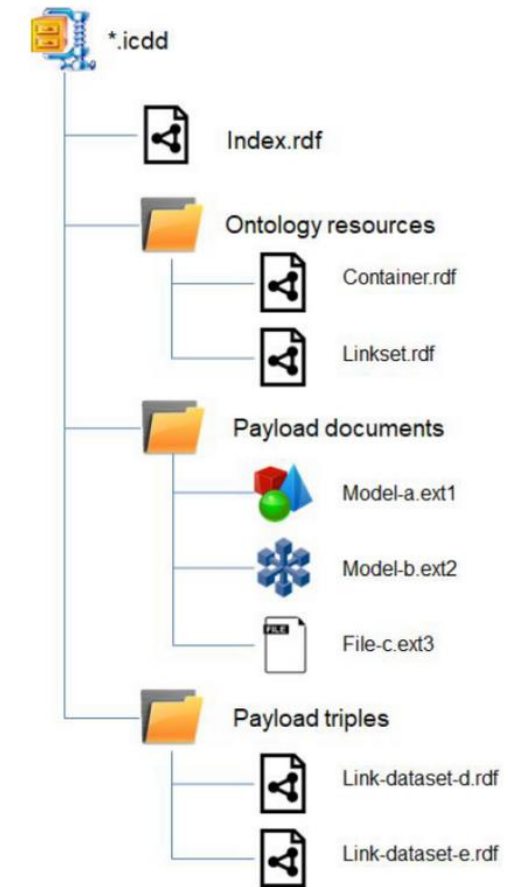
Require powerful access control rights! (TRUST)
Pod-specific “Satellites” (one or more)



2. SUB-DOCUMENT LINKING

CONTEXT: SUB-DOCUMENT LINKING - ICDD

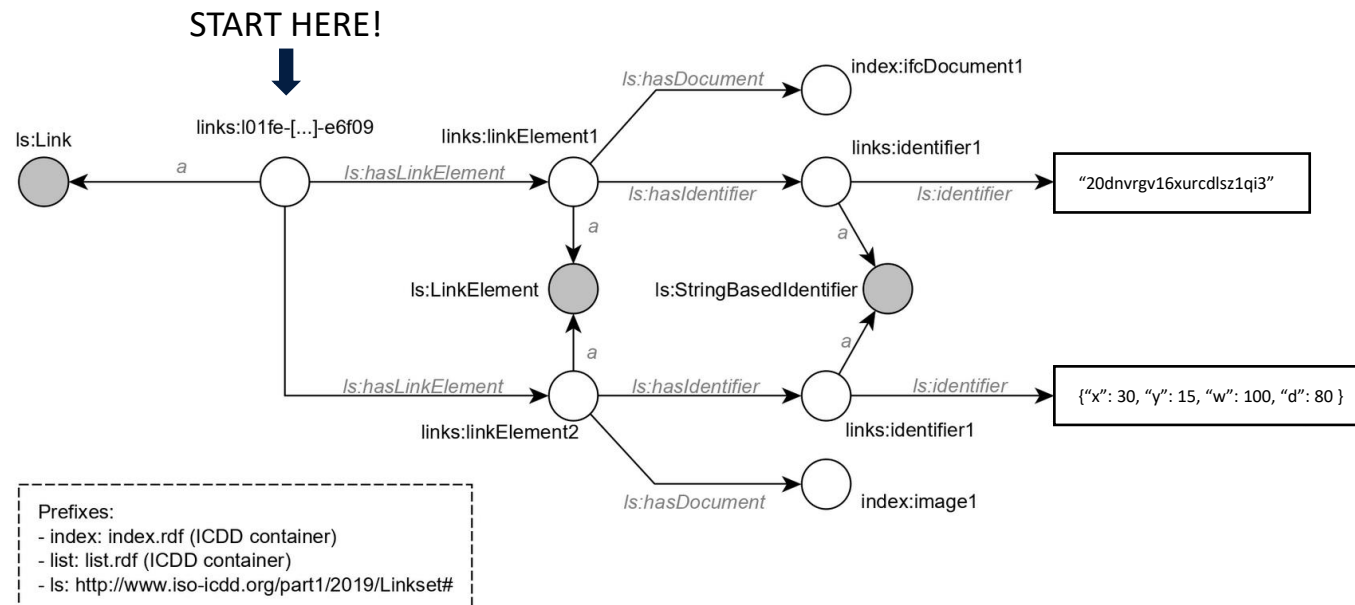
- ❖ Information Container for linked Document Delivery (ICDD, ISO 21597)
- ❖ RDF vocabulary to semantically integrate non-RDF resources
- ❖ ZIP container with dump of entire project
 - ❖ Final point of data exchange
 - ❖ Not a database
 - ❖ Not a CDE



© ISO 21597

CONTEXT: SUB-DOCUMENT LINKING - ICDD

- ❖ Information Container for linked Document Delivery (ICDD, ISO 21597)
- ❖ ZIP container with dump of entire project
- ❖ RDF vocabulary to semantically integrate non-RDF resources



REFERENCE REGISTRY

- ❖ Linking dataset-specific contexts to project-wide “abstract concepts”
 - ❖ Dataset ➡ Local project
 - ❖ Local project ➡ Global project
- ❖ Enriching a *concept* ➡ “Door X has parameter P”
- ❖ Enriching the *representation* of a concept ➡ “Geometry Y, which represents Door X, has an insufficient LOD”
- ❖ Reference registry = dataset
 - ❖ RDF-based distribution
 - ❖ JSON/BSON

CONCEPT ENRICHMENT: EXAMPLE

❖ ARCHITECT

- ❖ creates a *local* concept
- ❖ Creates a *semantic* reference where he says this is a wall
- ❖ Creates a *geometric* reference of this element

❖ ENGINEER

- ❖ opens the geometric reference of the architect in a Web Viewer
- ❖ uses it as a proxy to create a *local alias* (owl:sameAs) of the concept in his own Reference Registry
- ❖ creates a *semantic* reference where he indicates the element's properties (e.g. U-value 0,4 W/m²K)

❖ ...

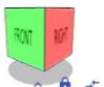
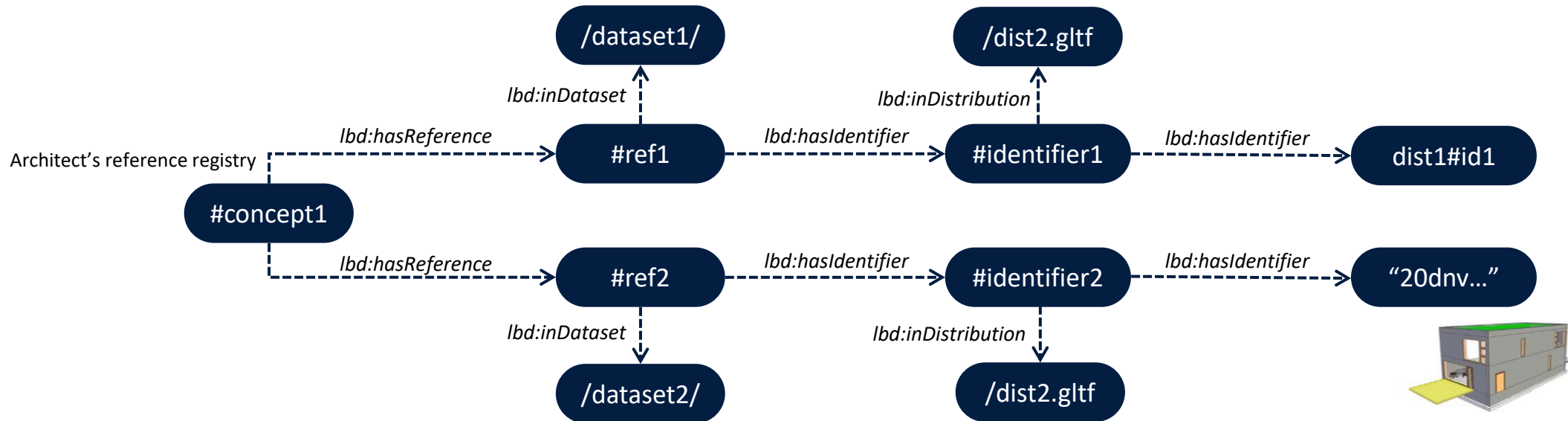
❖ {other stakeholder}

- ❖ Uses the aggregation of earlier enrichments for performing own tasks and responsibilities

CONCEPT ENRICHMENT: EXAMPLE

❖ ARCHITECT created a *local* concept <#concept1>

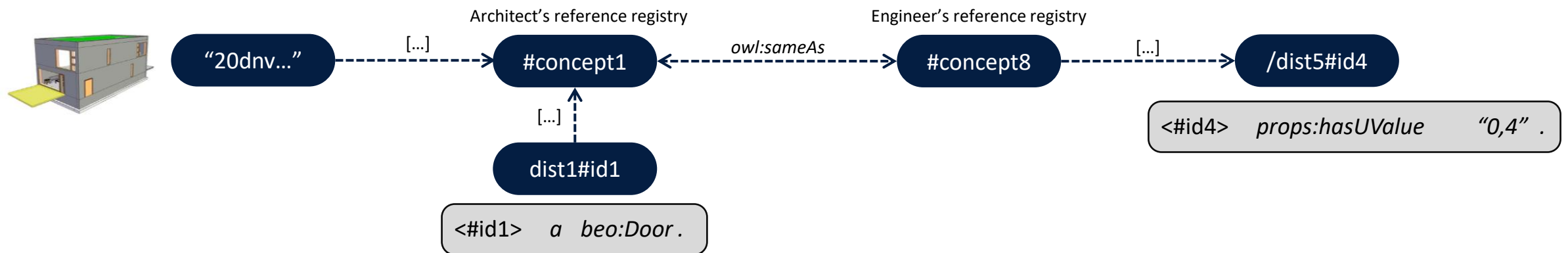
- 1) With a *semantic reference*: in dataset “/dataset1/” in distribution /dist1\$.ttl <#id1> a beo:Wall
- 2) With a *geometric reference*: in dataset “/dataset2/” in distribution /dist2\$.gtf “20dnv...”



CONCEPT ENRICHMENT: EXAMPLE

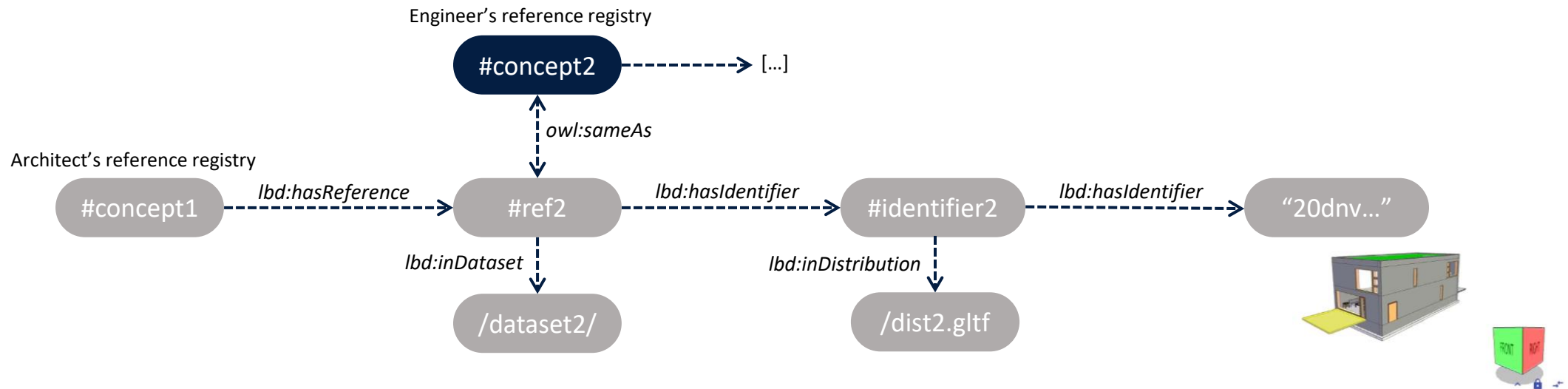
❖ ENGINEER

- ❖ opens the geometric reference of the architect (e.g. in a Web Viewer)
- ❖ uses it as a proxy to derive the abstract concept from the architect's Reference Registry
- ❖ creates a *local alias* (owl:sameAs) of the concept in his own Reference Registry
- ❖ Informs the satellite of the architect an alias was created (**backlinking**)
- ❖ creates a *semantic* reference where he indicates the element's properties (e.g. U-value 0,4 W/m²K)



REFERENCE ENRICHMENT

- ❖ Say something about the reference itself rather than its contents




3. END-USER INTERACTION

CONTEXT: INTERACTION WITH DATA

- ❖ Service-to-service communication: checking and validation, simulations, ...
- ❖ Design decisions still need GUI
- ❖ Can we have a *dynamically generated* GUI?
 - ❖ Based on available project data (geometry, semantics, images, spreadsheets, Web APIs ...)
 - ❖ Based on the needs of the task at hand

CONTEXT: MODULE FEDERATION

- ❖ Webpack 5: Module federation / microfrontends
- ❖ Standalone plugins, deployed on the Web
- ❖ Can also be used in configuration with others (Container application)
- ❖ Run-time loading  dynamically load UI depending on needs
- ❖ Minimal exchange of data between plugins
 - ❖ Auth session
 - ❖ Project ID
 - ❖ “active datasets”
 - ❖ Currently selected elements
 - List of abstract concepts (see Section 3 [Sub-document Linking])
 - Plugin ‘knows’ which representations/manifestations it can handle

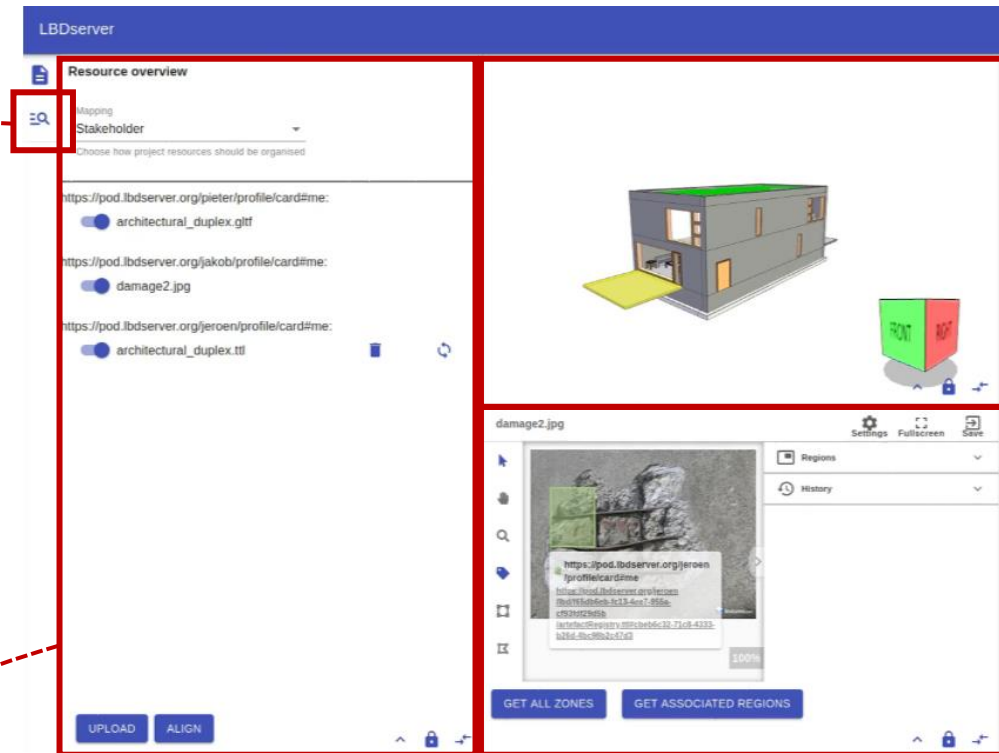
E.g. For task X, I will need:

- a 3D viewer that can display IFC models
- an image annotator
- a SPARQL query module
- ...

```
}  
}  
},  
"viewer": {  
  "url": "https://consolidproject.github.io/pluginViewer/remoteEntry.js",  
  "scope": "viewer",  
  "module": "./index",  
  "dimensions": {  
    "x": 600,  
    "y": 0,  
    "h": 450,  
    "w": 660  
  }  
},  
"imageannotator": {  
  "url": "https://consolidproject.github.io/pluginImageViewer/remoteEntry.js",  
  "scope": "imageannotator",  
  "module": "./index",  
  "dimensions": {  
    "x": 600,  
    "y": 450,  
    "h": 450,  
    "w": 660  
  }  
}  
}
```

<https://lbd-dev.org/resourceOverview/remoteEntry.js>

<https://3dsolutions.be/viewerplugin/remoteEntry.js>



<https://lbd-dev.org/resourceOverview/remoteEntry.js>

<https://another.com/imgAnnotate/remoteEntry.js>

LBDserver

Resource overview

Mapping

Stakeholder

Choose how project resources should be organised

https://pod.lbdserver.org/pieter/profile/card#me:

architectural_duplex.gltf

https://pod.lbdserver.org/jakob/profile/card#me:

damage2.jpg

https://pod.lbdserver.org/jeroen/profile/card#me:

architectural_duplex.ttl

No glTF models selected

UPLOAD

ALIGN

You can visit <https://lbdserver.org> for more information. When citing this research, please refer to one of the following papers: [...]

AUTHENTICATION

The LBDserver demo projects can be read and queried publicly. However, in most cases projects will not be open to the public - authentication is required. You can get a federated Web identity at the Solid Identity Provider of your choice. You can also [set up such Identity Provider yourself](#). Having a Web Identity and a personalised online data vault ("Pod"), you can start creating your own federated LBDserver projects.

You are logged in as:
<https://pod.lbdserver.org/jeroen/profile/card#me>

SIGN OUT

DEMO PROJECTS

Projects can be found via aggregators, which basically contain pointers to existing projects. Aggregators are basically LDP (Linked Data Platform) containers. If you have a dedicated LBDserver project folder configured in your Solid Pod: that is an aggregator. Other aggregators could group projects based on location, typology etc.

Activate a project from a public aggregator:

aggregator

<https://pod.lbdserver.org/jeroen/lbd/>

GET PROJECTS


<https://pod.lbdserver.org/jeroen/lbd/f65db6eb-fc13-4ee7-955a-cf93fdf29d5b/>


Active


MY PROJECTS

MODULE FEDERATION

END-USER INTERACTION

 GHENT
UNIVERSITY

 RWTH AACHEN
UNIVERSITY

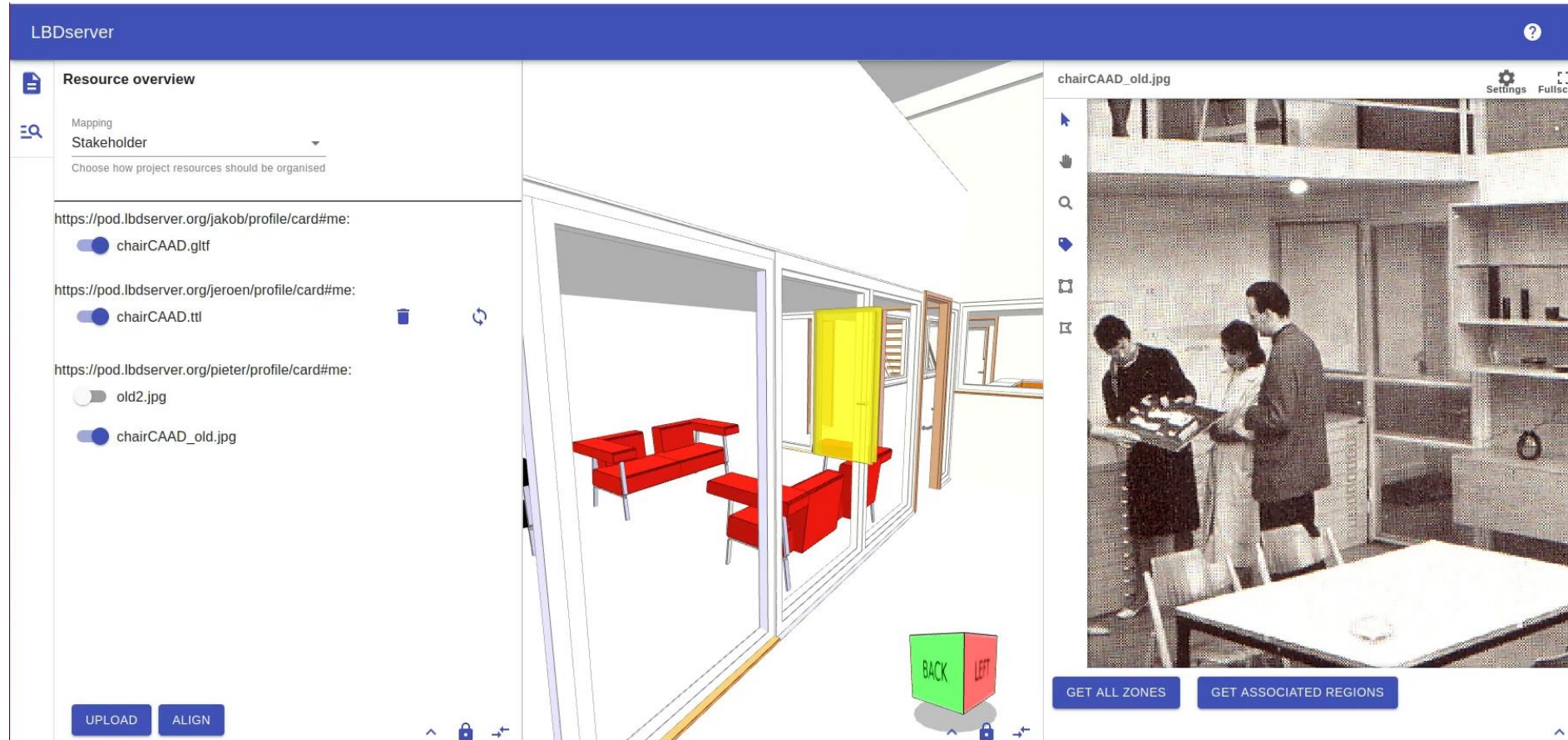
 fwo

4. PROOF-OF-CONCEPT

TRY IT OUT

- ❖ Very experimental!
- ❖ Small server (limited resources)
- ❖ <http://demo.lbdserver.org>

DEMO I



DEMO II

LBDserver

SETUPPROJECTSTOREAUTH

Project overview

http://localhost:5001/fm/profile/card

damages.ttl

http://localhost:5000/projectarchitect/profile/card

gravensteen.gltf

http://localhost:5000/owner/profile/card

notes.ttl

UPLOAD/CREATE

Damage Demo

Target graph

ASSIGN DAMAGE

SHOW DAMAGES


advanced




LEFT

! Outdated implementation – same concepts

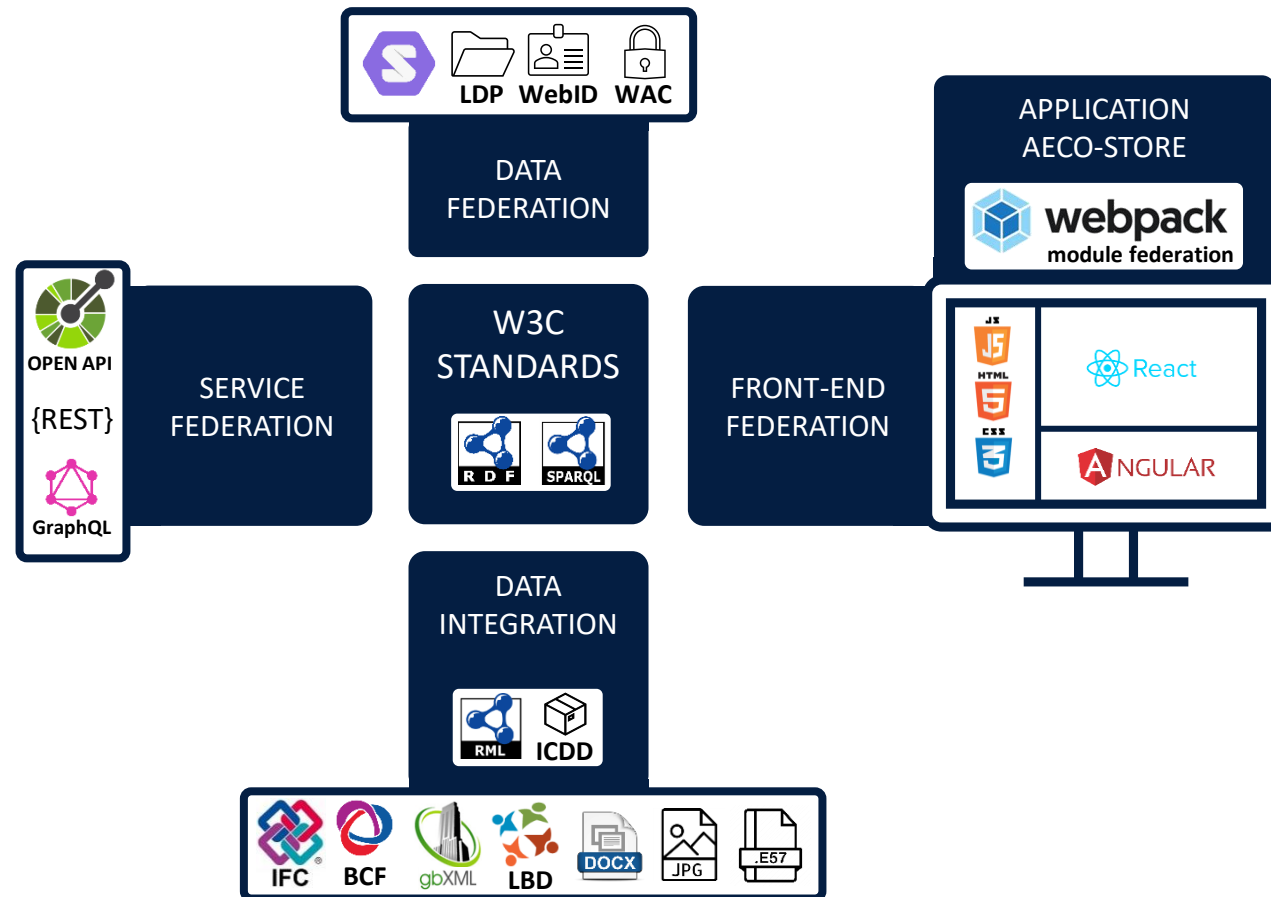
 GHENT
UNIVERSITY

 RWTHAACHEN
UNIVERSITY

 fwo

5. CONCLUSION

TECHNOLOGICAL OVERVIEW OF THE LBDSERVER



SUMMARY

- ❖ “Federated multi-model” based on Solid (AUTH) and well-established data storage technologies (DCAT, LDP)
- ❖ Abstract “things” have dataset-specific representations that can be managed at different stakeholder Pods
- ❖ Disparate UI modules can communicate with one another using abstract things & filtered manifestations
- ❖ UI modules of different origin can be configured in a “shopping cart” configuration

FUTURE WORK

ACADEMIC

- ❖ How to handle *dynamic* phases (e.g. design) in contrast with static phases?
- ❖ Real-time connection & syncing with BIM authoring tools
- ❖ More standardised API approach, less ad-hoc
- ❖ Paper: submitting for Special Issue of Semantic Web Journal

IMPLEMENTATION

- ❖ Publish and document LBDserver backend services (CSS/satellite-API/MongoDB/Fuseki)
- ❖ Document npm module “consolid” for communication between client and servers
- ❖ Document creation of micro-front-end modules on top of the LBDserver

CONTACT

Main Research

Jeroen Werbrouck

PhD Researcher (FWO Flanders)

Ghent University | RWTH Aachen

[Jeroen \[dot\] werbrouck \[at\] ugent \[dot\] be](mailto:Jeroen[dot]werbrouck[at]ugent[dot]be)

Promotors

Erik Mannens (Ghent University | IMEC)

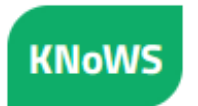
Pieter Pauwels (Ghent University | TU Eindhoven)

Jakob Beetz (RWTH Aachen)

Project Info

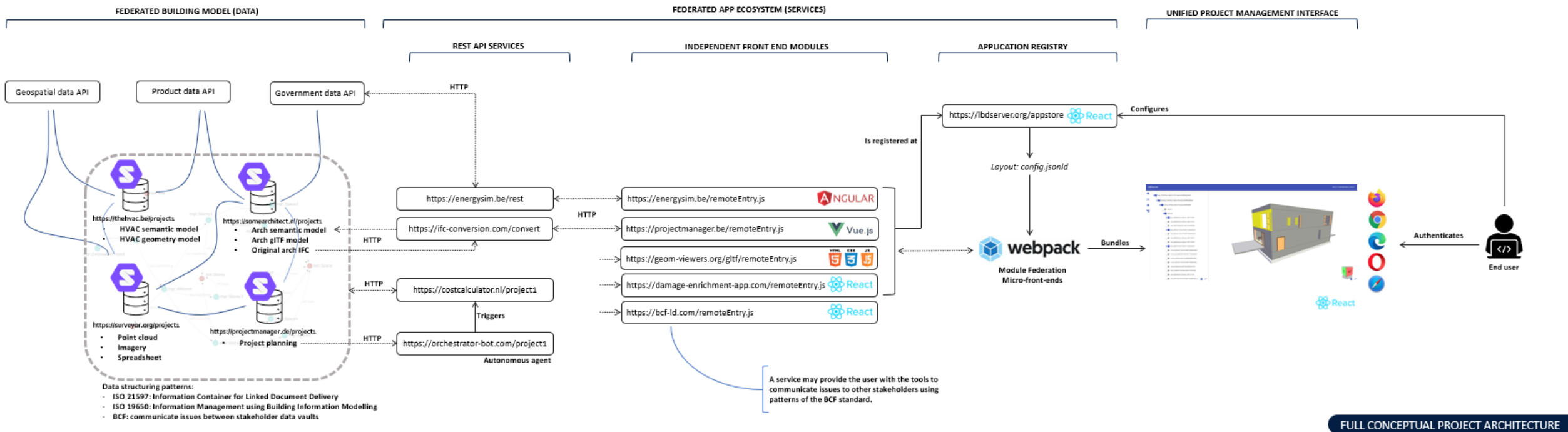
<https://lbdserver.org>

<https://github.com/ConSolidProject/>

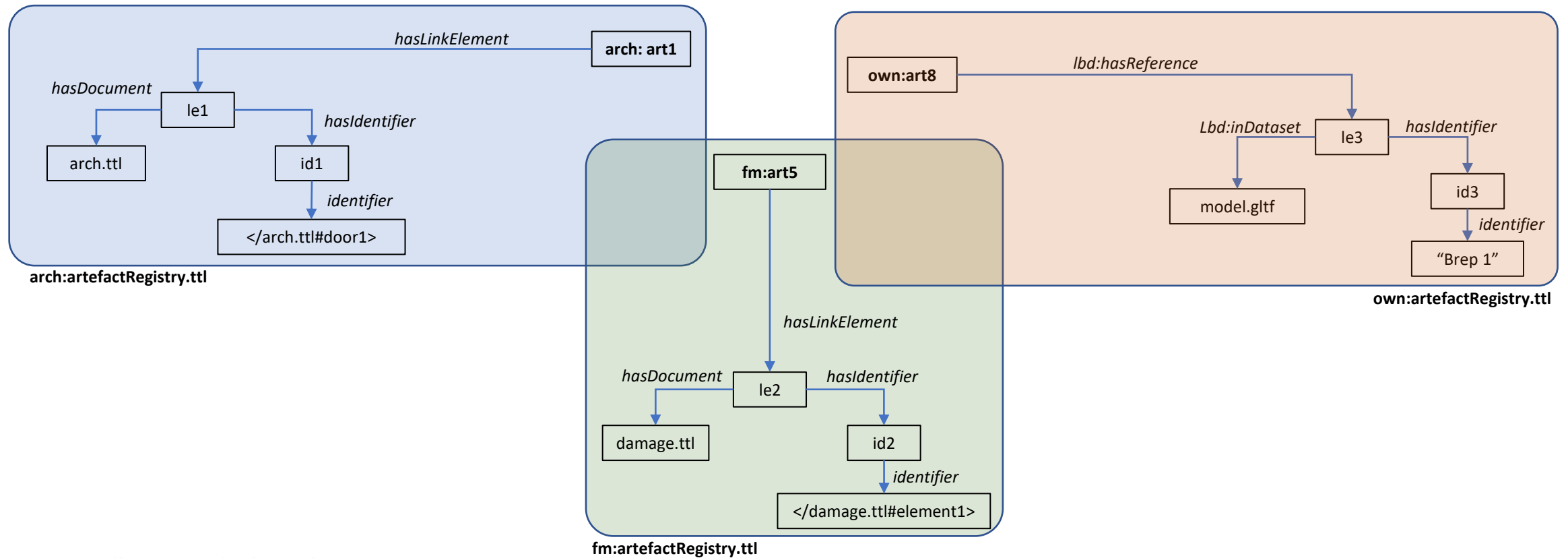


THANK YOU!

OVERVIEW

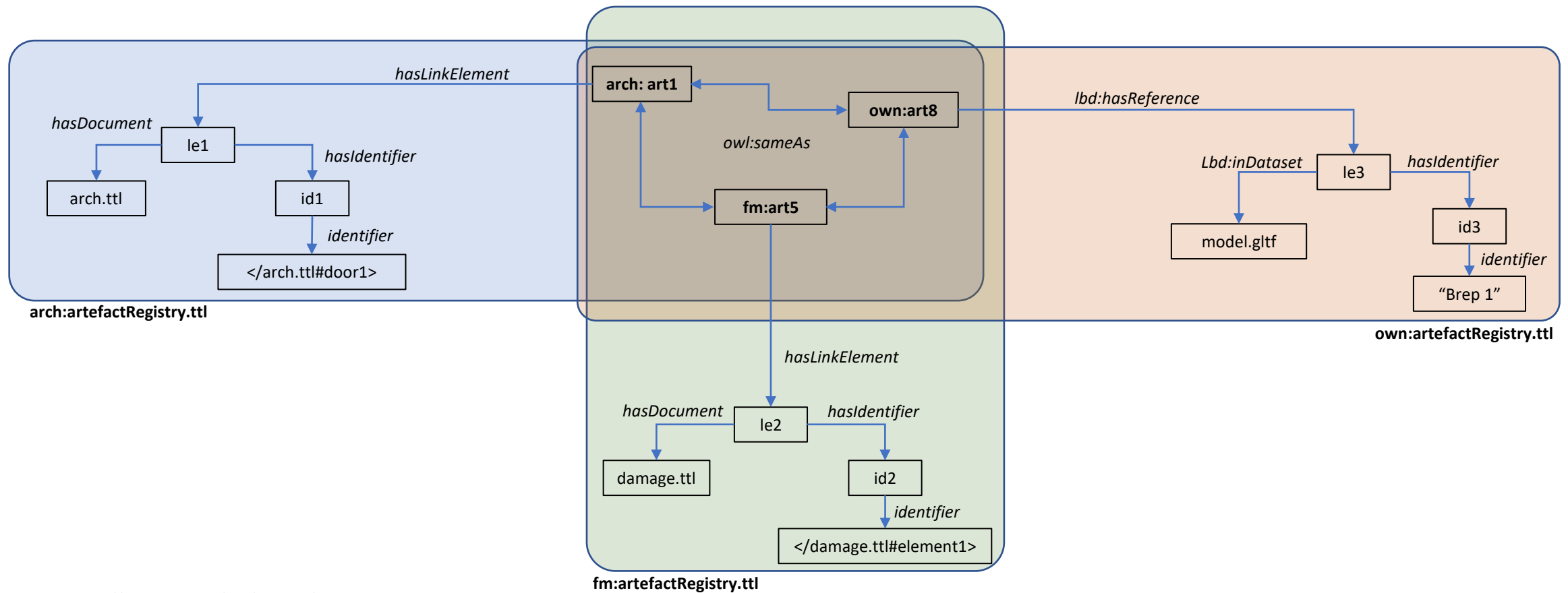


REFERENCE REGISTRY: LOCAL



@prefix arch: <https://archoffice.de/lbd/abc123/> .
@prefix fm: <https://fm-bureau.fr/lbd/abc123/> .
@prefix own: <https://pod.inrupt.net/lbd/abc123/> .

REFERENCE REGISTRY: GLOBAL ALIGNMENT



@prefix arch: <https://archoffice.de/lbd/abc123/> .

@prefix fm: <https://fm-bureau.fr/lbd/abc123/> .

@prefix own: <https://pod.inrupt.net/lbd/abc123/> .