
SEMIC 2024
conference
Brussels 27 June
interoperable
europe
from Vision to Reality
be EU
belgium24.eu
European Commission

Linked Data Event Streams: From Pilots to Standards

Pre-conference workshops part one

Anastasia Sofou – SEMIC, DIGIT
Pieter Colpaert – UGhent and SEMIC, DIGIT

Agenda

LDES an introduction
Pieter Colpaert - UGent

LDES for Flemish Data Spaces
Sander Van Dooren – Digital Flanders

LDES for DCAT-AP: DCAT-AP feeds
Matthias Palmér – MetaSolutions
Ulrika Domellöf Mattsson – Digg, Sweden
Simon Steuer – The Publications office

LDES for cultural heritage
Tim Thomassen – Rijksmuseum

LDES at the EU Railway Agency
Marina Aguado – ERA
Julian Rojas – Ugent

LDES for Vocab Terms
Marc Portier – VLIZ

LDES and OASC MIMs
Thimo Thoeye – OASC

LDES in the future
Emiel Dhondt – SEMIC, DIGIT

Introduction to LDES

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

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.

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

What is a LDES? - Example

Classical model

Data supplier

2019

Name	Living place
Willem	Paris

copy →

2020

Name	Living place
Willem	Paris
Ann	Lyon

copy →

2021

Name	Living place
Willem	Strasbourg
Ann	Lyon

copy →

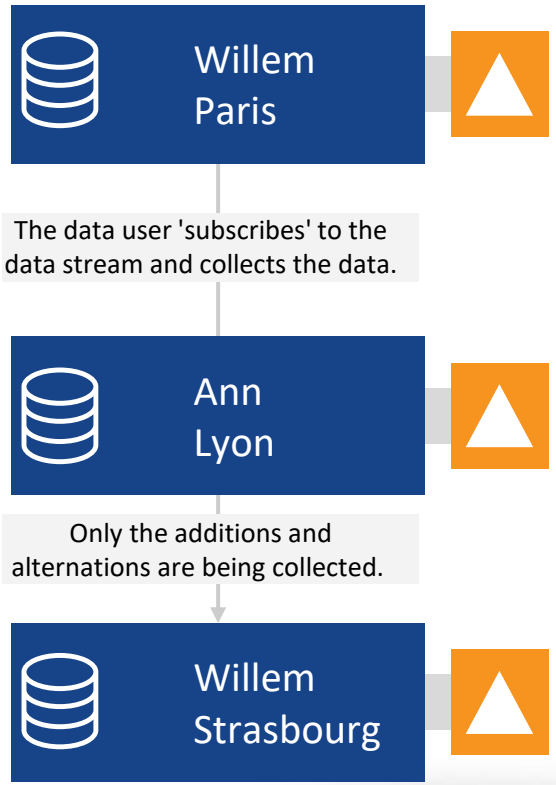
Data user

Name	Living place
Willem	Paris

Name	Living place
Willem	Paris
Ann	Lyon

Name	Living place
Willem	Strasbourg
Ann	Lyon

Linked Data Event Stream



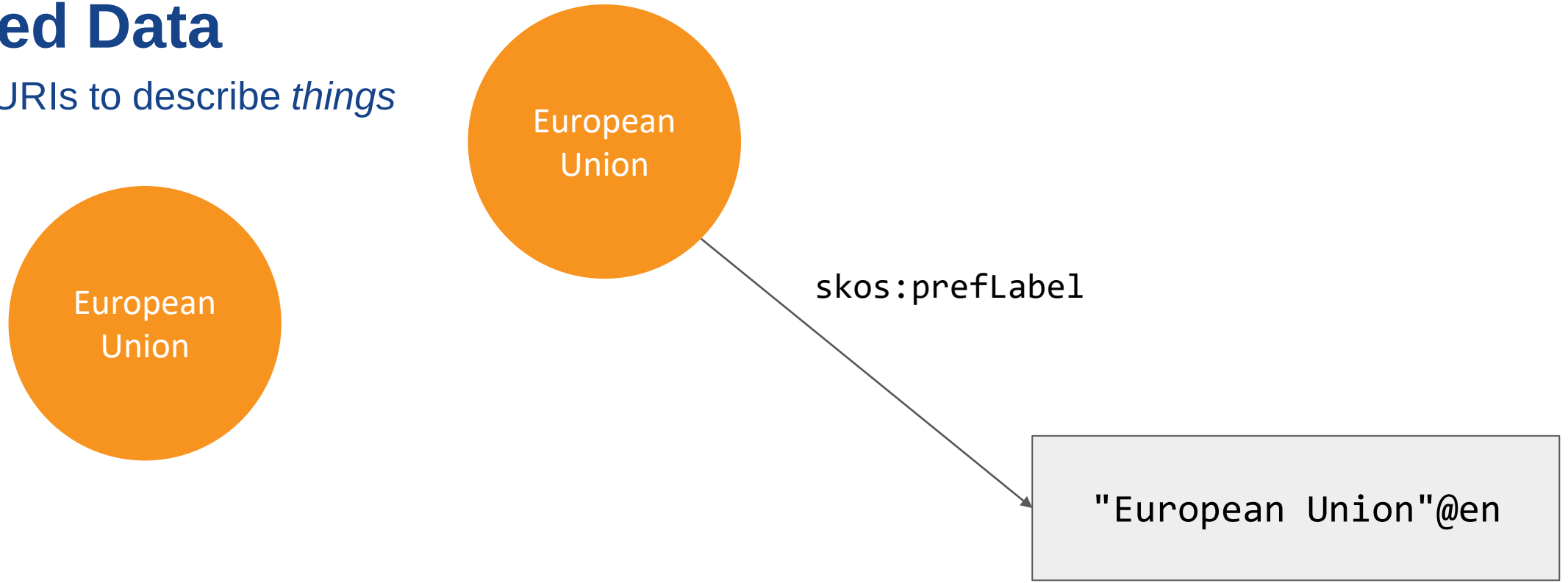
LDES specification overview

Combination of two ideas
Linked Data and **Event Sourcing**

Linked Data

1. Use URIs to describe *things*

2. Use triples to do statements about these *things*

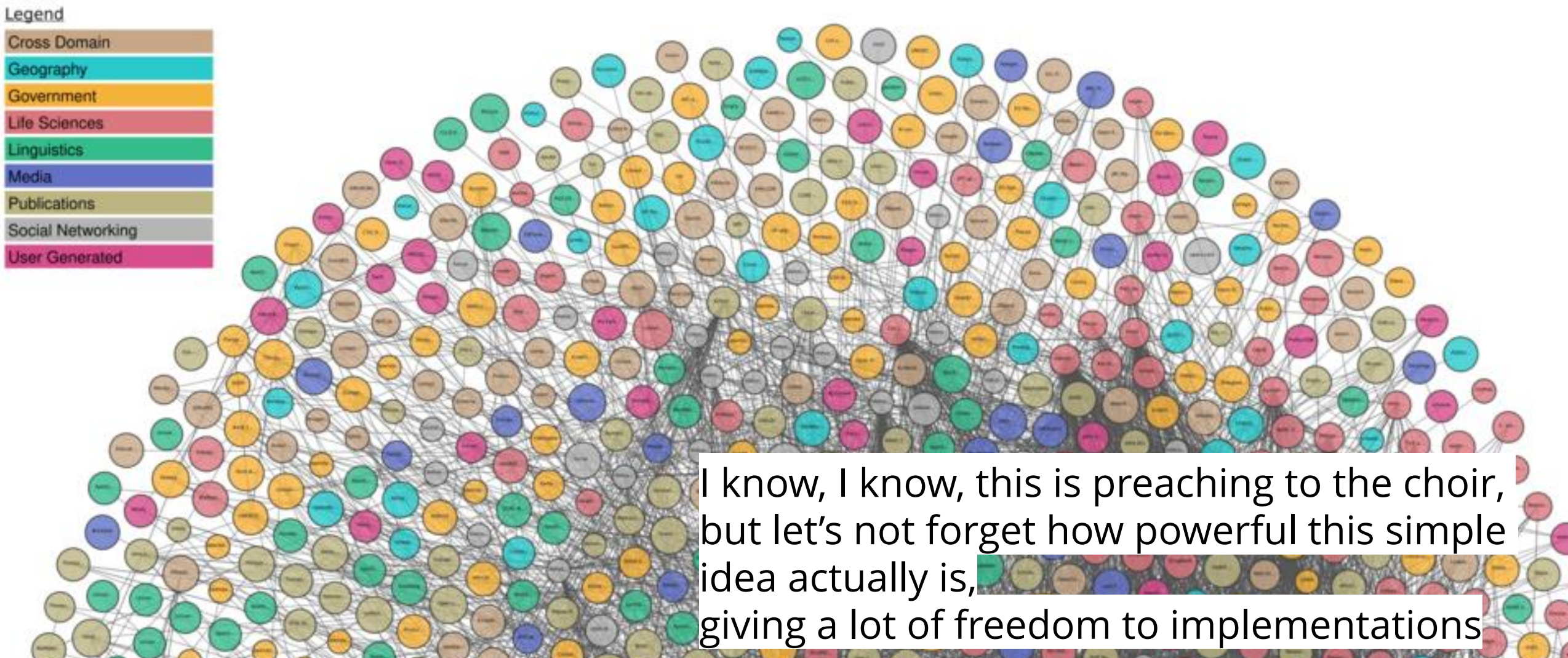


Encode Linked Data triples anywhere using…

1. Turtle
2. JSON-LD
3. CSV on the Web
4. RDF snippets in HTML pages
5. Using the RDF mapping language
6. etc.

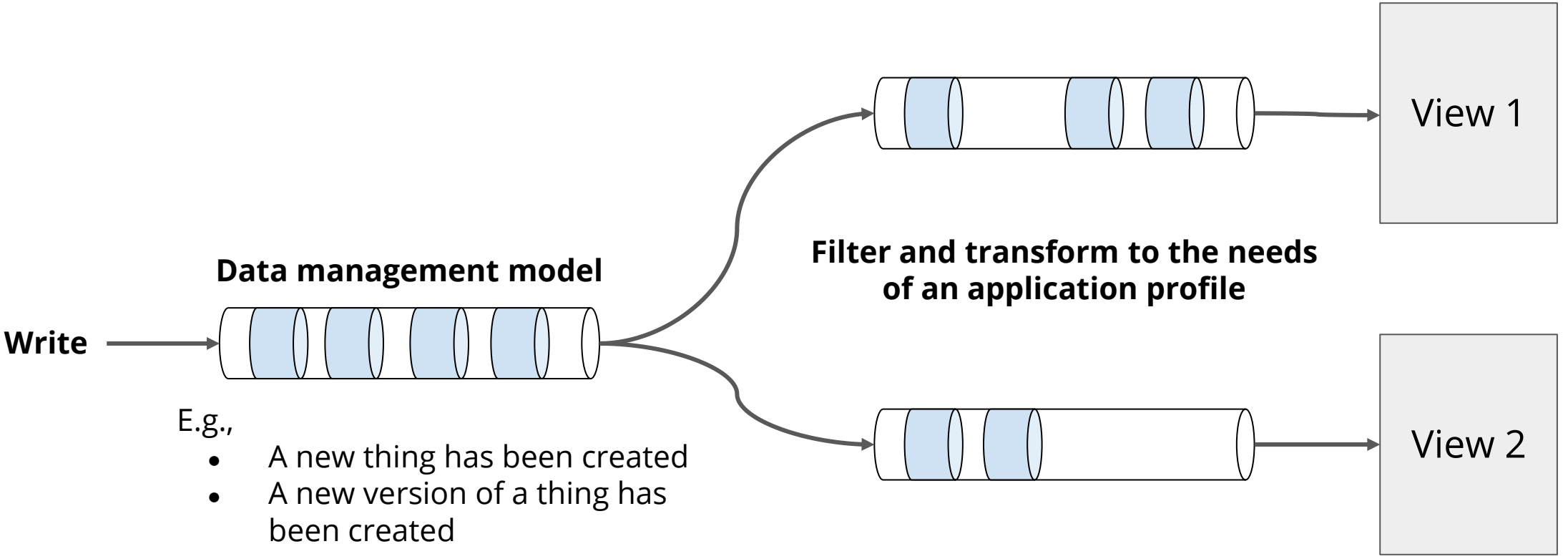
Creating semantic interoperability across data formats

Building a Linked Data Cloud on the Web



I know, I know, this is preaching to the choir, but let's not forget how powerful this simple idea actually is, giving a lot of freedom to implementations

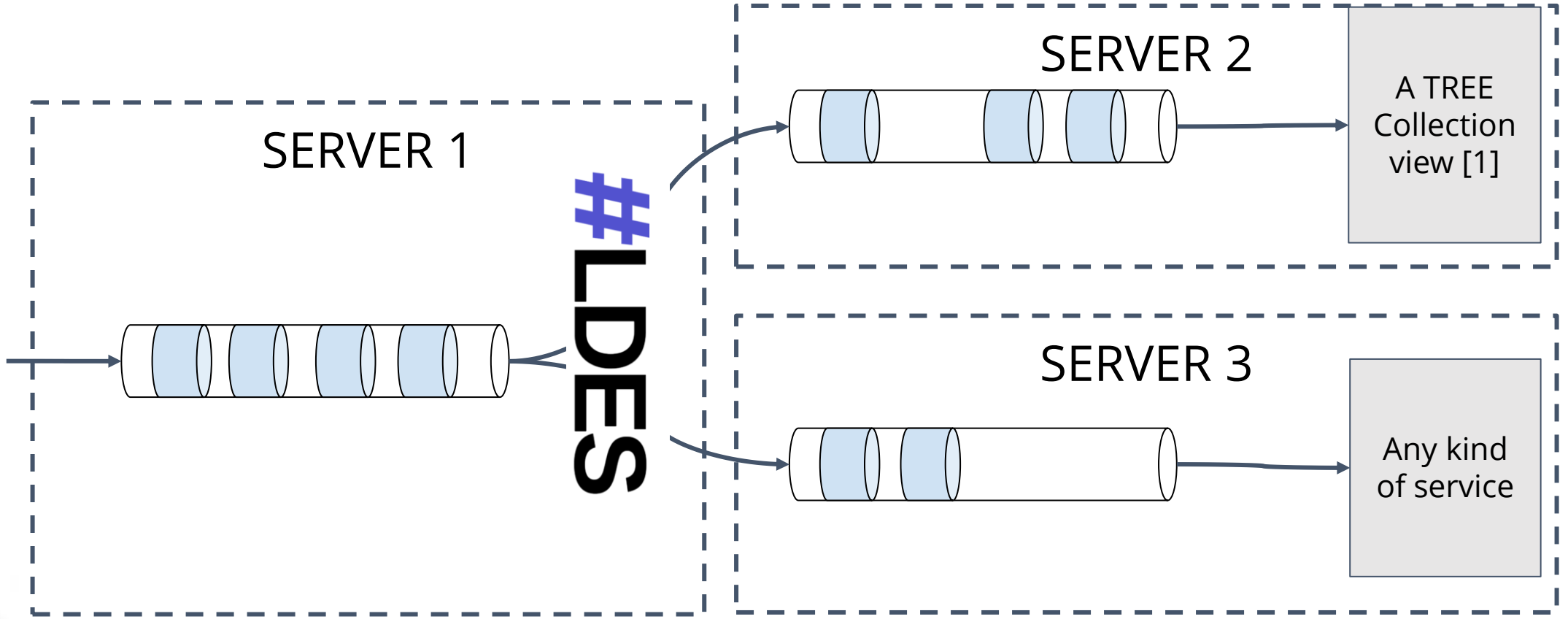
Event Sourcing – how APIs get created



E.g.,

- A new thing has been created
- A new version of a thing has been created

LDES – Event Sourcing across the Web



[1] Colpaert, Pieter. "Building materializable querying interfaces with the TREE hypermedia specification." (2022).

An LDES of street names

```
<C1> a ldes:EventStream ; # → the dcat:Dataset
     tree:member <streetname1> .
<streetname1> rdfs:label "Station Road" .
```

A specific `dcat:Dataset` called `ldes:EventStream` contains your members

Linked Data Event Streams

Explicitly talk about things:

1. In their time context

2. As part of a well-defined collection of items maintained by an organization

An LDES of street names

```
<C1> a ldes:EventStream ;
     tree:member <streetname1> .
<streetname1> rdfs:label "Station Road" .
```

Problem
street names can change!

The member `<streetname1>` is not immutable

An LDES of street names

```
<C1> a ldes:EventStream ;
     tree:member <streetname1-v1>, <streetname1-v2> .
<streetname1-v1> rdfs:label "Station Road" ;
                 dcterms:isVersionOf <streetname1> ;
                 dcterms:created "2020-01-01T00:10:00Z"^^xsd:dateTime .
<streetname1-v2> rdfs:label "Station Square" ;
                 dcterms:isVersionOf <streetname1> ;
                 dcterms:created "2021-01-10T00:10:00Z"^^xsd:dateTime .
```

Solution:
Introduce the concept of a version object that is immutable

Add triples linking the versions together

An LDES of street names

```
<C1> a ldes:EventStream ;
     ldes:timestampPath dcterms:created ;
     ldes:versionOfPath dcterms:isVersionOf ;
     tree:shape mystandards:StreetnamesShape1;
     tree:member <streetname1-v1>, <streetname1-v2> .
<streetname1-v1> rdfs:label "Station Road" ;
                 dcterms:isVersionOf <streetname1> ;
                 dcterms:created "2020-01-01T00:10:00Z"^^xsd:dateTime .
<streetname1-v2> rdfs:label "Station Square" ;
                 dcterms:isVersionOf <streetname1> ;
                 dcterms:created "2021-01-10T00:10:00Z"^^xsd:dateTime .
```

You can indicate what properties are being used for what, and indicate whether the members will conform to a certain shape

The LDES vocabulary

Defines terms needed for describing your event source:

- **Event Streams**
  An ever-growing collection of immutable members
- **Event Source**
  A view designed to keep other views in-sync
- **Retention policies**
  A view description can decide not to publish all members of the LDES. A retention policy documents what members are kept.
- **Snapshots**
  A potentially filtered list of members from an event stream until a certain point
- **Version materializations**
  The conversion of immutable members to “state objects”

The LDES specification

Explains what an LDES Server **MAY** publish

- Kept as easy and versatile as possible
- Want to be compatible with related specs with similar goals
- using the TREE specification to create materializable interfaces

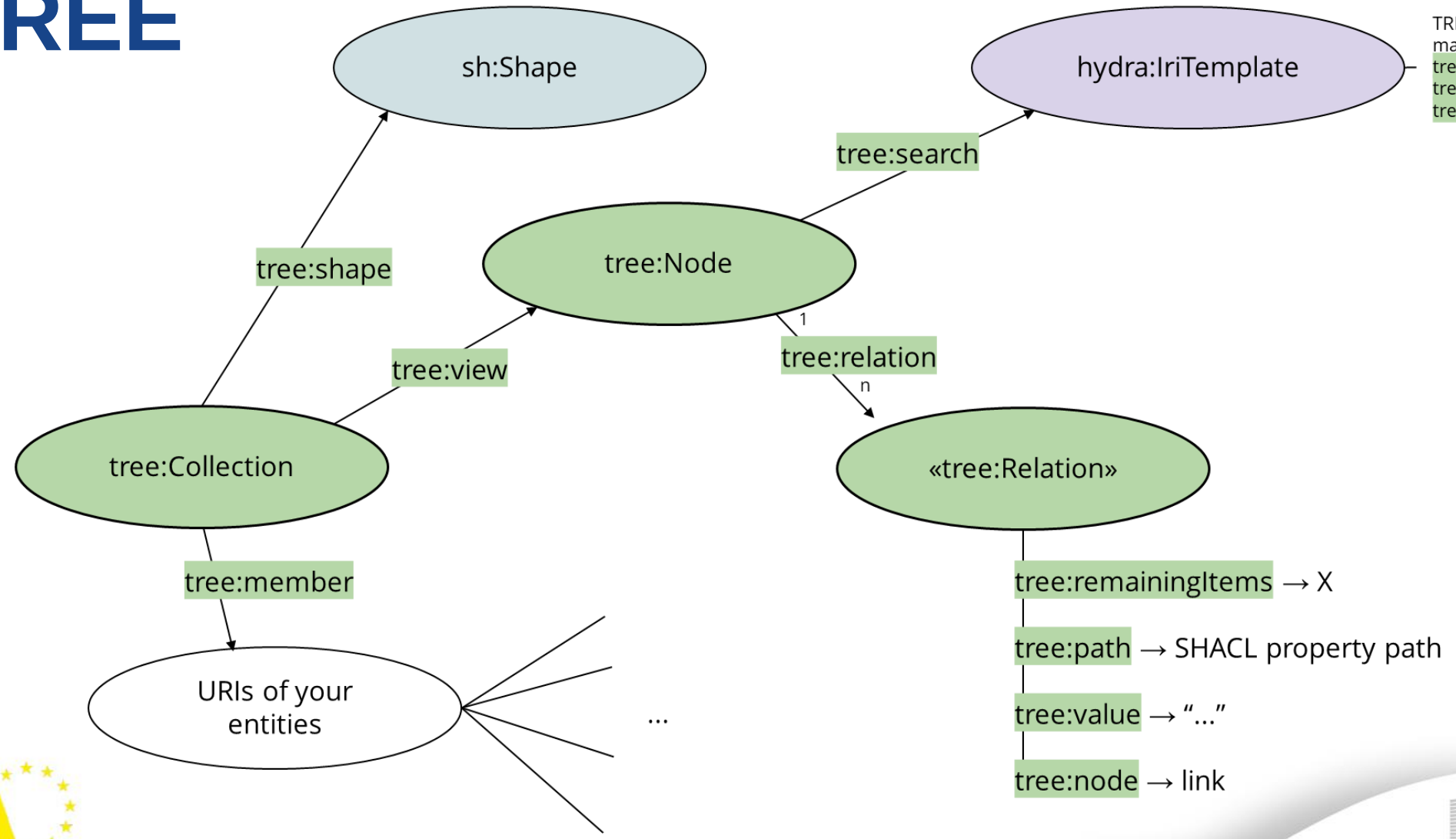
Explains what an LDES Client **MUST** support

- Focussing on keeping replicated views in-sync
- Using the TREE specification for guiding link traversal

The TREE hypermedia specification

```
<C1> a ldes:EventStream ;
     tree:view <?page=1> ;
     tree:member <streetname1-v1>, <streetname1-v2> .
<?page=1> tree:relation [
    a tree:GreaterThanRelation ;
    tree:node <?page=2> ;
    tree:path dct:created ;
    tree:value "2022-04-08T12:00Z"
] .
```

TREE


sh:Shape
hydra:IriTemplate
TREE specific variable mapping properties:
tree:timeQuery, tree:zoom, tree:latitudeTile, tree:longitudeTile
tree:search
tree:shape
tree:Node
1
tree:relation
n
tree:view
tree:Collection
«tree:Relation»
tree:member
tree:remainingItems → X
tree:path → SHACL property path
tree:value → "..."
tree:node → link
URIs of your entities
...

LDES as part of Data Spaces in Flanders

Pre-conference workshops part one

Sander Van Dooren
Digital Flanders (VSDS)

Linked Data Event Streams

01

What is LDES

The why and what of LDES

02

An opinionated implementation

Server, client and data transformation building blocks:
Batteries included.

03

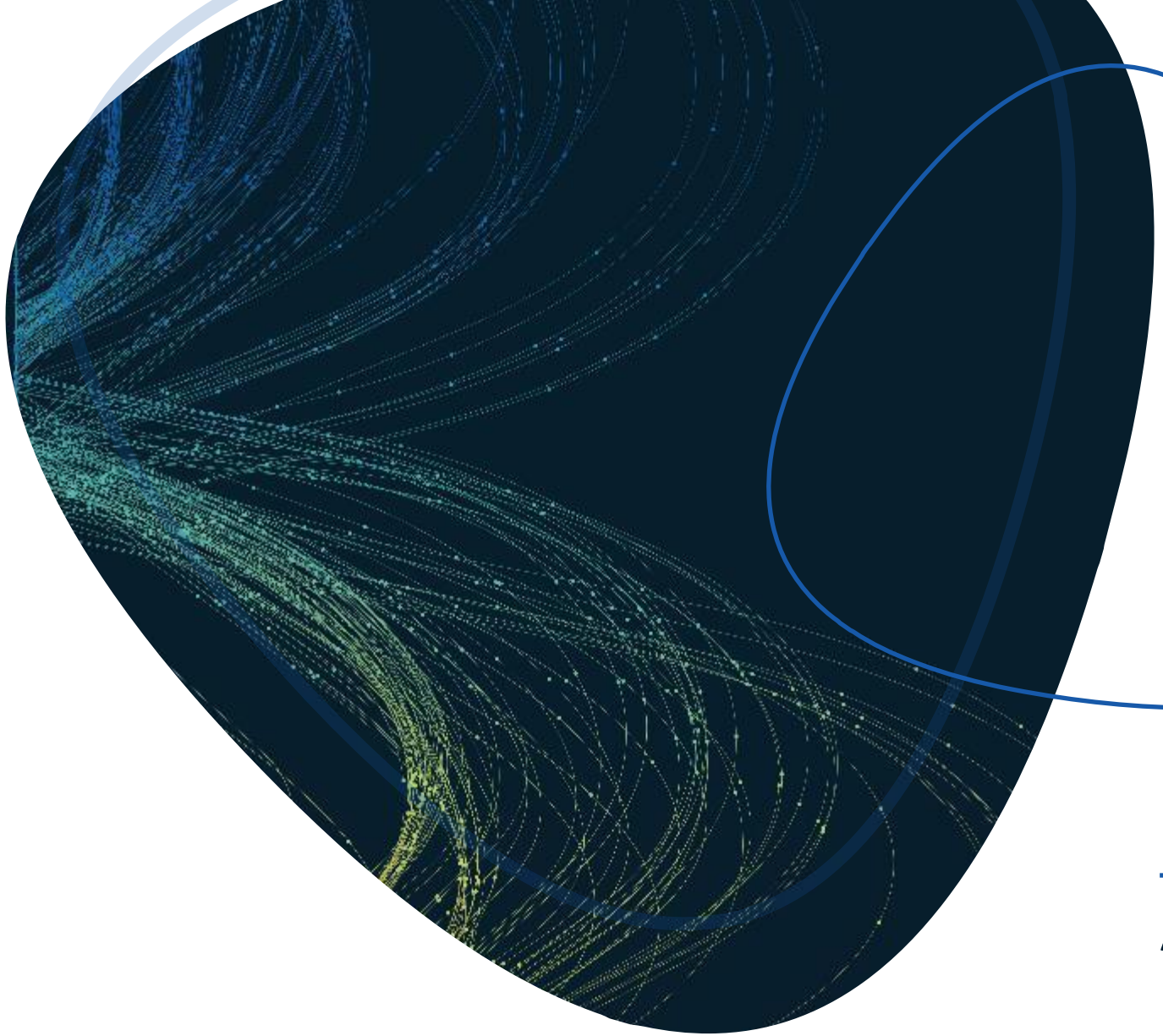
LDES in dataspaces

The role of LDES as a data plane in IDSA compliant dataspaces

04

Future of LDES

A personal view on a potential future



LDES

An overview of the Linked Data Event Stream protocol

What is a stream in LDES?

A continuous and ordered sequence of data events

A log of changes

Immutable, continuous log of changes to the data

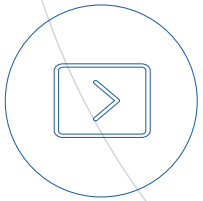
A member is complete

A version object (member) contains the entire state of an object

A stream is ordered

The order in the log is fixed

Writes are append-only



1.Replication & Synchronization

2.SCALABILITY

Optimized for HTTP caching

Fair cost balancing

3.INTEROPERABILITY

Open specification

Builds on existing web standards

4.TIMELINESS

Streaming

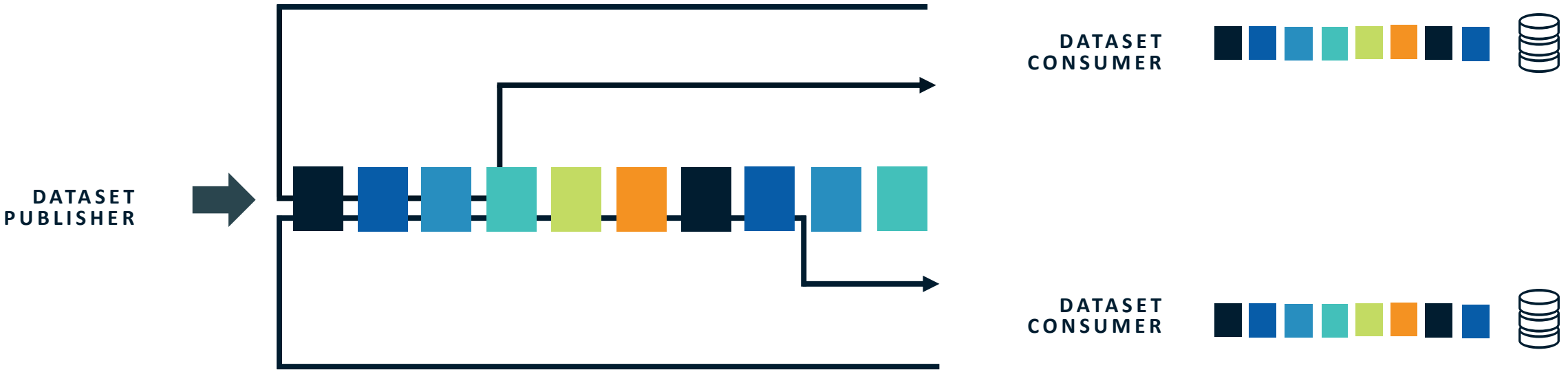
Optimized for near real-time applications

5.DECENTRALIZATION

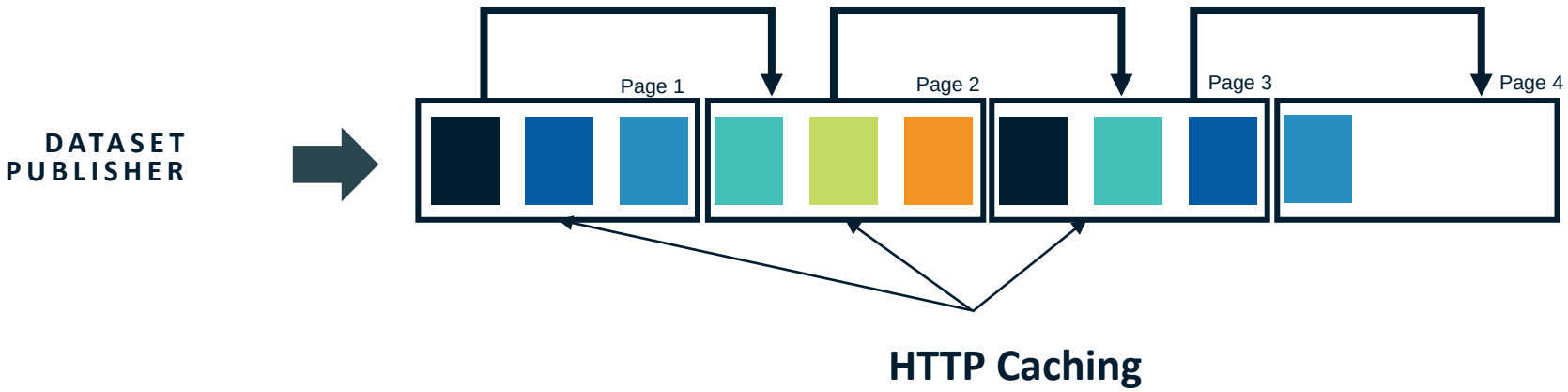
Coherent data publishing through identifier reuse

Core Concepts

Sharing near real-time data as streams, using fragmentation to break the stream into manageable chunks, applying retention policies to manage data lifespan, and incorporating metadata for enhanced traceability, interoperability, and context.



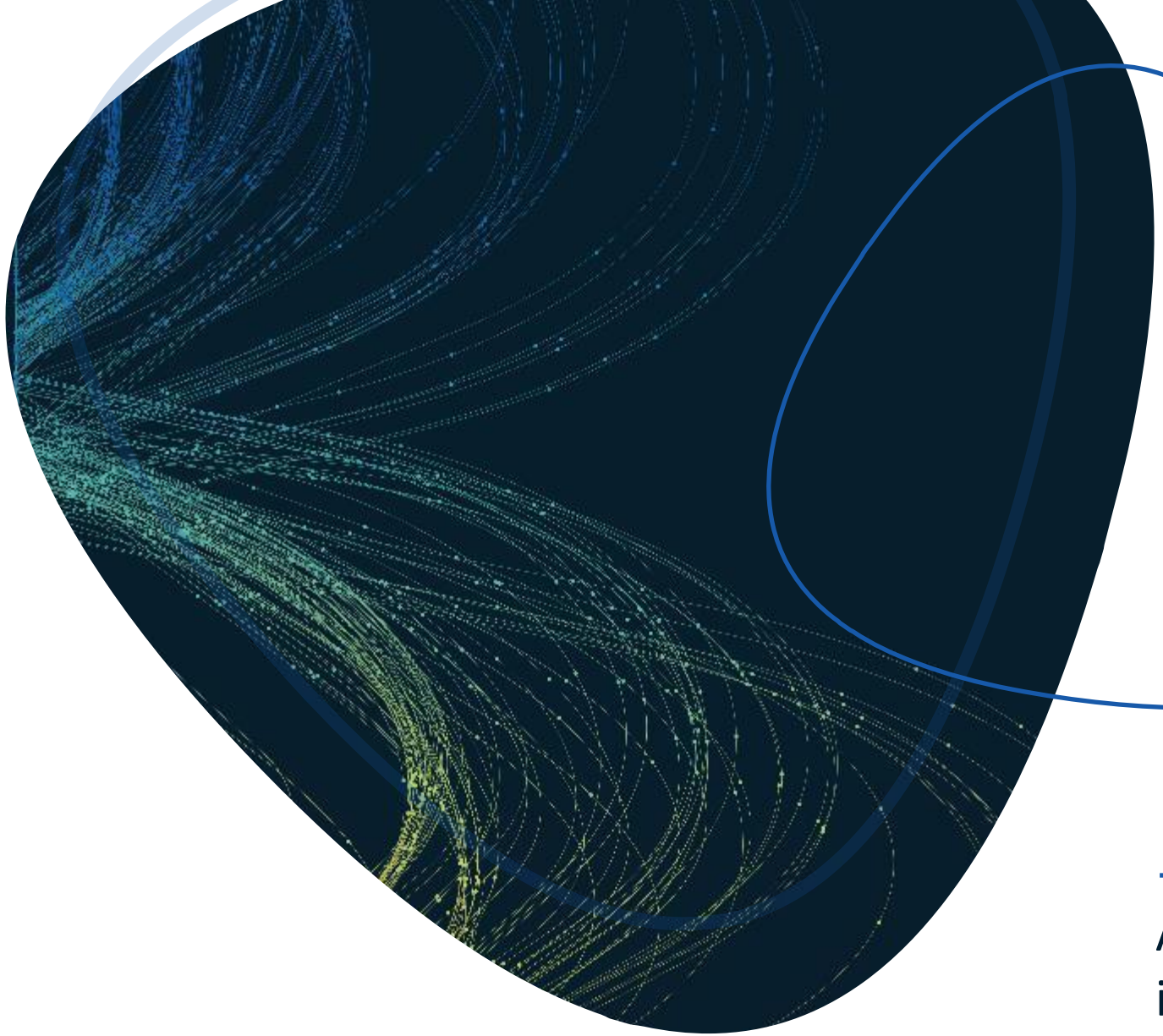
30


DATASET
PUBLISHER
Page 1
Page 2
Page 3
Page 4
HTTP Caching

DATASET PUBLISHER

Linked Data Event Streams are linked data first and foremost

Linked Data Event Streams is a SEMIC specification



Implementation

An opiniated implementation of LDES in the Flemish Smart Dataspace (VSDS)

Opinionated

Predictability

We focus on replication scenarios, such as base registries
Each client holds same data after materializing the log
Optimized for OLTP workloads

Embody best practices

Design choices make it easy to do the right thing.
Build in assumptions to prevent (common) mistakes.

LDES Server

ADMIN API

Configure multiple streams
Attach DCAT metadata
Define retention and fragmentation

Prometheus endpoint for monitoring

INGEST API

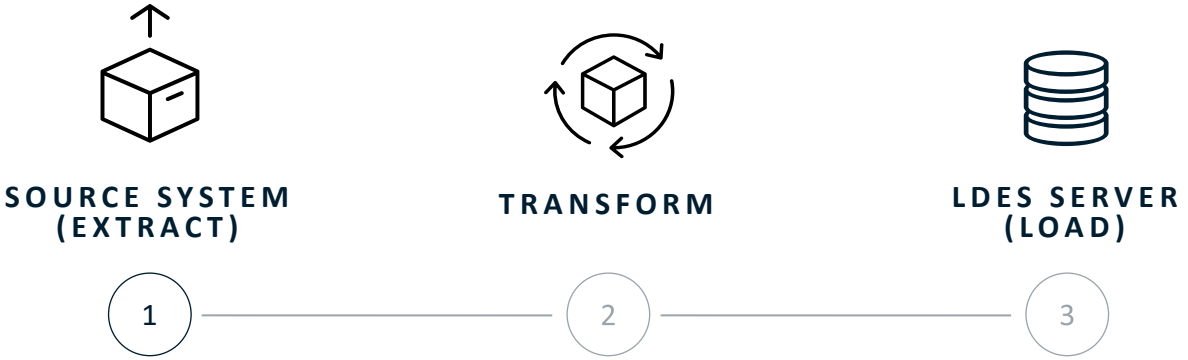
Endpoint per stream
SHACL validation
HTTP IN
(Kafka IN)

FETCH API

Serves LDES fragments to clients

TECHNOLOGY

JAVA (Spring)
DB: Postgresql
Open Source: EUPL-1.2



LINKED DATA INTERACTIONS

Simple

A simple data pipeline task runner for straightforward flows.

YAML based configuration

APACHE NiFi

Complete

A comprehensive suite of data transformation blocks.

Easy to use graphical user interface

Supported Transformers

01

SOURCES

LDES CLIENT (!)
HTTP
KAFKA
AMQP

02

TRANSFORMERS

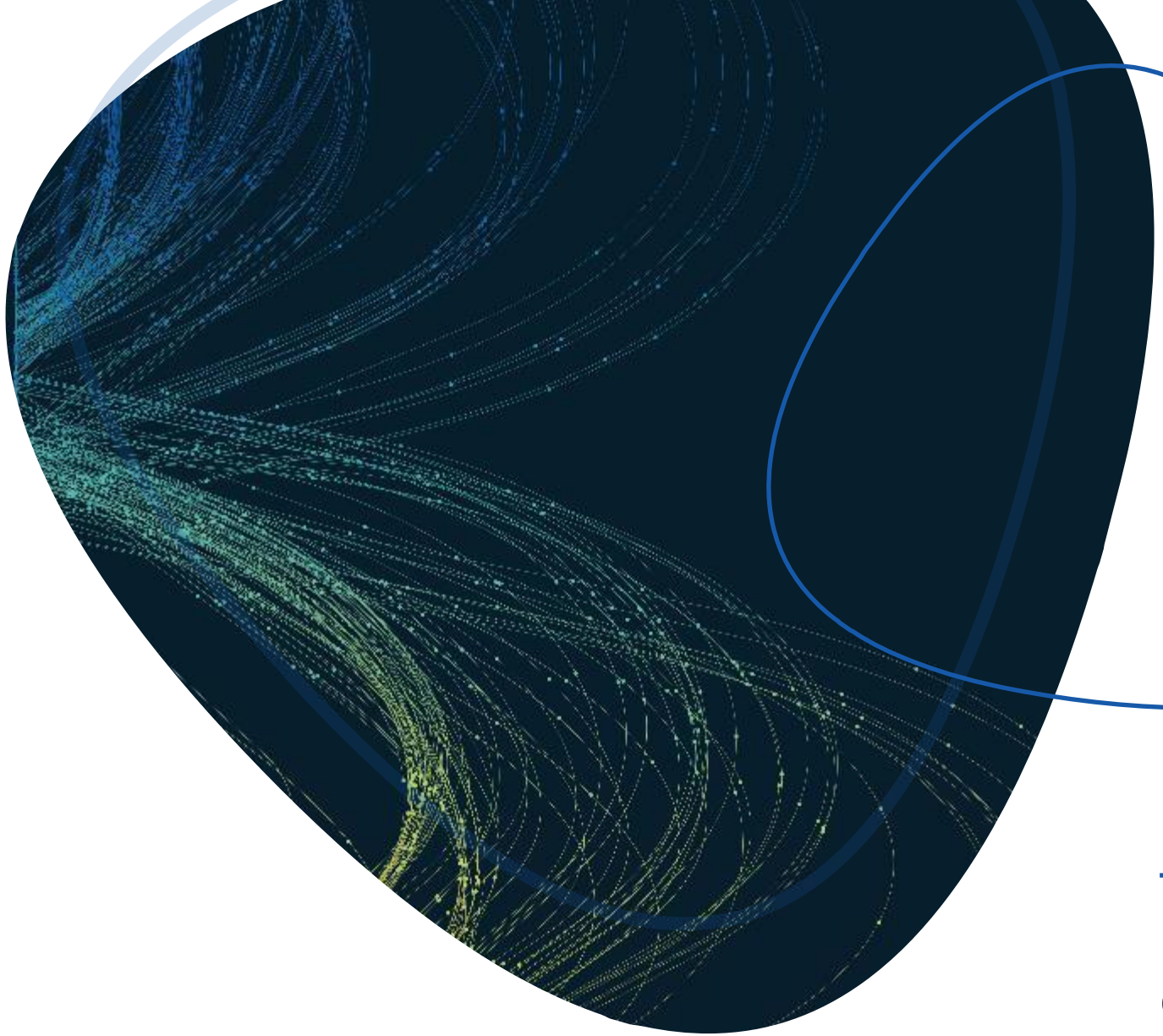
RML
SPARQL CONSTRUCT
NGSI v2 -> NGSI-LD

03

SINKS

HTTP
KAFKA
TRIPLESTORE (RDF4J contract)

The
Linked Data Event Stream
client is a source
in a data pipeline.



Dataspaces

Bridging the gap between IDSA dataspace connectors and LDES

Fitting LDES in the IDSA landscape

Data plane

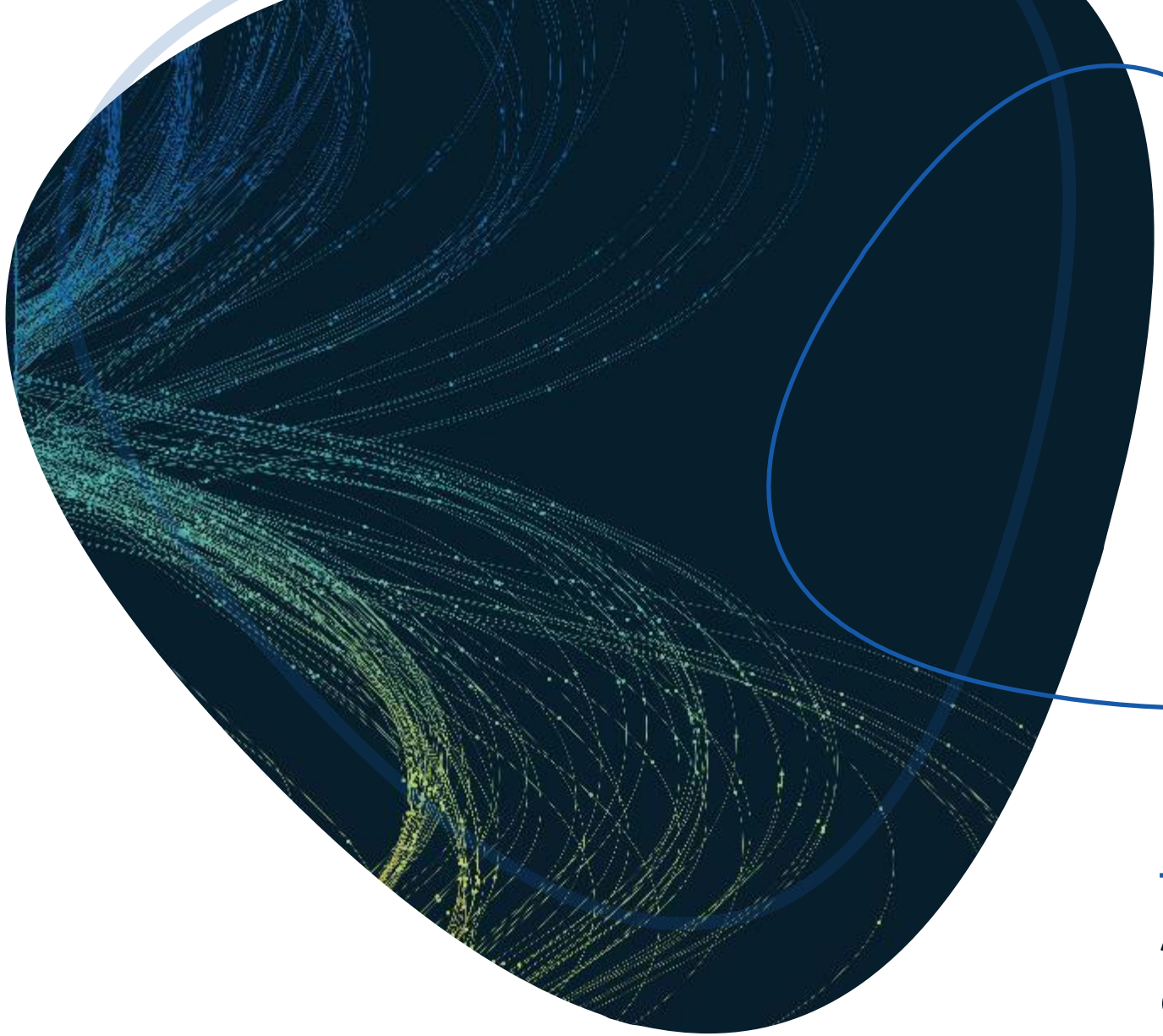
- A data plane extension for the Eclipse Dataspace Connector
- HTTP headers and status codes

Client wrapper

- Extension of the LDES Client
- Implements the IDSA Transfer Process Protocol

End-to-end

- Testing of LDES over IDSA Transfer Process Protocol
- Alternative: API gateway



Future

A personal reflection on the future of LDES

Future of LDES?

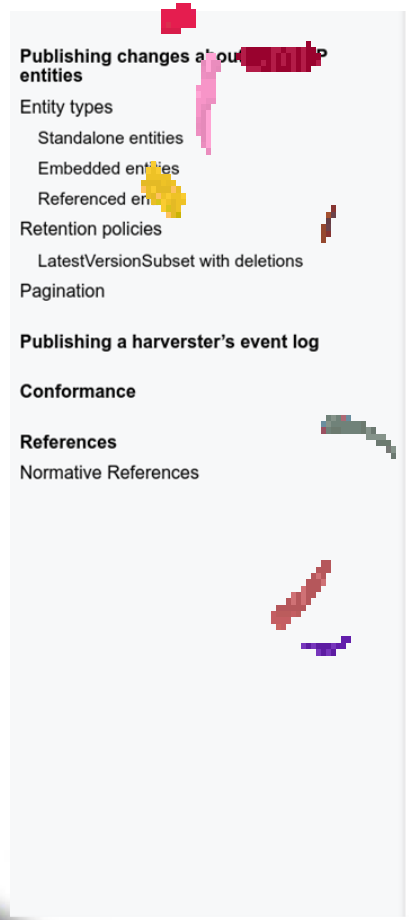
Disclaimer:

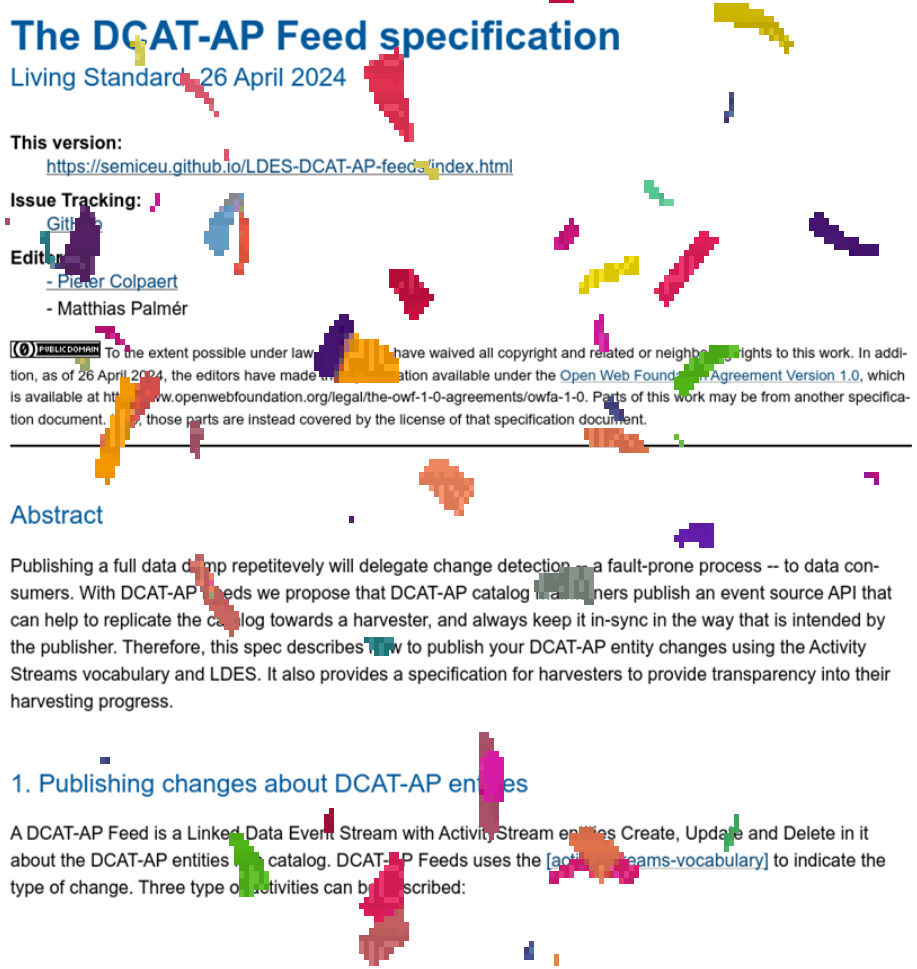
Personal opinion!

https://informatievlaanderen.github.io/VSDS-Tech-Docs/

Thank you!

DCAT-AP feeds





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

PUBLIC DOMAIN To the extent possible under law, the editors have waived all copyright and related or neighboring rights to this work. In addition, as of 26 April 2024, the editors have made this specification available under the Open Web Foundation Agreement Version 1.0, which is available at http://www.openwebfoundation.org/legal/the-owf-1-0-agreements/owfa-1-0. Parts of this work may be from another specification document. If so, those parts are instead covered by the license of that specification document.

Abstract

Publishing a full data dump repetitevely 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:

A LDES Profile for DCAT-AP

Contents of the standard

1. Identify an element as Create, Update or Delete
2. Classify an entity as Stand-alone, Embedded or Referenced
3. LDES description
4. A retention policy
5. A smart and straightforward pagination
6. DCAT-AP feeds SHACL shapes
7. JSON-LD and TRiG examples

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.

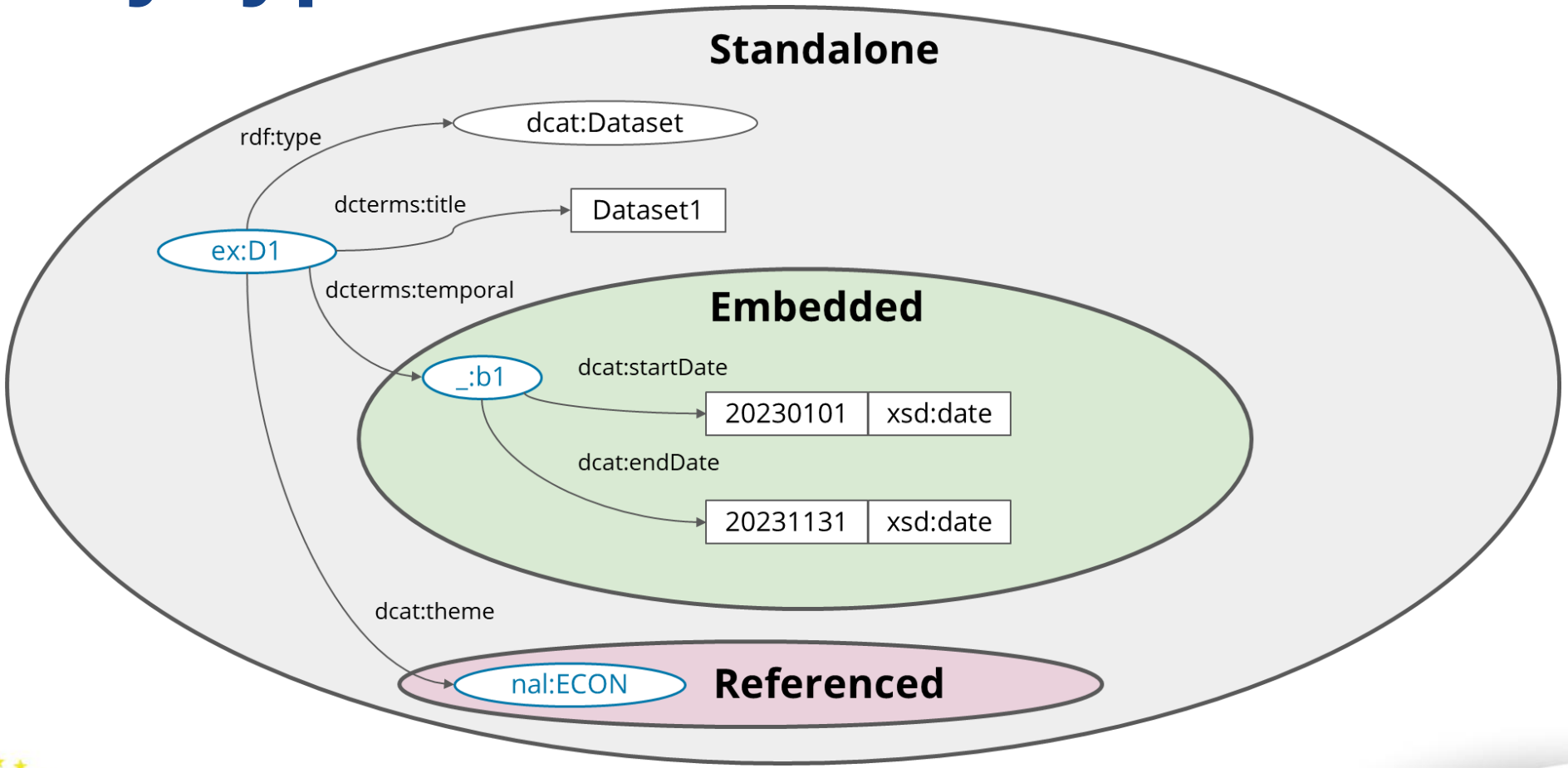
§ 1.1. Entity types

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

Entity types



Which entities do we have?

Standalone

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

Embeded

- 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

What is part of one update?

- 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

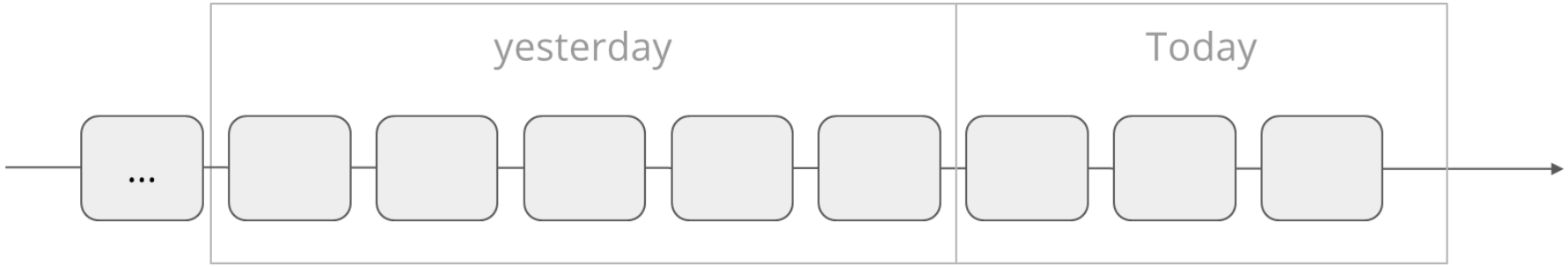
```
{
  "@id": "#Feed",
  "@type": "EventStream",
  "title": "My DCAT-AP Feed",
  "timestampPath": "published",
  "versionOfPath": "object",
  "view": {
    "@id": ""
  },
  "member": [  {
      …
  ]
}
```

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

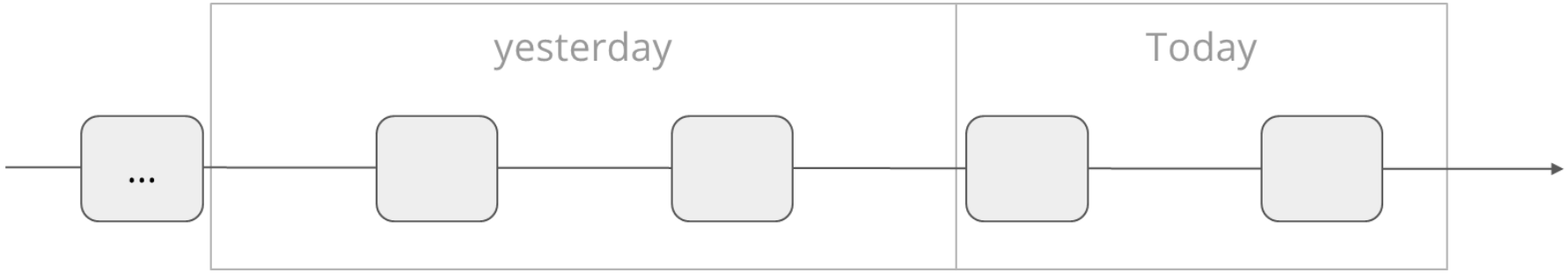
LDES supports retention policies

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": [  {
      …

  ]
}
```

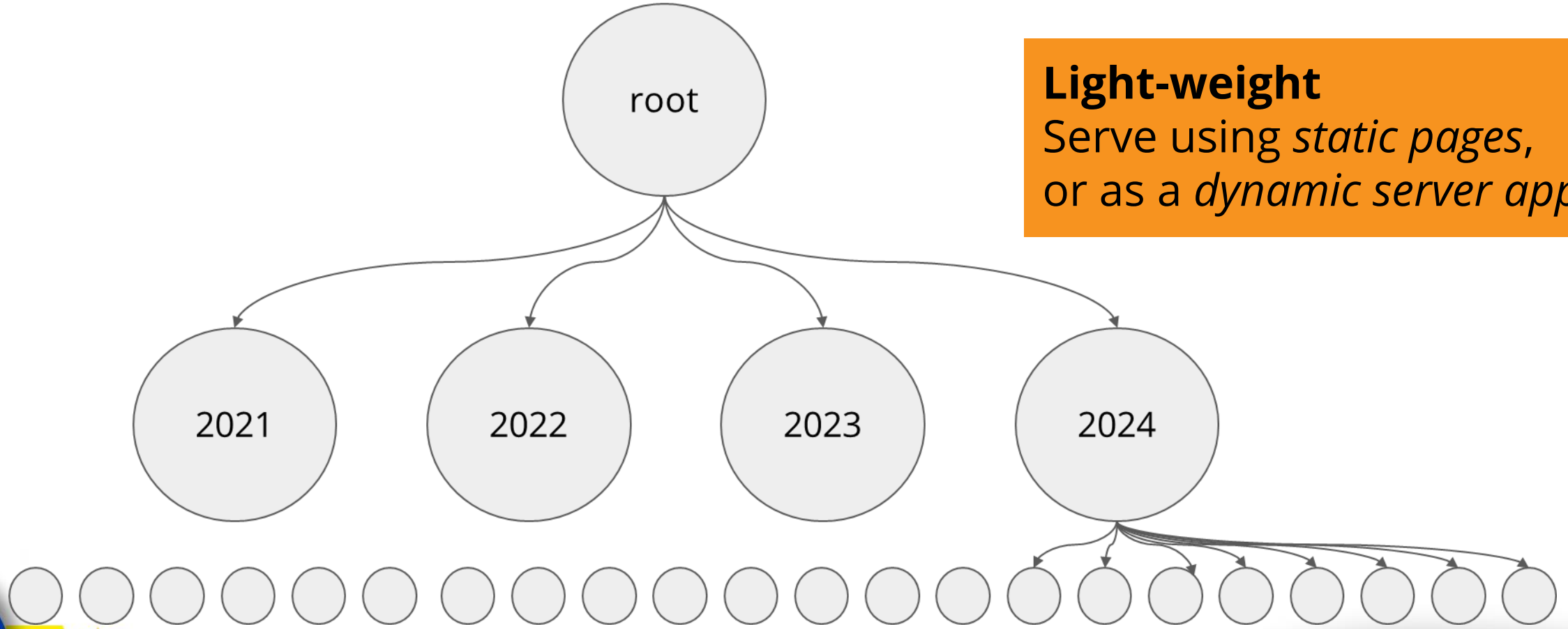
```
<#Feed> tree:view <> ;
        ldes:timestampPath as:published ;
        ldes:versionOfPath as:object .
<> ldes:retentionPolicy [
        a ldes:LatestVersionSubset ;
        ldes:amount 1

      ] .
```

Problem

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

Indicating we only keep the latest version



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

What is part of one update?

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) ;
```

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

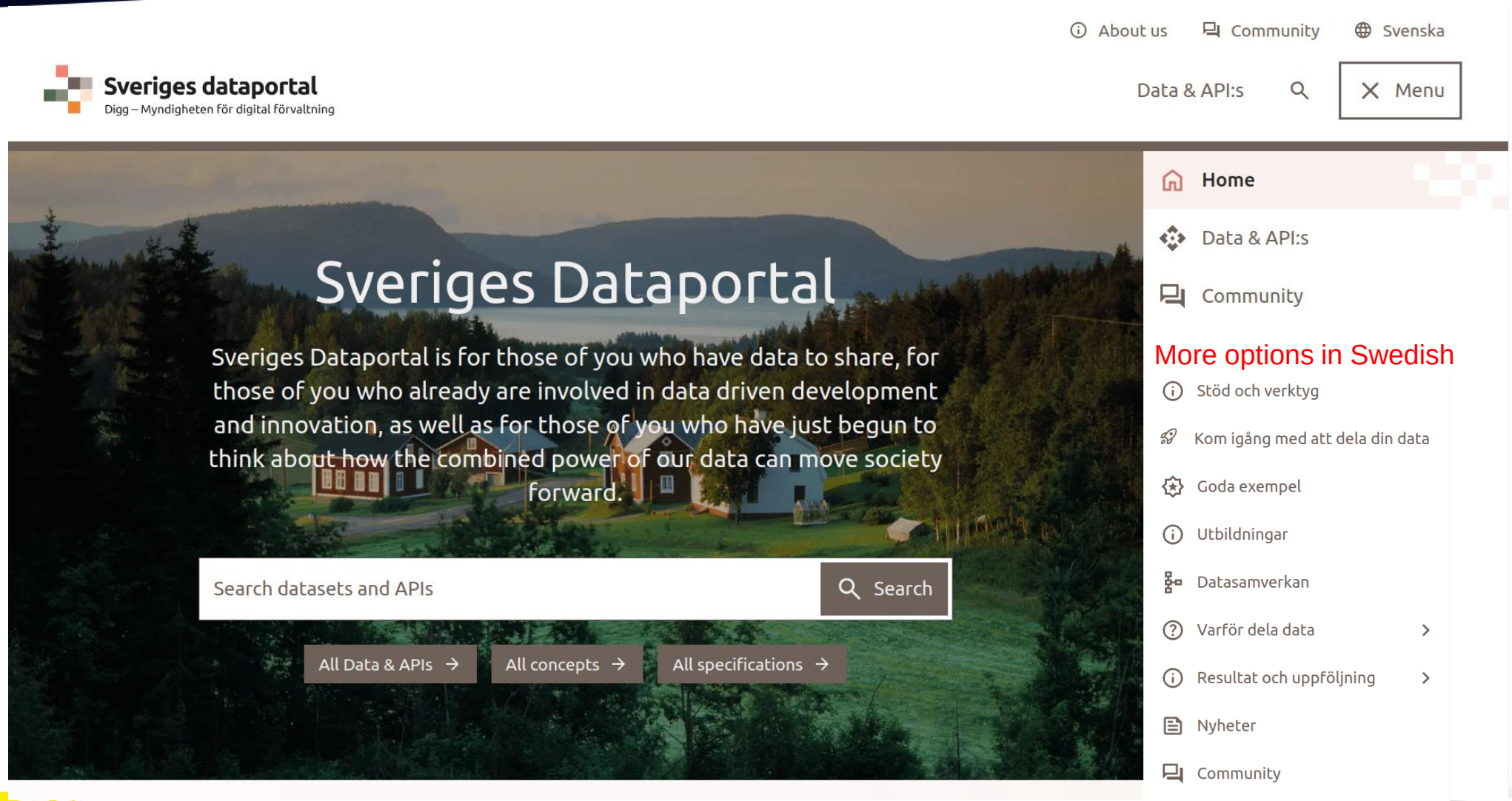
But most importantly

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

- The Swedish dataportal and the business case
- Implementation report
- Harvesting reports

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 options in Swedish
Stöd och verktyg
Kom igång med att dela din data
Goda exempel
Utbildningar
Datasamverkan
Varför dela data
Resultat och uppföljning
Nyheter
Community

Harvesting from three types of catalogs

- Shared editing platform (national instance)
  - 20 catalogs, mostly small municipalites
- Shared domain catalogs, for example:
  - National Geodata catalog
  - National Statistical catalog
  - Joint Scientific data catalog
- Individual catalogs
  - 100 catalogs, Municipalities, Agencies, NGOs / businesses

Business case for exploring LDES (Digg)

1) Synchronization issues with data.europa.eu
   - Minimize discrepancies in dataset search
   - Quicker detection of problems
   - Quicker updates
- 2) Use of standards
   - Validate that the use of the harvesting protocol is followed
   - Define more clearly what is needed to be harvested
- 3) Harvesting scalability

Swedish implementation report

Current harvesting infrastructure

Process

- One file per catalog
- 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)

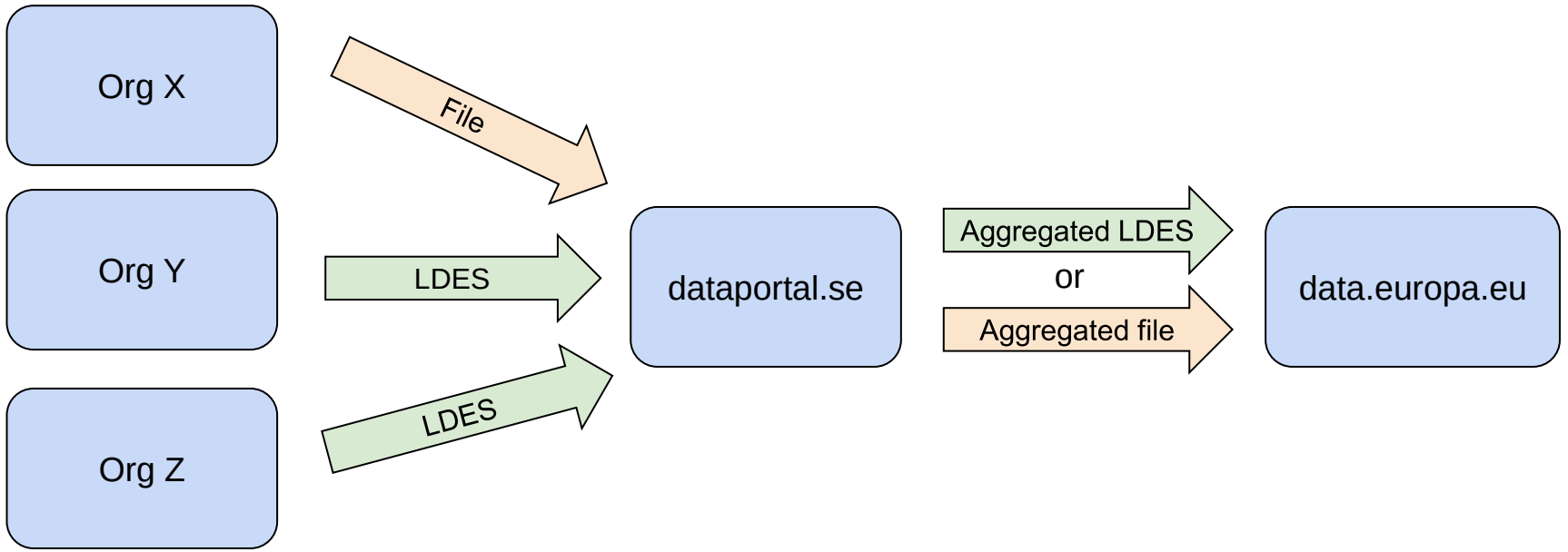
Why DCAT-AP feeds (LDES)

Some of the issues have solutions already

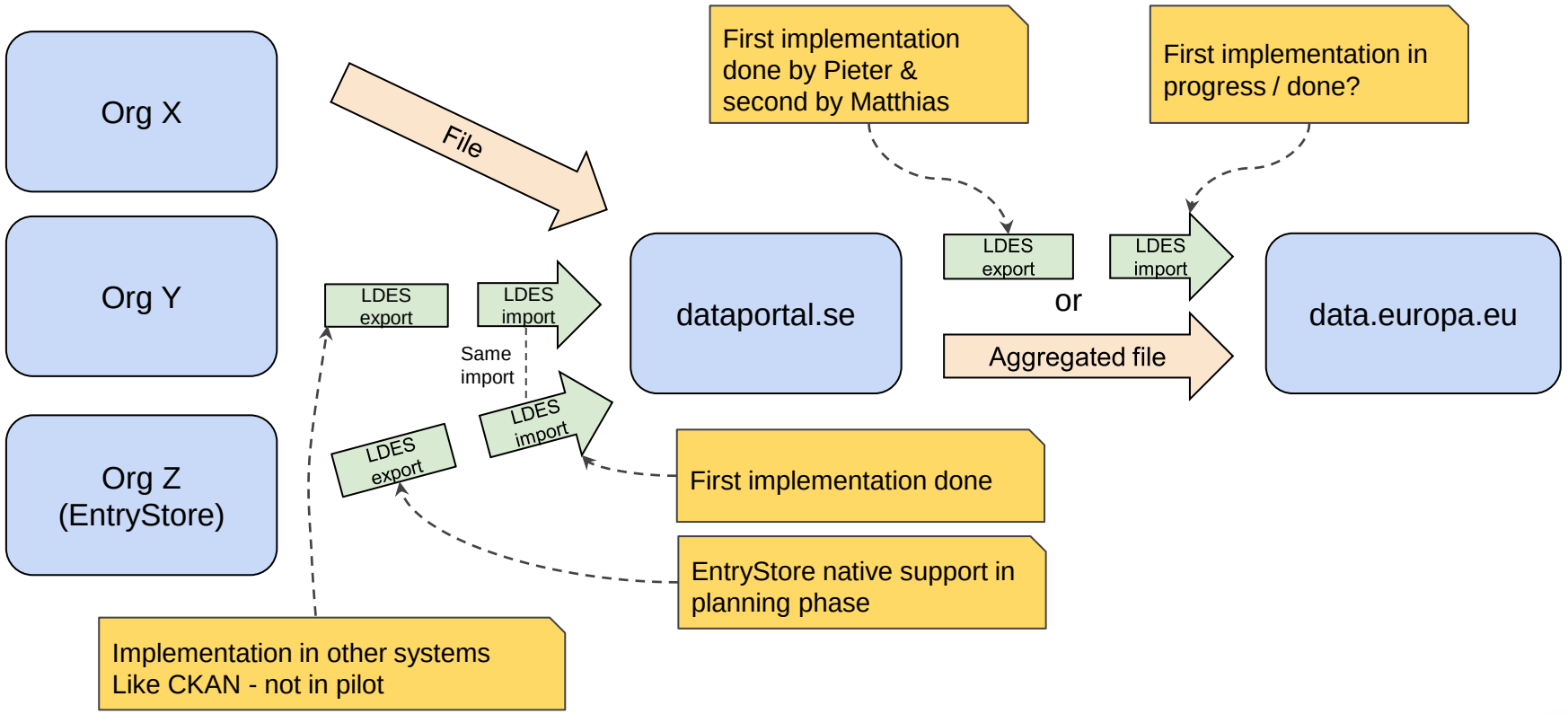
BUT

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

Target harvesting infrastructure



Target harvesting infrastructure (status)



Headache 1 - deletions

No record of deletes in some systems.

Solution 1 - add support in original system

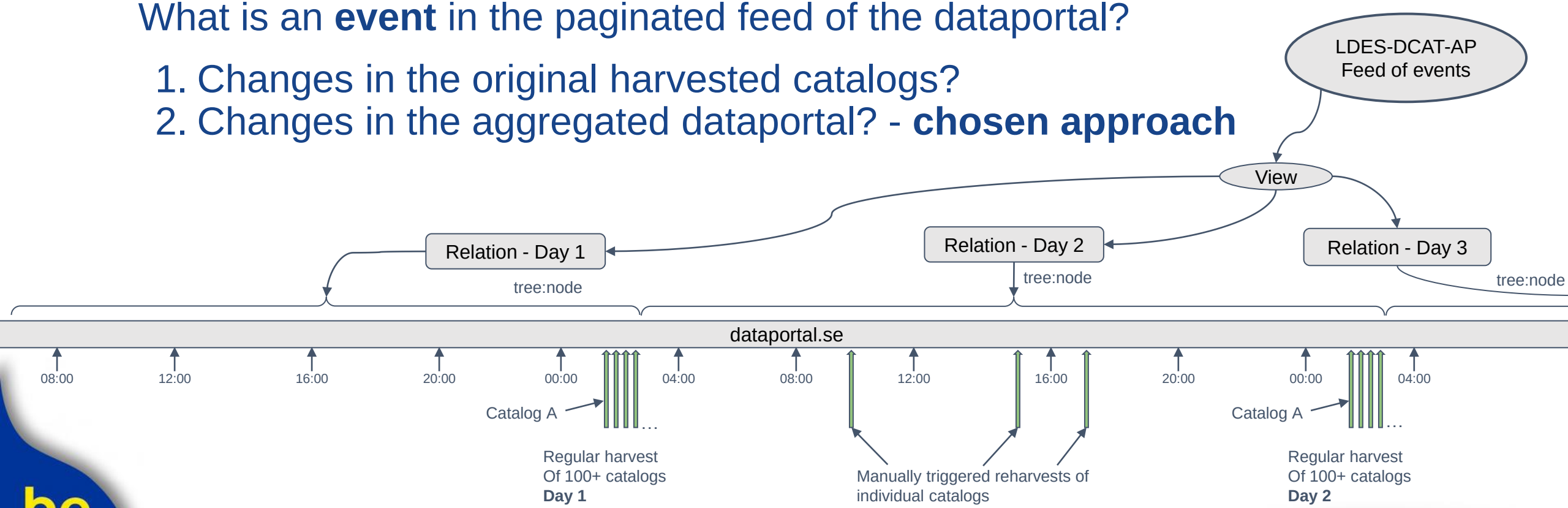
Solution 2 - do a wrapper which keeps an index - **chosen approach**

How long de we keep the records of deleted entities? - **indefinitely**
(but only URIs that have been deleted, this is part of the index and feed)

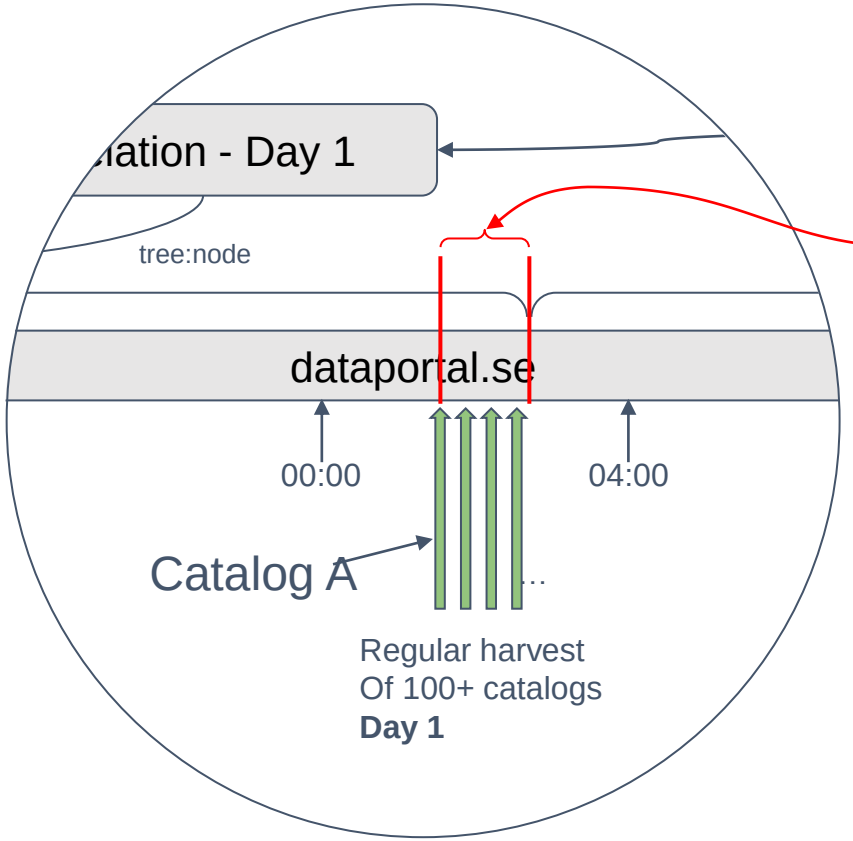
Headache 2 - what does the event correspond to?

What is an **event** in the paginated feed of the dataportal?

1. Changes in the original harvested catalogs?
2. Changes in the aggregated dataportal? - **chosen approach**



Headache 2 - explained



Harvesting process is not instantaneous

Catalog A may have datasets that are updated in the indicated period.

One solution is to have relations with overlapping time periods OR treat events as the changes in the portal (**chosen approach**).

Note: overlapping relations is possible, but leads to higher demans on the consumer.

Extra benefit: We do not need to store the modification dates from the original catalog.

Harvesting report - high level per catalog / feed

Amount of **primary entities**

- Added
- Updated
- Removed
- Unchanged

Two potential levels:

- Only the numbers
- Listing of the entities that are added etc. (Listing via their URIs)

We define primary entities simply as those we are interested in getting a report for

Most likely:

- Dataset
- Datasetseries
- Data service

Unlikely:

- Catalog
- Distribution (counted as part of dataset?)
- Contactpoints
- Publishers
- License Documents

Harvesting report - detailed report

Detailed report per catalog / feed

Cardinality report:

- Missing mandatory fields
- Missing recommended fields?

Expression errors:

- Wrong datatypes
- Wrong nodetypes
  (e.g. literals when expecting URIs)
- Pattern mismatches
  (e.g. expecting mailto: for foaf:mbox)
- Wrong URIs
  (e.g. ADMS status => NAL status in DCAT3)

Use SHACL Validation Reports?

Do we need an overarching structure in RDF, e.g. similar to EARL
(EARL is for accessibility reports)

Thank you!

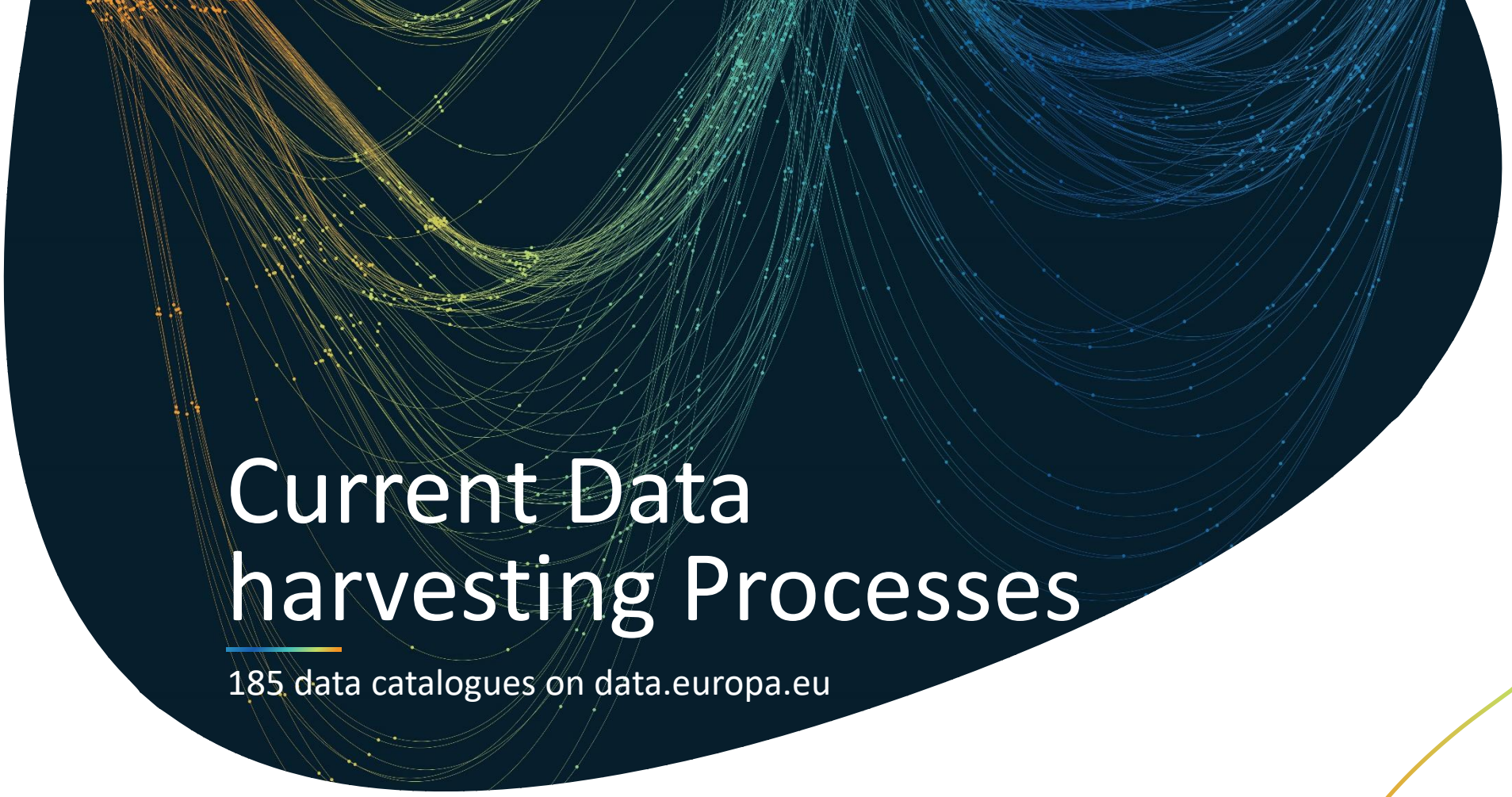
26 JUNE 2024

data.europa.eu - Linked data events stream

Dr. Simon Steuer,
Head of Sector, Publications Office of the EU

SEMIC2024

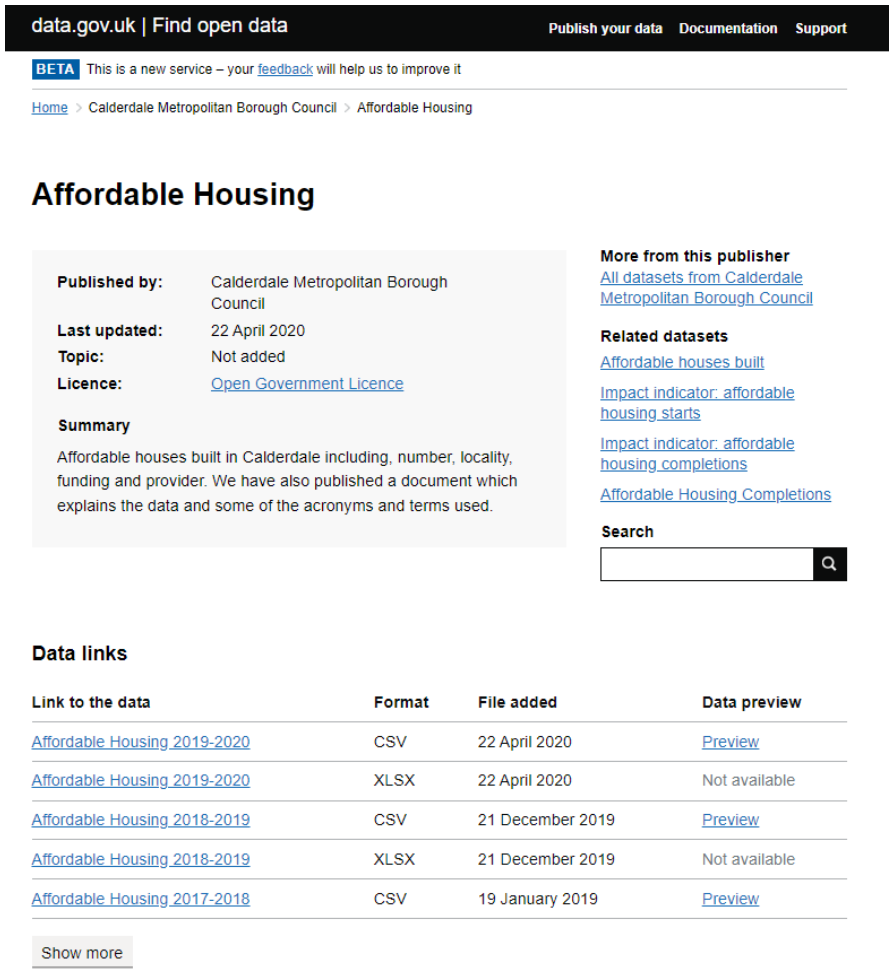
interoperable europe

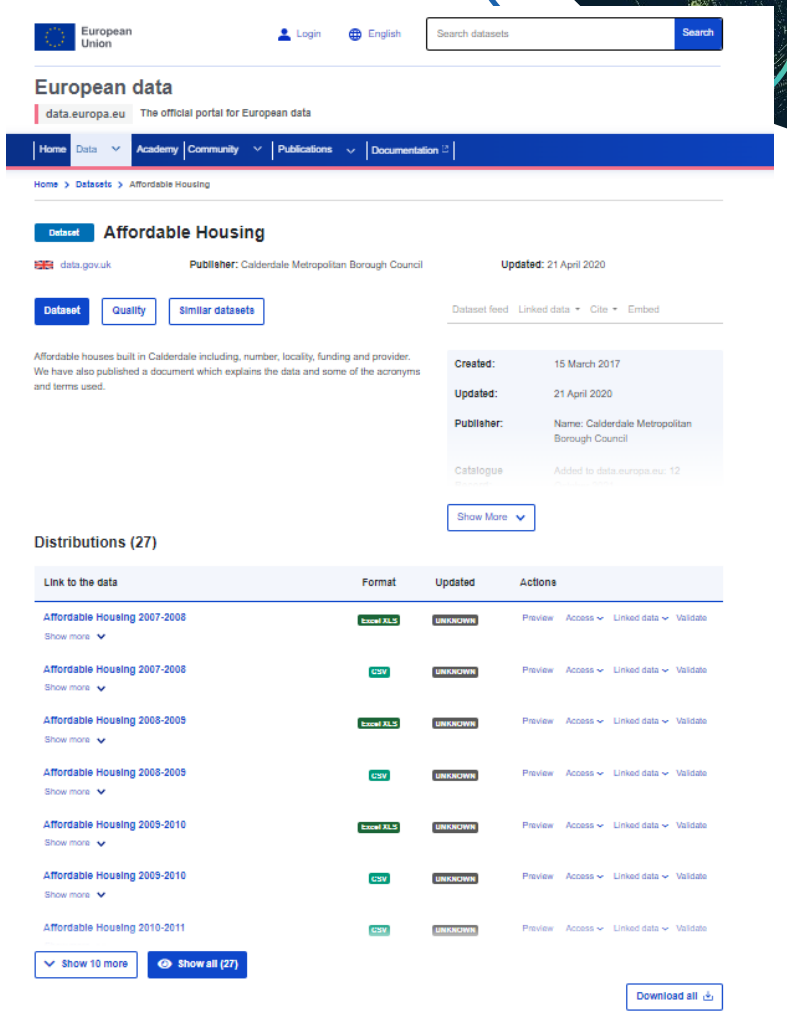


Current Data harvesting Processes

185 data catalogues on data.europa.eu

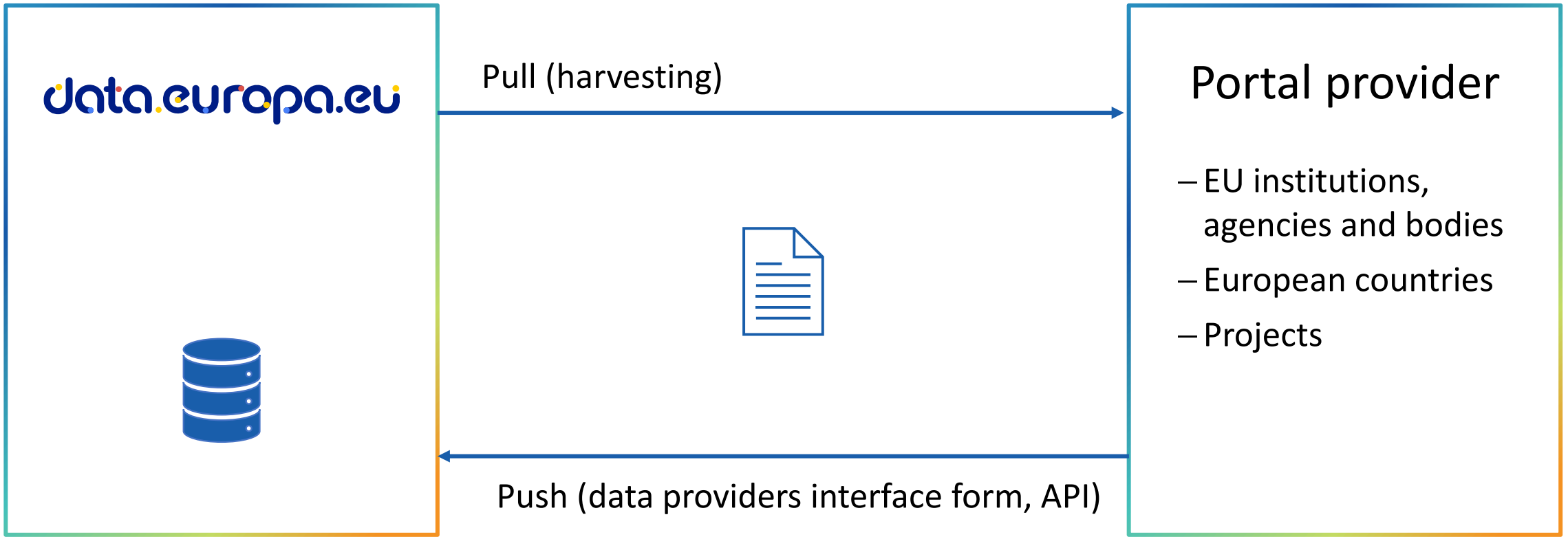
Current Data harvesting Processes



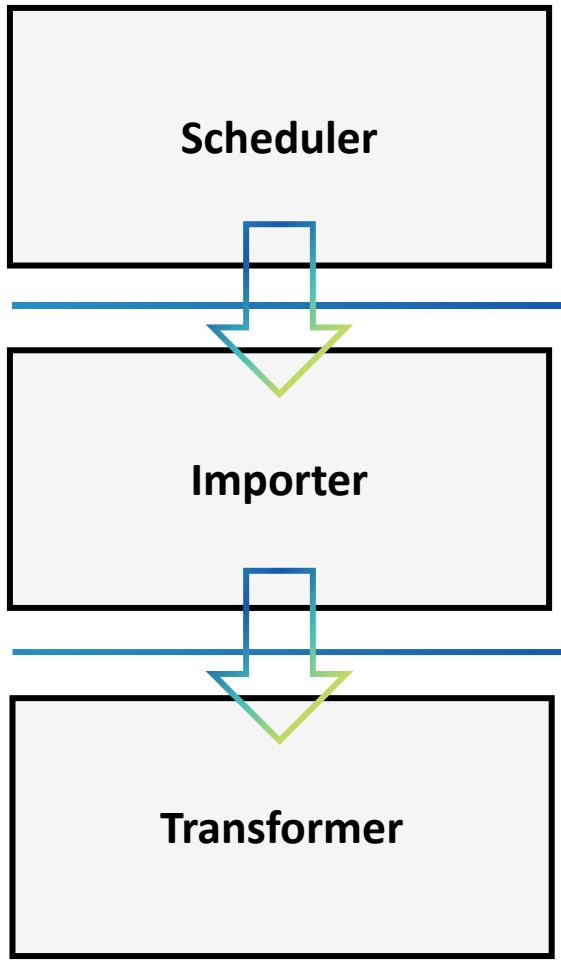


European Commission

Metadata retrieval



Data acquisition

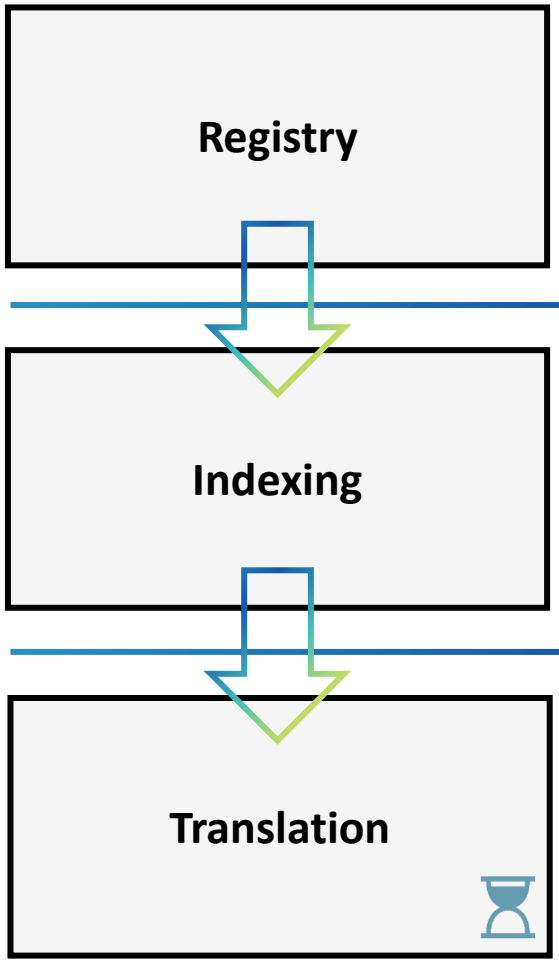


- Main entry point for the service orchestration.
- Periodically triggers the harvesting process, defined as a **pipeline descriptor.**
- Frequency: hourly, daily, weekly… depends on data provider

- 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)**

- 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

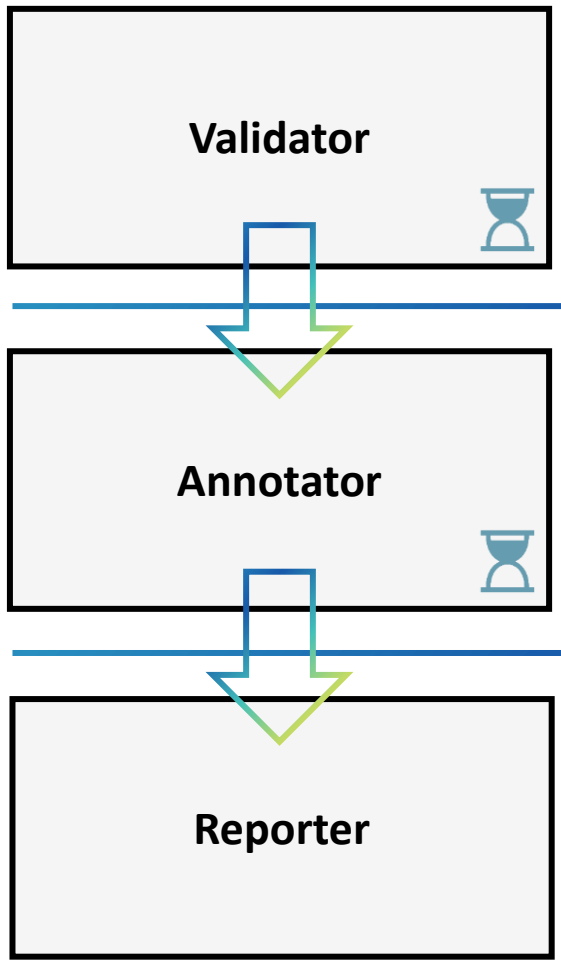


- Middleware and abstraction layer to interact with the triple store (**Virtuoso**).
- RESTful interface for RDF (**Turtle, JSON-LD, N-Triples, RDF/XML, Notation3**).
- Application of URI schemata, generation of unique IDs and inter-linking.

- 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**.

- 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



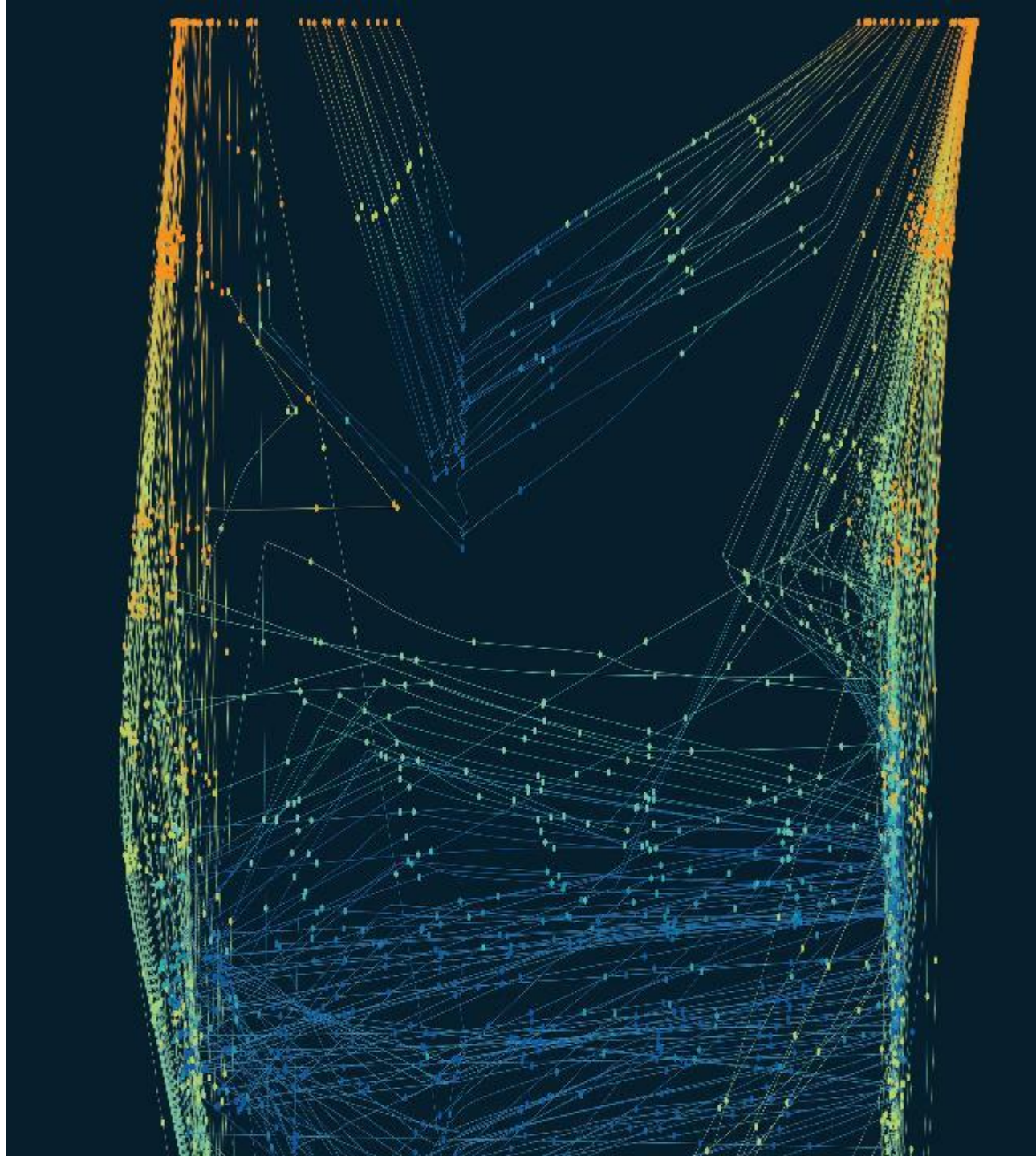
- Application of the **W3C SHACL**.
- Results include detailed information violations.
- Applied rules can also be extended or replaced (Built-in **DCAT-AP**).
- Accessibility tests on each linked distribution (the actual data).

- 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.

- 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

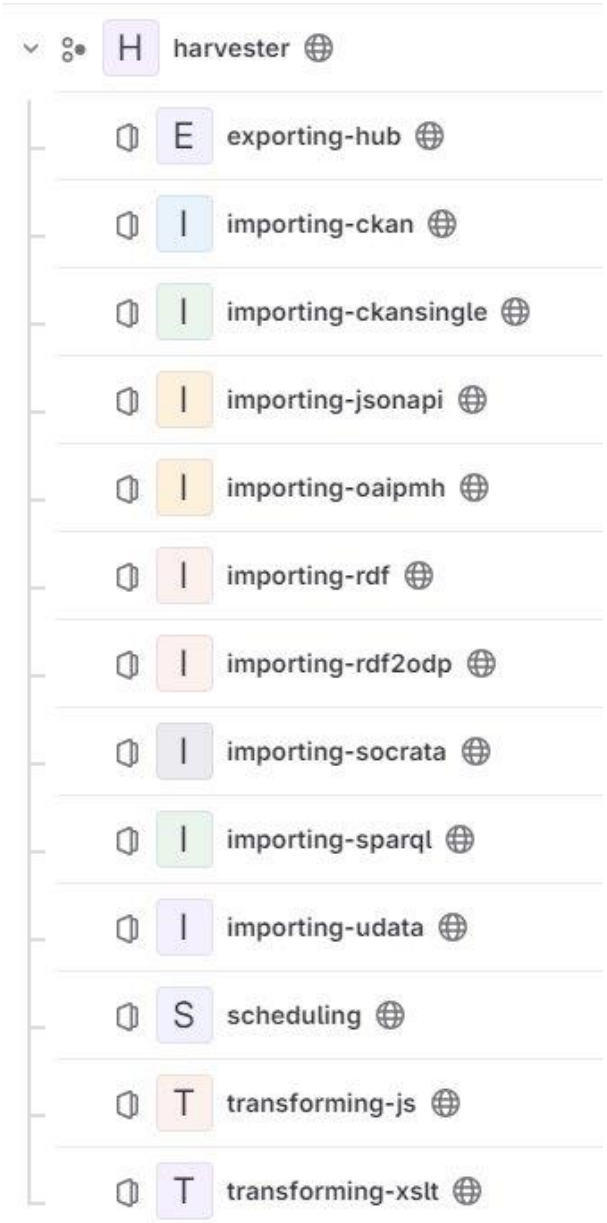
- When harvesting, we always store the original identifier in *dct:identifier*
- For internal handling, we create an additional identifier
- A URIRef based on the original identifier (our baseURI + “normalised” identifier)
- If the new identifier already exists, we add an increment at the end

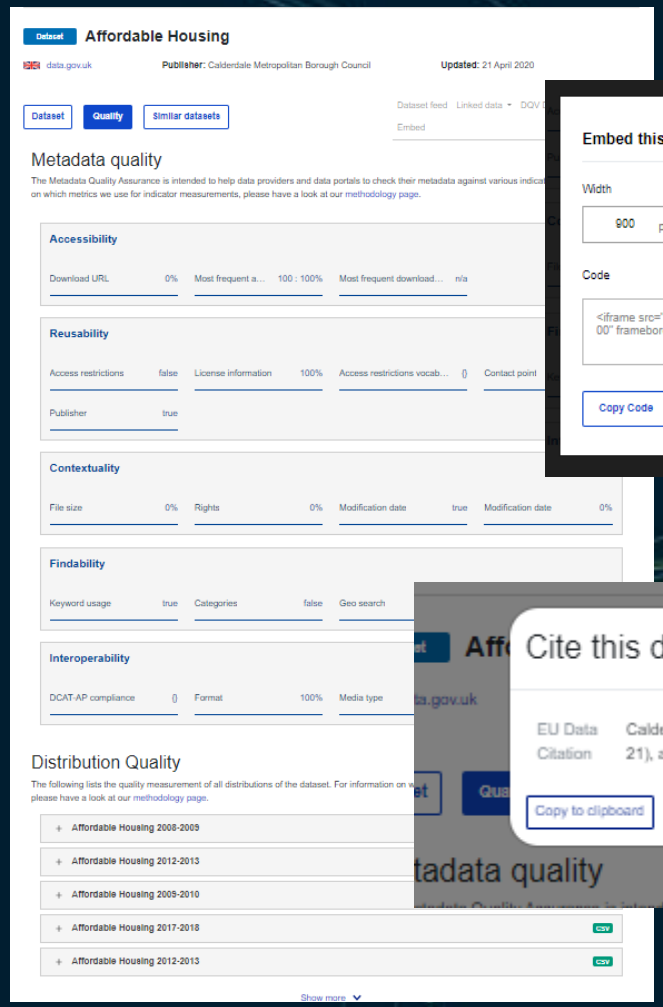


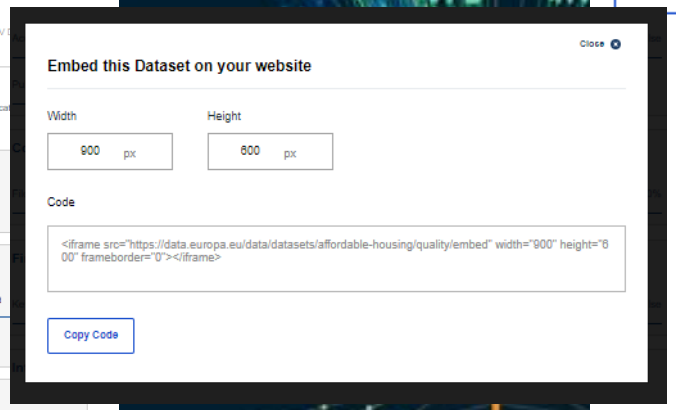
- A pipeline orchestration is described by a **descriptor**: a plain JSON document.
- It includes list of **segments**, where each segment describes a step aka a service.
- The descriptor is a compilation and self-contained description of a data processing chain.
- Each microservice must expose an endpoint 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.

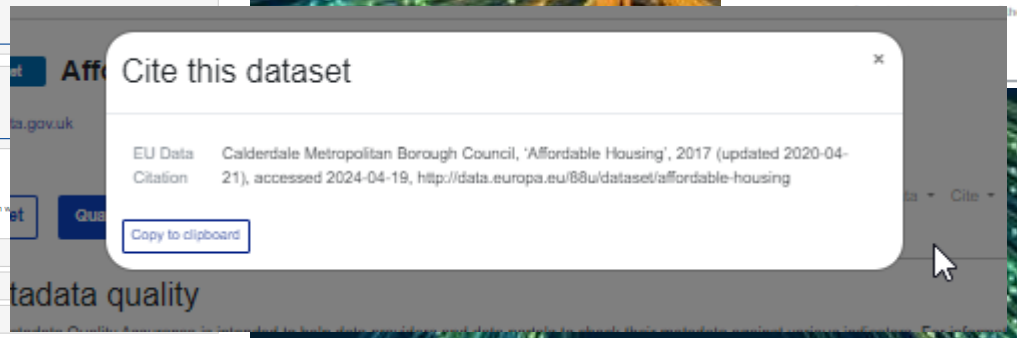
Software stack

- 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**

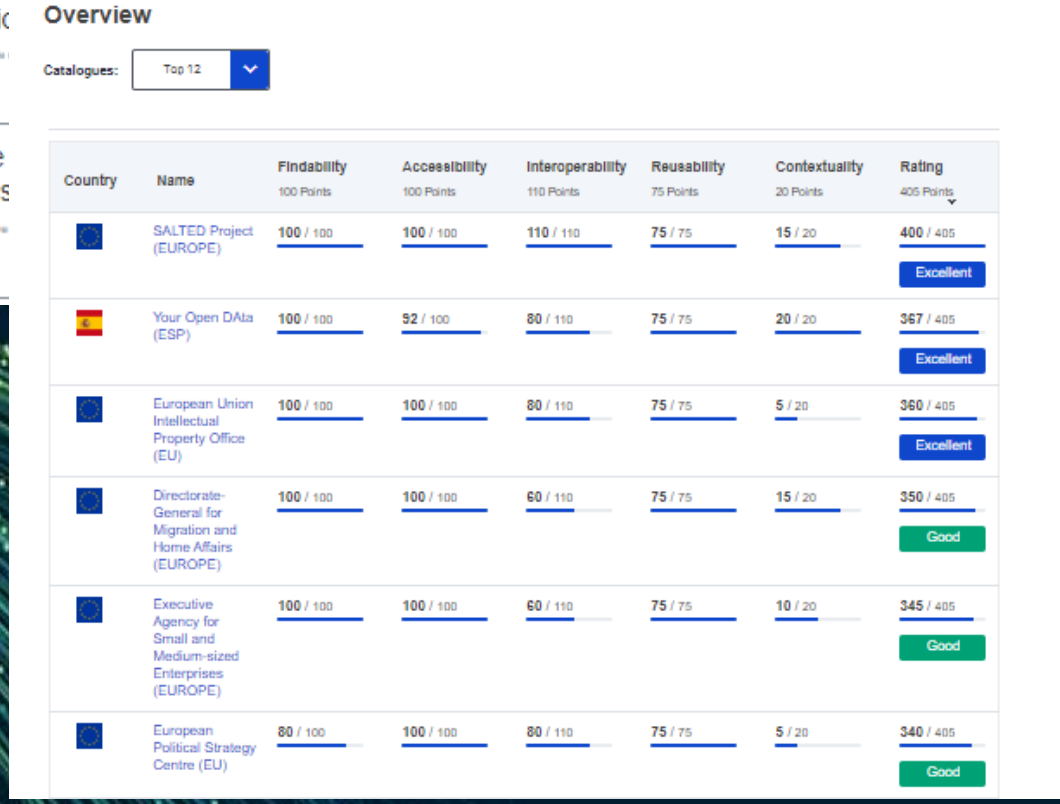


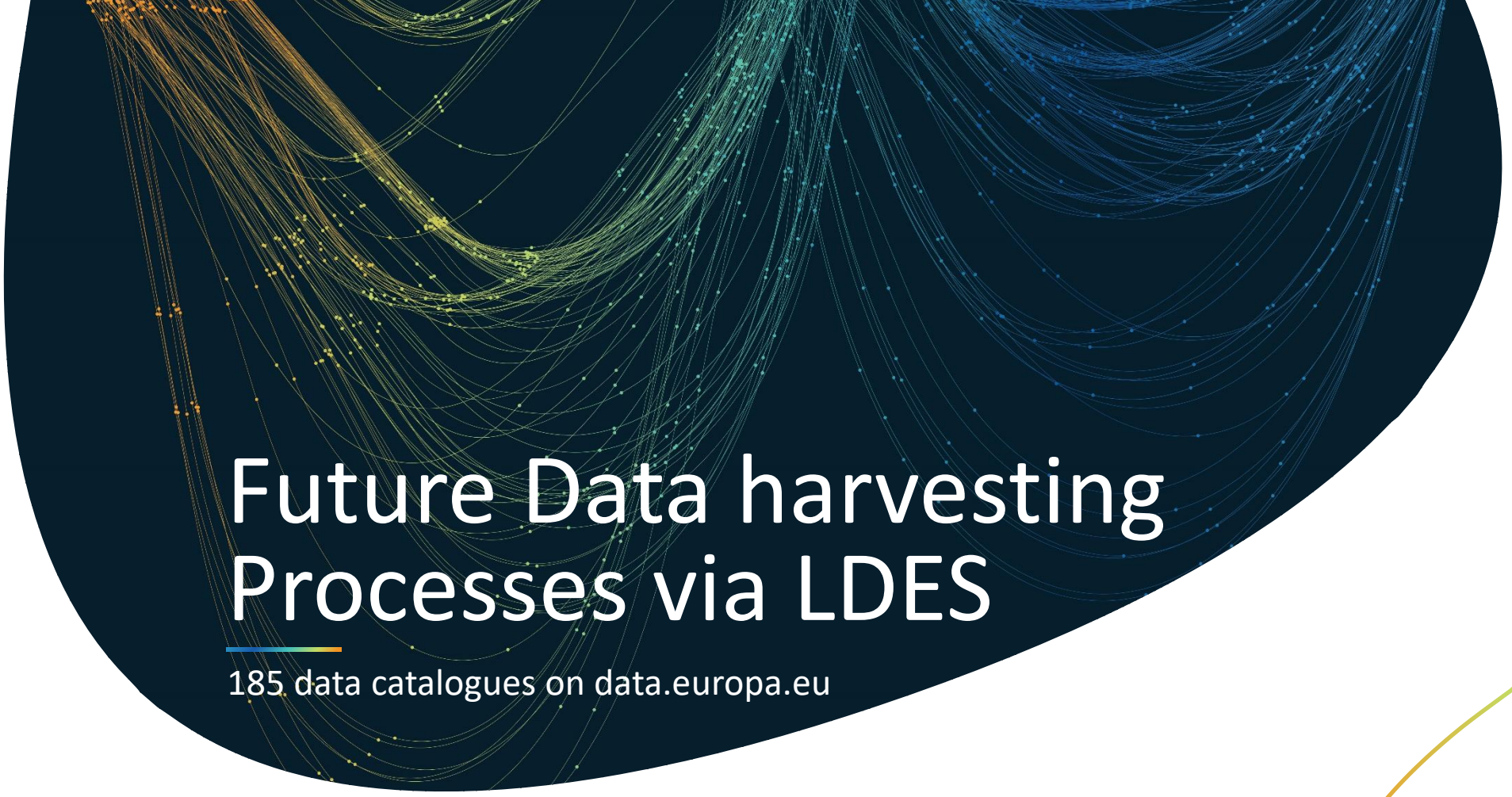

Dataset
Affordable Housing
data.gov.uk
Publisher: Calderdale Metropolitan Borough Council
Updated: 21 April 2020
Dataset
Quality
Similar datasets
Metadata quality
Accessibility
Reusability
Contextuality
Findability
Interoperability
Distribution Quality
Affordable Housing 2008-2009
Affordable Housing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

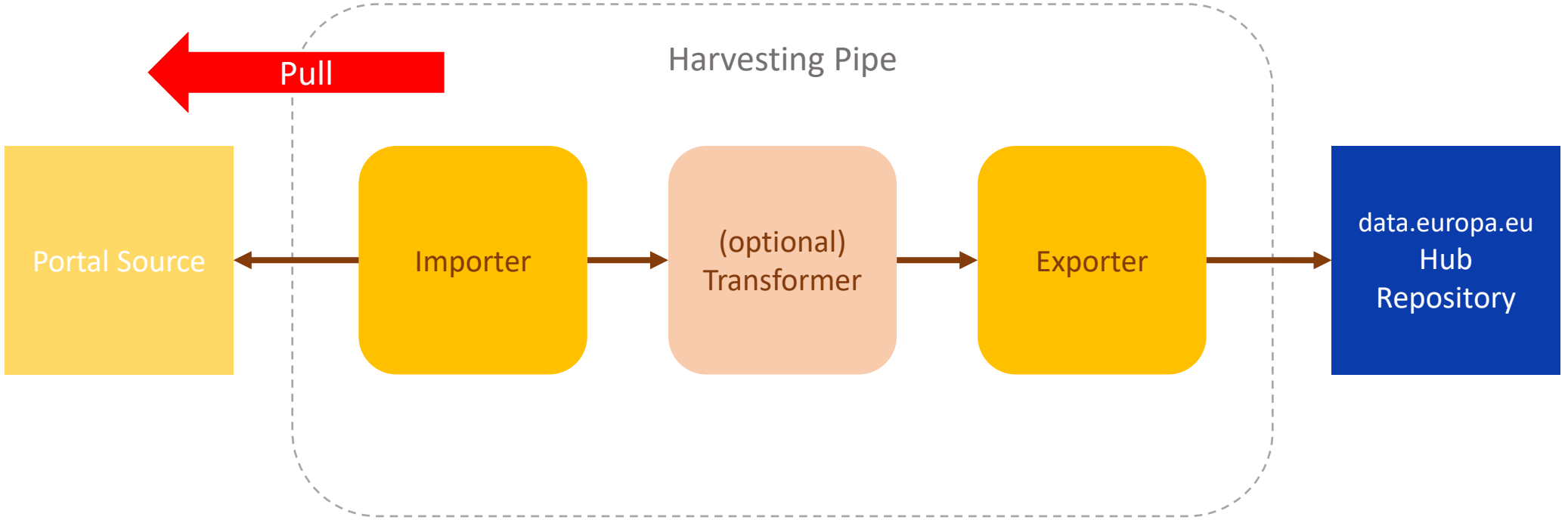


Future Data harvesting Processes via LDES

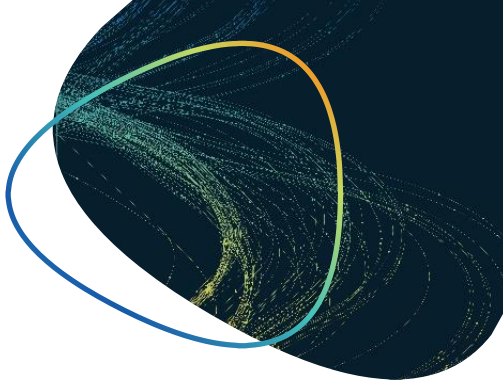
185 data catalogues on data.europa.eu

Harvesting on data.europa.eu

- Importer *pulls* always all datasets from a source, exporter checks if update of a dataset is really necessary
- Deletion happens as final phase at the end and is a comparison between identifiers



Technical Integration



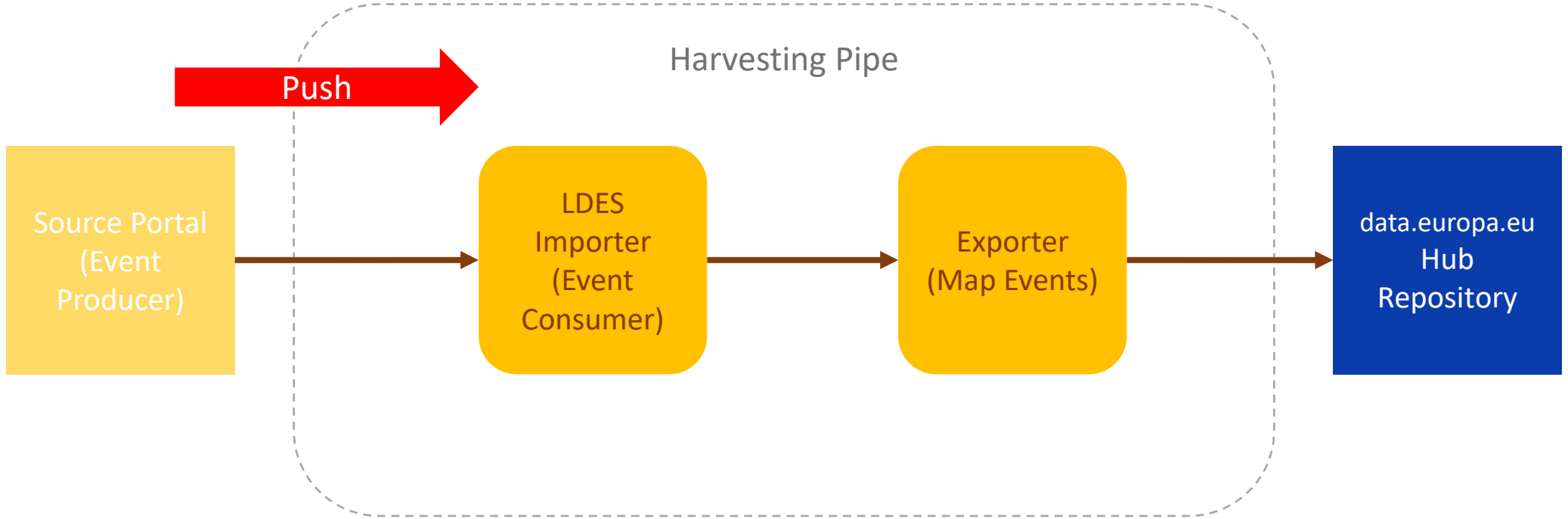
Two components involved

- New LDES Importer *pushes* Events to data.europa.eu
  - Receives events and maps them to data.europa.eu harvesting pipe: Consuming pipe descriptor and feed that pipe with LDES events

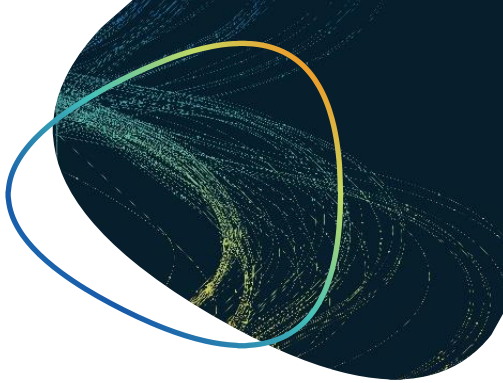
- Extended Hub Exporter processing LDES Events
  - Receives pipes with embedded LDES Events and map them to corresponding hub repository API calls

Technical Integration

- LDES Importer *pushes* single events to the harvester

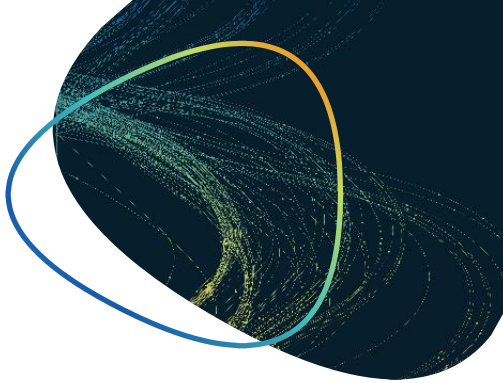


Technical Integration



- Only the following events are currently possible due to data.europa.eu back-end restrictions:
  - Create dataset
  - Update dataset
  - Delete dataset
- Distribution only events cannot be applied
  - Add distribution
  - Update distribution
  - Delete distribution
- Patching just a set of triples is not possible at the moment

Technical Integration

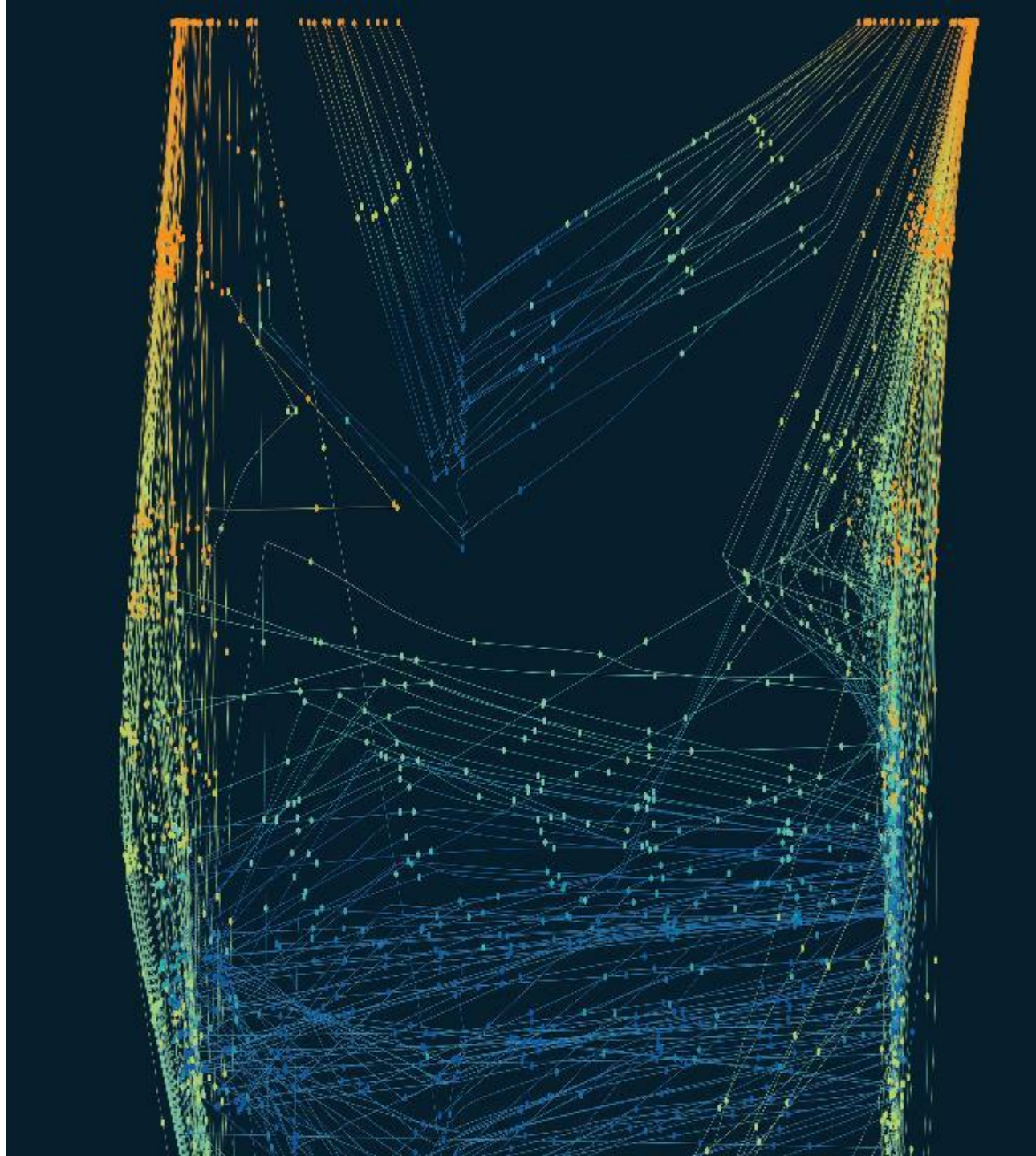


- Reduced load of the harvesting pipe. Only relevant changes are passed
- Reduced complexity on data.europa.eu side, because no check of update or not (triples hash) is necessary
- Deletion does not depend on successful run before comparison of identifiers can happen

- For the moment: Events are still collected and processed regularily and not immediately when happening (caching)

Future data harvesting process

- Add one more step to check for LDES metadata
- Reduce the harvesting to the new, updated and deleted datasets only
- Reducing load even more: Implementing smaller events e.g. for Distributions and Patches
- Reducing delay even more: Processing Events when they are “produced”
- Offer more details about history of datasets
- First tests are ongoing for data.europa.eu (currently in Pre-Production Environment) with the help of DIGIT



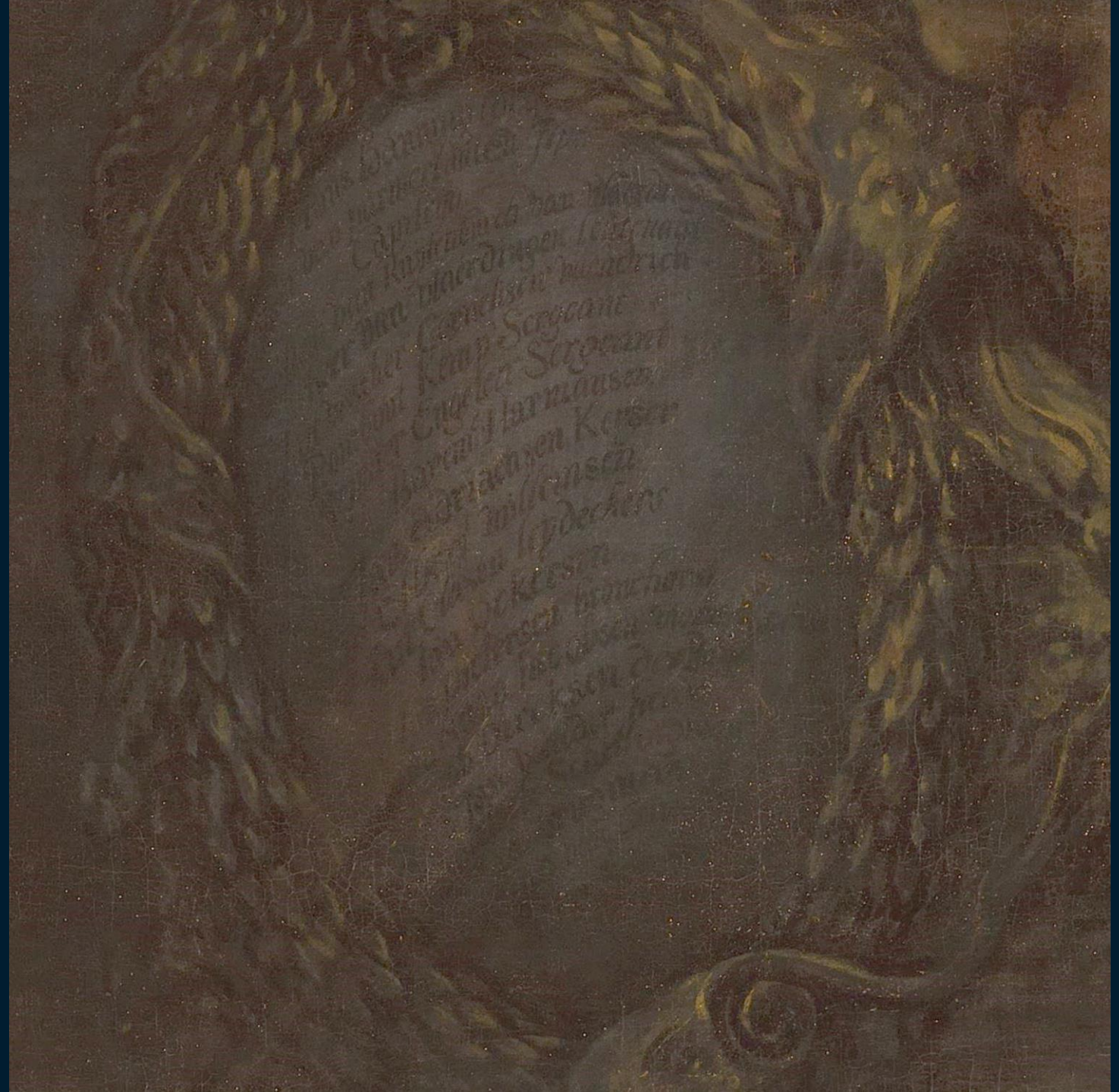

European Commission
26
JUNE
2024
THANK YOU !
SEMIC – Phase 15
interoperable
europe


European Commission
Linked Data Event Streams at
RIJKS MUSEUM
interoperable
europe

Tim Thomassen

Software Developer

Web- and API development, cloud computing and integration engineering



1.000.000

Objects

450.000

books

800

Meters of documentation

17

Terabytes of research data

1.000.000

Objects

450.000

books

800

Meters of documentation

17

Terabytes of research data

Library System

Collection Management System

Document Management System

Agile methodology

Fast development cycles

Main focus of work

Data, code and infrastucture

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

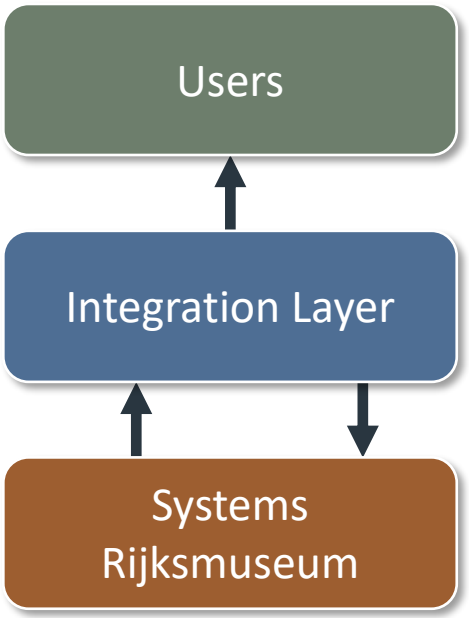
Create predictable data services

Validation

Guarantee data quality

Synchronisation

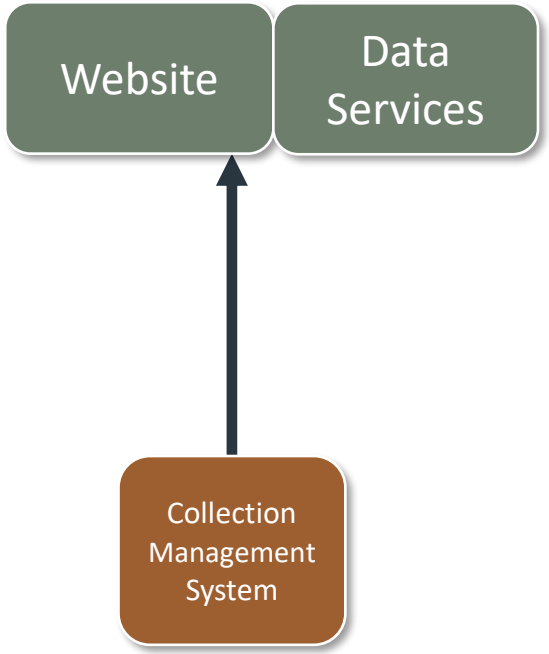
Keep data up-to-date



Standardisation

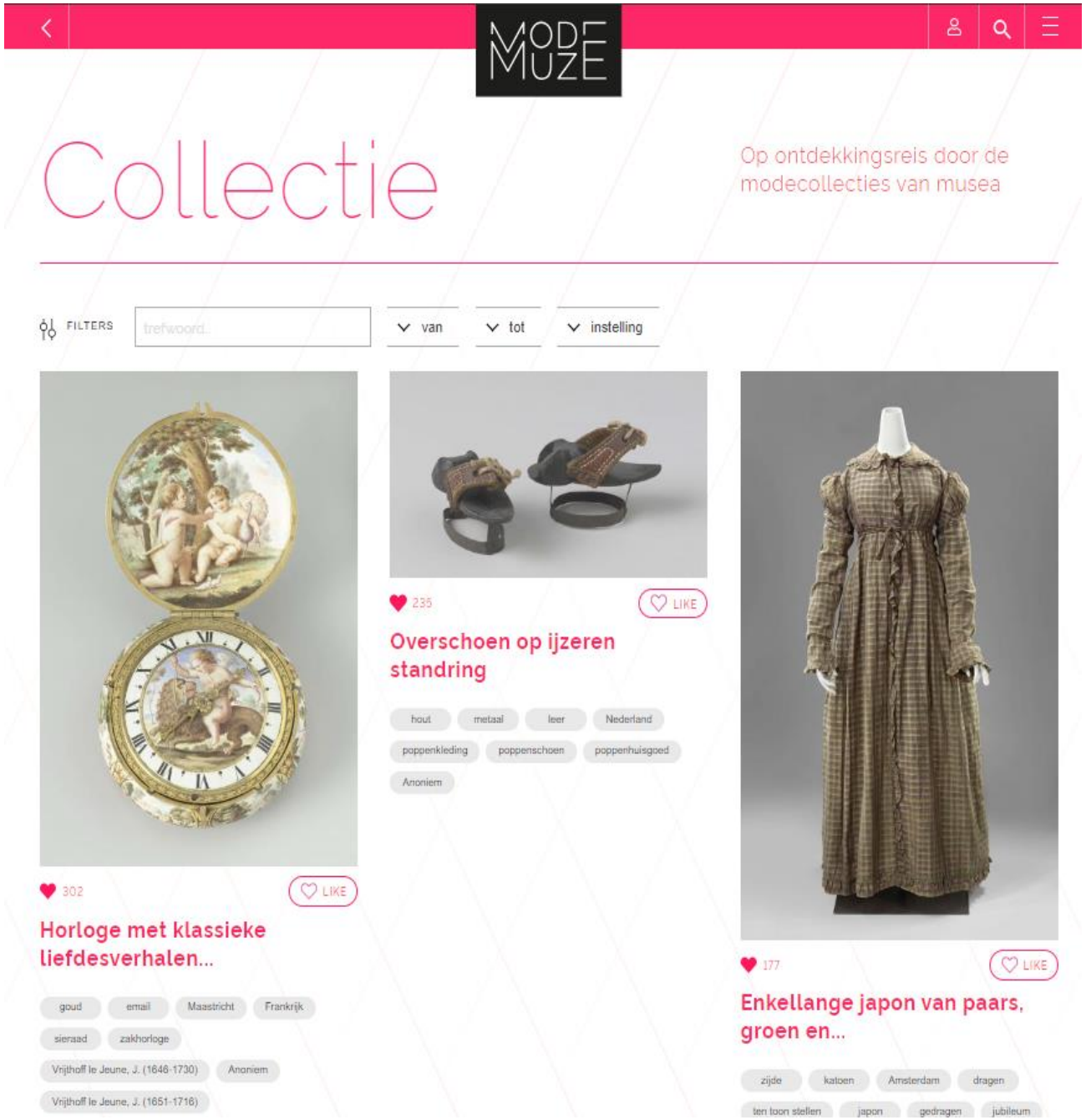
Current situation

- **System specific** data structures and communication protocols
- Data services managed by **external web design company**



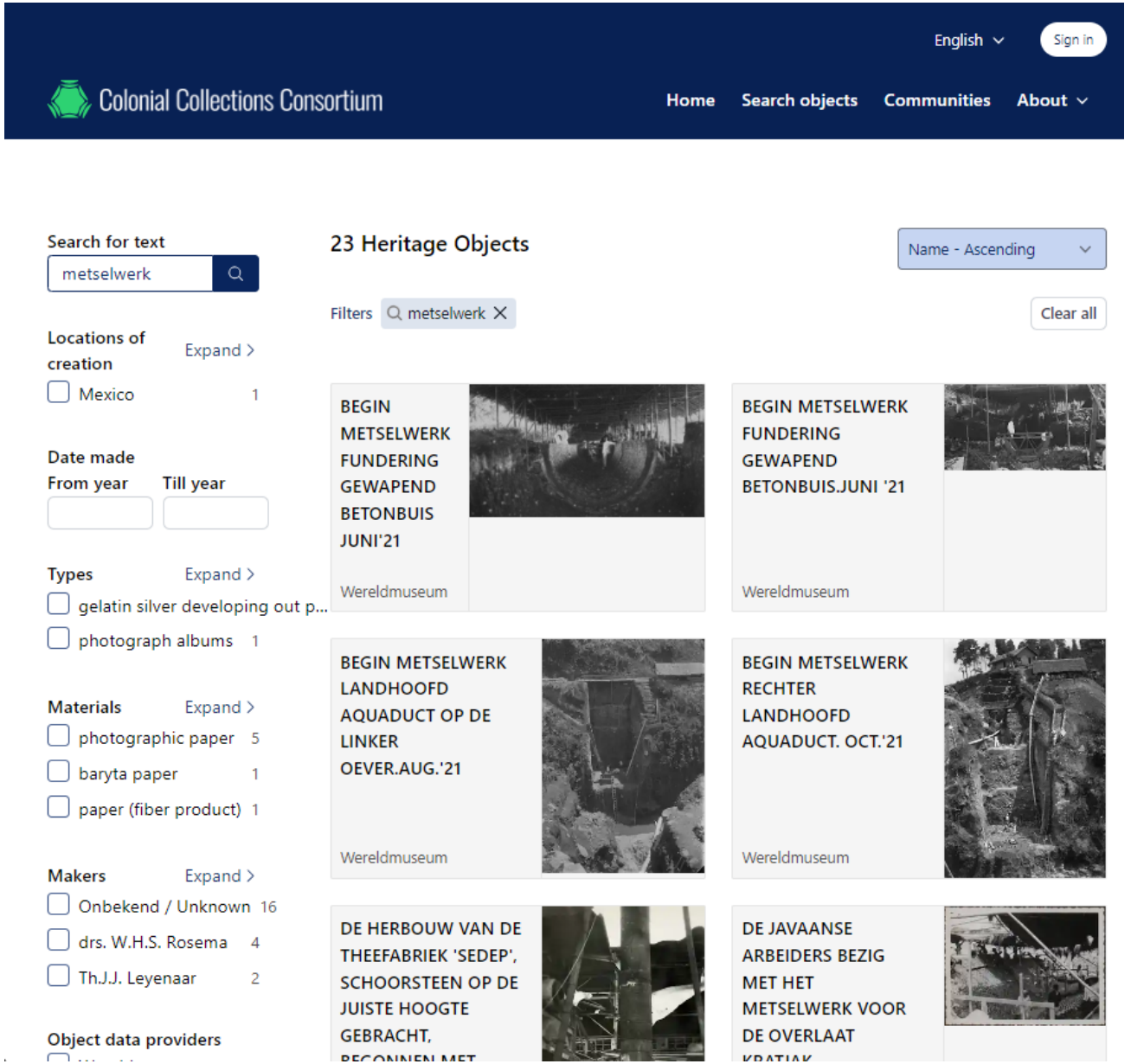
Aggregators

- **Modemuse** (OAI-PMH)
- **Europeana** (OAI-PMH)
- **KVVAK** (JSON)



Against Opacity Datahub

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



Standardisation

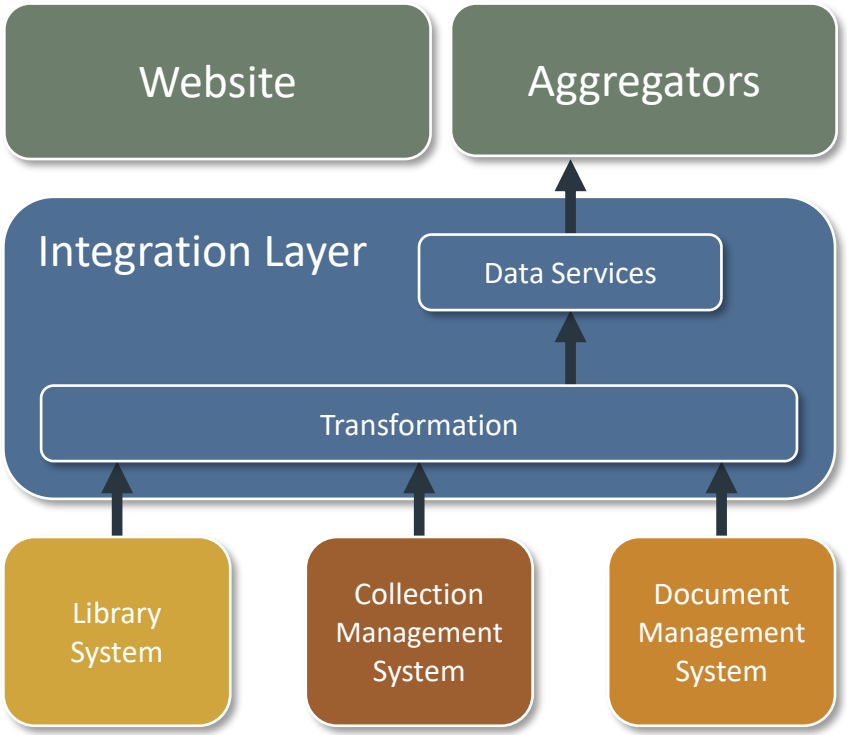
Future situation

System specific data structures 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

Microservices Architecture

How

Microservices

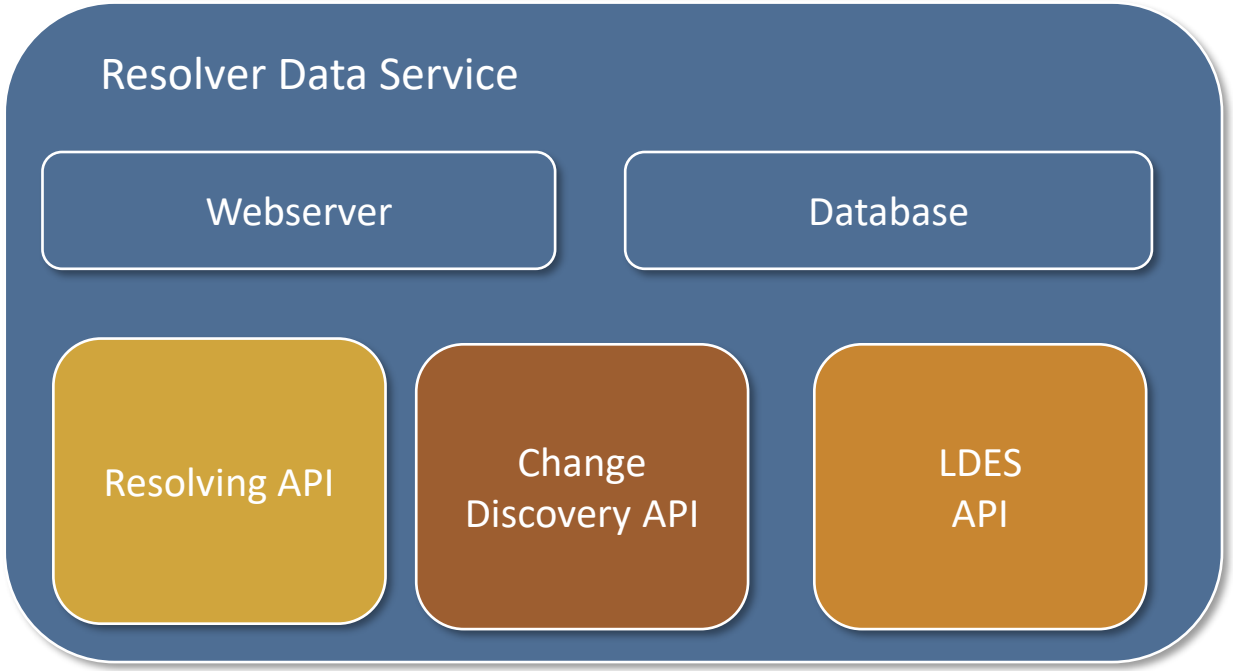
Software split into small parts

Docker Containers

Software packaged so it can be easily deployed on servers

Kubernetes Cluster

Establish relations between parts, improve reliability and make software scalable



Common setup

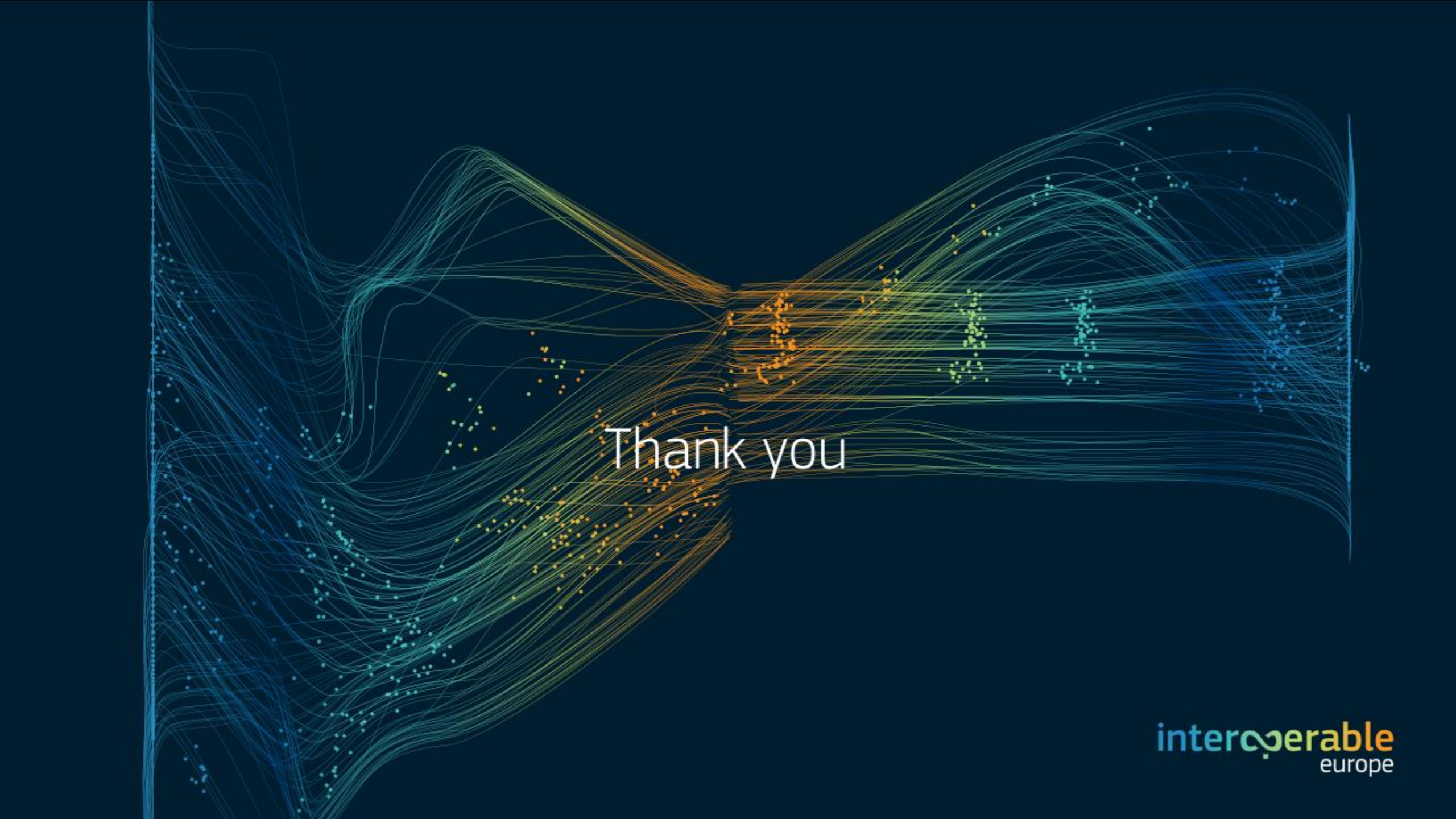
- Webserver
- Database

```
# 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

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

Multiple API configurations

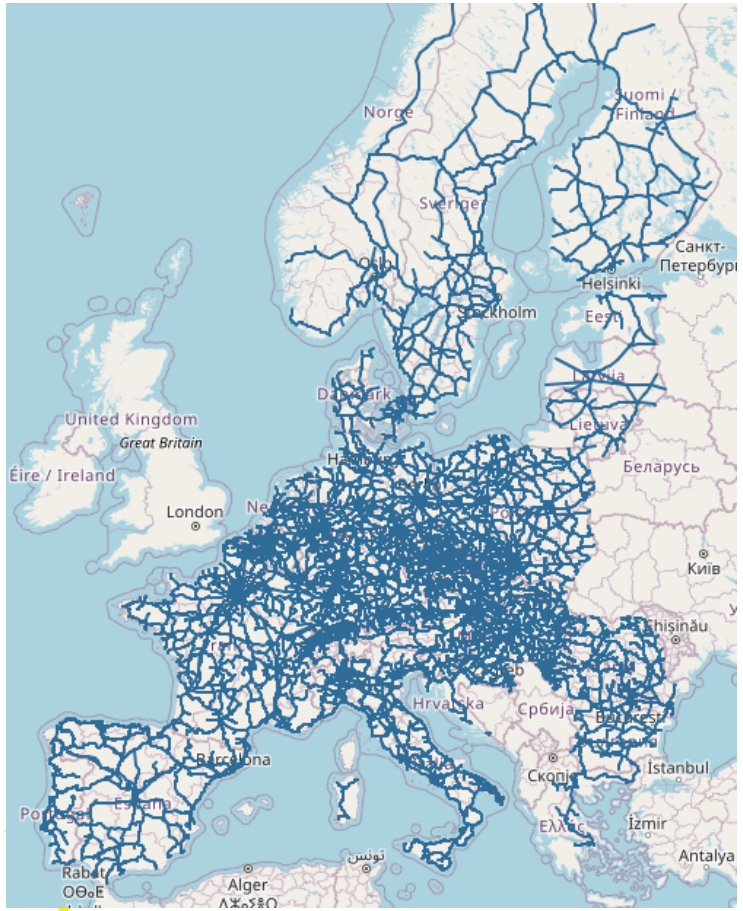


interoperable europe

LDES @ERA

Marina Aguado – European Union Agency for Railways
Julian Rojas Melendez – IMEC/Ghent University

European Railway Network



ERA KG

- More than 47 million triples
- More than 31k lines of mappings
- More than 100 SHACL shapes
- +270k track segments described
- +50k stations described
- +50k geo-referenced objects (lat/long)
- +2k Vehicle Types described
- 27 countries covered (EU countries)

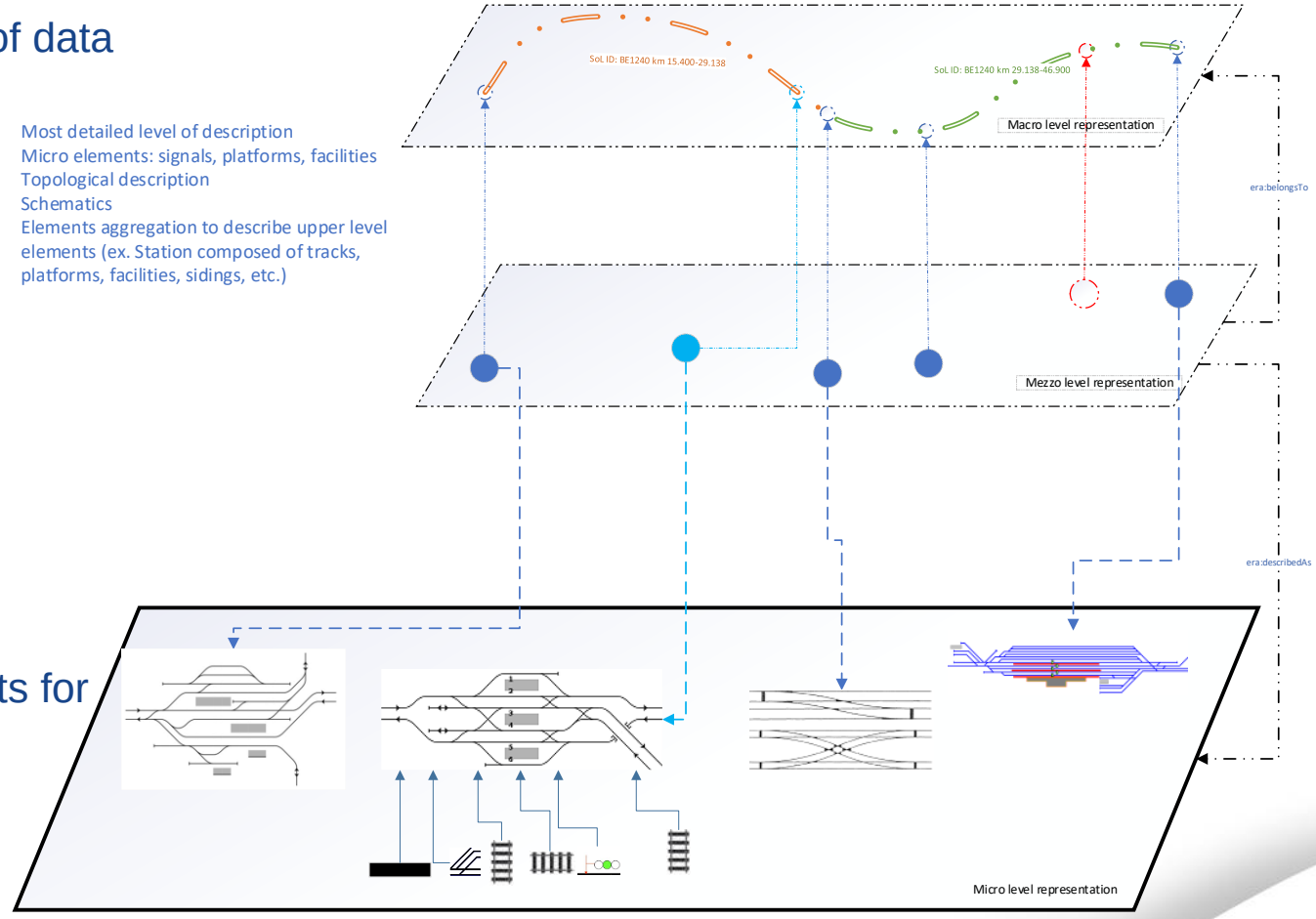
ERA — Map explorer (europa.eu).
https://data-interop.era.europa.eu/

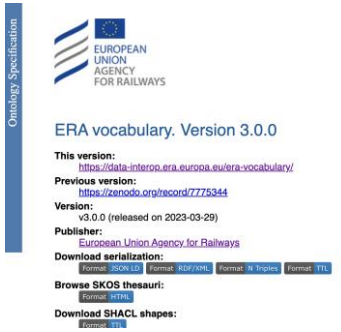
ERA — Route Compatibility Check (europa.eu)

GIS component .. zooming in and out

- Different levels of granularity of data

Most detailed level of description
Micro elements: signals, platforms, facilities
Topological description
Schematics
Elements aggregation to describe upper level elements (ex. Station composed of tracks, platforms, facilities, sidings, etc.)





- Topological and topographical representation of railway infrastructure

- Model the temporal constraints for the technical characteristics

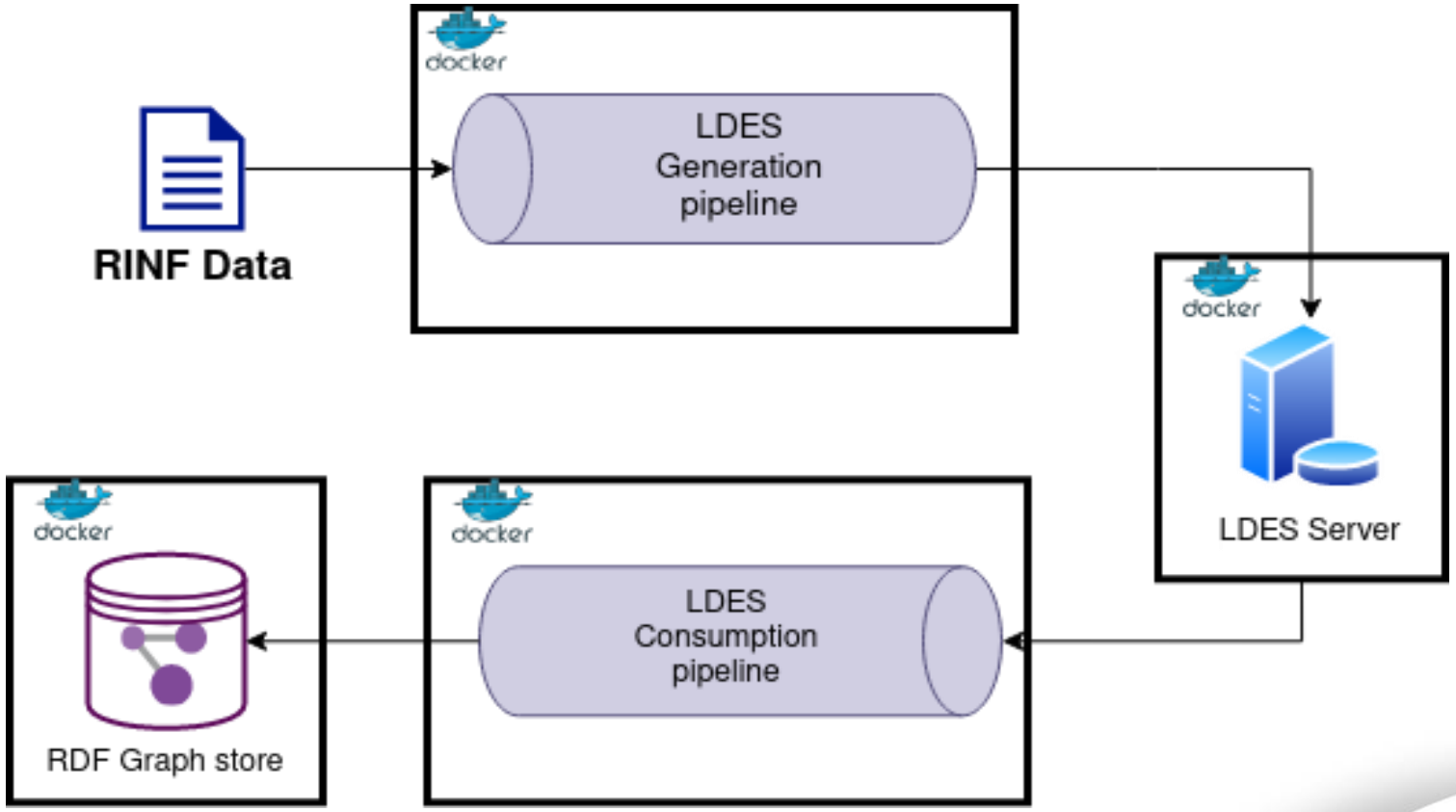
A Time Machine Functionality for Linked data

Traceability of changes in the railway network for data provision and data consumption
Notification of changes to consumers
Query of changes to identify potential risk factors in drivers
Backup changes
Timestamping data sets and route compatibility checks / routebooks…

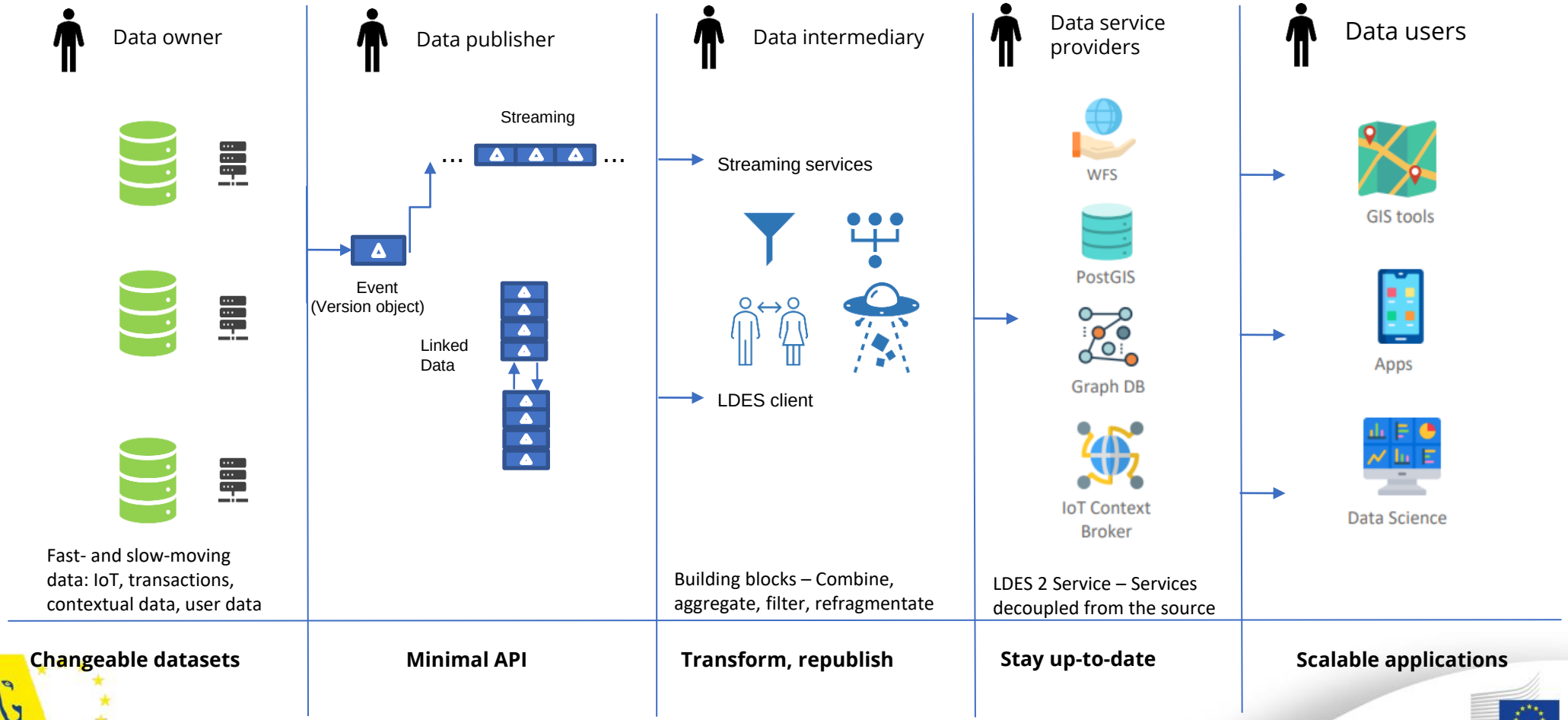
Technical challenges to overcome

- Generic stateful (delta-aware) approach for RDF generation
- Efficient continuous LDES writing/reading
- Generic and reusable pipeline creation and execution

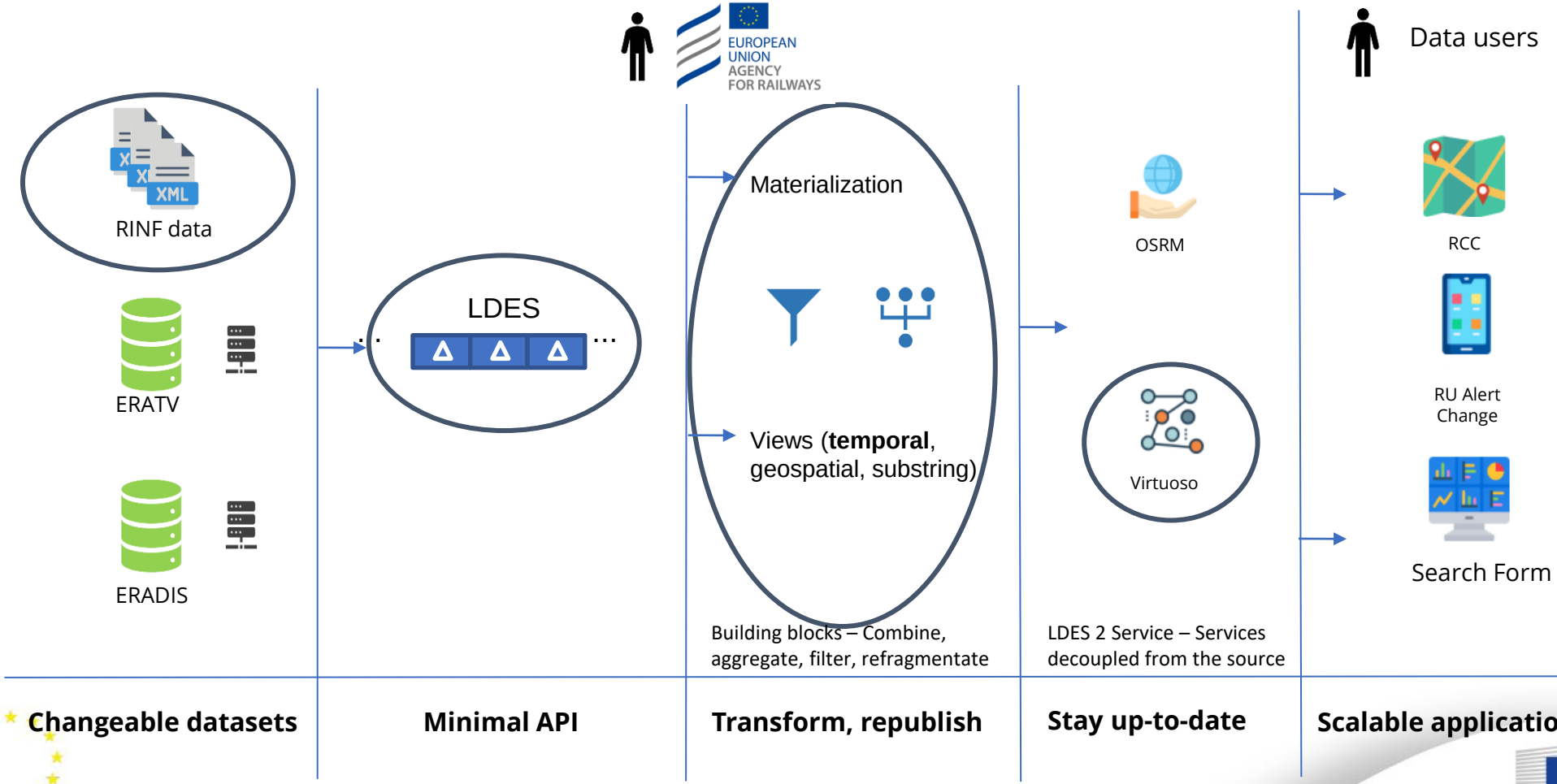
Implementation



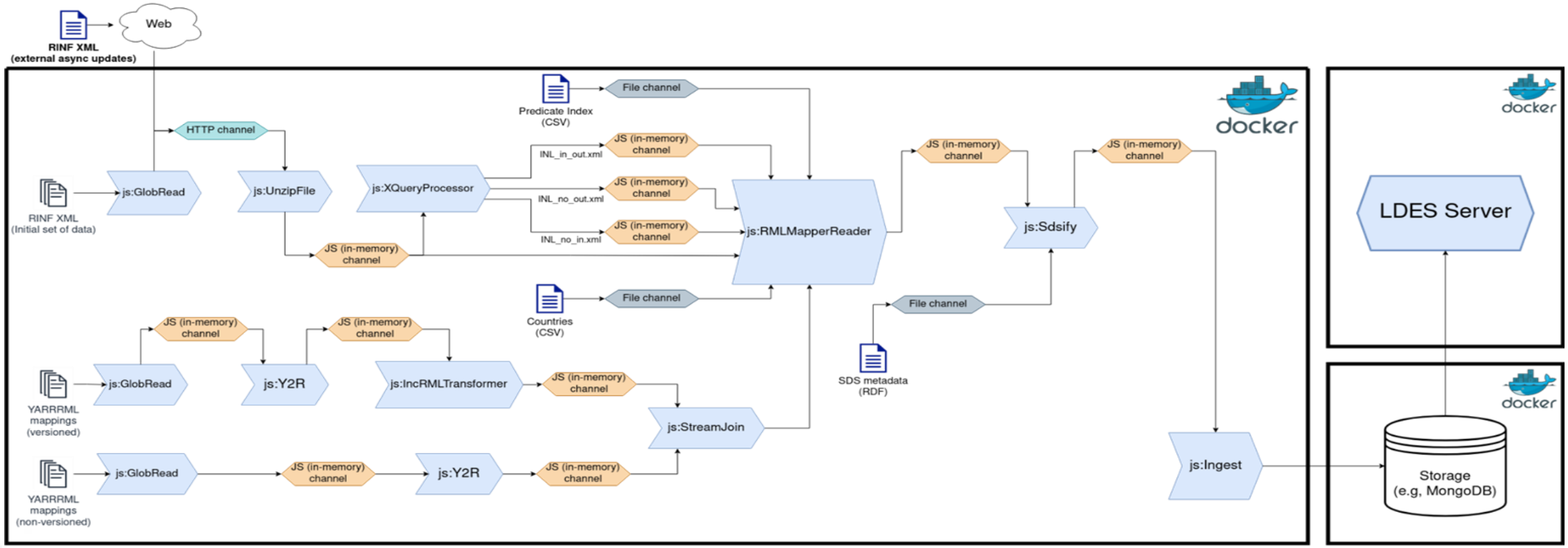
Generation Pipeline



ERA's LDES ecosystem



Generation Pipeline



Demo Time

Thank you!


SEMIC 2024
conference
Brussels 27 June
interoperable
europe
from Vision to Reality
be EU
belgium24.eu
European Commission

LDES For Vocab Terms

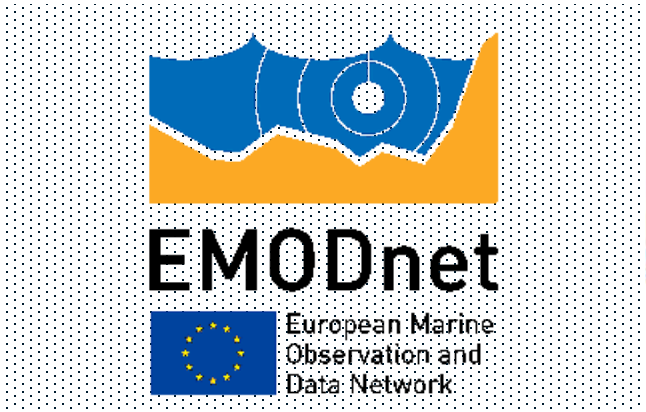
2024-06-26 pre-Semic.eu 2024

Marc Portier, ir.
Open Science Team Lead, VMDC, VLIZ vzw

Term Lookup Services
& Translation Management
for Marine VOCABs

130

credits



MareGraph
Towards an Interoperable
Marine Knowledge Graph

DIGITAAL VLAANDEREN

Vlaamse overheid

imec GHENT UNIVERSITY

Consiglio Nazionale delle Ricerche

MARIS

BODC

VLIZ

Topics

- Building the Science Knowledge Graph (SKG)
- Term Lookup Service & Widget
- LDES for Vocab Terms
- Translation Management

The «Felicity Jones» effect

There is only one web.
Not a separate one for Science.

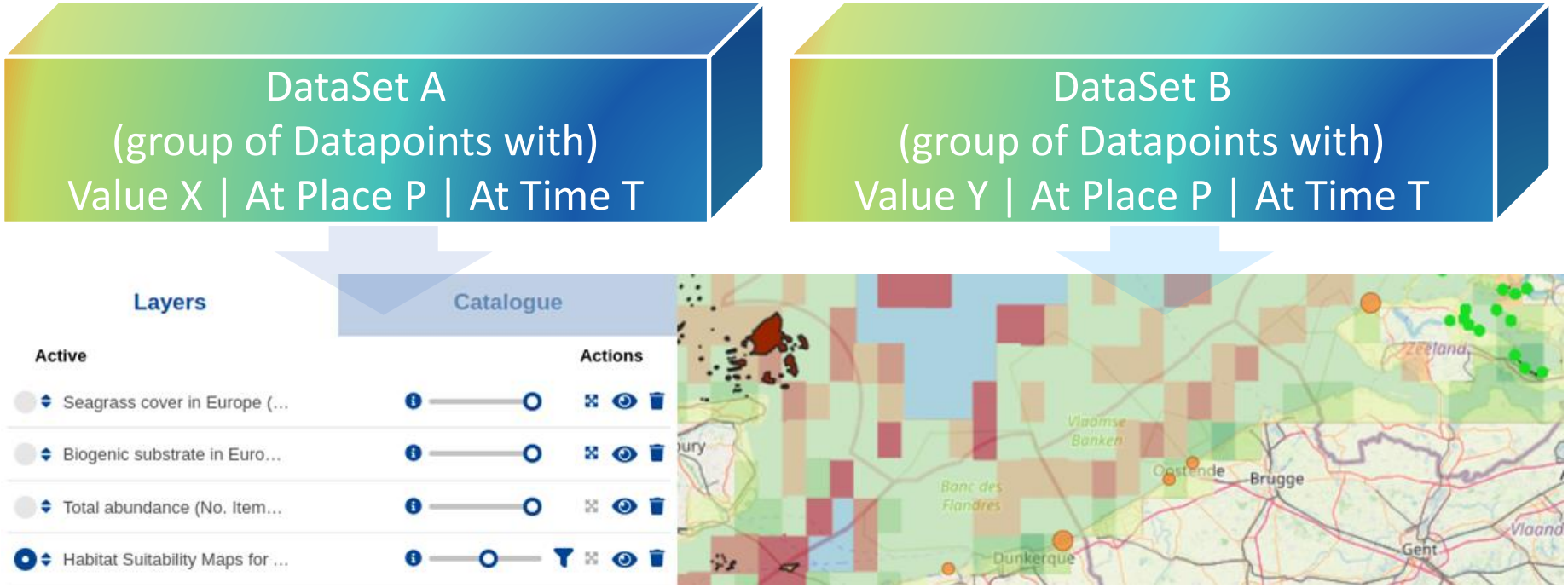
#RuleZero ::
If you're doing it (only) for the scientists, you're doing it wrong!

Compared to this Academia / Research is failing twice
(1) itself: wasting expert resources on trivial work
(2) society: public platforms not fed by quality research data

OpenScience / OPSCI-1
provide a relevant slice of the global research dataset that is as simple as a google search

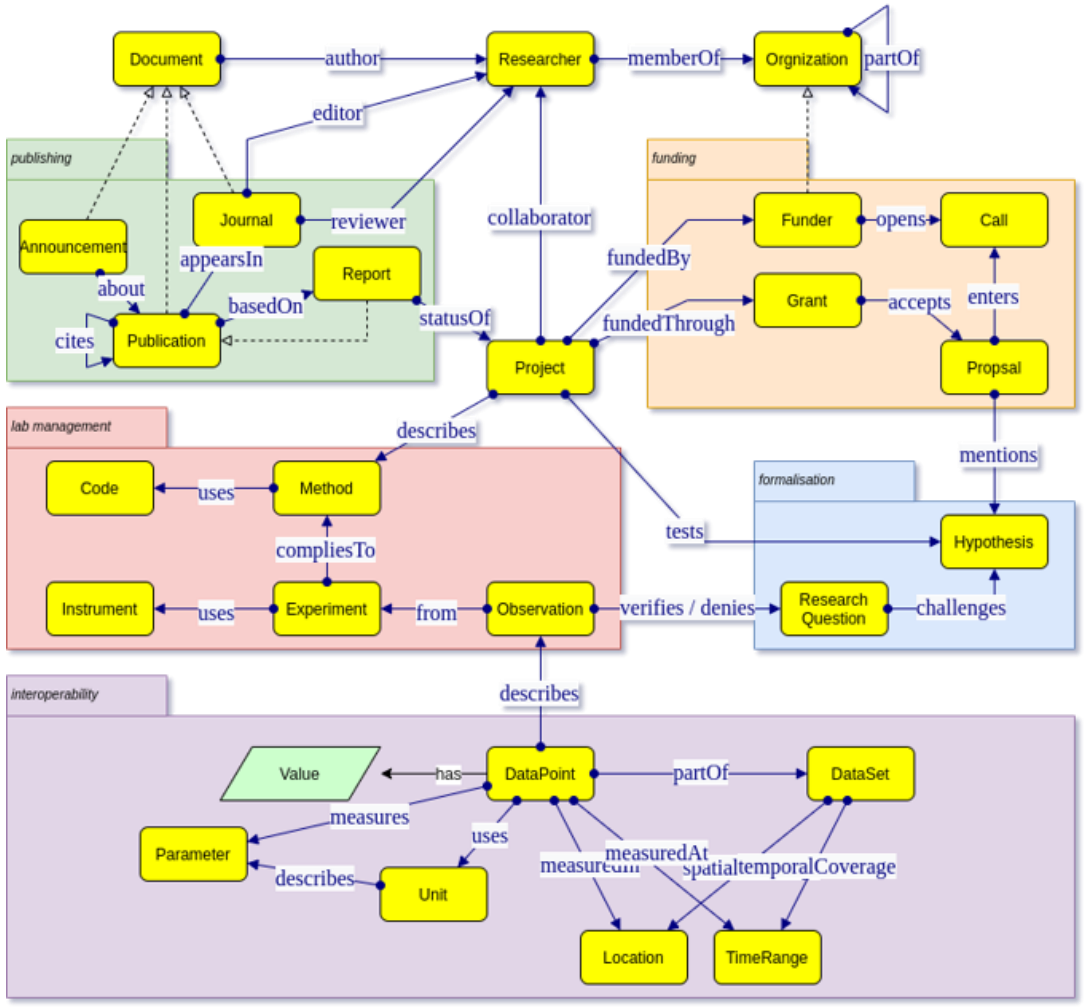
Known approach in Data Analysis

- Compare different values (like layers in a map)
  --> datapoints with some shared geo-temporal properties
- To discover some hypothesis connection / causal relation



Connecting the dots

- The case for LOD in Open Science
- Adding all aspects of observations:
  - Where, When,
  - Value, Unit, Instrument, Platform, Procedure,
  - Taxname, Class, Function, Attribute, Trait,
  - Maintenance, Storage, Lab, Datasets,
  - Question, Hypothesis,
  - Project, Funding, Grant, Organisation, Person...
- … and how they are connected



Providing more inference options

- To extend the range of "coincidence" axes / dimensions
  - Not only geo-temporal (Where and When)
  - But along all conceivable links
- Enables: to detect more "connections"
- Feeds the Research Cycle:

  ⇝ [*Statistics*] ⇝ [*Coincidence*] ⇝ [*Hey, that's funny*]

  ⇝ [*Hypothesis*] ⇝ [*Causation*] ⇝ [*Knowledge*] ⇝ [*Policy*]
- Supporting unpredictable rehashing, mashups, ...

MareGraph.EU

MareGraph
Towards an Interoperable Marine Knowledge Graph

https://marineregions.org/

Gazetteer

Placenames (~70k), typology, geometries, relations.

Since 2021: LOD + LDES (collab with KNoWS / IMEC)

- https://marinespecies.org/
- Taxname Register
- Names (~500k), Classification, Vernaculars, Documented (Proof), Habitats, Traits

https://eurobis.org/

BioDiversity Data

Occurences (~30M), Date&Geo Bound, Events, CDT Measurements

Workflow for DWC-A ingest and publication

137

LOD publication of High Value Datasets

Making a tighter web of data

Interlinked with other Existing Reference Sets

- BODC / NERC Vocabulary Service
  - Measurements
  - Instruments, platforms, ...
- SeaDataNet Directories
  - Projects, Organisations,
  - Cruises, Obersving Systems, …
- Common Research Platforms: orcid.org, ror.org, ...

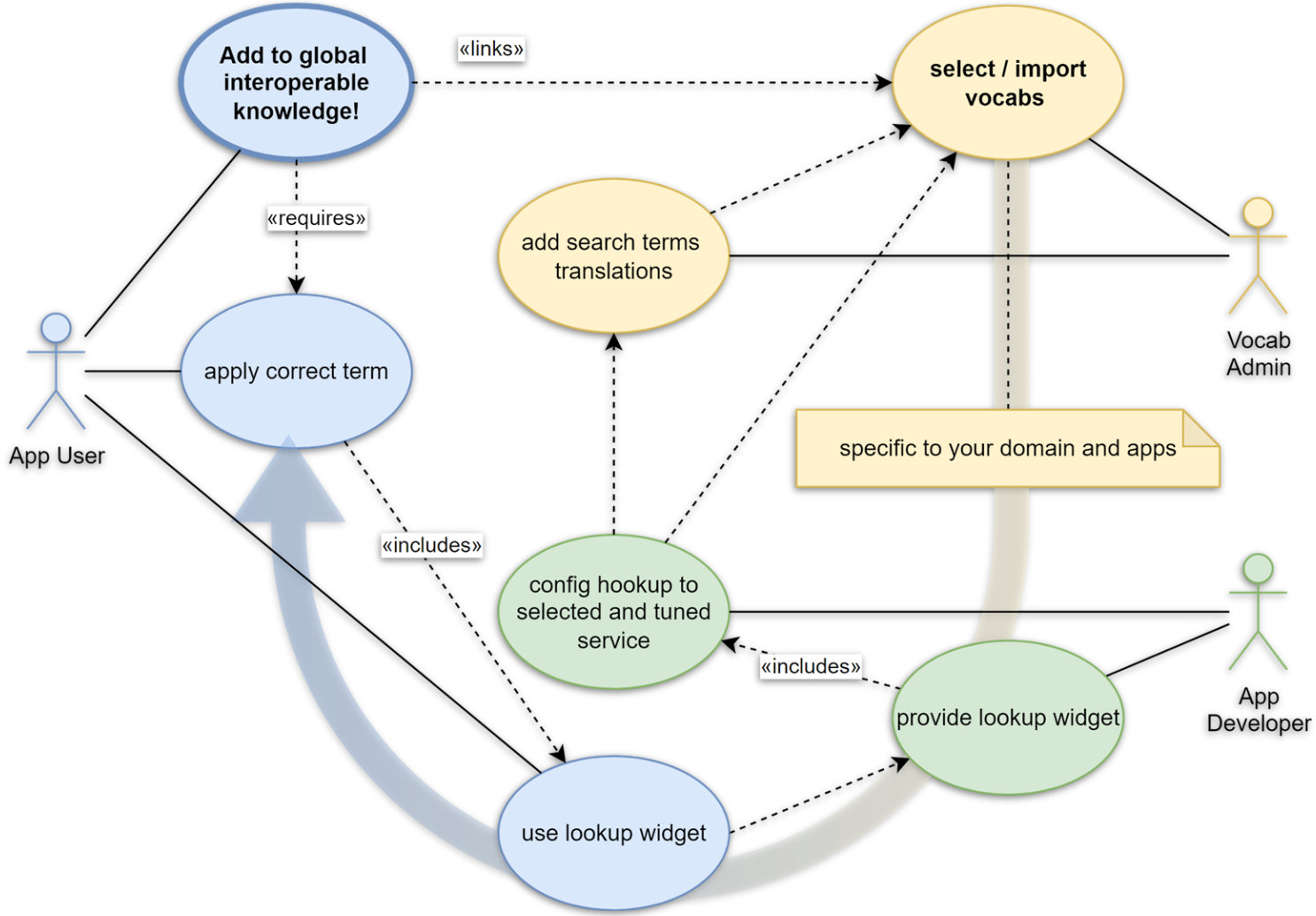
Globally connected:

- to the IODE / ODIS / Ocean InfoHub

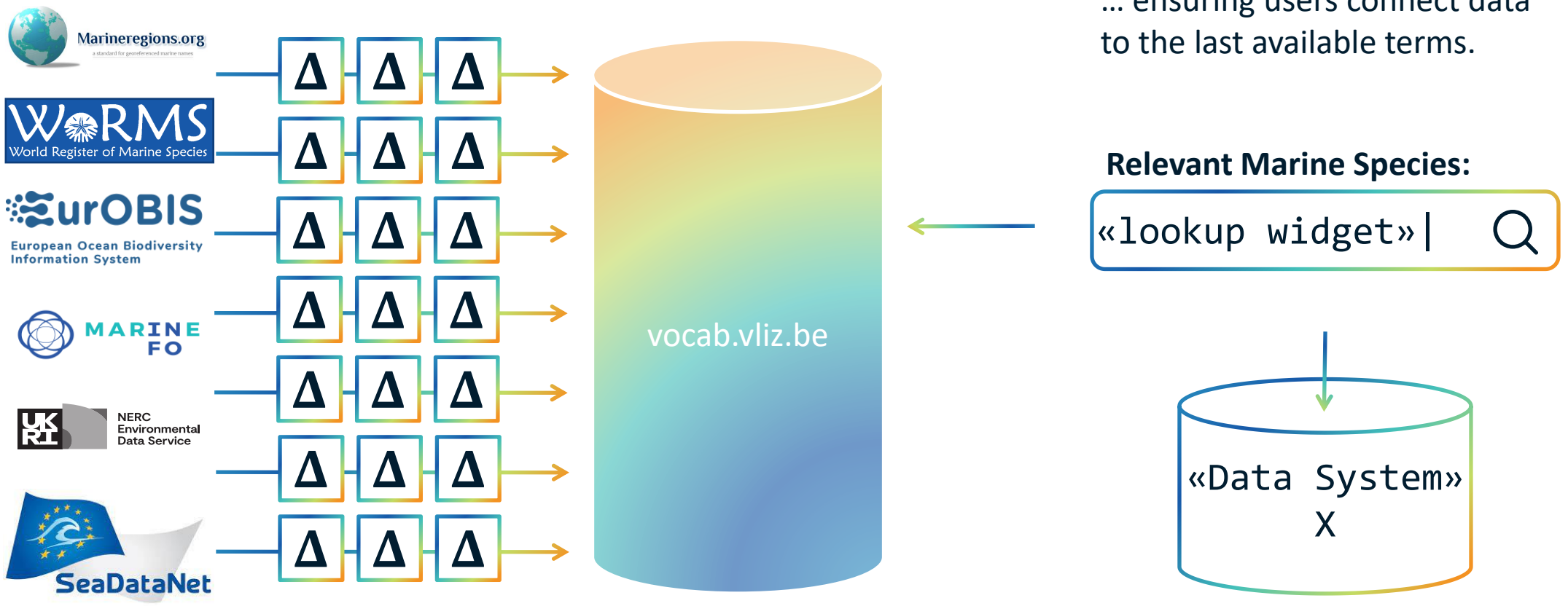
Vocab Term Lookup Service & Widget

Top level Flow

1. «admin»
   Add vocabs to service
2. «dev»
   - Define Selector Widgets
   - Embed those in Apps
3. «end user»
   - Pick term in natural language
   - Stored as URI (linked concept) in the data system

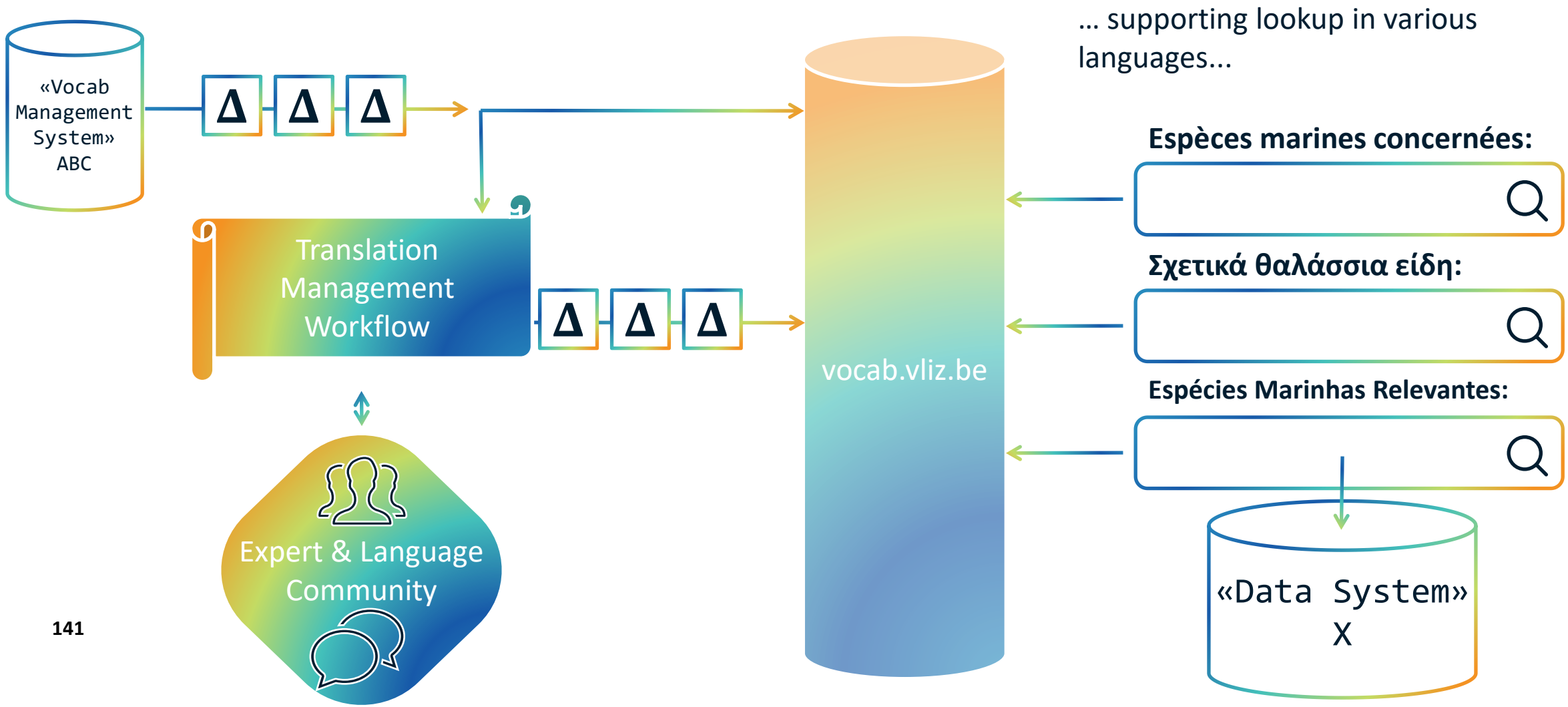


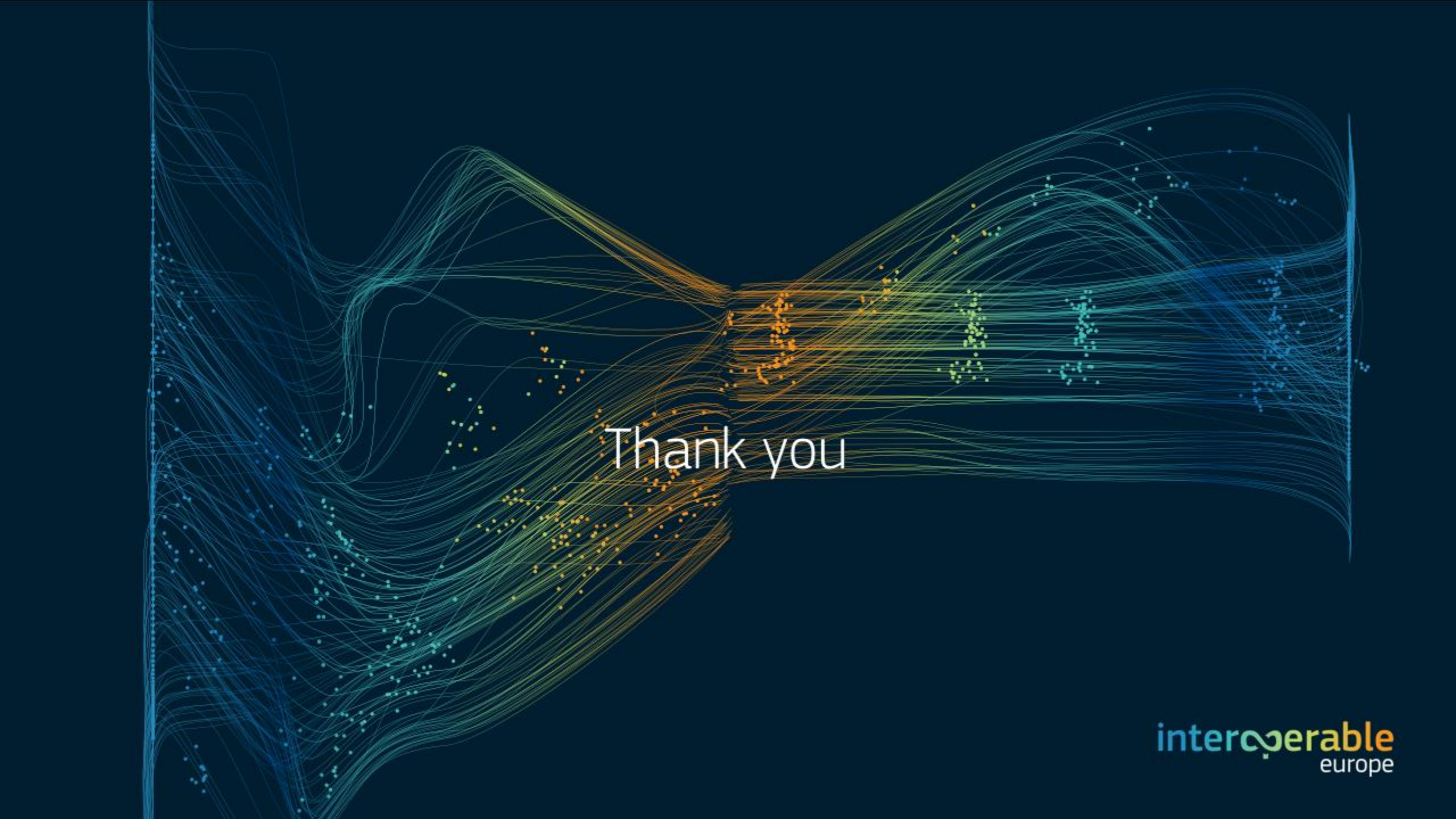
#LDES feeds sync up the vocab server …



… creating effectively connected distributed linked datasets …

Next phase : Multilingual Support



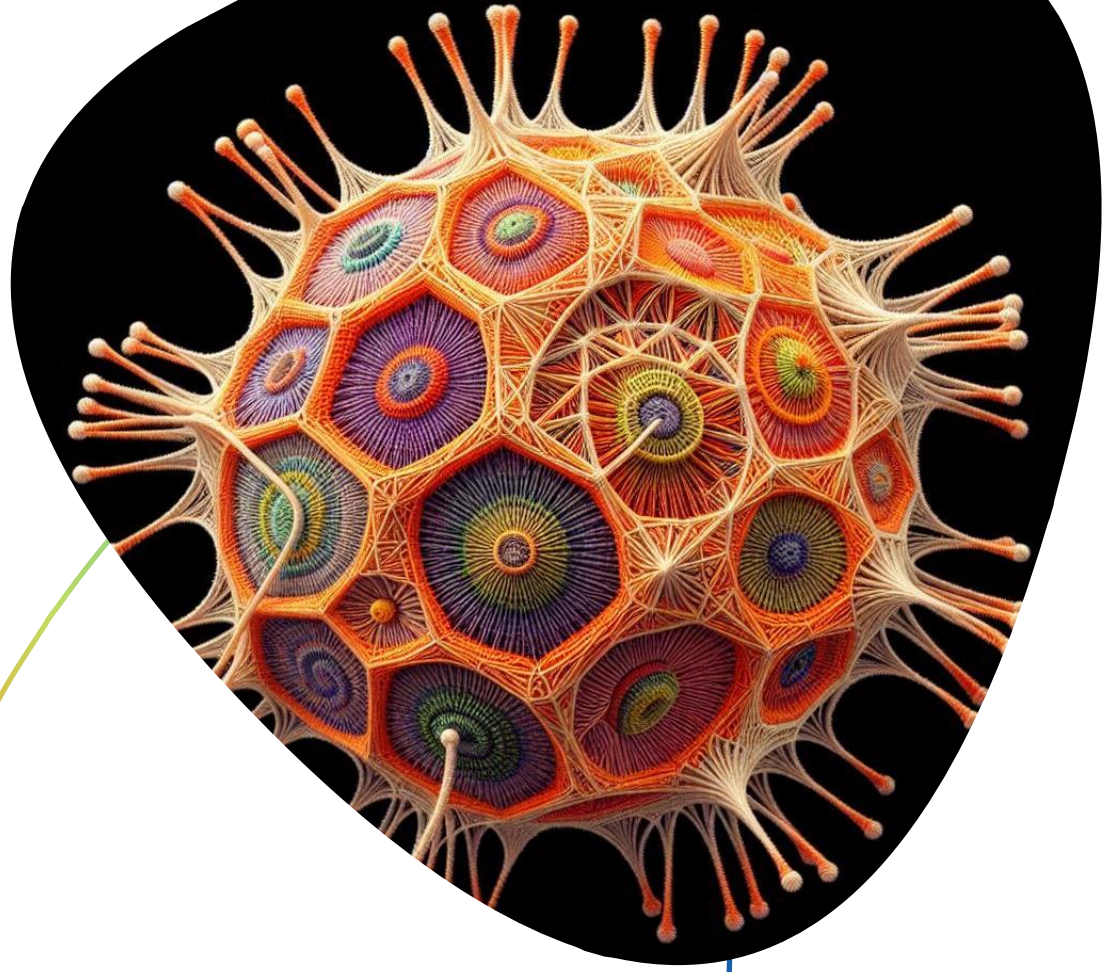


interoperable europe

LDES AS A MINIMUM INTEROPERABILITY MECHANISM

What are MIMs?

The minimal interoperability mechanisms (MIMs) emerged to enable a minimal but sufficient level of interoperability for data, systems, and services specifically in the context of smart city solutions. By facilitating this minimal yet essential level of interoperability, MIMs pave the way for the development of a cohesive global market and collaboration centered around solutions, services, and data.



What are MIMs?

The minimal interoperability mechanisms (MIMs) emerged to enable a minimal but sufficient level of interoperability for data, systems, and services specifically in the context of smart city solutions. By facilitating this minimal yet essential level of interoperability, MIMs pave the way for the development of a cohesive global market and collaboration centered around solutions, services, and data.

What are MIMs?

The minimal interoperability mechanisms (MIMs) emerged to enable a minimal but sufficient level of interoperability for data, systems, and services specifically in the context of smart city solutions. By facilitating this minimal yet essential level of interoperability, MIMs pave the way for the development of a cohesive global market and collaboration centered around solutions, services, and data.

MINIMAL INTEROPERABILITY

"Good enough"

INTEGRATION

In a cost-effective way

What are MIMs?

The minimal interoperability mechanisms (MIMs) emerged to enable a minimal but sufficient level of interoperability for data, systems, and services specifically in the context of smart city solutions. By facilitating this minimal yet essential level of interoperability, MIMs pave the way for the development of a cohesive global market and collaboration centered around solutions, services, and data.

MINIMAL INTEROPERABILITY

"Good enough"

PROCUREMENT

In between Standards and Principles

INTEGRATION

In a cost-effective way

What are MIMs?

The minimal interoperability mechanisms (MIMs) emerged to enable a minimal but sufficient level of interoperability for data, systems, and services specifically in the context of smart city solutions. By facilitating this minimal yet essential level of interoperability, MIMs pave the way for the development of a cohesive global market and collaboration centered around solutions, services, and data.

The Y-MIM Structure

Capabilities

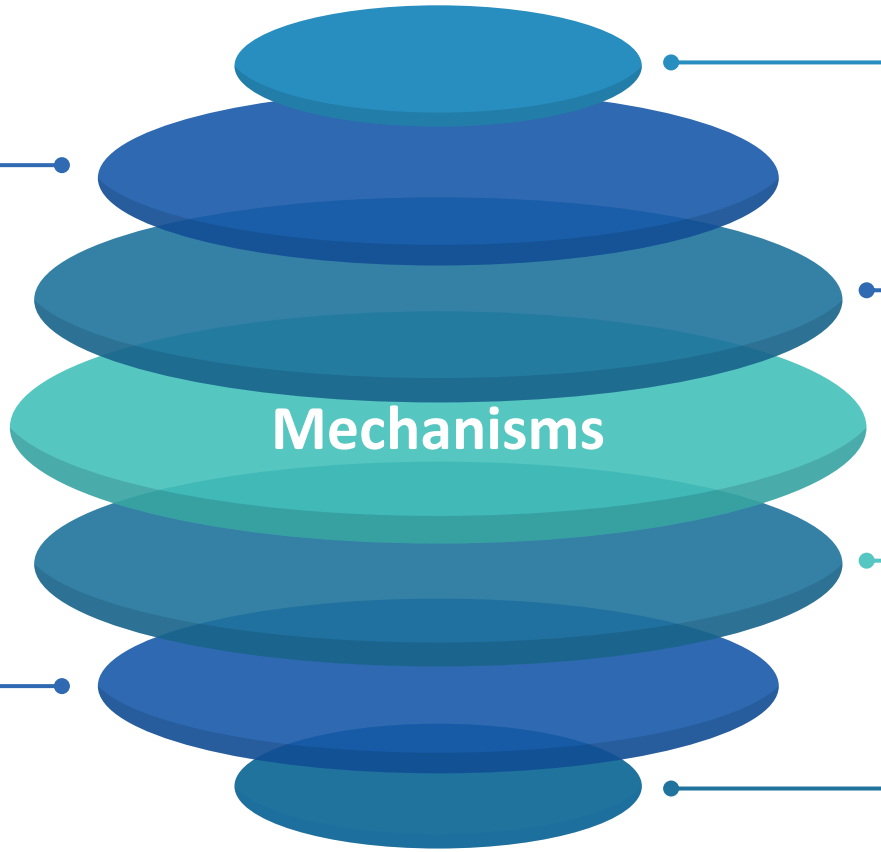
Capabilities required to realize the objectives

Mechanisms

Possible ways of implementing solutions that adhere to all requirements

Compliance & Conformance

Test suites and approaches to validate solutions against specifications



Objectives

Objectives oif the MIM

Requirements

Requirements for solutions that provide capabilities

Specifications

Used within the mechanism

Interoperability Guidance

Ways to move from one mechanism to another

MIM 1 DEVELOPMENT HISTORY

2015 — Conception of OASC

2017 — Incorporation of OASC as a not for profit

Dec 2019 — Launch of “Minimum Interoperability Mechanisms” (MIMs)

Jan 2020 — Launch of MIM1 as ”Context Information Management”

ADOPTION OF MIMS Plus

Jun 2021 — **Launch of the Living-in.eu Tech Subgroup**

MIM1 Incorporated in the living-in.eu Tech Subgroup with NGSI-LD focus. Also mention of Core Vocabularies, SAREF and oneM2M

Jan 2022

Adoption of MIMs Specification v4.0

Includes ETSI NGSI-LD as a single specification for MIM1

Jan 2023

Adoption of MIMs Specification v5.0

ETSI Testing Task Force TTF launched on NGSI-LD

Jun 2023

Adoption of MIMs Specification v5.5

Introduction of the Y.MIM format. Introduction of Mechanisms as a means to describe multiple implementations.

First draft of OGC and LDES based Mechanisms

Jan 2024

Adoption of MIMs Specification v6.0

Election of Flanders region as MIM1 Champion

Jun 2024

Complete specification for 3 Mechanisms within MIM1. Updated Objectives, Capabilities and Requirements to reflect those

- **ETSI NGSI-LD**
- **OGC SensorThings**
- **LDES**

Today

Full set of Compliance & Conformance testing tools for each Mechanism. Start of building a Knowledge Base for Interoperability Guidance

Call to action: Please provide your case studies!

OBJECTIVES

The first section of any MIM should be a short description of the desired outcome of the implementation of a particular MIM. This should provide the basis to facilitate making the case for implementing that MIM to the key decision makers.

- To enable context information from different systems within or across organisations, such as cities or communities, originating from heterogeneous sources, to be brought together using a Web based API.
- To enable comprehensive and integrated use, reuse and sharing of data as well as management of context information
- To turn data into a strategic resource

CAPABILITIES

The Capabilities section should provide a short description of the set of functionalities needed within a MIM to enable the objective to be achieved.

- C1: Applications are able to access data from different sources (such as cities, communities and vertical solutions).
- C2: Applications are able to use both current and historical data, use geospatial querying and be automatically updated when the source data changes.
- C3: Applications can discover and retrieve data relevant to their context from a variety of sources

MECHANISMS

This is the key section of the MIM, where the requirements of the MIM are listed. It should provide a description of the set of processes needed to enable the capabilities required in a MIM to be achieved.

- NGSI-LD
- OGC SensorThings
- LDES

SPECIFICATIONS

The Specifications section should provide a description of one or more alternate sets of tried and tested methodologies that can be used to deliver the mechanism covered in the MIM. These may be taken from formal standards documentation or may be drawn from emerging or de facto standards.

- ETSI NGSI-LD
- ISO 19156 Observations, Measurements, and Samples
- LDES

INTEROPERABILITY GUIDANCE

This section should provide a description of the ways to enable a level of interoperability between different sets specifications so there is at least a basic level of interoperability with systems that use a different set of specifications to comply with the MIM.

- Case Studies
    - Porto Digital
    - Flanders Smart Data Space

COMPLIANCE & CONFORMANCE

Here information will be provided to the delivery team to help them be sure that their implementation conforms to the MIM, to industry to help them know how to demonstrate that their products and services comply with the MIMs and to procurement officers to understand how they can check that the proposals they are assessing are MIMs compliant.

- European Interoperability Testbed
    - NGSI-LD Test Suite
    - LDES Test Suite
- OGC Compliance Test Suites
- Flanders' Exploratory LDES tests

How does LDES Satisfy MIM1 Requirements?

Context can be managed through the web

The Linked Data Event Streams (LDES — https://w3id.org/ldes/specification) specification uses HTTP and RDF as its web based interface for the re-use of domain models, the definition of the schema (SHACL), API interface descriptions (TREE hypermedia — https://w3id.org/tree/specification), context and instance data.

Information from all sources should use the same concepts […]

Each LDES contains information on how to the member objects are structured based on well-defined SHACL shapes. Across the Web, it promotes the re-use of Linked Data vocabularies.

The Web Based API should support retrieval of latest data

An LDES is an append-only log of members, and thus by-default a server keeps the full history. On top of a view, it may document a retention policy in which the server indicates data will be removed from the server after a certain period of time, or amount of members. Third parties should read retention policies to understand what subset of the data is retrievable.

`..+ Should support retreivel of historic data

See R3: LDES provides historical and live data in the same interface.

[…] Should support geospatial querying

Geospatial functionality can be achieved in two ways:
1.Either you use a LDES to Service pipeline in which you replicate the full dataset into a geospatial software of choice.
2.You publish a geospatially fragmented Linked Data Event Streams (see https://informatievlaanderen.github.io/VSDS-LDESServer4J/configuration/fragmentations/geospatial)

[…] Should support subscription to changes

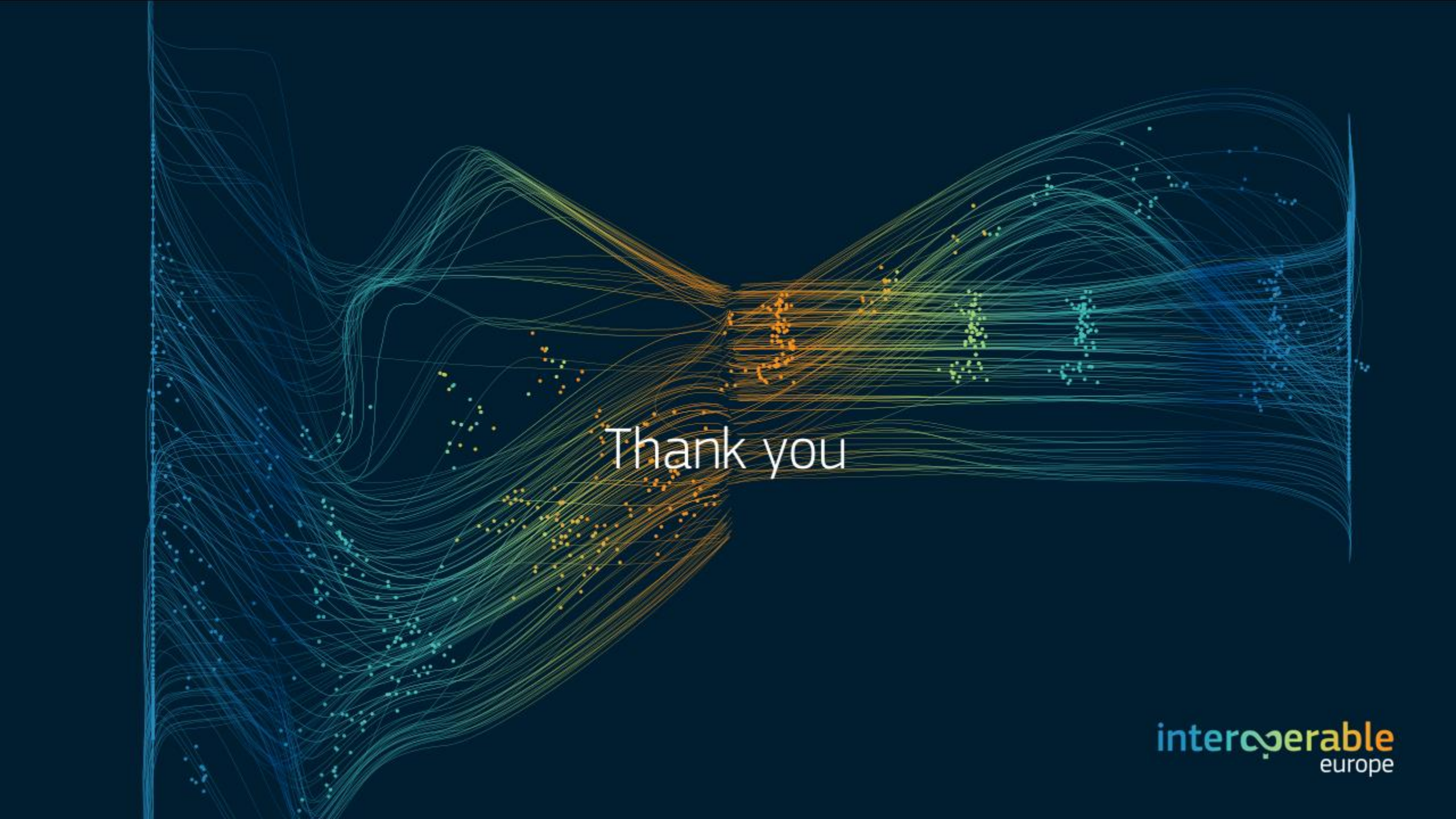
The LDES event source is a specialized view for replication and synchronization in the same interface as R4 and R5. Using an LDES client, an agent can stay up to date with the latest changes.

Relevant data sources should be discoverable and retrievable according to their context

Data discovery works via DCAT-AP data portals that indicate that their dcat:Dataset is also an ldes:EventStream, and that their dcat:DataService is also a tree:ViewDescription. On an ldes:EventStream, there will be a SHACL shape defined that shows what properties are being used within the members. Using that, you can select the properties of interest and use that dataset in other contexts as well.

Specific subsets of data relevant to the context should be retrievable from within larger data sets

LDES is built on top of the TREE hypermedia specification that allows to fragment event streams in search trees. Such search trees then allow client to specifically stay in-sync, or replicate the full history, of a subset. It is up to the server to decide what granularity and type of fragmentations to publish.



interoperable europe

The future of LDES

The future of LDES

LDES is going through a preparatory phase in collaboration with Digital Flanders

Please visit the GitHub to contribute, post your issues and requests

As of October: Working Group, Governance, Update of the Specification

Thank you!