European Commission

30 April 2024

LDES and its Pilots

DIGIT.B2 - Interoperability.

interoperable europe

Workshop practicalities

Audio

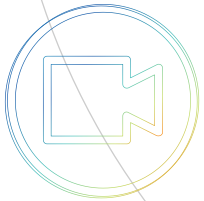
Click on ‘connect audio’ but please mute your microphones

Chat

You can also share your questions for the Q&A session via the chat

Recording

The workshop will be recorded



Objectives of this webinar

01

Update the community

Highlight new LDES Pilots to the community

02

A glimpse of the future

Provide the community with information with what is to come.

Agenda

01

Introduction to DCAT-AP feeds

By Matthias Palmér, Metasolutions and Agency for Digitalisation Sweden & Pieter Colpaert, imec, Ugent and SEMIC

02

The pilot goes on: DCAT-AP feeds at Data.Europa.EU

Simon Steuer, Publications office of the EU

03

LDES @ Rijksmuseum

Tim Thomassen, Maxdoro and Rijksmuseum

04

The future of LDES

Anastasia Sofou, SEMIC

A Linked Data Event Stream (LDES)

A publication technology to share or aggregate information with or from multiple parties

Allowing everyone to replicate and stay up-to-date regarding the unique source of truth

interoperable europe

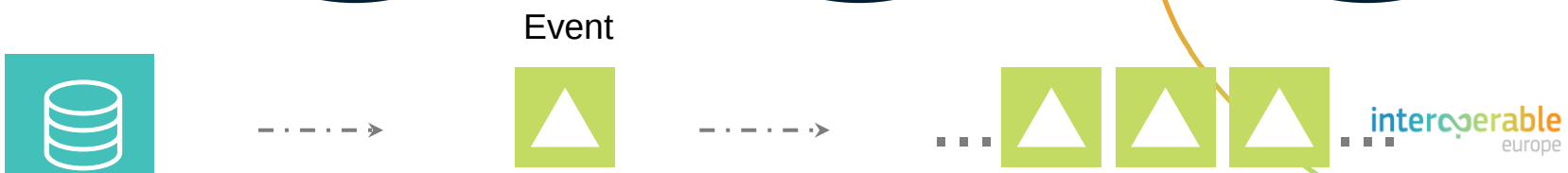
What is a LDES?

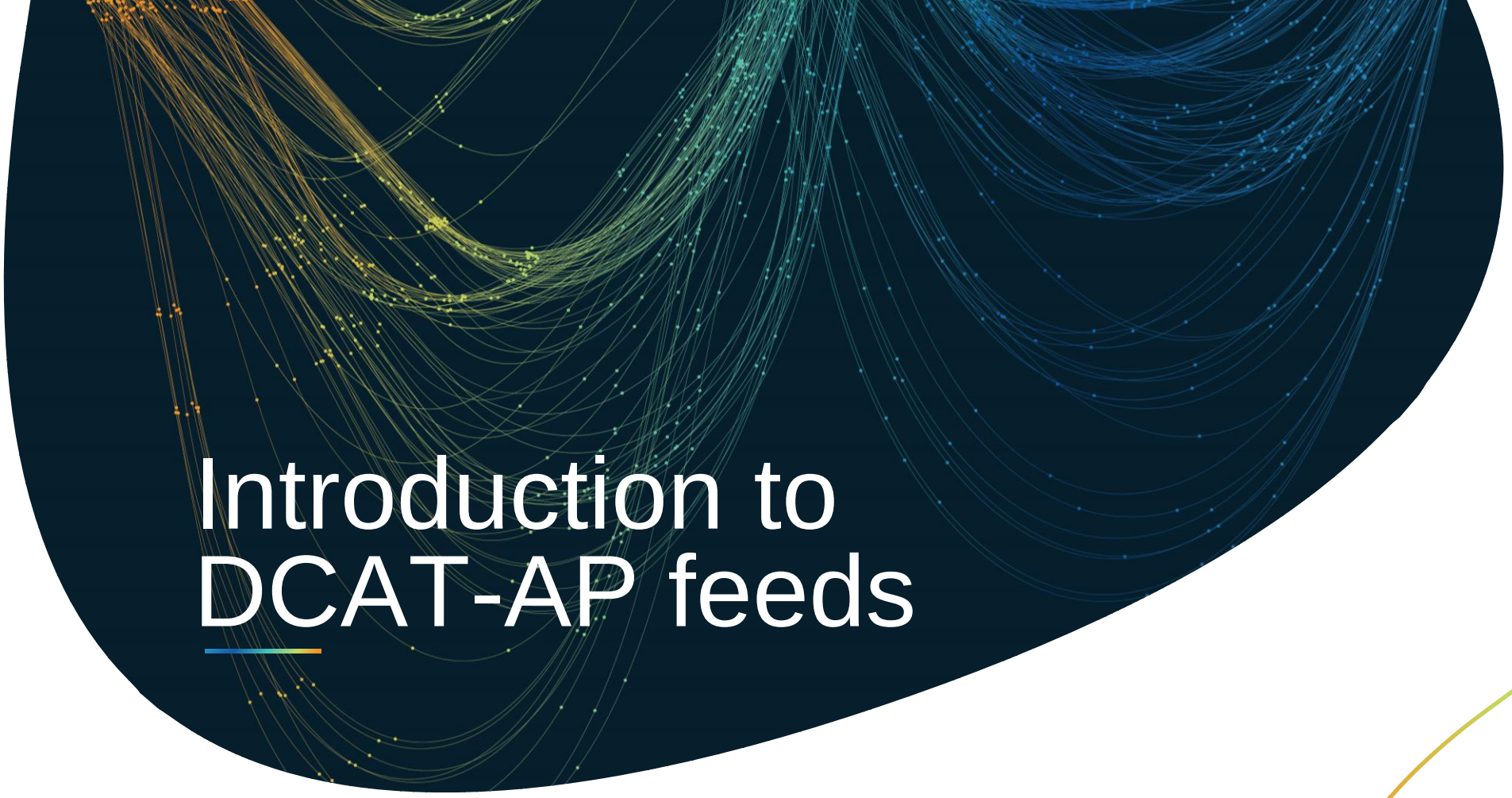
A Linked Data Event Stream (LDES) is a collection of immutable objects whereby you do not change the data itself but simply add new data records to the stream. It represents a publication strategy to publish and make data discoverable in a cost-effective and flexible manner.

LDES helps you to structure your data as stream data, enabling you and your users to keep track of what changed at the data level, independently from the data format.

It allows data users to:

- Have up to date data
- Be aware of changes
- Access to historic data
- Relate historic data to current data





Introduction to DCAT-AP feeds

MYNDIGHETEN FÖR
DIGITAL FÖRVALTNING
Agency for Digital Government

About us

Mattias Ekhem
- Architect
- Digg

Matthias Palmér
- PhD
- Consultant Digg
- CTO Metasolutions AB

Sveriges dataportal

Digg – Myndigheten för digital förvaltning

Agenda

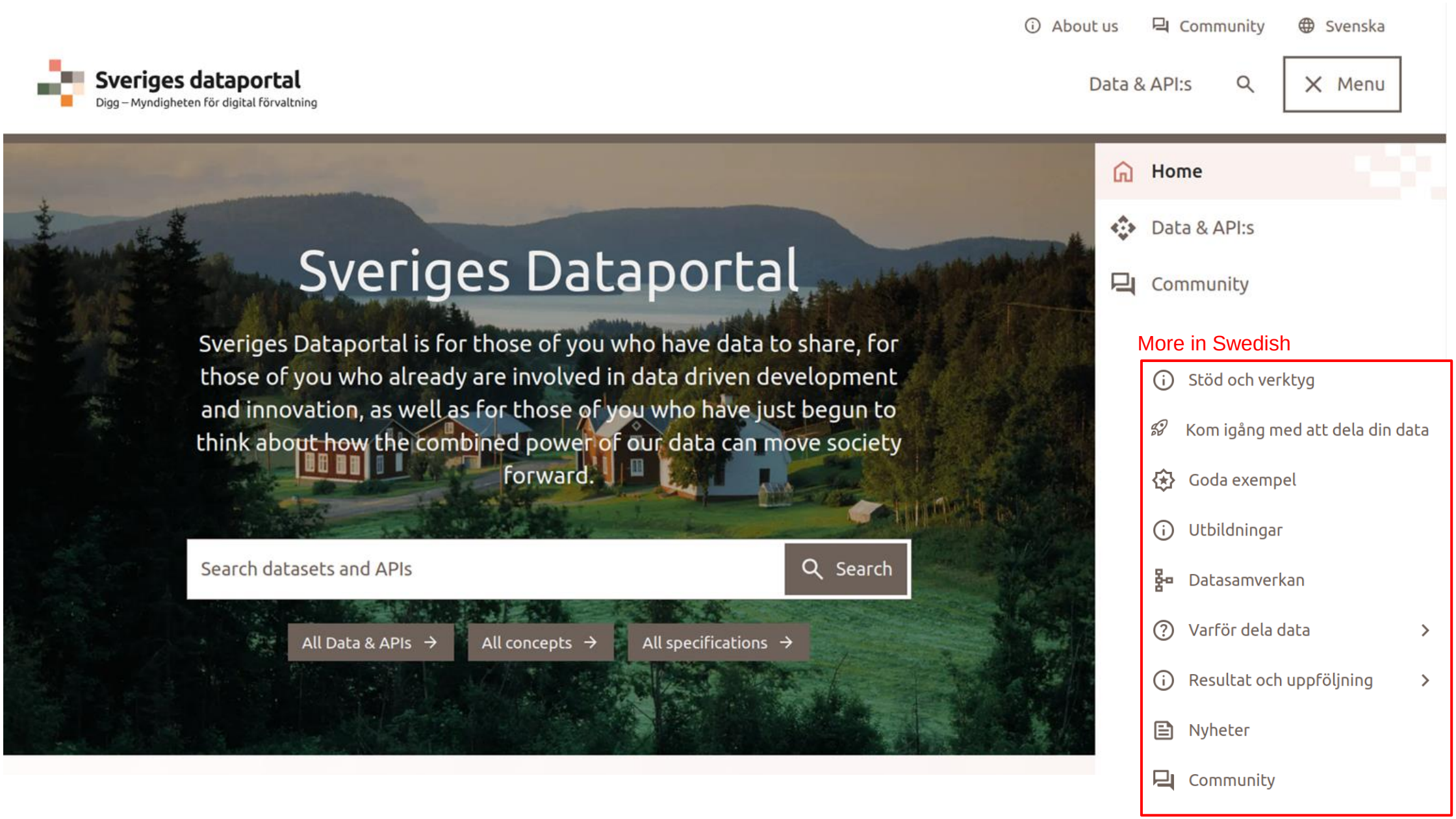
- The Swedish dataportal and the business case
- What is all the fuss about harvesting
- Implementation report
- Future issues

The Swedish dataportal

Been around since 2014

Maintained by the Swedish Agency for digital Government (Digg)

The Swedish dataportal contains the national registry for datasets (Open data Directive) as well as support for data users and data producers.


About us
Community
Svenska
Sveriges dataportal
Digg – Myndigheten för digital förvaltning
Data & API:s
Menu
Sveriges Dataportal
Sveriges Dataportal is for those of you who have data to share, for those of you who already are involved in data driven development and innovation, as well as for those of you who have just begun to think about how the combined power of our data can move society forward.
Search datasets and APIs
Search
All Data & APIs →
All concepts →
All specifications →
Home
Data & API:s
Community
More in Swedish
Stöd och verktyg
Kom igång med att dela din data
Goda exempel
Utbildningar
Datasamverkan
Varför dela data
Resultat och uppföljning
Nyheter
Community

Three ways of harvesting data catalogs

- Shared editing platform (national instance)
- Shared domain catalogs (e.g. geodata)
- Individual catalogs

Business case for exploring LDES (Digg)

Synchronization issues with data.europa.eu

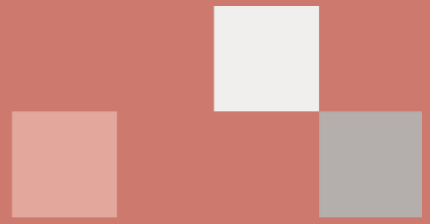
- Minimize discrepancies in dataset search
- Quicker detection of problems
- Quicker updates

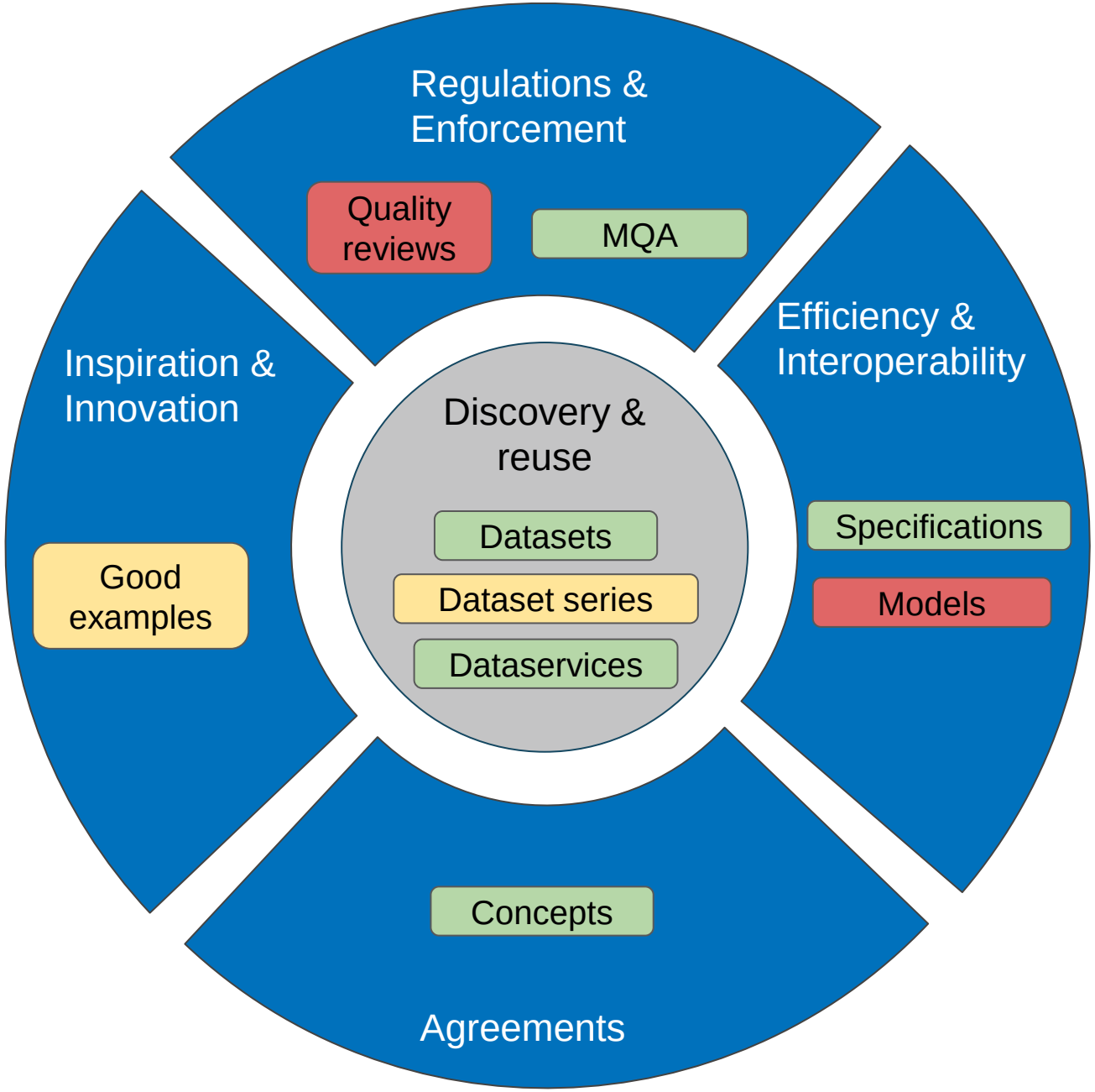
Use of standards

- Validate that the use of the harvesting protocol is followed
- Define more clearly what is needed to be harvested

Harvesting scalability

What is all the fuss about harvesting?

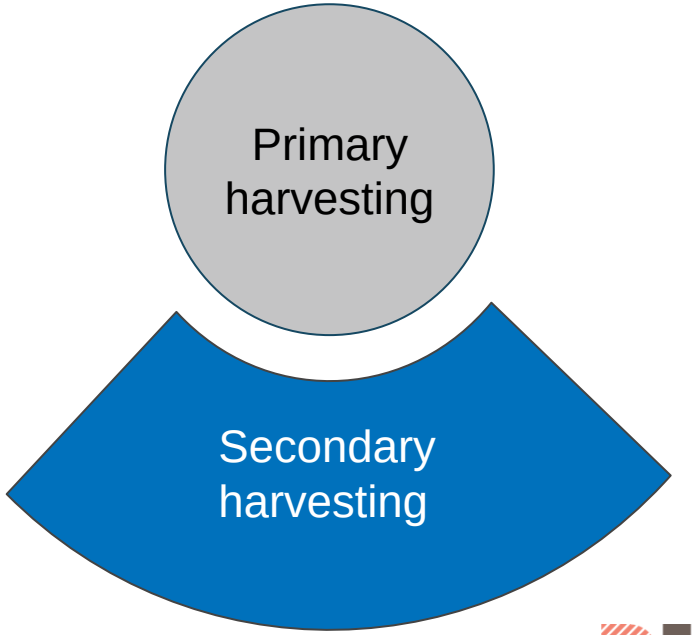




Established

Decided

Plans



Primary Harvesting (Data catalogs)

Process

- One file per source
- RDF/XML format
- Named graph extraction per main entity
- URI generation (sometimes)
- Fingerprinting metadata for detecting updates
- Validation of DCAT-AP-SE
- Harvesting report
- Notifications on errors

Issues

- DCAT-AP is a vocabulary,
- not a protocol
- RDF/XML is fragile
- Identifiers missing (URIs)
- Implicit when to update
- Scalability for large sources
- Reports missing from upstream harvesting (data.europe.eu)

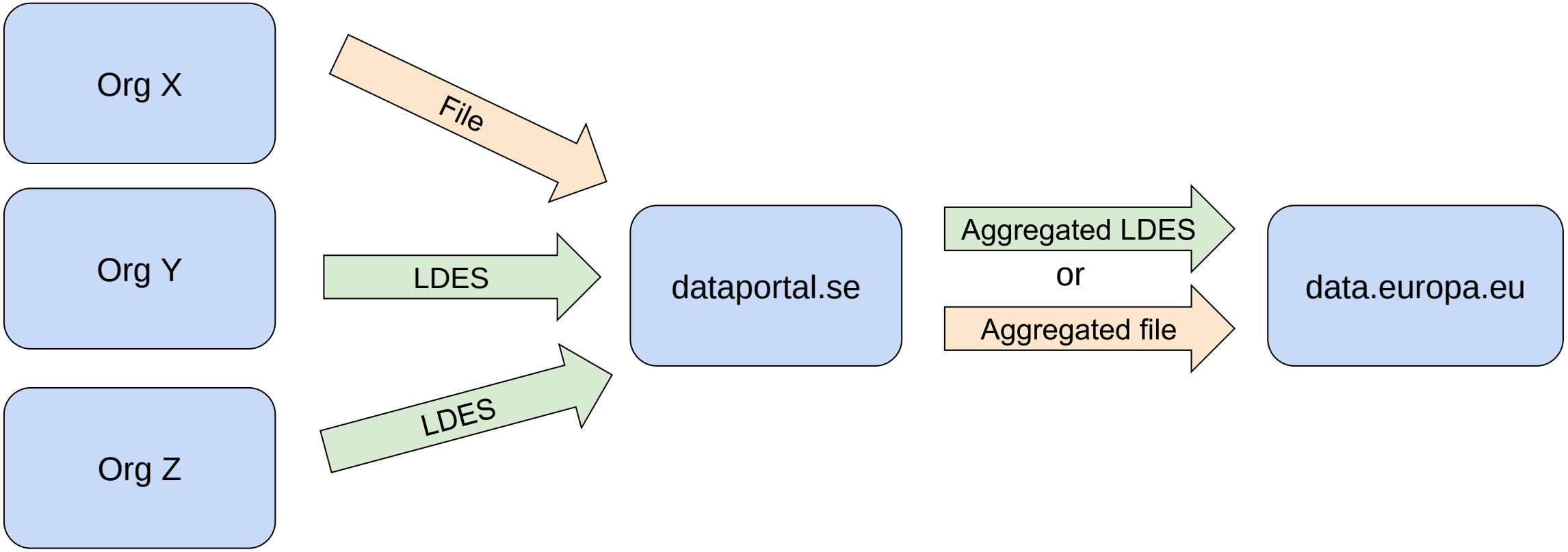
Why DCAT-AP feeds (LDES)

Some of the issues have solutions already

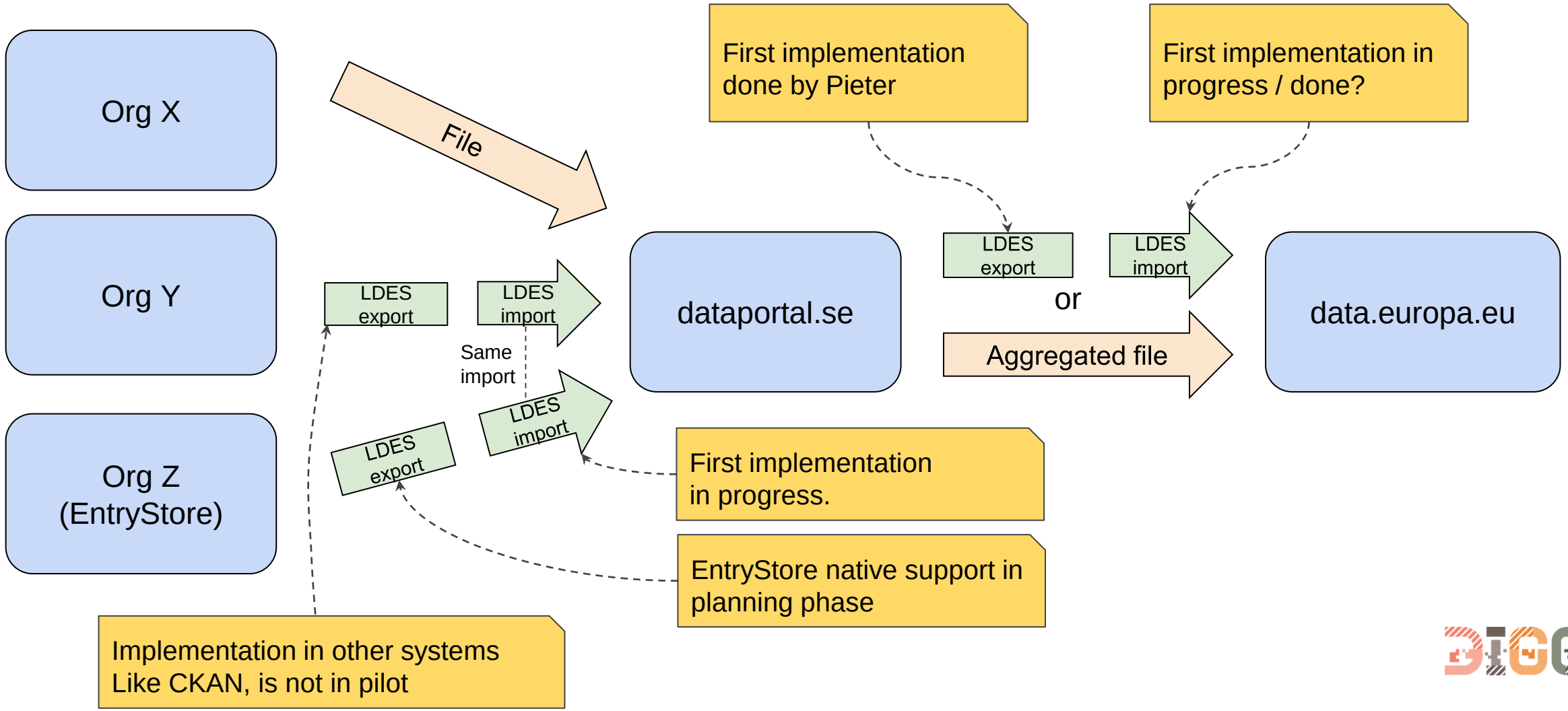
BUT

Some remain and we prefer that we solve things together and document the mechanism clearly.

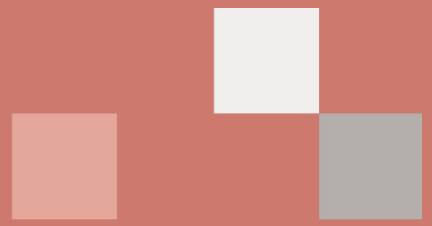
Implementation report (Sweden)



Implementation report (Sweden)



Future Issues



Main headache

No record of deletes in some systems.

Solution 1 - add support in original system

Solution 2 - do a wrapper which keeps an index

How long de we keep the records of deleted entities?

Standardized harvesting report

Agree on a information model, perhaps in RDF?

OR

Just agree on a JSON expression?

How detailed should it be?

How technical should it be, direct it towards:

- Portal providers?
- Data catalog providers?

Secondary harvesting

Do we need a complete new specification every time?

- SKOS Feed specification?
- PROF Feed specification?
- Etc.

Or, can it be "parametrized" by the entity types?

Backwards compatibility

- Continue support for file based harvesting?
- Risk of only increasing the burden of harvesting maintenance.
- Can we support an envisioned harvesting report for the file based harvesting?

Thank you

Telephone: 0046-771-11 44 00

E-mail: info@digg.se

www.digg.se

The DCAT-AP Feeds specification

Launching the first draft

Pieter Colpaert
Matthias Palmér

Timeline

Sept 2023

First encounter SEMIC-DIGG

Identified DCAT-AP harvesting as a pilot

But…

Matthias said he doesn't want to do LDES because LDES didn't support named graphs.

Oct

Named Graphs in LDES

The W3C TREE hypermedia community group added named graphs support in the specification, enabling this for LDES as well.

DIGG wants to move forward.

Nov

Official go

We created the skeleton of the specification

Dec

The Ghent meeting resulting in a first draft

Timeline

Febr 2023

A first implementation based on Sweden dumps

Mar

Putting in a lot of work

Apr

The webinar

Introducing the first official draft of DCAT-AP Feeds

Showing the first prototype implementation for Sweden

June

SEMIC2024

Launching the DCAT-AP Feeds 0.1 spec

Showing the first real implementation for Sweden

Showing implementations in other systems

TABLE OF CONTENTS

The DCAT-AP Feed specification

Living Standard, 26 April 2024

This version:

https://semiceu.github.io/LDES-DCAT-AP-feeds/index.html

Issue Tracking:

GitHub

Editors:

- Pieter Colpaert
- Matthias Palmér

Abstract

Publishing a full data dump repetitively will delegate change detection -- a fault-prone process -- to data consumers. With DCAT-AP Feeds we propose that DCAT-AP catalog maintainers publish an event source API that can help to replicate the catalog towards a harvester, and always keep it in-sync in the way that is intended by the publisher. Therefore this spec describes how to publish your DCAT-AP entity changes using the Activity Streams vocabulary and LDES. It also provides a specification for harvesters to provide transparency into their harvesting progress.

§ 1. Publishing changes about DCAT-AP entities

A DCAT-AP Feed is a Linked Data Event Stream with ActivityStream entities Create, Update and Delete in it about the DCAT-AP entities in a catalog. DCAT-AP Feeds uses the [activitystreams-vocabulary] to indicate the type of change. Three type of activities can be described:

Contents of the spec

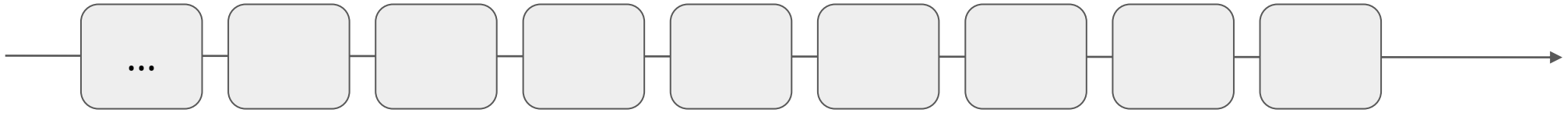
1. Telling whether something is a Create, Update or Delete
2. Stand-alone, embedded and referenced entities
3. The Linked Data Event Stream description
4. A retention policy
5. A smart but straightforward pagination of the feed
6. DCAT-AP Feeds SHACL shape for updates

We'll provide examples in both JSON-LD and TRiG (Turtle with named graphs)

```
{
  "@context": "context.jsonld",
  "@id": "#Dataset1",
  "@type": "dcat:Dataset",
}
```

```
# add prefixes
<#Dataset1> a dcat:Dataset .
```

A DCAT-AP *feed*



Every time there's an update, it should tell us how we can change our copy

We re-use the ActivityStreams vocabulary:

- `as:Create` and `as:Update` are "upserts"
- `as:Delete` indicates it has been deleted from the source

```
{
  "@id": "#DatasetEvent1",
  "@type": "Create",
  "object": "https://example.org/Dataset1",
  "published" : "2023-10-01T12:00:00Z"
}
```

```
<#DatasetEvent1> a as:Create ;
    as:object <https://example.org/Dataset1> ;
    as:published "2023-10-01T12:00:00Z"^^xsd:dateTime .
```

What triples do we upsert/remove?

We'll use **named graphs** for that

```
{
      "@id": "#Dataset1Event1",
      "@type": "Create",
      "object": "https://example.org/Dataset1",
      "published" : "2023-10-01T12:00:00Z",
      "@graph": {
          "@id": "https://example.org/Dataset1",
          "@type": "dcat:Dataset",
          …
      }
}
```

```
<#Dataset1Event1> a as:Create ;
    as:object <https://example.org/Dataset1> ;
    as:published "2023-10-01T12:00:00Z"^^xsd:dateTime .

<#Dataset1Event1>  {
    <https://example.org/Dataset1> a dcat:Dataset ;
        ## The (updated) representation of this particular
dataset
        ## ...
  }
```

Indicating these objects are part of a *feed*

⇒ an append-only log = a Linked Data Event Stream

```
{
  "@id": "#Feed",
  "@type": "EventStream",
  "title": "My DCAT-AP Feed",
  "member": [  {
          "@id": "#Dataset1Event1",
          "@type": "Create",
          "object": "https://example.org/Dataset1",
          "published" : "2023-10-01T12:00:00Z",
          "@graph": {
              "@id": "https://example.org/Dataset1",
              "@type": "dcat:Dataset",
          },
          …

  ]
}
```

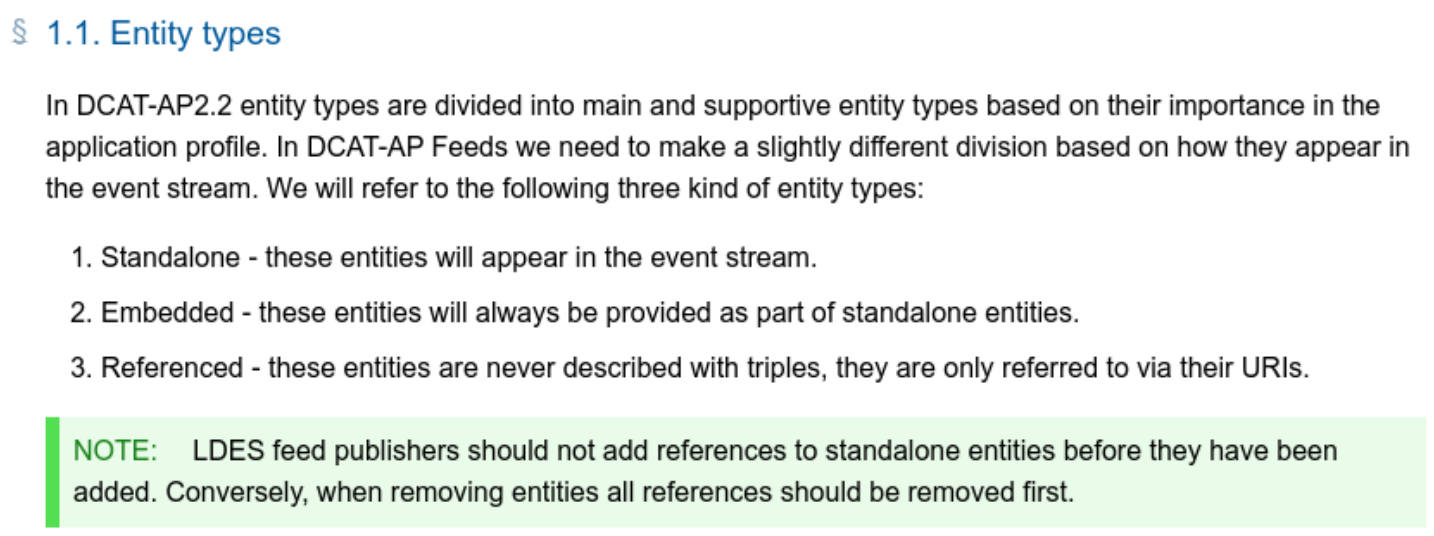
```
<#Feed> a ldes:EventStream ;
        dct:title "My DCAT-AP Feed" ;
        tree:member <https://example.org/Dataset1#Event1>
.

<#Dataset1Event1> a as:Create ;
    as:object <https://example.org/Dataset1> ;
    as:published "2023-10-01T12:00:00Z"^^xsd:dateTime .

<#Dataset1Event1>  {
    <https://example.org/Dataset1> a dcat:Dataset ;

}
```

What is part of one update?

DCAT-AP Feeds specifies a `dcat:Distribution` *should* be a **stand-alone** entity that has an IRI.

If you cannot provide an IRI, then it must appear as an **embedded** entity within the update about a `dcat:Dataset`.

Referenced entities should not be included in the feed.



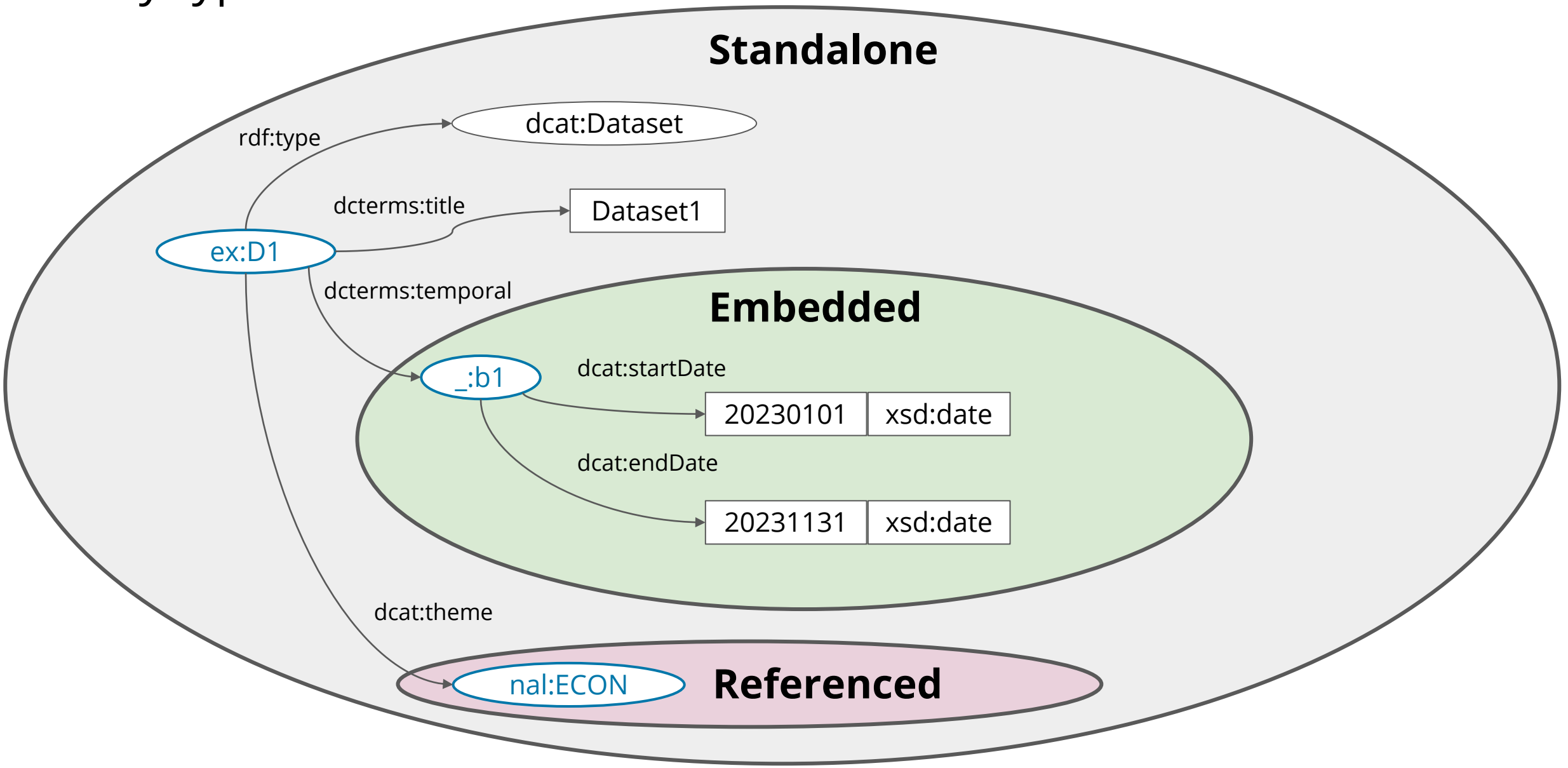
§ 1.1. Entity types

In DCAT-AP2.2 entity types are divided into main and supportive entity types based on their importance in the application profile. In DCAT-AP Feeds we need to make a slightly different division based on how they appear in the event stream. We will refer to the following three kind of entity types:

1. Standalone - these entities will appear in the event stream.
2. Embedded - these entities will always be provided as part of standalone entities.
3. Referenced - these entities are never described with triples, they are only referred to via their URIs.

NOTE: LDES feed publishers should not add references to standalone entities before they have been added. Conversely, when removing entities all references should be removed first.

Entity types



Which entities do we have?

Standalone

- dcat:Catalog
- dcat:Dataset
- dcat:Distribution
- dcat:DataService
- foaf:Agent
- vcard:Kind
- dcterms:LicenseDocument

Embedded

- spdx:Checksum
- dcterms:Location
- locn:Geometry
- dcat:Relationship
- prov:Activity
- dcat:Attribution
- spdx:ChecksumAlgorithm
- foaf:Document
- adms:Identifier

Referenced

- dcterms:LicenseDocument
- ConceptSchemes
- skos:Concept
  - frequency
  - theme
  - etc.
- dcterms:MediaType
- dcterms:PeriodOrTime
- odrl:Policy
- dcterms:ProvenanceStatement (?)
- dcterms:RightsStatement (?)
- dcat:Role
- dcterms:Standard

Adding some fields useful for LDES clients

- `timestampPath:` tells a client how the events need to be ordered
- `versionOfPath:` tells a client what the identifier is of what is represented
- `view:` must link to the current page (i.e. use a relative IRI):
  ⇒ tells the client this page is a (partial) view of the event stream

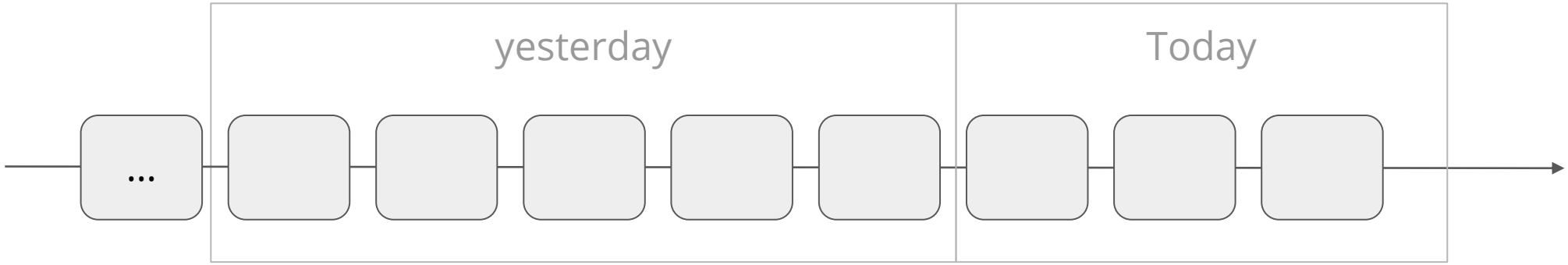
```
{
  "@id": "#Feed",
  "@type": "EventStream",
  "title": "My DCAT-AP Feed",
  "timestampPath": "published",
  "versionOfPath": "object",
  "view": {
    "@id": ""
  },
  "member": [  {
      …

 ]
}
```

```
<#Feed> a ldes:EventStream ;
        dct:title "My DCAT-AP Feed" ;
        ldes:timestampPath as:published ;
        ldes:versionOfPath as:object ;
        tree:view <> ;
        tree:member … .
```

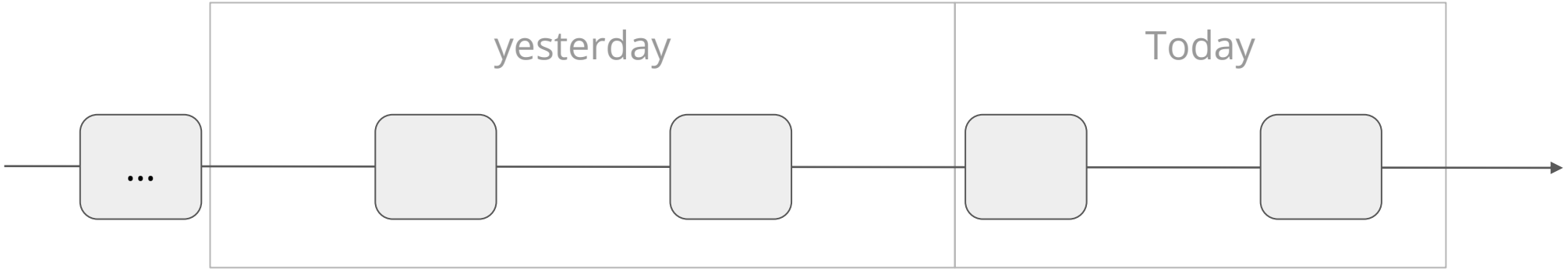
LDES supports retention policies

you can have multiple views of the same feed with different retention policies.

Full history view



Exactly the same LDES, but a view with a latest version subset only keeping the last version



= more efficient when harvesters are only interested in the latest state

Indicating we only keep the latest version

```
{
  "@id": "#Feed",
  "@type": "EventStream",
  "timestampPath": "published",
  "versionOfPath": "object",
  "view": {
    "@id": "",
    "ldes:retentionPolicy": {
      "@type": "ldes:LatestVersionSubset",
      "ldes:amount": "1"
    }
  },
  "member": [  {
      …

 ]
}
```

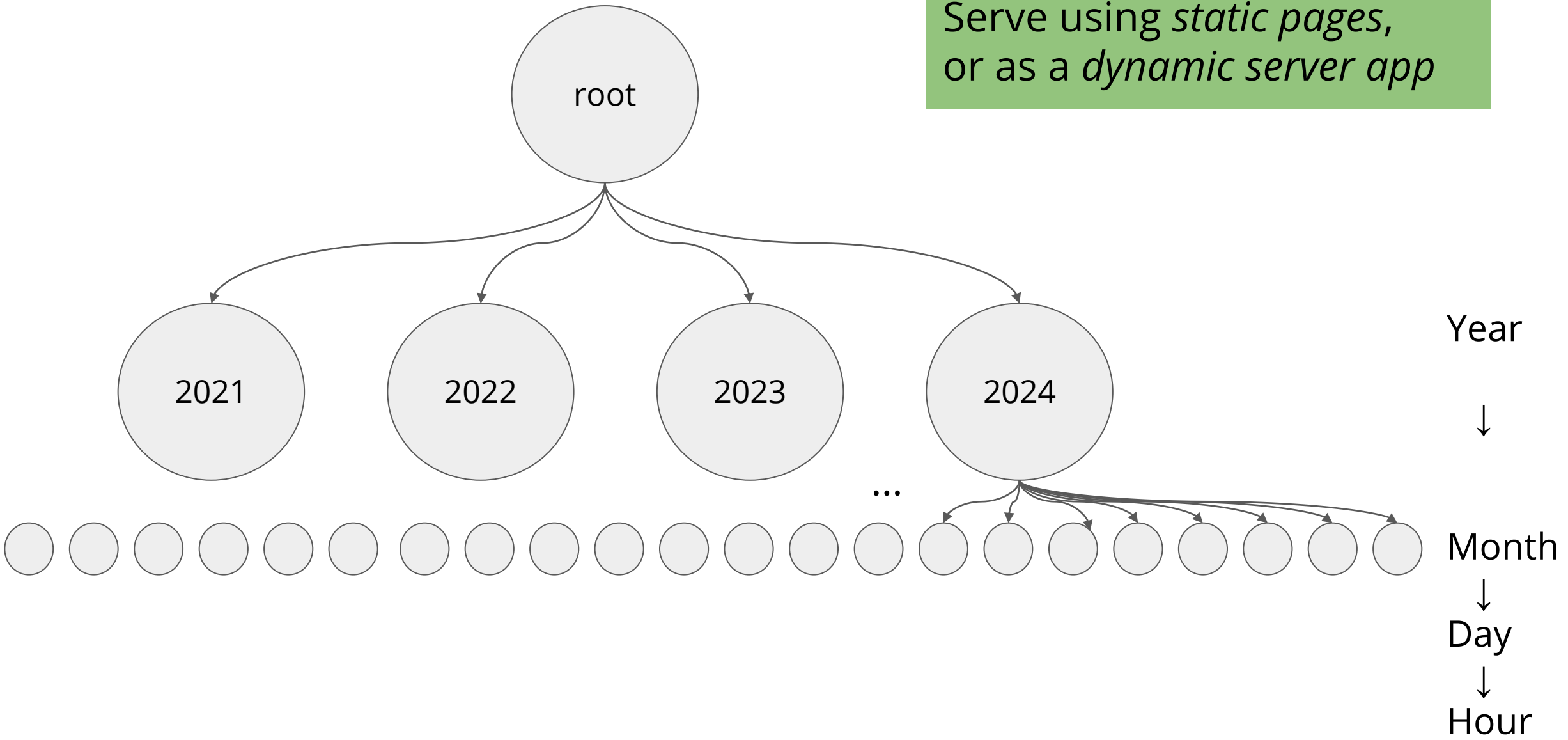
```
<#Feed> tree:view <> ;
        ldes:timestampPath as:published ;
        ldes:versionOfPath as:object .
<> ldes:retentionPolicy [
        a ldes:LatestVersionSubset ;
        ldes:amount 1

      ] .
```

Problem

dumping this in one file is efficient for first-time replication, but not for synchronization

Solution: a search tree

Light-weight
Serve using *static pages*,
or as a *dynamic server app*



Describing search trees using TREE relations

A hypermedia spec by the [W3C TREE community group]

```
{
  "@id": "#Feed",
  "@type": "EventStream",
  "timestampPath": "published",
  "versionOfPath": "object",
  "view": {
    "@id": "",
    "tree:relation": [ {
      "@type": "tree:GreaterThanOrEqualToRelation",
      "tree:path": "published",
      "tree:value": "2020-01-01T00:00:00Z",
      "tree:node": "2020.jsonld",
    },
    …
    ]
  },
  "member": [  {
      …

  ]
}
```

```
<#Feed> tree:view <> ;
<>    tree:relation [
        a tree:GreaterThanOrEqualToRelation ;
        tree:path as:published ;
        tree:value "2020-01-01T00:00:00Z"^^xsd:dateTime ;
        tree:node </2020.trig>
    ] ,
    [
        a tree:LessThanRelation ;
        tree:path as:published ;
        tree:value  "2021-01-01T00:00:00Z"^^xsd:dateTime ;
        tree:node </2020.trig>
    ]
.
```

SHACL shapes

Dedicated SHACL shapes for the DCAT-AP Feeds specification

Automatically syncs with the official DCAT-AP shapes, and extends it with how entities will appear in the feed

```
dcatapfeeds: owl:imports <https://semiceu.github.io/DCAT-AP/releases/3.0.0/html/shacl/shapes.ttl> .

dcatapfeeds:ActivityShape a sh:NodeShape ;
    sh:xone ( dcatapfeeds:UpsertShape dcatapfeeds:DeleteShape ) .

dcatapfeeds:UpsertShape a sh:NodeShape ;
    sh:closed true ;
    sh:ignoredProperties (rdf:type) ;
    sh:targetClass as:Create, as:Update ;
    sh:property [
        sh:path as:object ;
        sh:node dcatapfeeds:EntityShape ;
        sh:minCount 1 ;
        sh:maxCount 1 ;
    ] ;
    sh:property dcatapfeeds:PublishedPropertyShape .

dcatapfeeds:DeleteShape a sh:NodeShape ;
    sh:closed true ;
    sh:ignoredProperties (rdf:type) ;
    sh:targetClass as:Delete ;
    sh:property [
```

What we learned

Named graphs are an elegant addition to the LDES spec

More input for the LDES spec itself
A retention policy specifically for deletions should be added in the LDES spec: how long do you want to keep removals? Maybe also implicit removals need to be supported?

Domain specific primers?
Write primers like this for other domains as well, such as for Cultural Heritage

But most importantly

The DCAT-AP Feeds specification is now ready for your comments and implementations

Happy publishing!

The pilot goes on: DCAT-AP feeds @ Data.Europa.EU

Publications Office
of the European Union

DATA.EUROPA.EU - LINKED DATA EVENT STREAM

Dr. Simon Steuer,

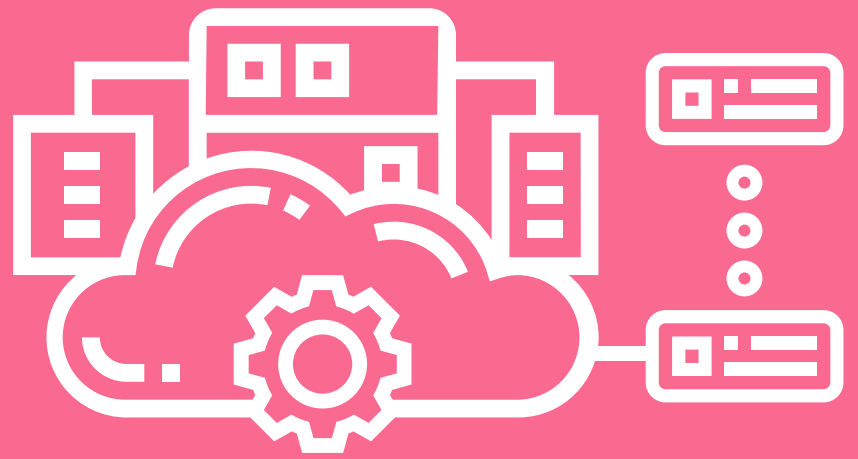
Head of Sector, Publications Office of the EU

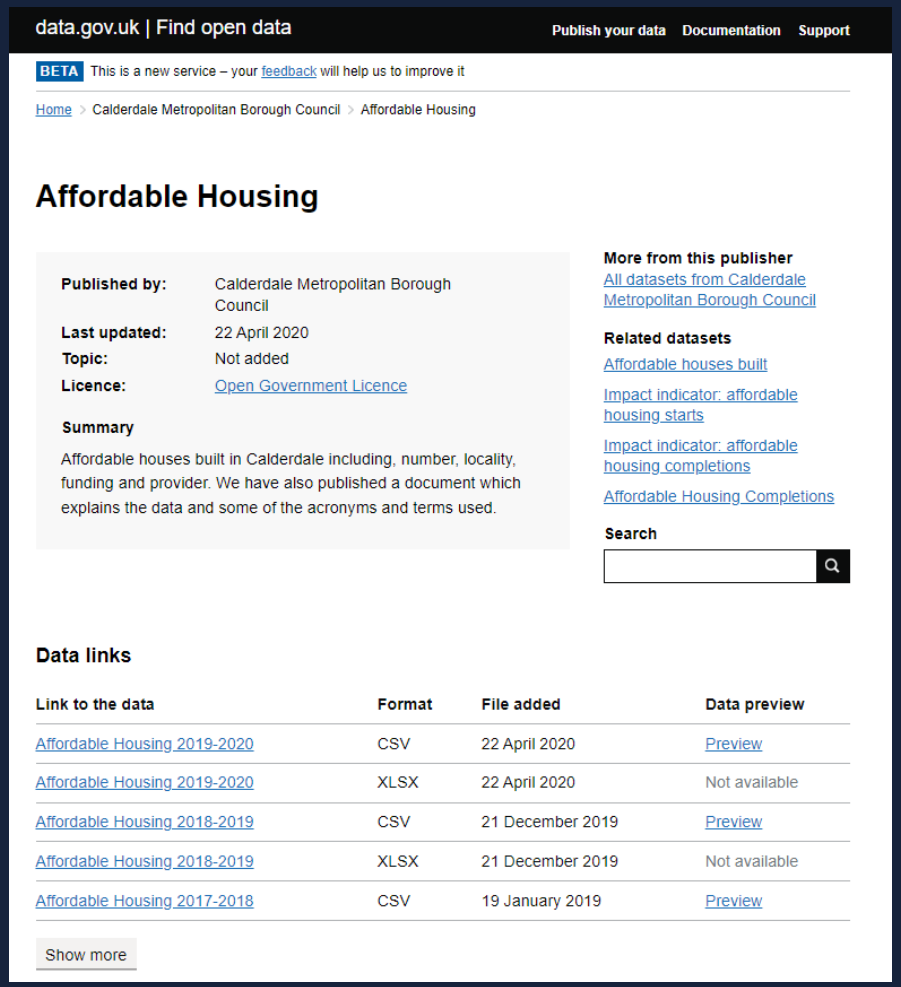
30 April 2024

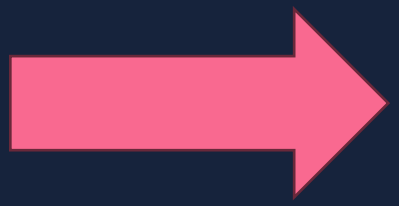
Publications Office
of the European Union

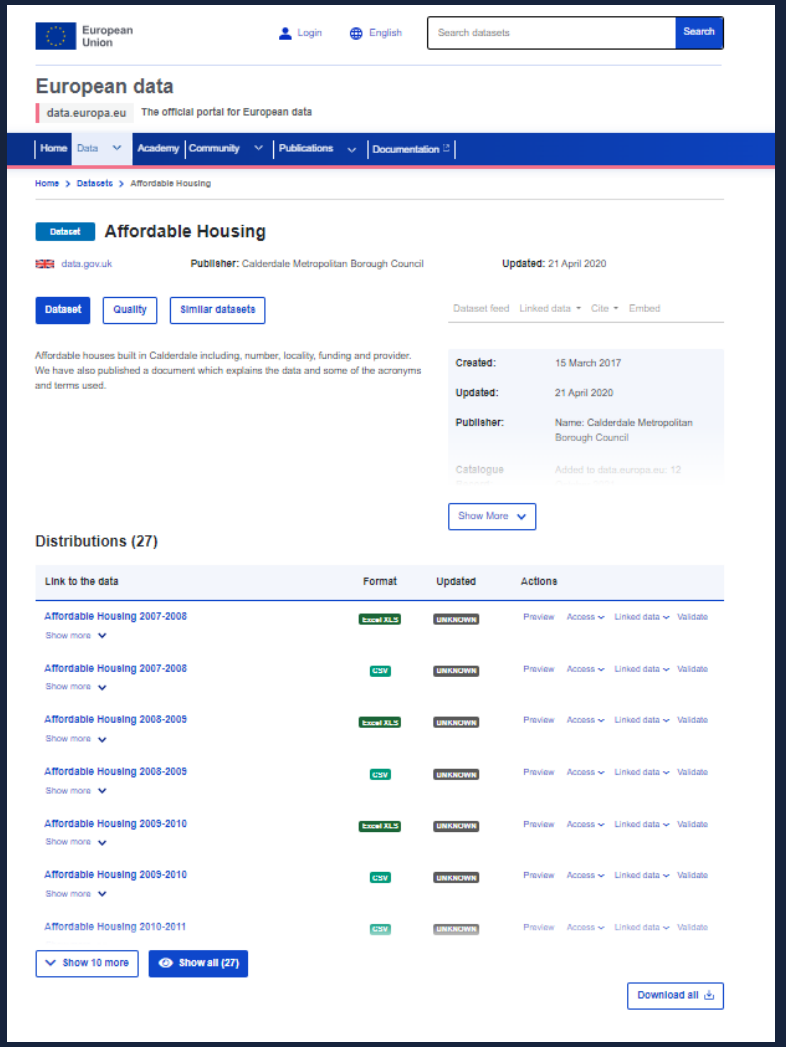
CURRENT DATA HARVESTING PROCESSES

183 data catalogues on data.europa.eu




data.gov.uk | Find open data
Publish your data
Documentation
Support
BETA This is a new service – your feedback will help us to improve it
Home > Calderdale Metropolitan Borough Council > Affordable Housing
Affordable Housing
Published by: Calderdale Metropolitan Borough Council
Last updated: 22 April 2020
Topic: Not added
Licence: Open Government Licence
Summary
Affordable houses built in Calderdale including, number, locality, funding and provider. We have also published a document which explains the data and some of the acronyms and terms used.
More from this publisher
All datasets from Calderdale Metropolitan Borough Council
Related datasets
Affordable houses built
Impact indicator: affordable housing starts
Impact indicator: affordable housing completions
Affordable Housing Completions
Search
Data links
Link to the data
Format
File added
Data preview
Affordable Housing 2019-2020
CSV
22 April 2020
Preview
Affordable Housing 2019-2020
XLSX
22 April 2020
Not available
Affordable Housing 2018-2019
CSV
21 December 2019
Preview
Affordable Housing 2018-2019
XLSX
21 December 2019
Not available
Affordable Housing 2017-2018
CSV
19 January 2019
Preview
Show more




European Union
Login
English
Search datasets
Search
European data
data.europa.eu
The official portal for European data
Home
Data
Academy
Community
Publications
Documentation
Home > Datasets > Affordable Housing
Dataset
Affordable Housing
data.gov.uk
Publisher: Calderdale Metropolitan Borough Council
Updated: 21 April 2020
Dataset
Quality
Similar datasets
Affordable houses built in Calderdale including, number, locality, funding and provider. We have also published a document which explains the data and some of the acronyms and terms used.
Created: 15 March 2017
Updated: 21 April 2020
Publisher: Name: Calderdale Metropolitan Borough Council
Show More
Distributions (27)
Link to the data
Format
Updated
Actions
Affordable Housing 2007-2008
Affordable Housing 2008-2009
Affordable Housing 2009-2010
Affordable Housing 2010-2011
Show 10 more
Show all (27)
Download all

Metadata retrieval

data.europa.eu

Pull (harvesting)

Push (data providers interface form, API)

Portal provider

- EU institutions, agencies and bodies
- European countries
- Projects

Data acquisition

Scheduler

- Main entry point for the service orchestration.
- Periodically triggers the harvesting process, defined as a **pipeline descriptor.**
- Frequency: hourly, daily, weekly… depends on data provider

Importer

- Retrieves the metadata from the source portal(s).
- Support for a variety of interfaces and data formats:
- **Responsive API that provides DCAT-AP and supports paging is preferred (e.g. RDF / XML)**

Transformer

- Applies lightweight scripting-based transformation rules.
- Rules are written in JavaScript or XSLT.
- The final output is “DCAT-AP-compliant” RDF.
- The scripts can be managed externally (e.g. in Git) to ensure maintainability.

Processing and storing

Registry

- Middleware and abstraction layer to interact with the triple store (**Virtuoso**).
- RESTful interface for RDF (**Turtle, JSON-LD, N-Triples, RDF/XML, Notation3**).
- Application of URI schemata, generation of unique IDs and inter-linking.

Indexing

- Responsible for managing the high-performance search index (**Elasticsearch**).
- “Flattening” of the DCAT RDF to simple JSON.
- Extracting literals from the data, e.g. from properties like title and description.
- Supports the use of existing and **vocabularies and ontologies**.

Translation

- Middleware to eTranslation
- Bundling literals from multiple datasets to an integrated request.
- Returns the translation by applying the native multi-language features of RDF.
- Translates description and title from datasets and distributions.

Quality evaluation

Validator

- Application of the **W3C SHACL**.
- Results include detailed information violations.
- Applied rules can also be extended or replaced (Built-in **DCAT-AP**).
- Accessibility tests on each linked distribution (the actual data).

Annotator

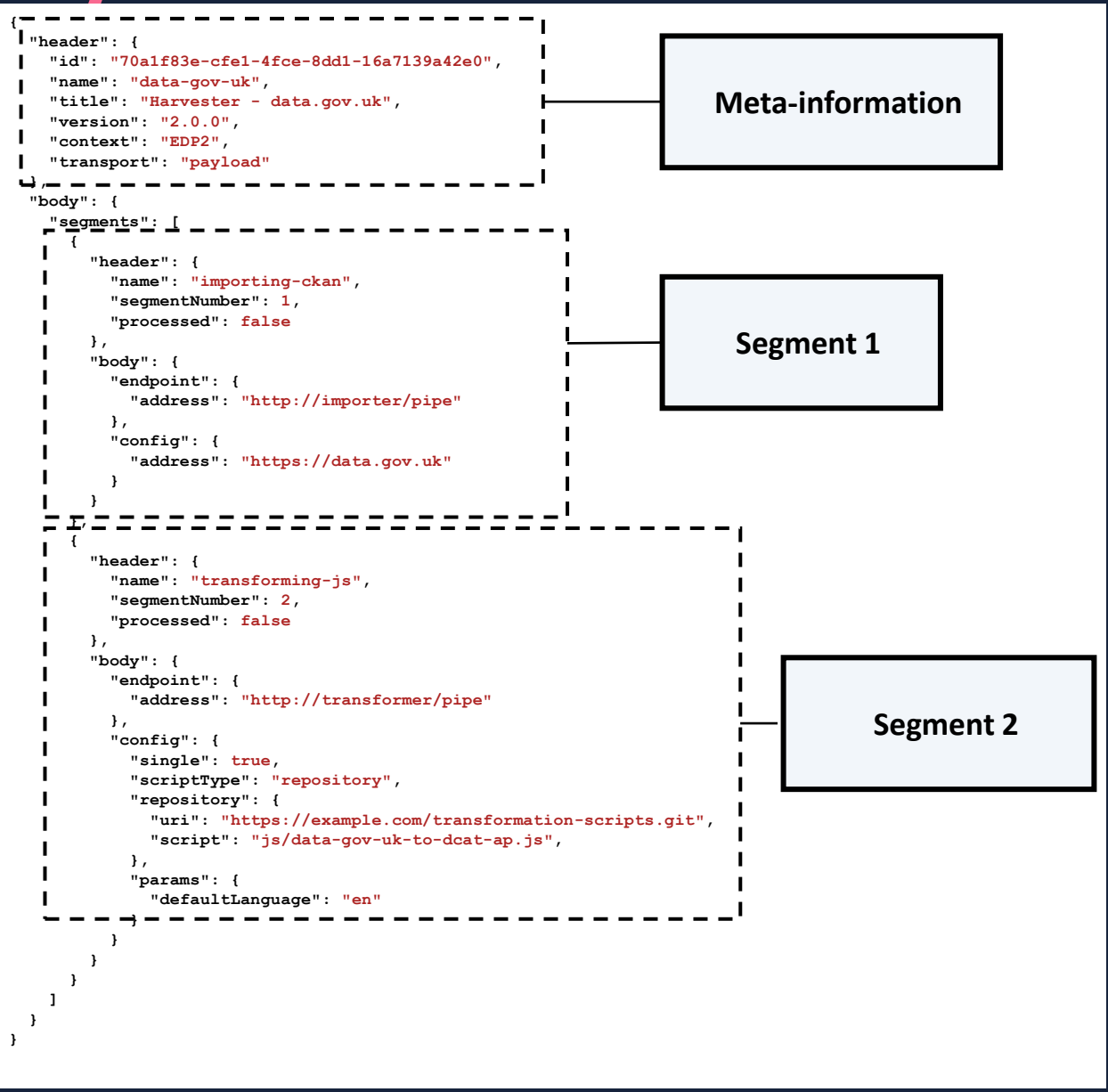
- Quality assessment for each dataset with a custom metrics scheme.
- Inspired by the **FAIR** principles.
- Completeness of the metadata, evaluating the format and type of data, availability of licensing information and linked distributions.

Reporter

- Applies W3C Data Quality Vocabulary (**DQV**) for creating quality reports.
- Based on the results of the Validator and Annotator.
- Attached as RDF to the concerned dataset in the triplestore.
- Offers a variety of human-readable versions (PDF, XLS, ODS, HTML).

Identifier handling

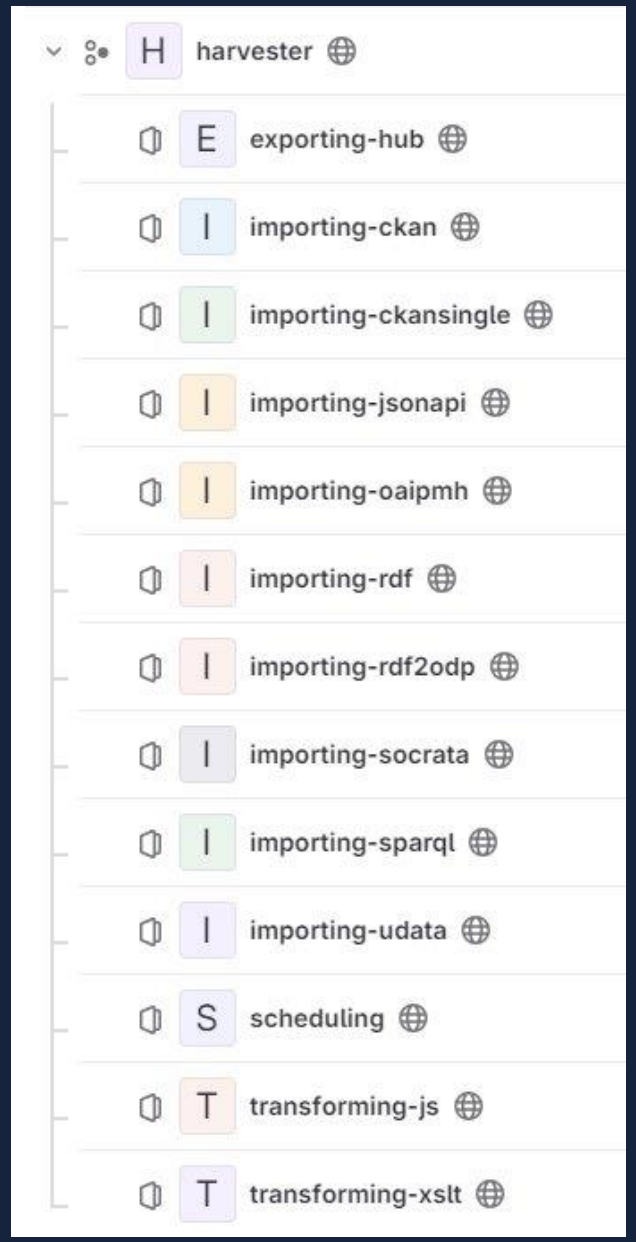
- When harvesting, we always store the original identifier in *dct:identifier*
- For internal handling, we create an additional identifier
- A URIRef based on the original identifier (our baseURI + “normalised” identifier)
- If the new identifier already exists, we add an increment at the end

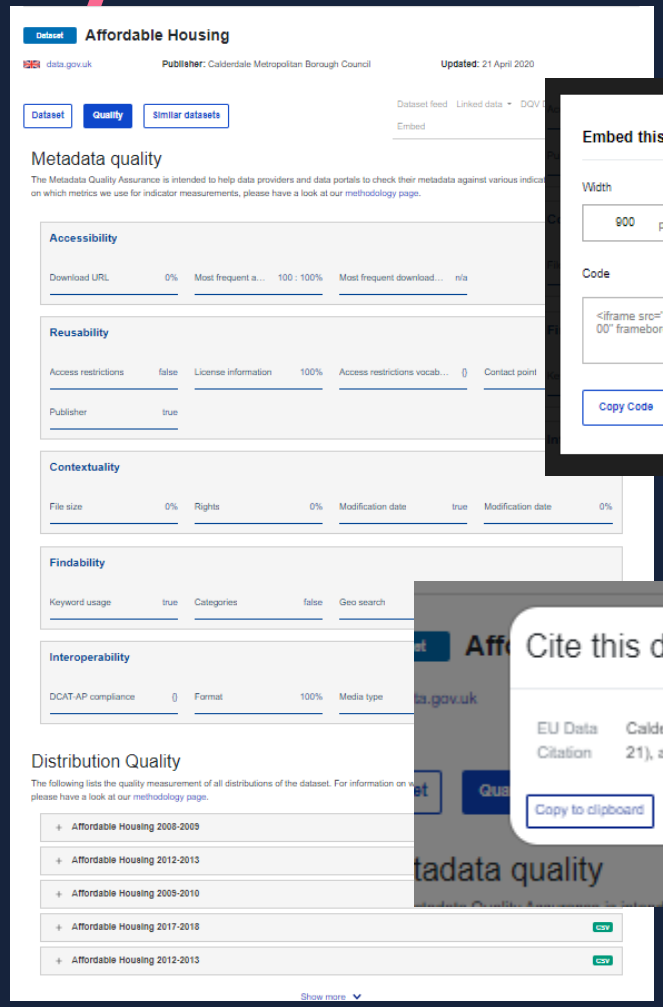


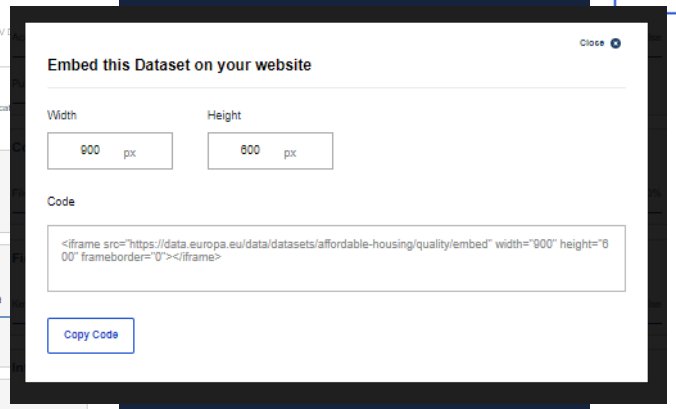
- A pipeline orchestration is described by a **descriptor**: a plain JSON document.
- It includes list of **segments**, where each segment describes a step aka a service.
- The descriptor is a compilation and self-contained description of a data processing chain.
- Each microservice must expose an endpoint to receive the descriptor and must be able to parse and execute its content.
- Data itself can be embedded directly into the descriptor or passed via a pointer to a separate data store.

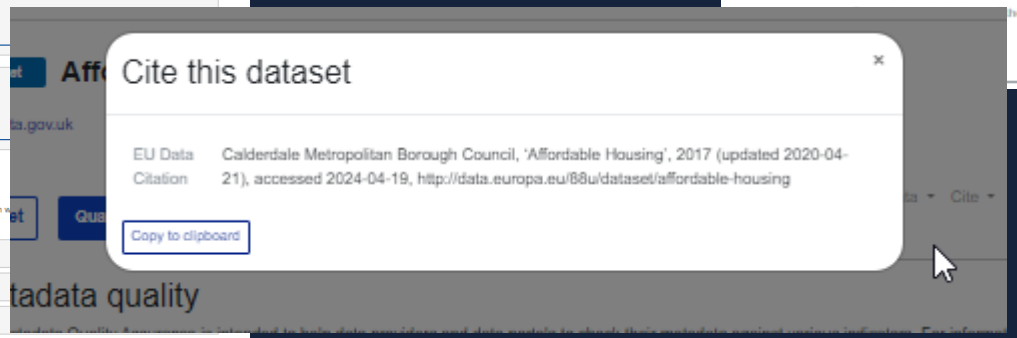
Software stack

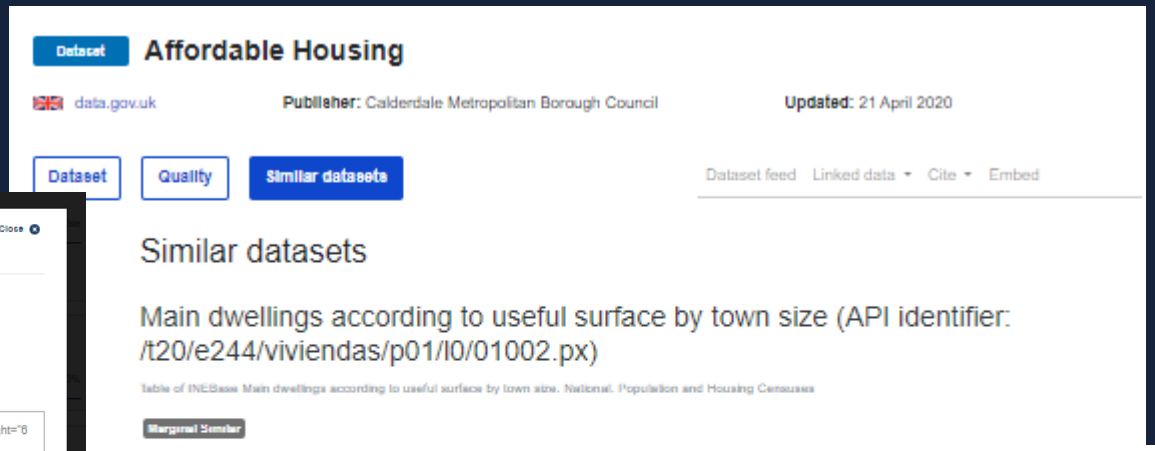
- Reactive Java framework **Vert.x** and employment of an asynchronous programming paradigm
- DevOps-based **Microservice** approach
- Deployment via **Docker** and support for container-orchestration like **Kubernetes**
- Virtuoso triple store a primary database and **Elasticsearch** as search server
- Modern Single-Page-Application frontend based on **Vue.js**

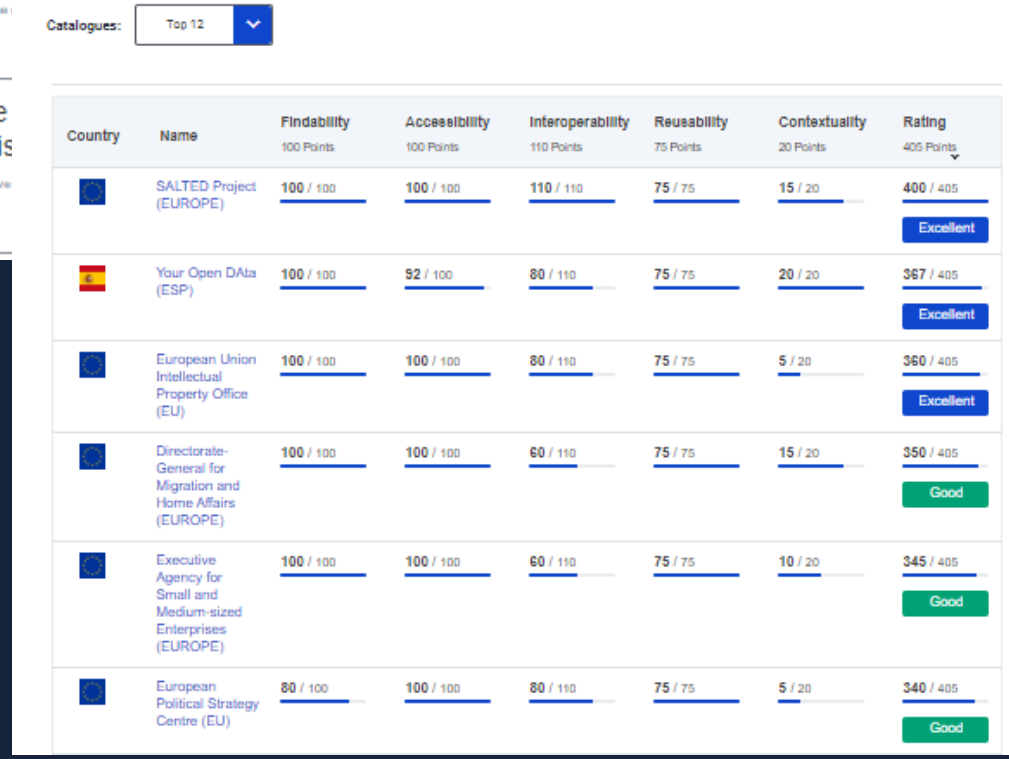



Dataset
Affordable Housing
data.gov.uk
Publisher: Calderdale Metropolitan Borough Council
Updated: 21 April 2020
Dataset
Quality
Similar datasets
Metadata quality
Accessibility
Reusability
Contextuality
Findability
Interoperability
Distribution Quality
Affordable Housing 2008-2009
Affordable Housing 2012-2013
Affordable Housing 2009-2010
Affordable Housing 2017-2018
Affordable Housing 2012-2013
Show more


Embed this Dataset on your website
Close
Width
900 px
Height
600 px
Code
<iframe src="https://data.europa.eu/data/datasets/affordable-housing/quality/embed" width="900" height="600" frameborder="0"></iframe>
Copy Code


Cite this dataset
EU Data Citation
Calderdale Metropolitan Borough Council, 'Affordable Housing', 2017 (updated 2020-04-21), accessed 2024-04-19, http://data.europa.eu/88u/dataset/affordable-housing
Copy to clipboard

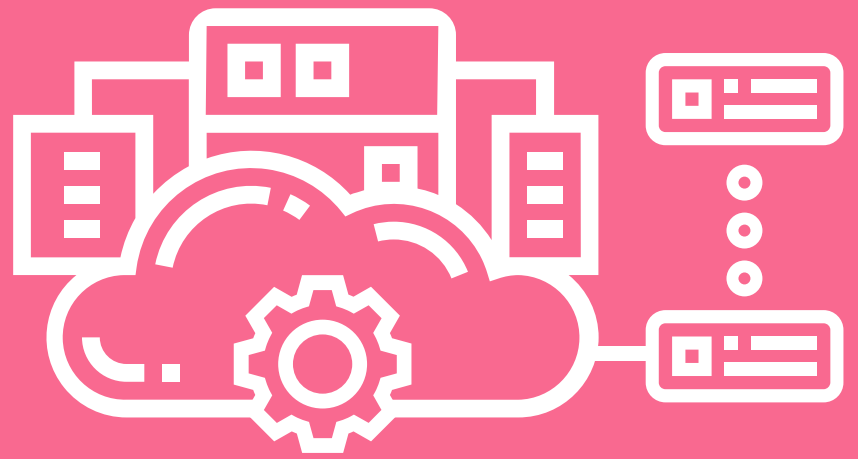

Dataset
Affordable Housing
data.gov.uk
Publisher: Calderdale Metropolitan Borough Council
Updated: 21 April 2020
Dataset
Quality
Similar datasets
Dataset feed
Linked data
Cite
Embed
Similar datasets
Main dwellings according to useful surface by town size (API identifier: /t20/e244/viviendas/p01/l0/01002.px)


Overview
Catalogues:
Top 12
Country
Name
Findability
100 Points
Accessibility
100 Points
Interoperability
110 Points
Reusability
75 Points
Contextuality
20 Points
Rating
405 Points
SALTED Project (EUROPE)
100 / 100
100 / 100
110 / 110
75 / 75
15 / 20
400 / 405
Excellent
Your Open DAta (ESP)
100 / 100
92 / 100
80 / 110
75 / 75
20 / 20
367 / 405
Excellent
European Union Intellectual Property Office (EU)
100 / 100
100 / 100
80 / 110
75 / 75
5 / 20
360 / 405
Excellent
Directorate-General for Migration and Home Affairs (EUROPE)
100 / 100
100 / 100
60 / 110
75 / 75
15 / 20
350 / 405
Good
Executive Agency for Small and Medium-sized Enterprises (EUROPE)
100 / 100
100 / 100
60 / 110
75 / 75
10 / 20
345 / 405
Good
European Political Strategy Centre (EU)
80 / 100
100 / 100
80 / 110
75 / 75
5 / 20
340 / 405
Good

Publications Office
of the European Union

FUTURE DATA HARVESTING PROCESSES

183 data catalogues on data.europa.eu



Future data harvesting process

- Add one more step to check for LDES metadata
- Reduce the harvesting to the new, updated and deleted datasets only
- Reduce the compute workload dramatically
- Offer more details about history of datasets
- First tests are ongoing for data.europa.eu with the help of DIGIT
- There will be more advantages than we now think of

THANK YOU

op-data-europa-eu@publications.europa.eu

Publications Office
of the European Union

LDES @ Rijksmuseum

Linked Data Event Streams at RIJKS MUSEUM

Tim Thomassen
t.thomassen@rijksmuseum.nl

30 April 2024

Rijksmuseum

National museum of the Netherlands

Collection

~ 1.000.000 objects

~ 450.000 books

~ 800 meters of documentation

~ 17 terabyte of research data

Library System

Collection Management System

Document Management System

RIJKS MUSEUM

Team

Agile methodology fast development cycles

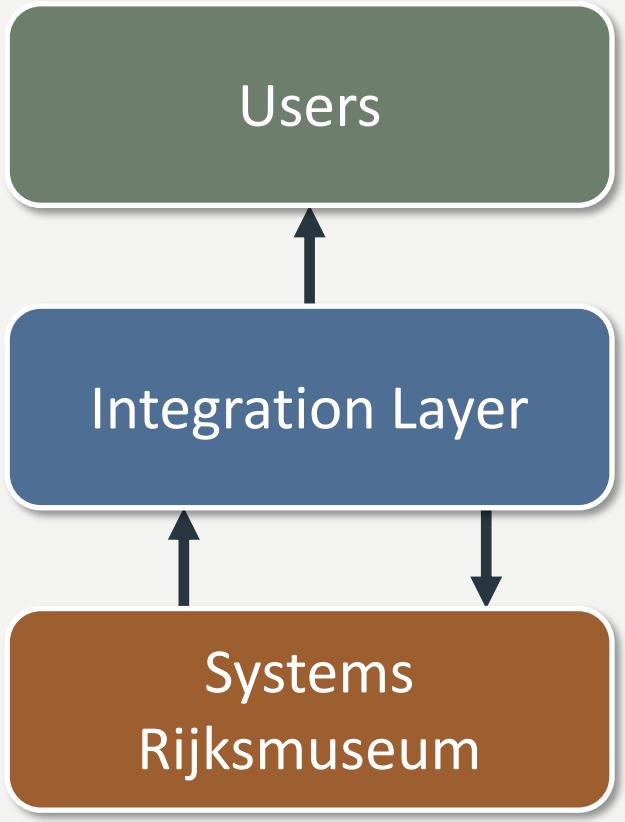
Main focus of work data, code, infrastructure

Role	Appointment
Architect	1.6 fte Consultant
Data Engineer	1 fte Rijksmuseum, 0.4 fte Consultant
DevOps Engineer	1.9 fte Consultant

Integration Layer

Infrastructure that **connects systems** and makes **data accessible**

Integration	Connect data from different domains
Standardisation	**Create predictable data services**
Validation	Guarantee data quality
Synchronisation	Keep data up-to-date

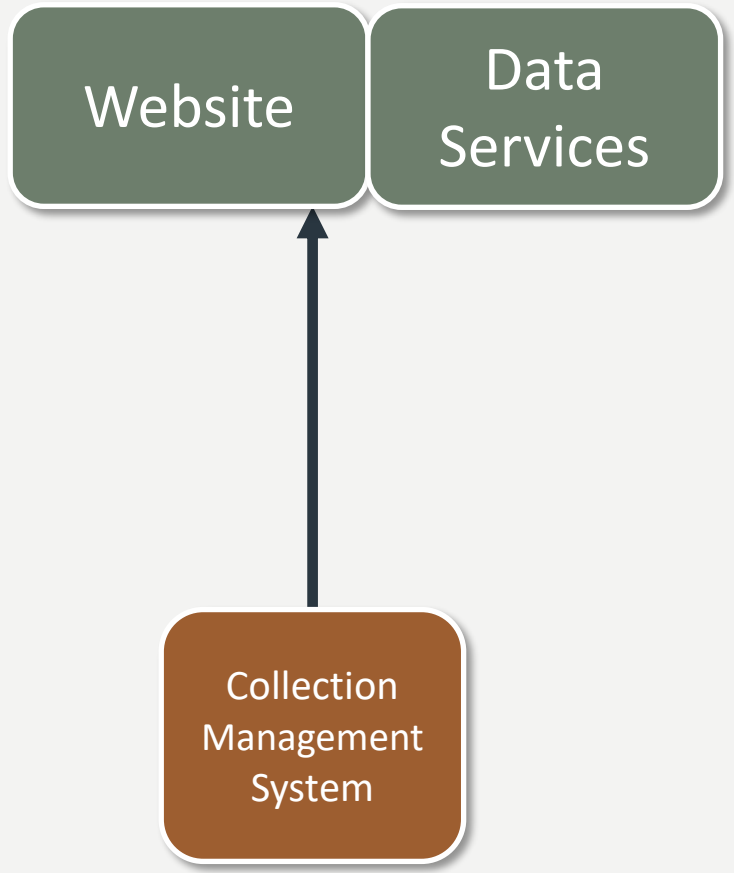


RIJKS MUSEUM

Standardisation current situation

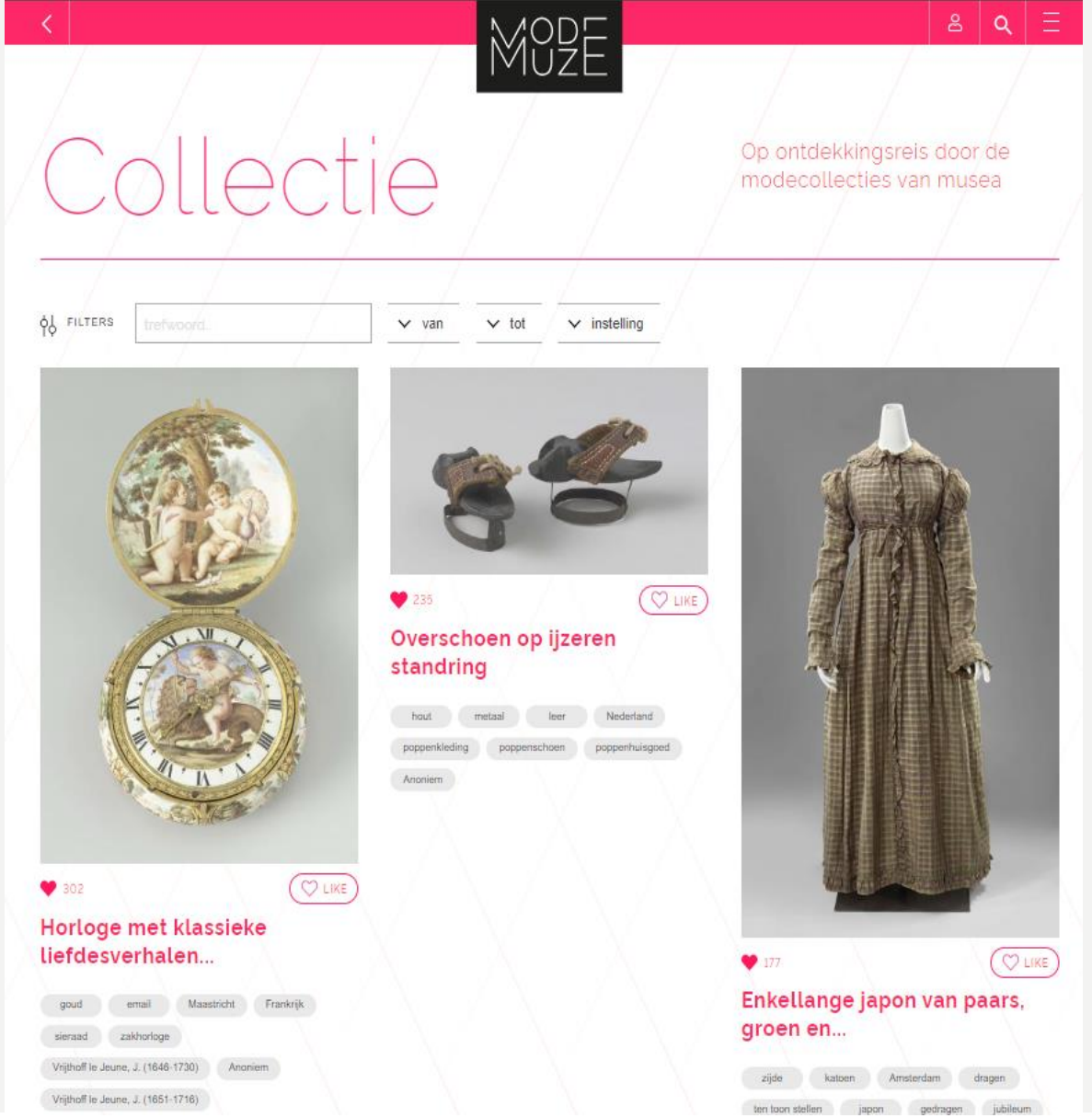
System specific data structures and communication protocols

Data Services managed by **external web design company**



Aggregators

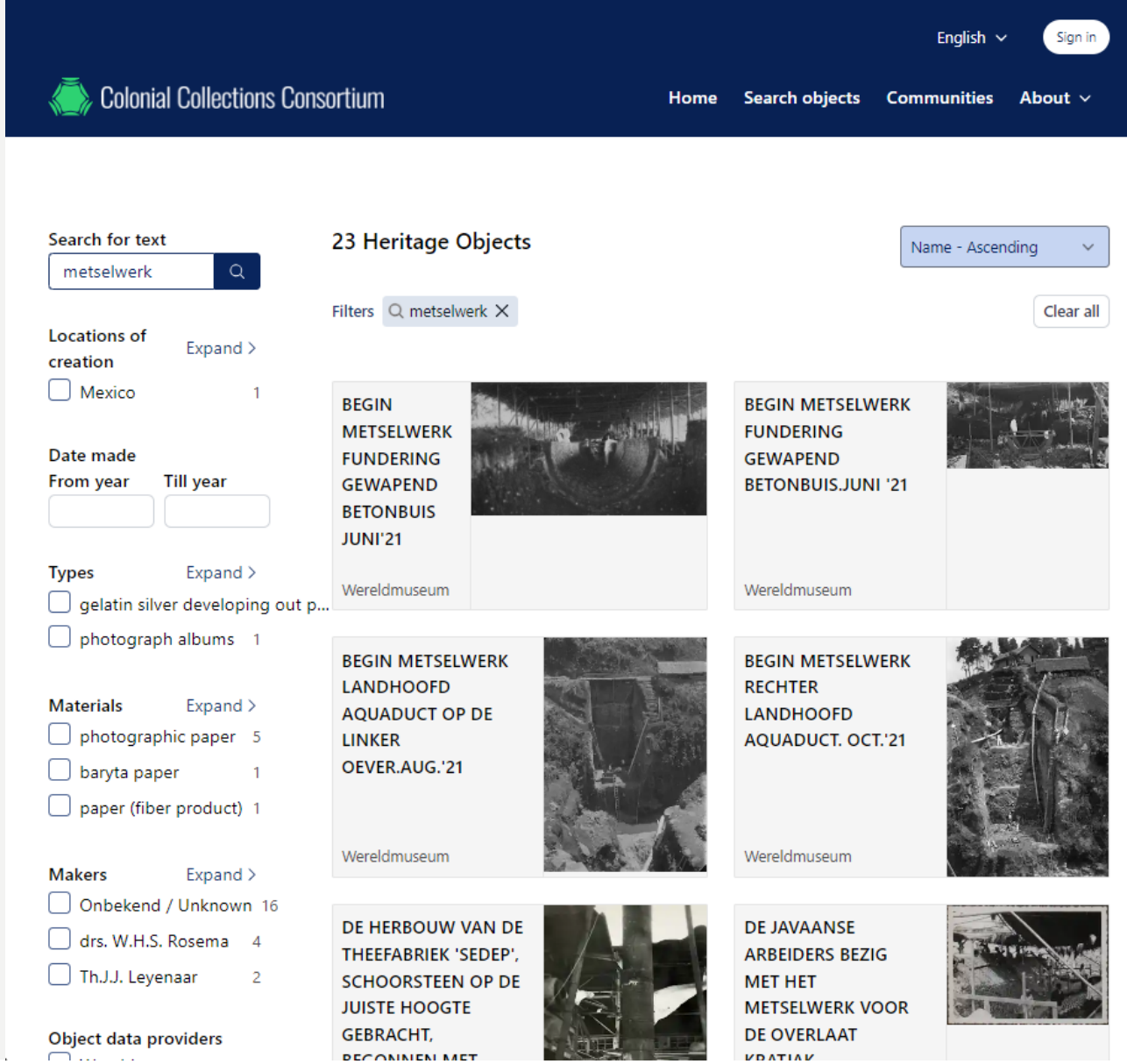
- Modemuze (OAI-PMH)
- Europeana (OAI-PMH)
- KVVAK (JSON)



RIJKS MUSEUM

Against Opacity Datahub

- Goal: Datahub for Colonial Heritage
- Use case for LDES



RIJKS MUSEUM

Standardisation future

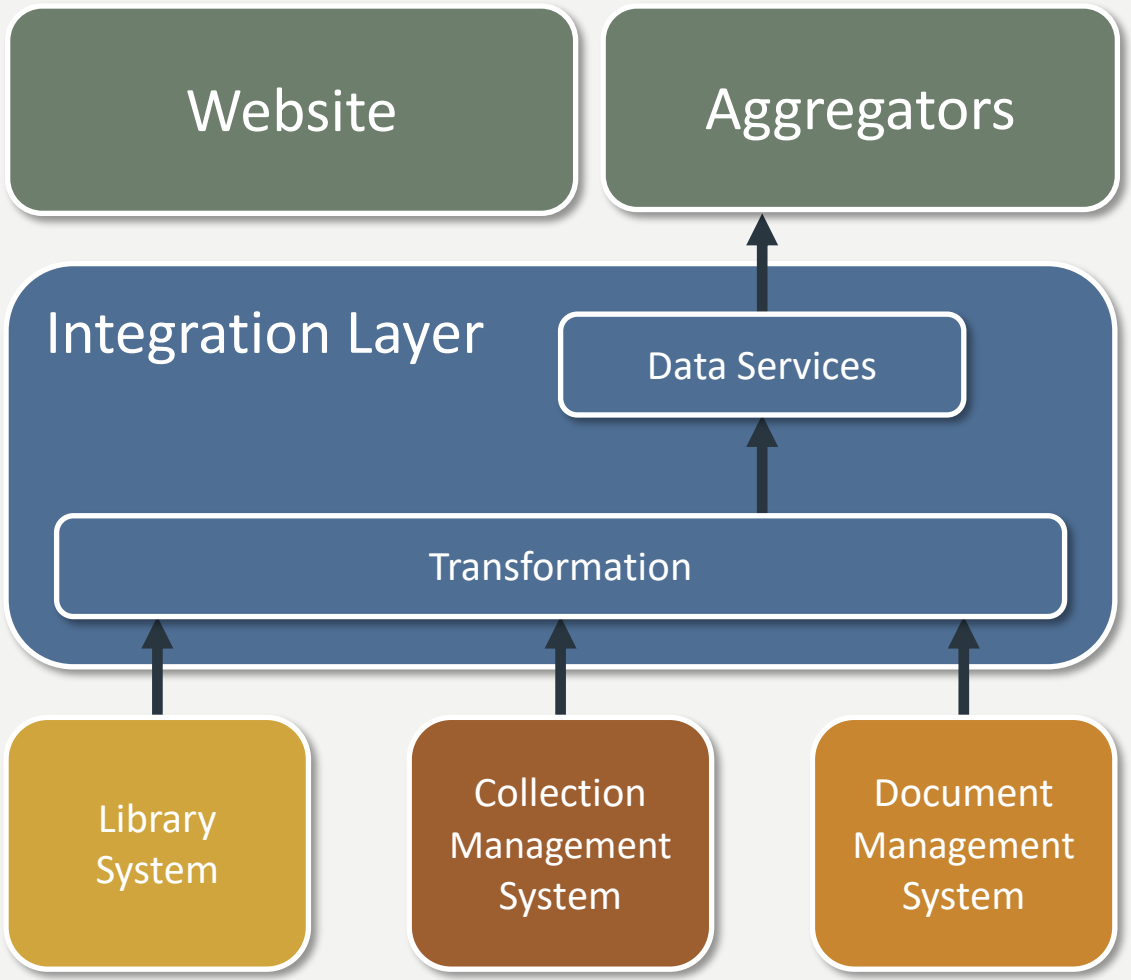
System specific data structure and communication protocols

Data Services managed by external web design company

Usable due to **following standards**

- Standardised protocols (e.g. LDES)
- Standardised data structures (e.g. Linked Art)

The Rijksmuseum **takes responsibility** for infrastructure



Infrastructure as Code how

Infrastructure as Code

servers integration layer are defined as code

Azure Cloud

servers managed by Microsoft

Continuous Deployment

changes are quickly and easily deployed on servers

RIJKS MUSEUM

Microservices Architecture how

Microservices

software split into small parts

Docker Containers

software packaged so it can be easily deployed on servers

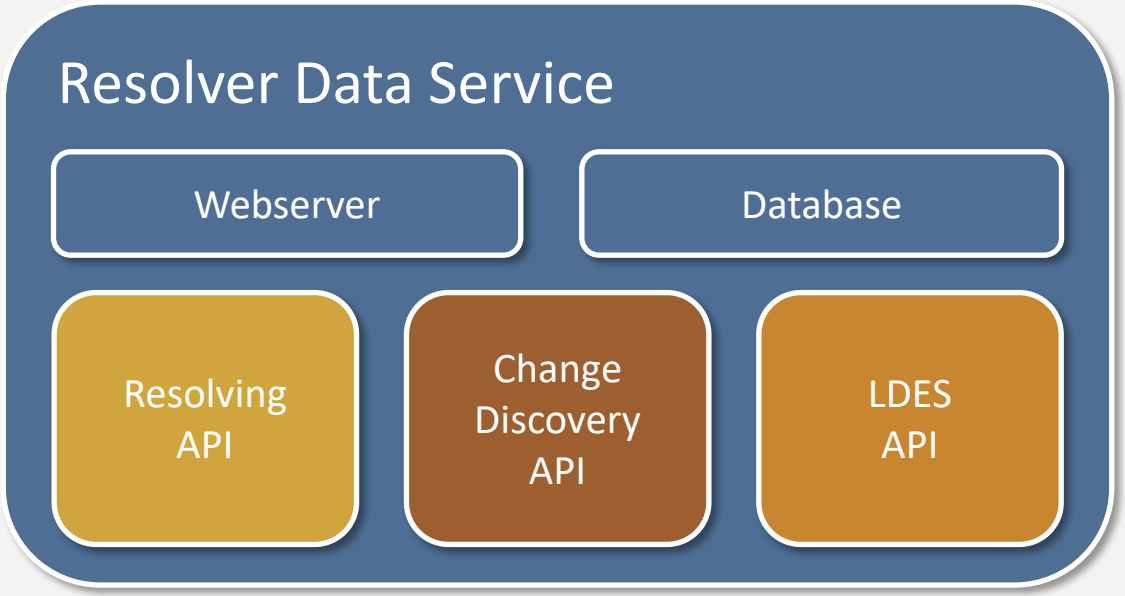
Kubernetes Cluster

establish relations between parts, improve reliability and make software scalable

RIJKS MUSEUM

Resolver Data Service

- Common setup
  - Webserver
  - Database
- Multiple API configurations




```python
# Resolving API
resolving_api = ResolvingAPI(engine, config)
app.register_blueprint(resolving_api.blueprint)

# Change Discovery API
cd_api_prefix = '/cd'
cd_api = ChangeDiscoveryAPI(engine, config, cd_api_prefix)
app.register_blueprint(cd_api.blueprint, url_prefix=cd_api_prefix)

# LDES API
ldes_api_prefix = '/ldes'
ldes_api = LDESAPI(engine, config, ldes_api_prefix)
app.register_blueprint(ldes_api.blueprint, url_prefix=ldes_api_prefix)
```

Demo

RIJKS MUSEUM

RIJKS MUSEUM

LDES Demo

The demonstrator(s)

Harvesting LDESs with DCAT-AP and cultural heritage data using the same toolchain

Pieter Colpaert

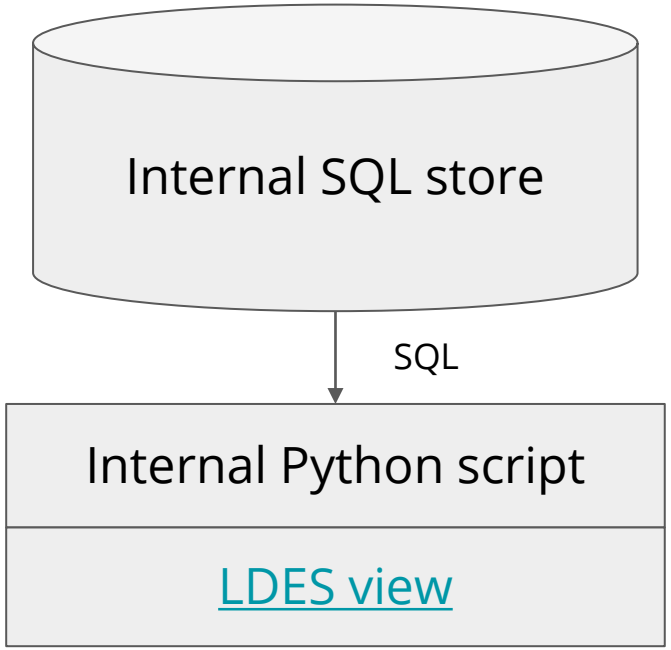
Content

1. **The Sweden publication PoC**
   Implementing a prototype static site Sweden DCAT-AP Feed from dumps
2. **The Rijksmuseum publication**
   Implementing a feed for Rijksmuseum
3. **The LDES client**
4. **Consumption pipelines**
   a. Sweden: Harvesting → validation pipeline
   b. Sweden: Piveau/consus importer
   c. Rijksmuseum: “Against Opacity” subsets

Sweden DCAT-AP feeds PoC

Now also available as a generic component: https://github.com/rdf-connect/dumps-to-feed-processor-ts

Rijksmuseum



Instead of DCAT-AP, we're exchanging data in the CIDOC domain, yet we also implemented activity streams with named graphs here.

IIIF-CD template changed to LDES template #18

Edit | Code

Open **pietercolpaert** wants to merge 50 commits into `main` from `ldes`

Conversation 19 | Commits 50 | Checks 4 | Files changed 23 | +911 −104

pietercolpaert commented on Feb 9

This pull request should not be merged (I don't have the permissions to open up a draft PR)! It's an overview of the untested changes (we were unable to run the code due to restricted access) we did to instead of exposing a IIIF-CD feed, to describe the interface using Linked Data Event Streams.

What we would need to do now:

- Put this branch in a separated environment, and test it out. As the code is untested, we might have made some obvious mistakes (Rijksmuseum). Some feedback loops might be necessary
- Review the pagination and test whether an LDES client can retrieve this in this way (SEMIC)
- Optimize the pagination and discovery information of the LDES (SEMIC)

Reviewers

DylanVanAssche

tthomassen-rm

Still in progress? Learn about draft PRs

Assignees

No one—assign yourself

Labels

None yet

The LDES client: a generic tool for harvesters

Get the LDES client in typescript from Github or NPM: [rdf-connect/ldes-client]

```
npx ldes-client https://www.pieter.pm/dcat/sweden/feed.ttl
```

```
npx ldes-client --basic-auth HIDDEN:FORNOW
https://acc.data.rijksmuseum.nl/ldes/collection.json
```

Replicates and always stays in-sync with the source

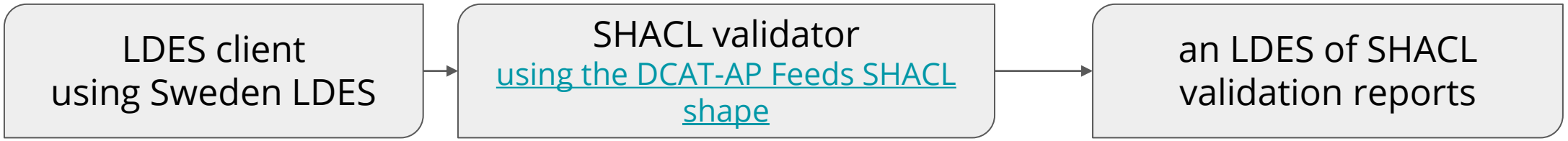
#LDES

Creating consumption pipelines using RDF Connect

1. While we're harvesting, we can validate
2. Building a piveau/consus importer (a tool used by data.europa.eu)
3. Creating subsets of Rijksmuseum for "Against Opacity"

#RDF CONNECT

Validating


LDES client
using Sweden LDES
SHACL validator
using the DCAT-AP Feeds SHACL shape
an LDES of SHACL validation reports

The validation feed provides automated feedback to member states on why certain datasets won’t show up in the European portal

```
<validation#Feed> a ldes:EventStream ;
                  ldes:timestampPath prov:generatedAtTime ;
                  tree:member <#1> , … .

<#1> a sh:ValidationResult;
      prov:generatedAtTime "2024-04-26T16:50:01Z"^^xsd:dateTime ;
      sh:focusNode <https://datakatalog.helsingborg.se/store/3/resource/476>;
      sh:resultMessage "Value does not have datatype xsd:decimal";
      sh:resultPath <http://www.w3.org/ns/dcat#byteSize>;
      sh:resultSeverity sh:Violation;
      sh:sourceConstraintComponent sh:DatatypeConstraintComponent;
      sh:sourceShape [];
      sh:value "2022-02-09"^^xsd:date .
```

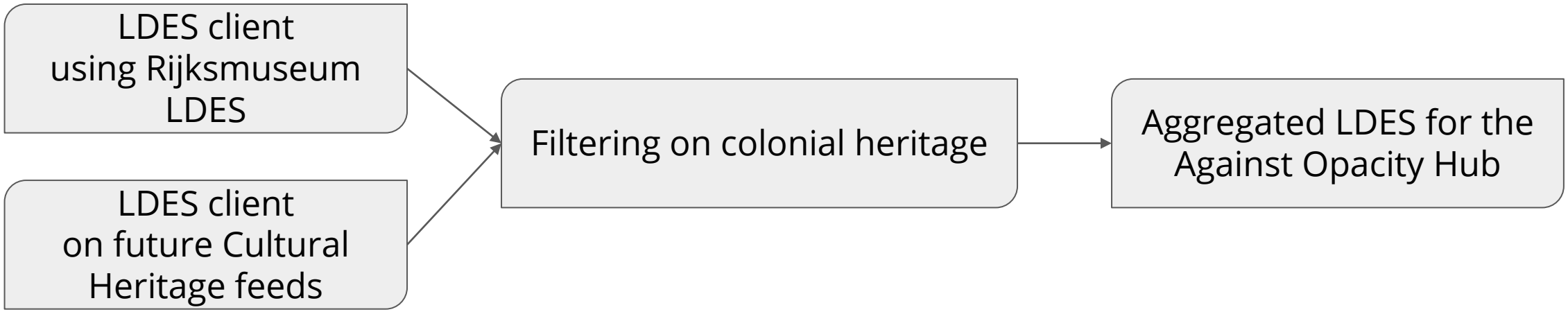
Building our own consus importer for data.europa.eu

Consus can now work with the create, update, deletes given by the member states

The importer wraps the LDES client in a service that provides the data to piveau

https://github.com/rdf-connect/piveau-consus-importing-ldes

Against Opacity Hub



Live demo in June? Join us online during the SEMIC2024 pre-conference!

fin.

Your turn

Question time

- **Do you have questions for any of our speaker?**
- **Would LDES be interesting for you?**
- **Do you use LDES? tell us more!**
- **What do you want to see in LDES' future?**

Next steps

- An update to these and other LDES implementations at **SEMIC2024 on 26 June**. Online attendance possible.
- In September/October a **LDES standardisation track** will start. Digital Flanders will organise a preparatory track until then.
- **Post issues** and **join the Working Group** for the Track. Reach out to digit-semic-team@ec.europa.eu


Thank you

interoperable europe

innovation ∞ govtech ∞ community

Stay in touch

(@InteroperableEU) / Twitter

Interoperable Europe - YouTube

Interoperable Europe | LinkedIn

DIGIT-INTEROPERABILITY@ec.europa.eu

https://joinup.ec.europa.eu/collection/interoperable-europe/interoperable-europe