Chris Dijkshoorn &
Edward Anderson

RIJKS MUSEUM

Integration Layer

A Linked Data Infrastructure for Sustainable Data Services

ELAG 2022

The Rijksmuseum

National museum of the Netherlands

Collection

- 1.000.000 objects
- 450.000 books
- 800 meters of documentation
- 17 terabytes of research data

Research Services responsible for all data about collection and organisation

RIJKS MUSEUM

Collection Data Architecture

Design by consultancy company

Integration Layer to address

- Siloed data
- Make-do connections

Durable design decisions

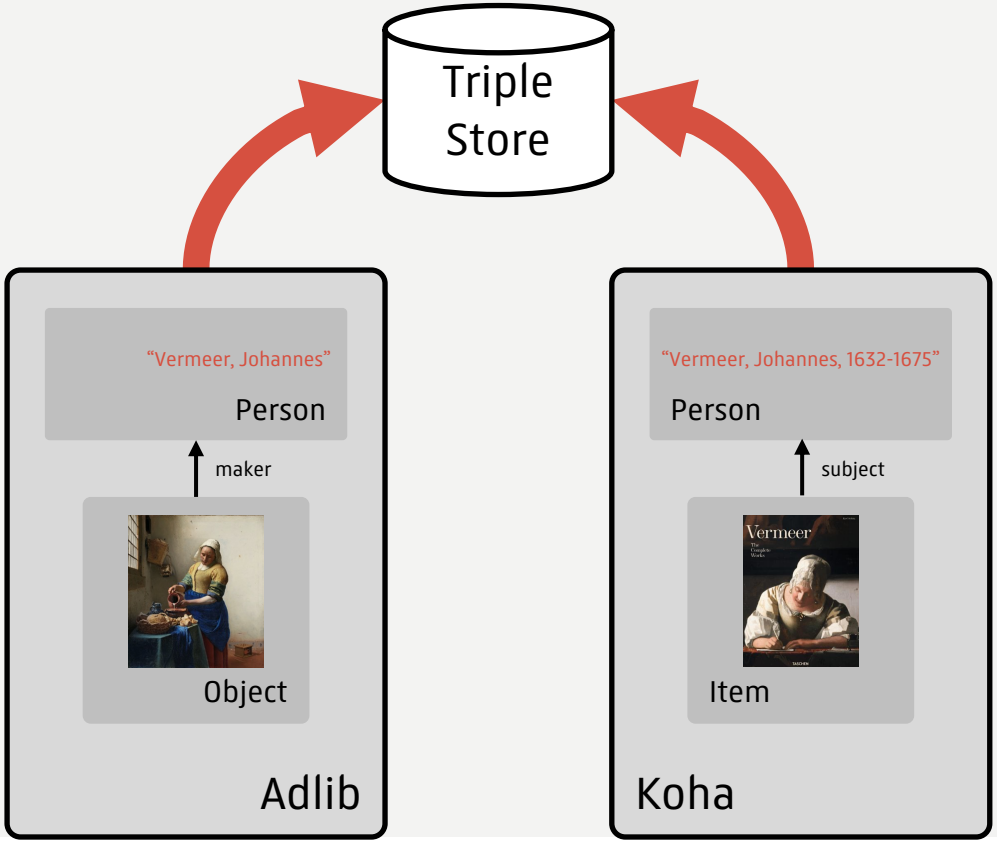
RIJKS MUSEUM

Linked Data

Institutional multi-domain integration

Domain-specific data models

Source systems remain authoritative



RIJKS MUSEUM

Persistent Identifiers

PIDs stored in source systems

URI strategy

https://{type}.rijksmuseum.nl/{hidden semantics}{reference}

PID Vermeer in Library domain

https://id.rijksmuseum.nl/310117784

Domain	Concept	Integer
rijksmuseum	actor	110
collection	object	200
collection	person	210
library	item	301
library	person	310

Triple Store

Domain	Concept	Integer
rijksmuseum	actor	110
collection	object	200
collection	person	210
library	item	301
library	person	310

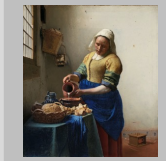
https://id.rijksmuseum.nl/**210**2529

label

"Vermeer, Johannes"

Person

maker



https://id.rijksmuseum.nl/2006417

Object

Adlib

https://id.rijksmuseum.nl/**310**11784

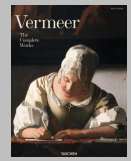
label

"Vermeer, Johannes, 1632-1675"

Person

subject

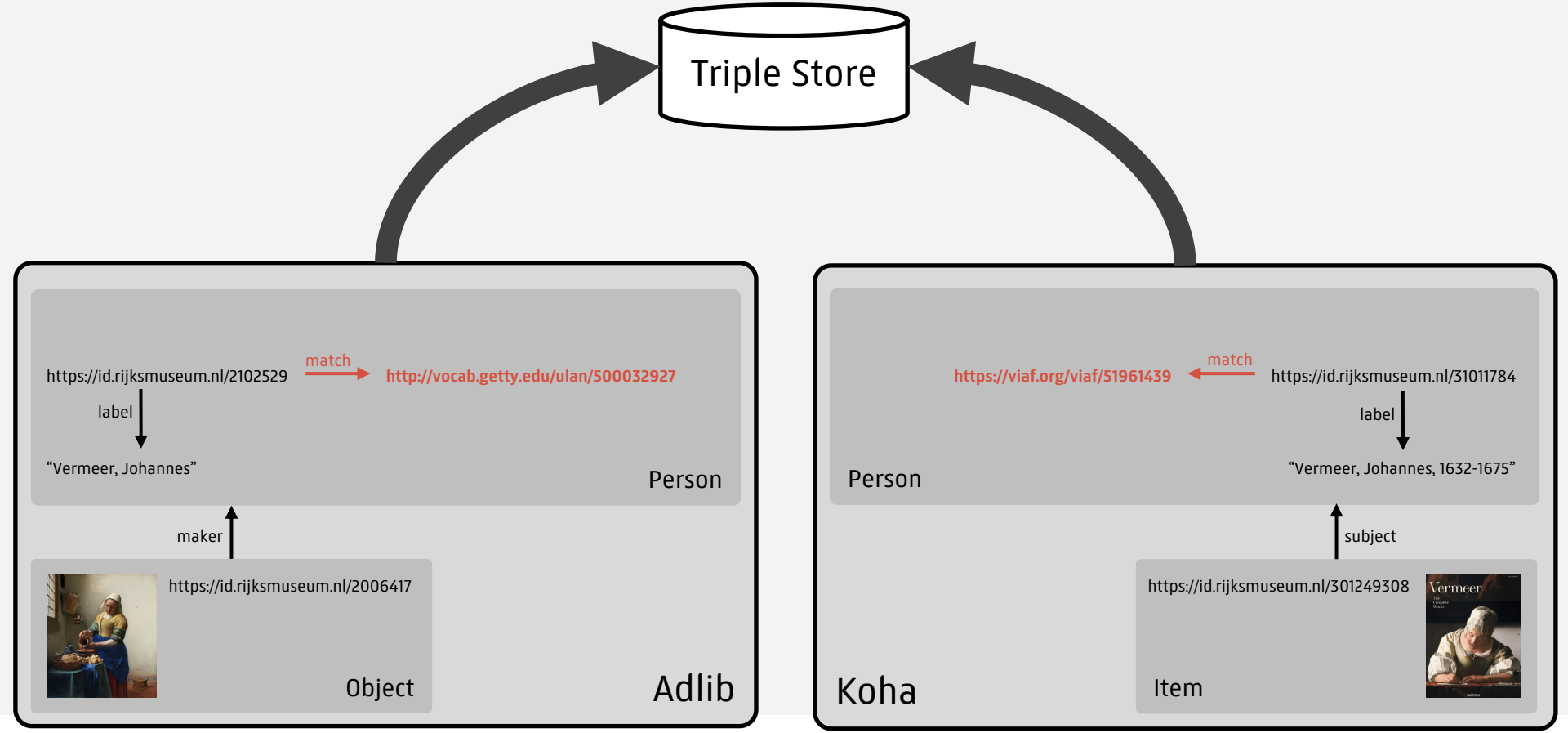
https://id.rijksmuseum.nl/301249308



Item

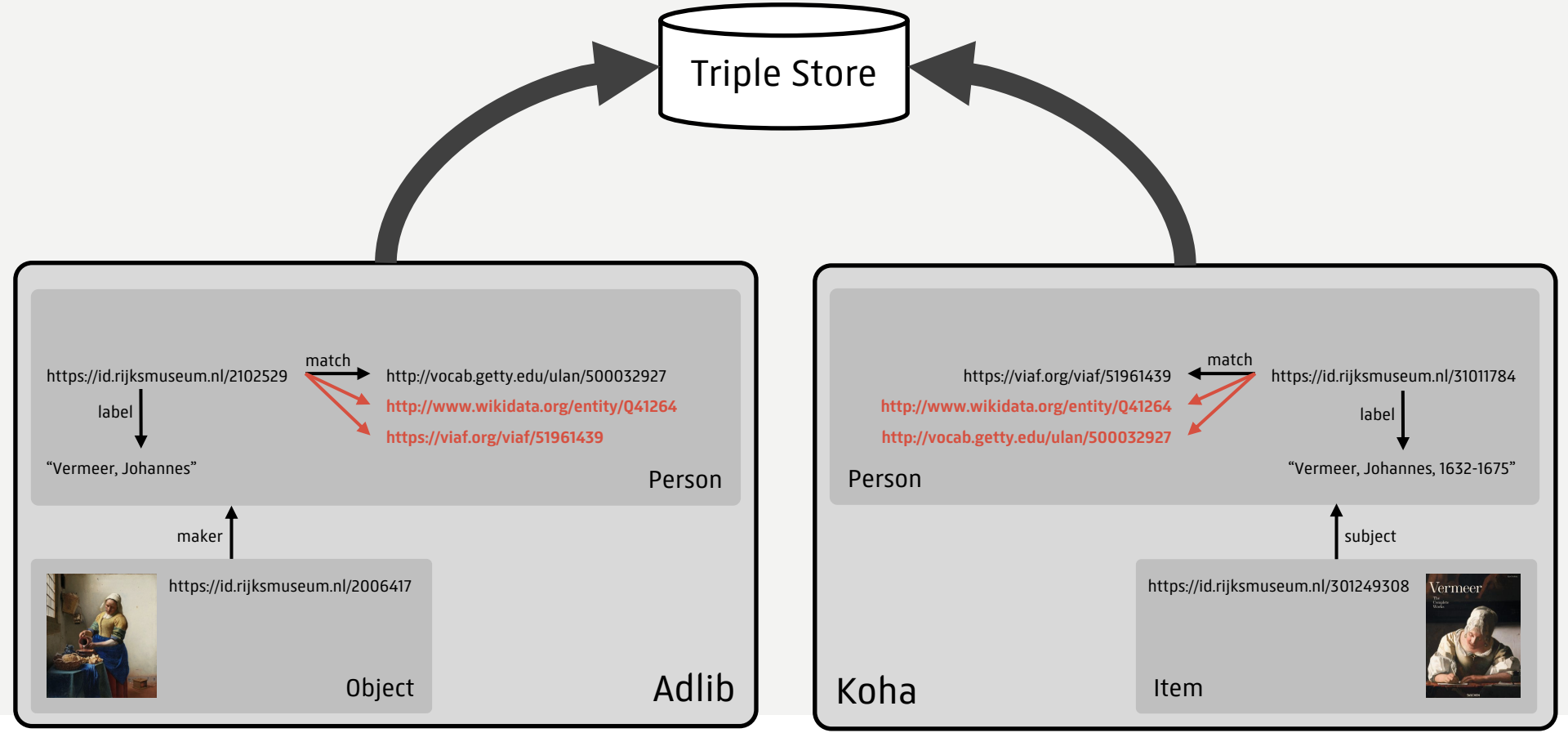
Koha

RIJKS MUSEUM



Match with external sources

RIJKS MUSEUM



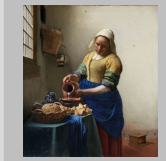
Automatically extend matches

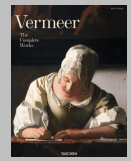
RIJKS MUSEUM

Reconciliation Daemon
Two identifiers concern same resource:
https://id.rijksmuseum.nl/31011784
https://id.rijksmuseum.nl/2102529

Triple Store

Reconciliation Daemon

https://id.rijksmuseum.nl/2102529
label
"Vermeer, Johannes"
match
http://vocab.getty.edu/ulan/500032927
http://www.wikidata.org/entity/Q41264
https://viaf.org/viaf/51961439
Person
maker

https://id.rijksmuseum.nl/2006417
Object
Adlib

https://viaf.org/viaf/51961439
http://www.wikidata.org/entity/Q41264
http://vocab.getty.edu/ulan/500032927
match
https://id.rijksmuseum.nl/31011784
label
"Vermeer, Johannes, 1632-1675"
Person
subject
https://id.rijksmuseum.nl/301249308


Item
Koha

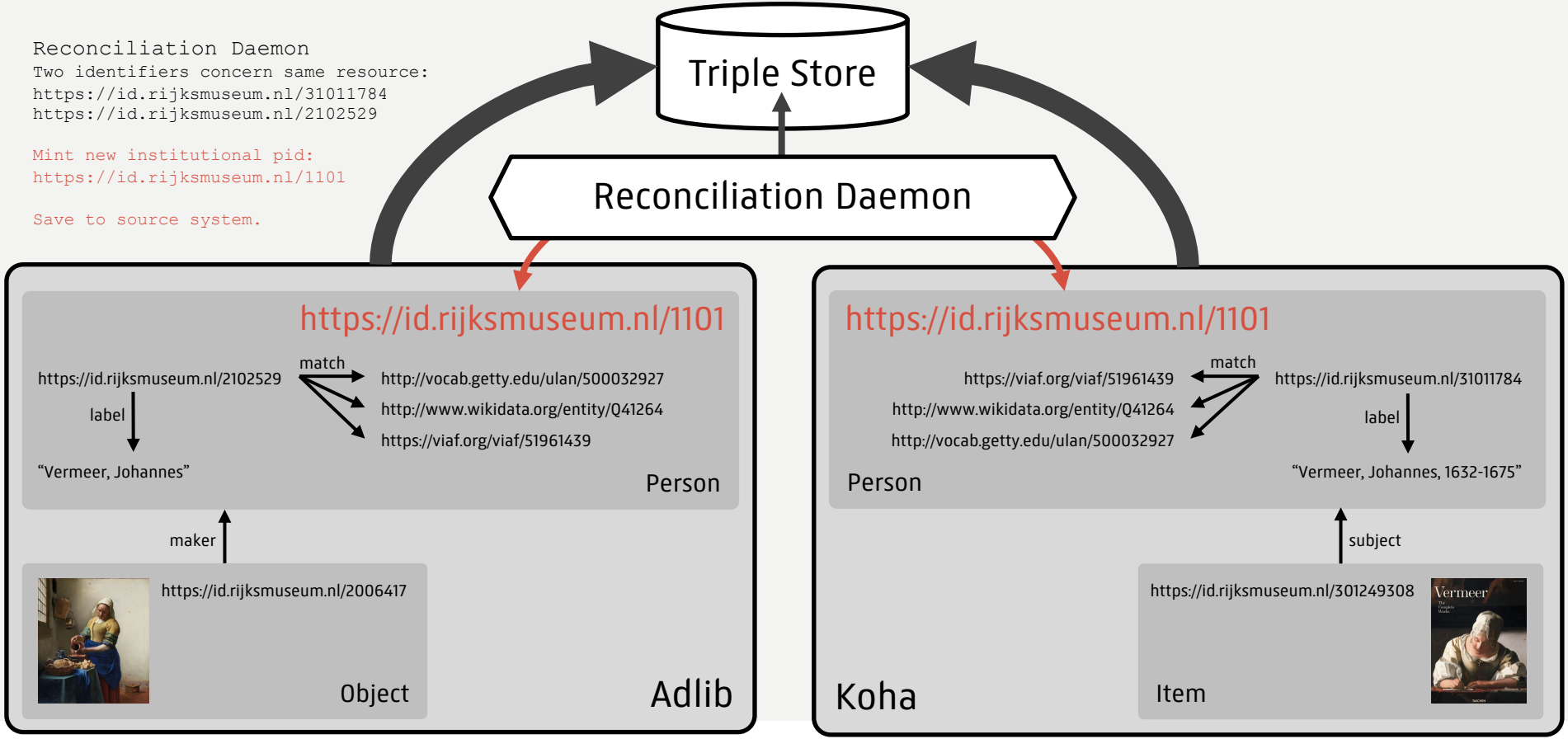
Reconcile based on alignments

RIJKS MUSEUM

Reconciliation Daemon
Two identifiers concern same resource:
https://id.rijksmuseum.nl/31011784
https://id.rijksmuseum.nl/2102529

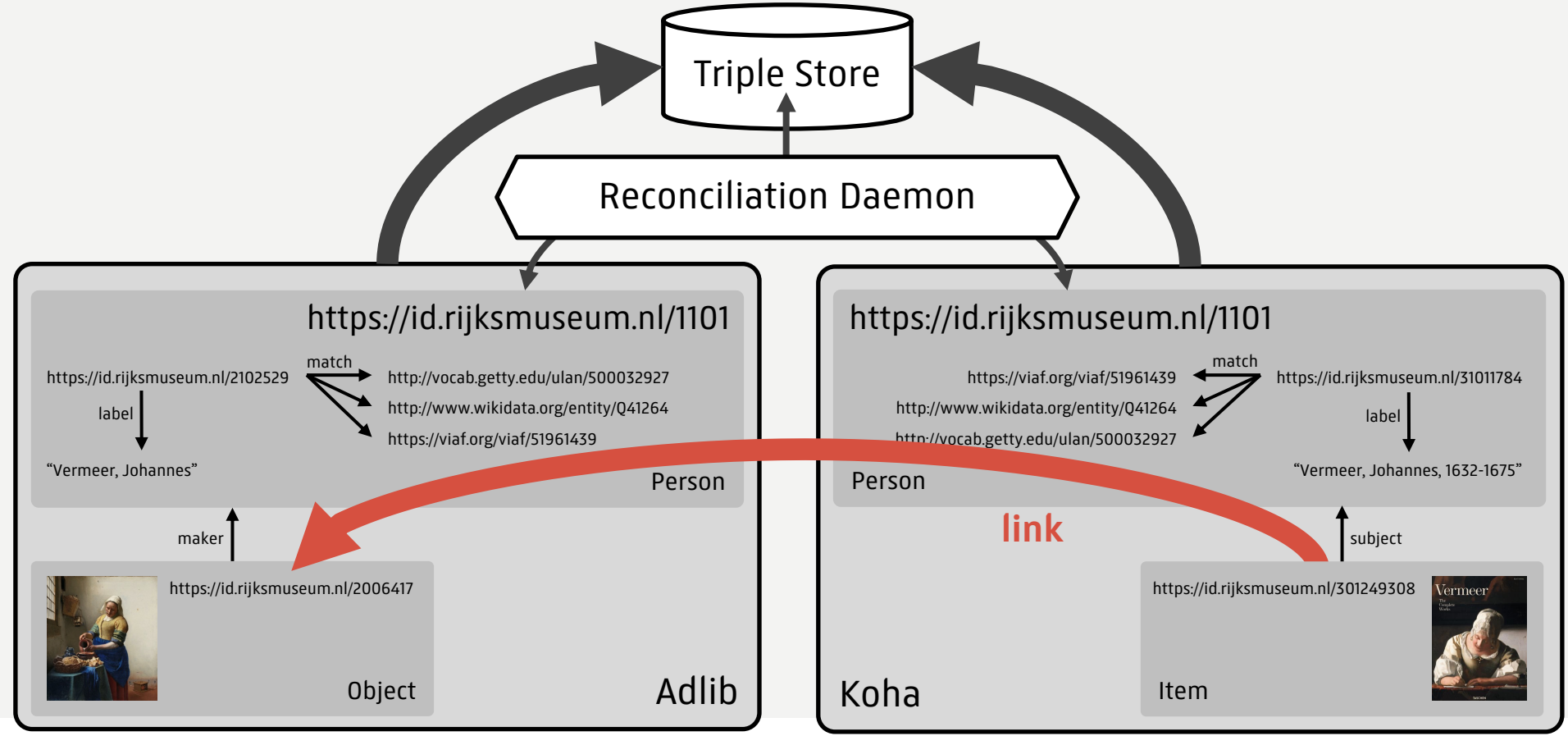
Mint new institutional pid:
https://id.rijksmuseum.nl/1101

Save to source system.



Save institutional identifier to source systems

RIJKS MUSEUM



Create direct links between objects

RIJKS MUSEUM

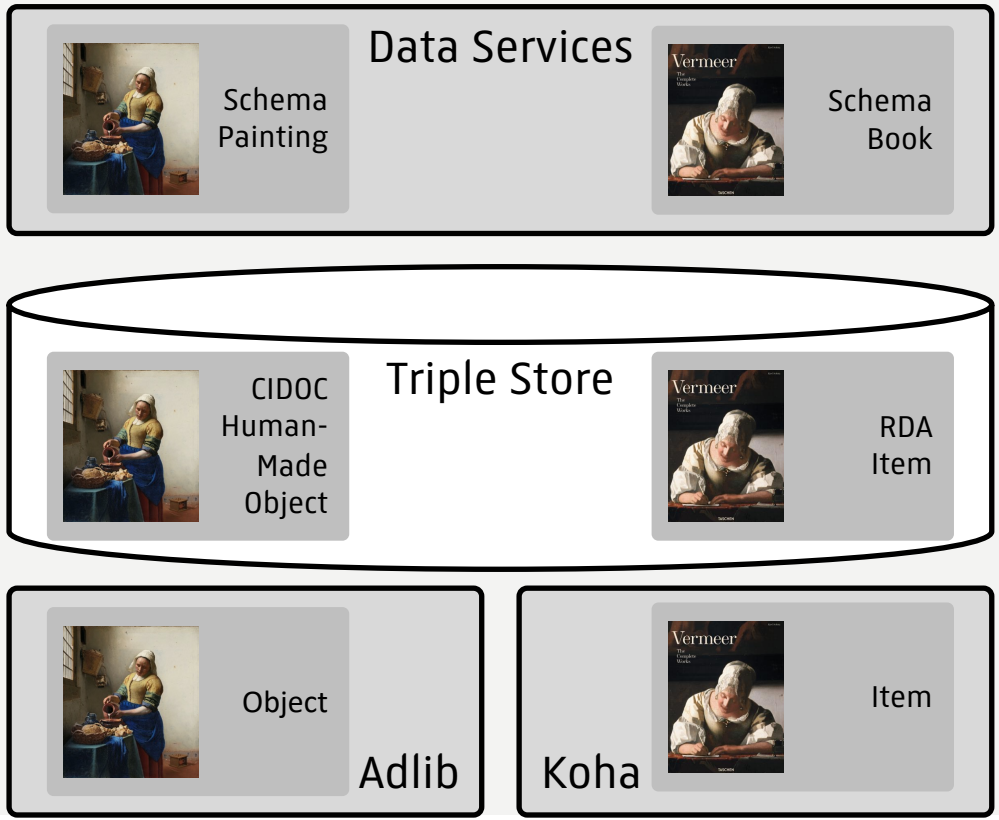
Data models

Domain models in Triple Store

- collection: CIDOC Conceptual Reference Model
- archive: Records in Context
- library: RDA/RDF

Multiple target models for Data Services

- Schema.org
- Dublin Core
- Europeana Data Model
- Linked Art



RIJKS MUSEUM

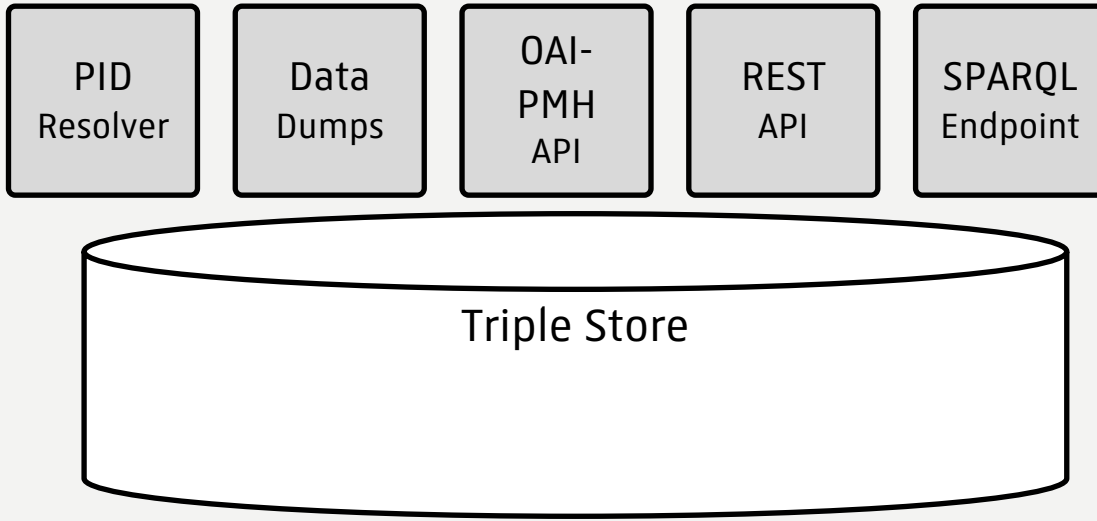
Data Services

Range of services, different

- query expressivity to define subsets
- richness results

Essential initial user:
website developers

Microservices architecture



Implementing an Integration Layer

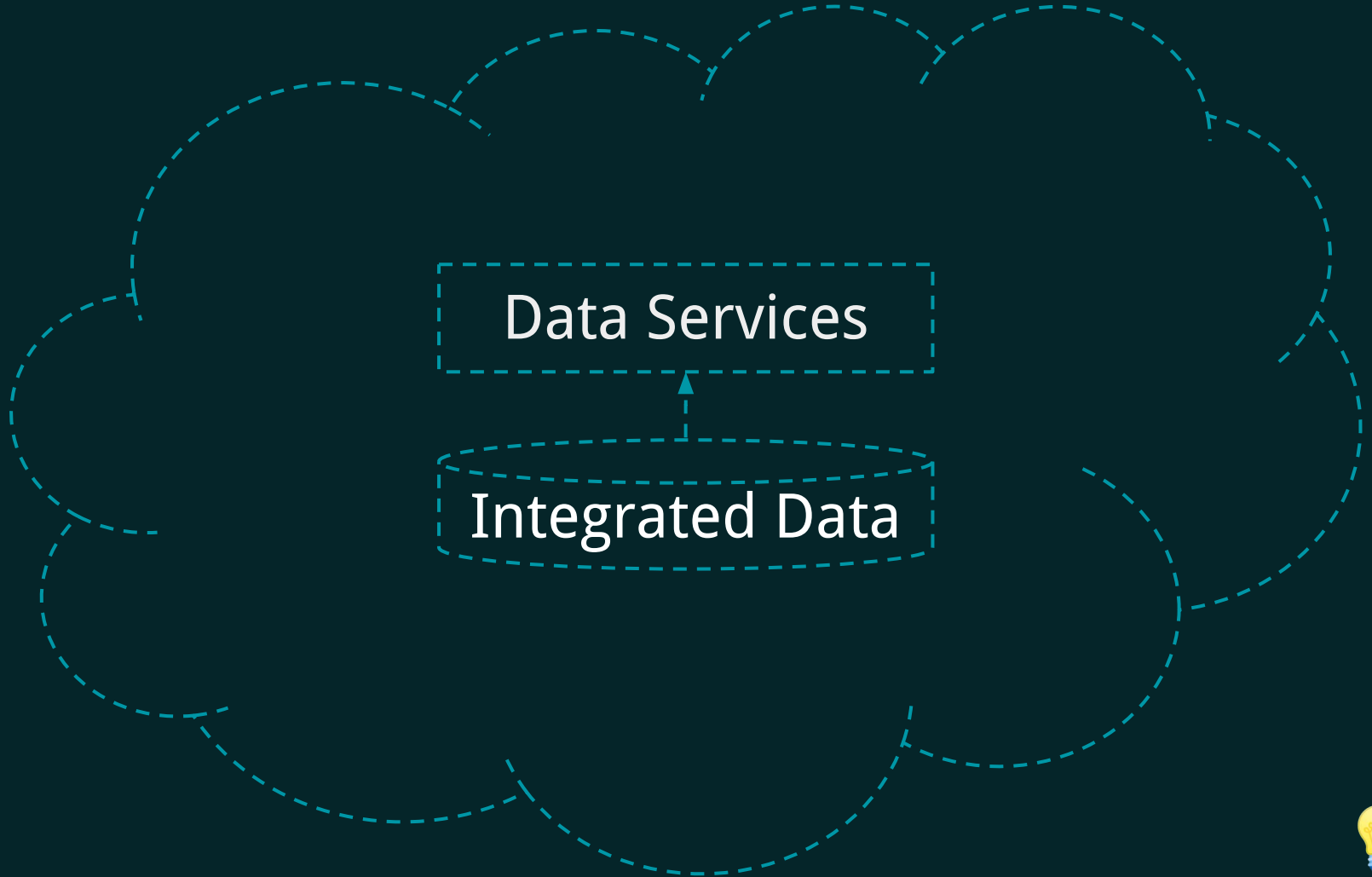
This page intentionally left blank

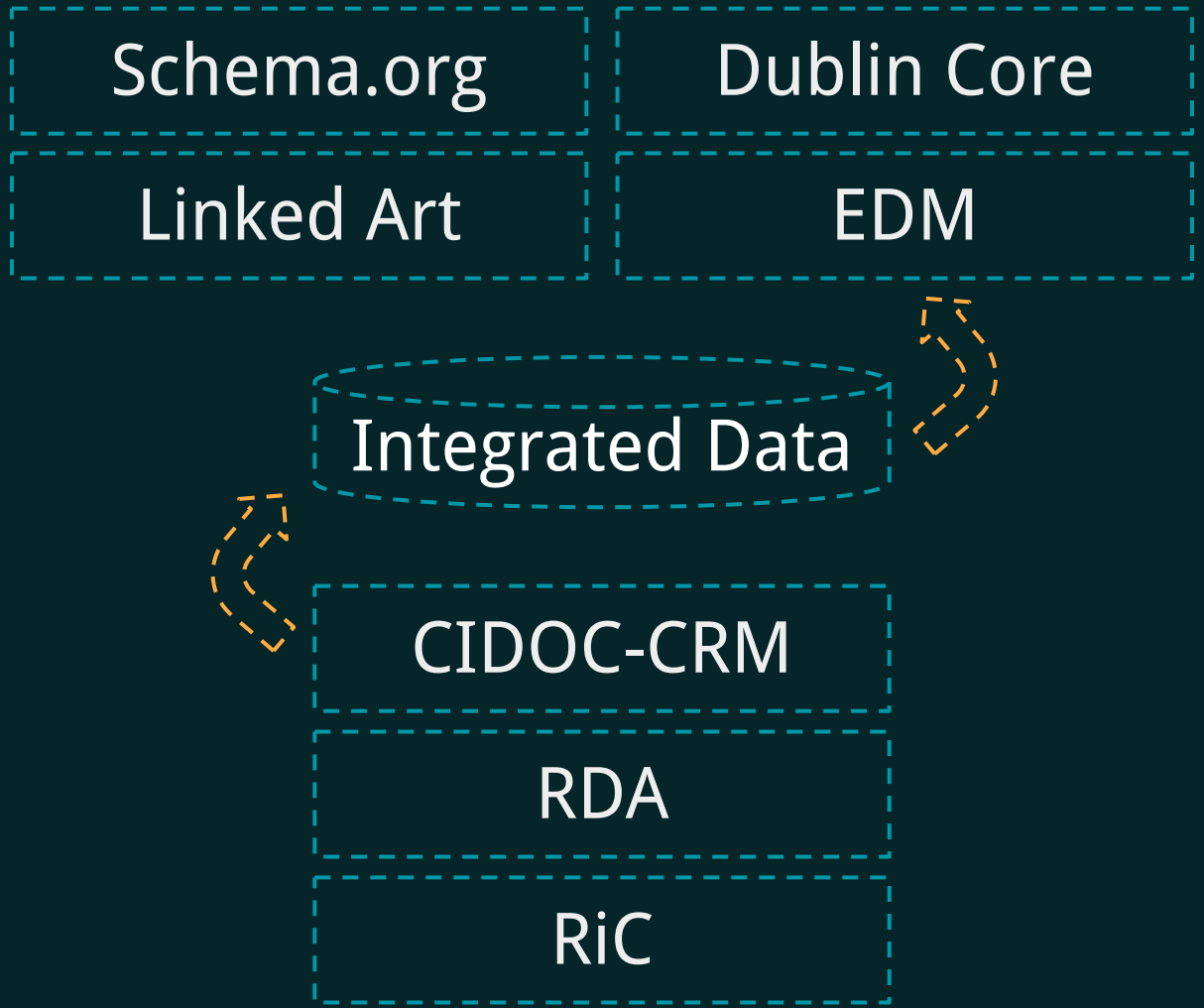
Šī lapa ar nolūku ir atstāta tukša

💡 What did we want?

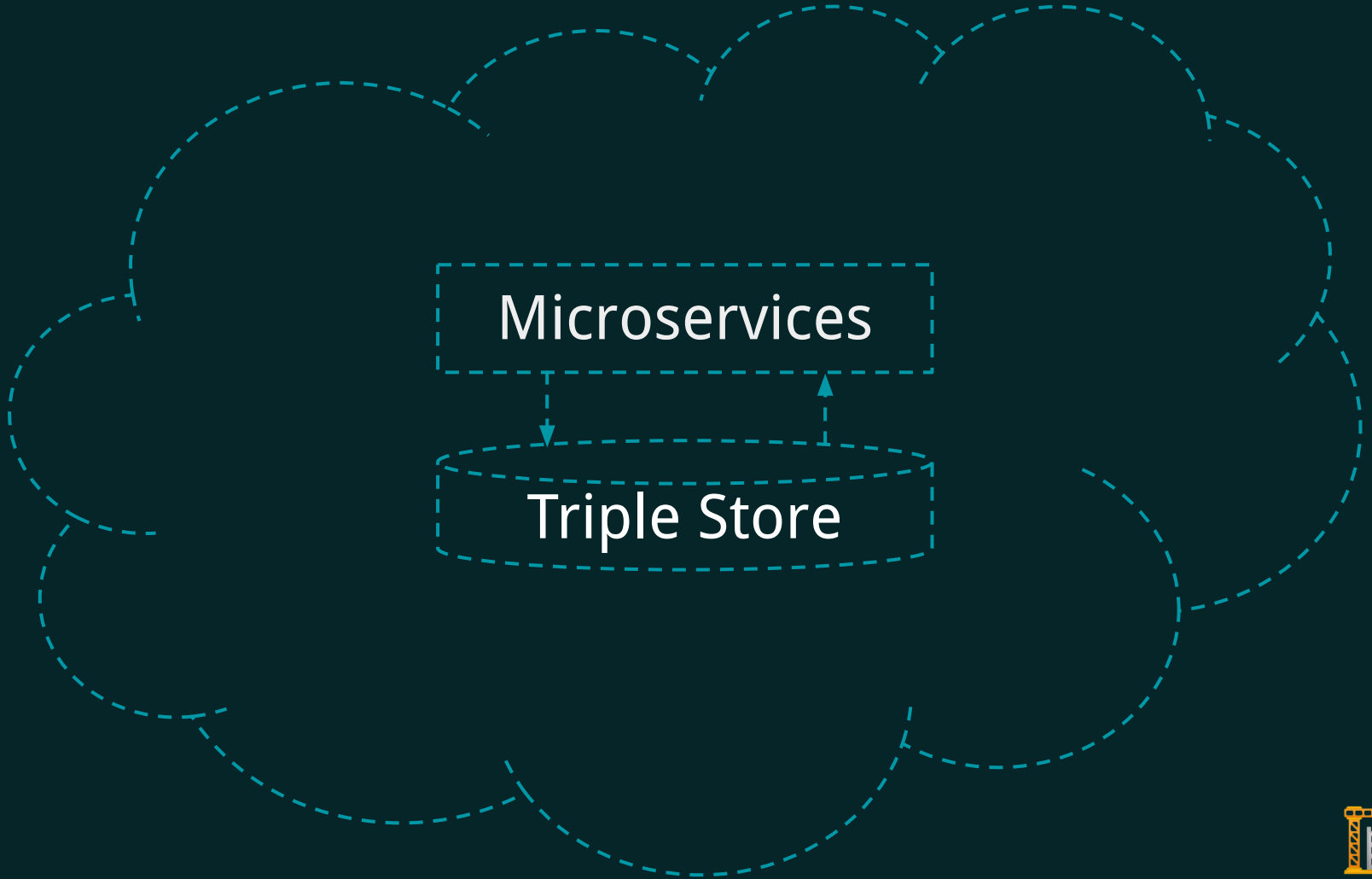
🏗️ What did we build?

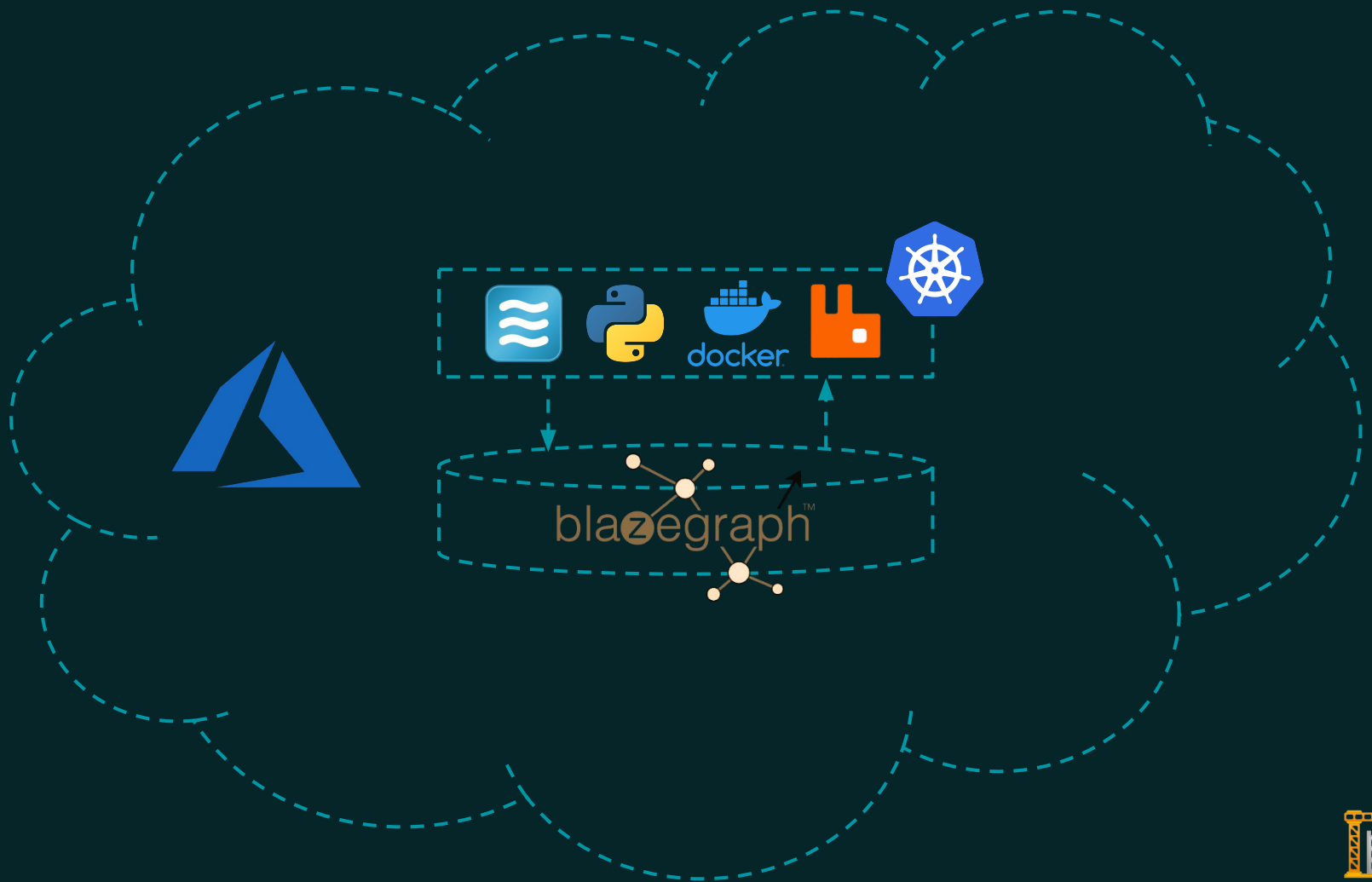
🤔 What did we learn?


Data Services
Integrated Data

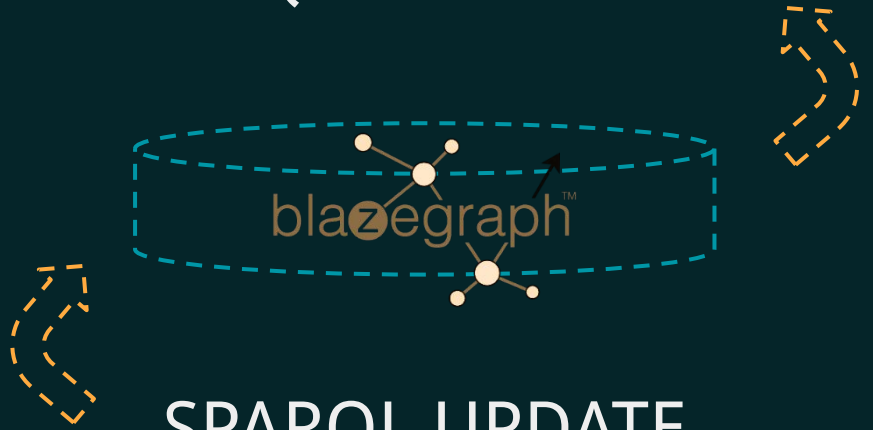

Schema.org
Dublin Core
Linked Art
EDM
Integrated Data
CIDOC-CRM
RDA
RiC

What did we build?


Microservices
Triple Store


docker
blazegraph™

SPARQL CONSTRUCT

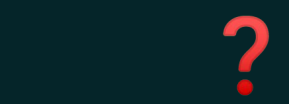


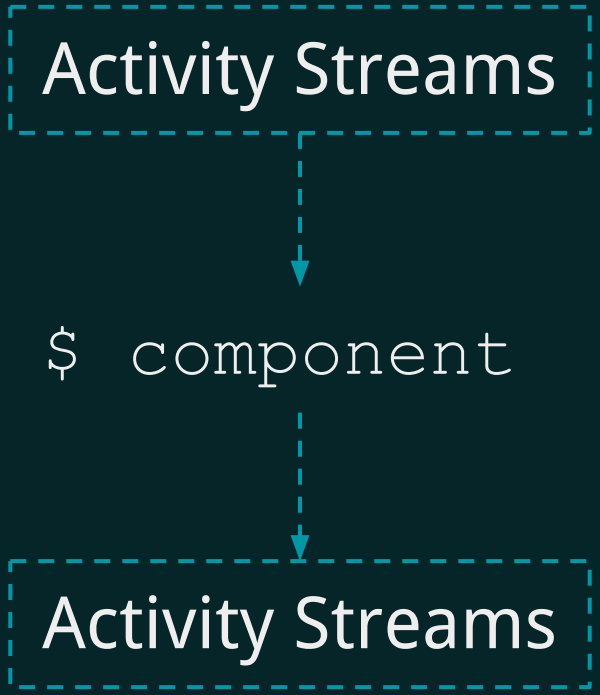
SPARQL UPDATE

$ component

```
$ pip install component
```

```
$ component --help

$ component --version
```



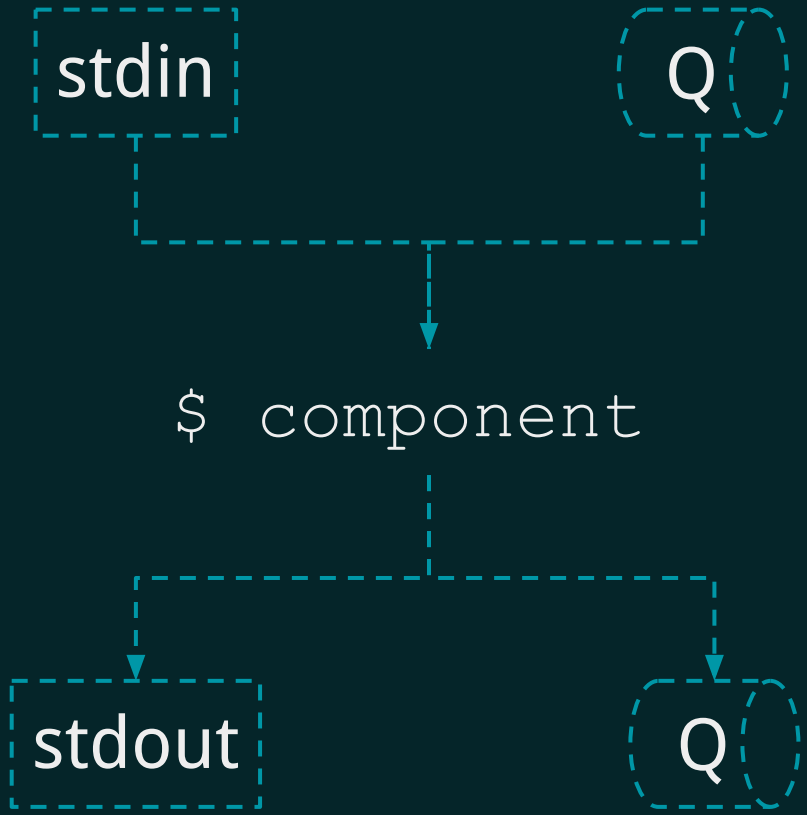

Activity Streams
$ component
Activity Streams

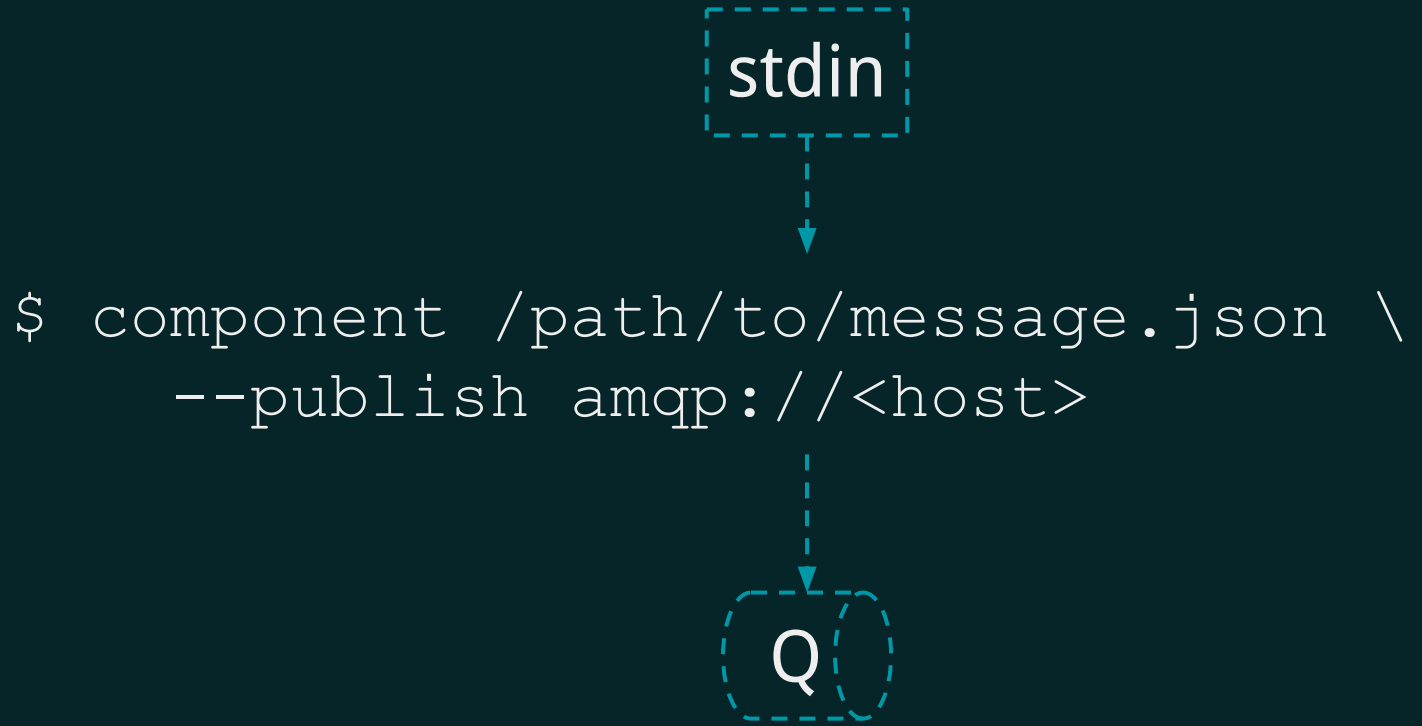
```
$ echo '{"type": "Update"}' | component -
{
   "type": "Create"
}
```

```
$ component /path/to/message.json
{
   "type": "Update"
}
```

```
$ component *.json
{
   "type": "Update"
}
{
   "type": "Delete"
}
```

```
$ component-a *.json | component-b -
{
   "type": "Create"
}
```


stdin
Q
$ component
stdout
Q


stdin
$ component /path/to/message.json \
--publish amqp://<host>
Q


```
$ component --subscribe amqp://<host>
{
   "type": "Announce"
}
```

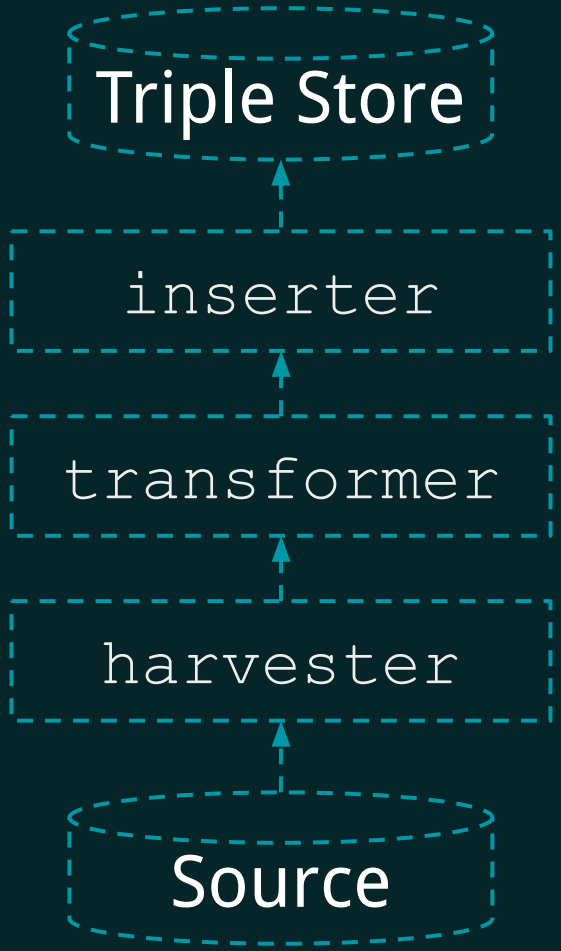

```
$ component --subscribe amqp://<host> \
            --publish amqp://<host>
```

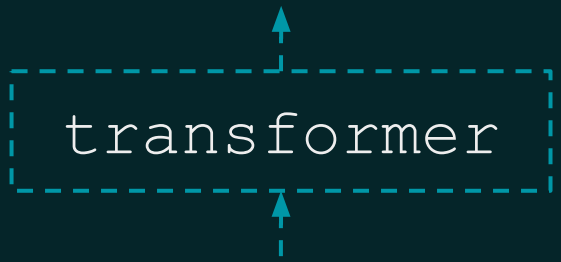
README.md

Inherit from the `Component` base class to create a new Integration Layer application.

```
from argparse import ArgumentParser
from component_framework import (
    ActivityStreamMessage,
    Component,
    Configuration,
    MessageStatus
)


class Application(Component):

    def on_configure(self, config: Configuration) -> None:
        ...

    def on_setup_arguments(self, parser: ArgumentParser) -> None:
        ...

    def on_start(self) -> None:
        ...

    def on_receive_message(self, message: ActivityStreamMessage) -> MessageStatus:
        ...

    def on_stop(self) -> None:
        ...


if __name__ == '__main__':
    app = Application()
    app.run()
```

Demos

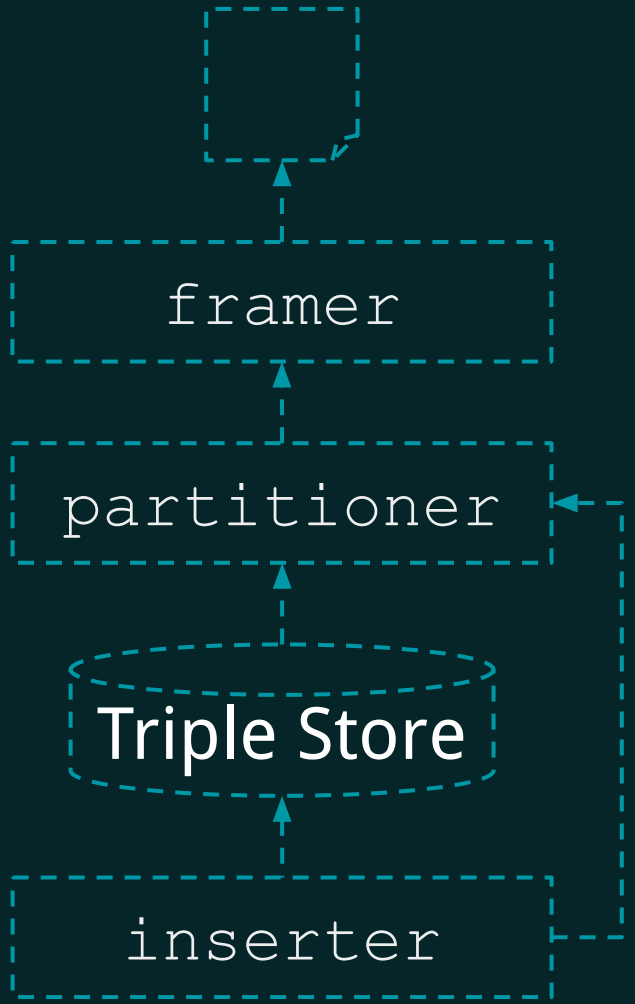

Triple Store
inserter
transformer
harvester
Source


transformer

README.md

```
from xml_transformations.adlib.cidoc_crm import HumanMadeObject


record = '''
<record>
    <PersistentIdentifier>https://id.rijksmuseum.nl/2005216</PersistentIdentifier>
    <title>
        <title.gb>Officers and other civic guardsmen of District II in Amsterdam, under the command
    </title>
</record>
'''

template = HumanMadeObject()
graph = template.transform(record)
print(graph)
```

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:crm="http://www.cidoc-crm.org
    <rdf:Description rdf:about="https://id.rijksmuseum.nl/2005216">
        <crm:P1_is_identified_by>
            <crm:E33_E41_Linguistic_Appellation>
                <crm:P190_has_symbolic_content>Officers and other civic guardsmen of District II in
                <crm:P72_has_language rdf:resource="http://vocab.getty.edu/aat/300388277"/>
            </crm:E33_E41_Linguistic_Appellation>
        </crm:P1_is_identified_by>
    </rdf:Description>
</rdf:RDF>
```


framer
partitioner
Triple Store
inserter


partitioner
Triple Store

16 lines (14 sloc) | 343 Bytes

```
PREFIX crm: <http://www.cidoc-crm.org/cidoc-crm/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

CONSTRUCT
{
    ?PhysicalObject dc:publisher "Rijksmuseum" .
}
WHERE
{
    BIND ( ?Subject as ?PhysicalObject )

    {
        ?PhysicalObject rdf:type crm:E22_Human-Made_Object .
    }
}
```

17 lines (15 sloc) | 464 Bytes

```
PREFIX crm: <http://www.cidoc-crm.org/cidoc-crm/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

CONSTRUCT
{
    ?Subject dc:title ?Text .
}
WHERE
{
    ?Subject crm:P1_is_identified_by ?Name .

    ?Name rdf:type crm:E33_E41_Linguistic_Appellation ;
        # Getty AAT: "preferred terms".
        # crm:P2_has_type <http://vocab.getty.edu/aat/300404670> ;
        crm:P190_has_symbolic_content ?Text .
}
```

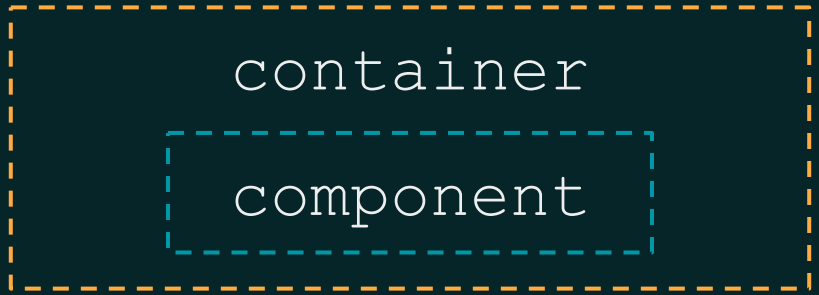
27 lines (24 sloc) | 751 Bytes

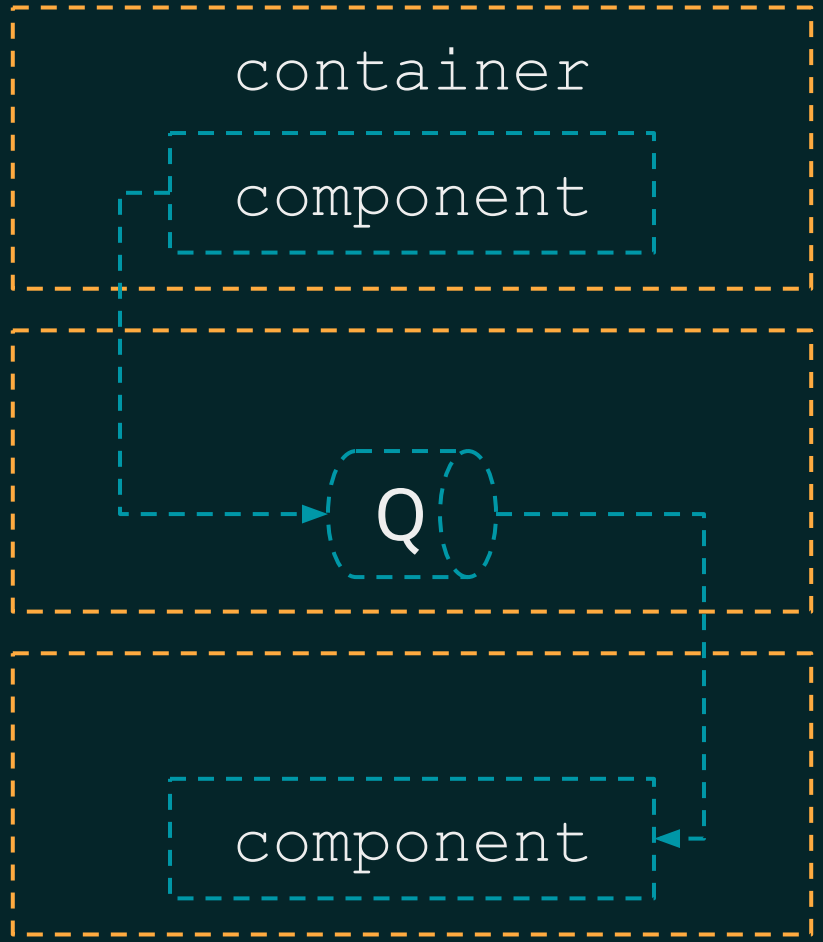
```
PREFIX crm: <http://www.cidoc-crm.org/cidoc-crm/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

CONSTRUCT
{
    ?Subject dc:identifier ?Handle .
    ?Subject dc:identifier ?ObjectNumber .
}
WHERE
{

    ?Subject crm:P1_is_identified_by ?Identifier .
    ?Identifier rdf:type crm:E42_Identifier

    {
        ?Identifier crm:P190_has_symbolic_content ?Handle ;
            # Getty AAT: "persistent identifiers".
            crm:P2_has_type <http://vocab.getty.edu/aat/300387580> .
    }
    UNION
    {
        ?Identifier crm:P190_has_symbolic_content ?ObjectNumber ;
            # Getty AAT: "accession numbers".
            crm:P2_has_type <http://vocab.getty.edu/aat/300312355> .
    }
}
```

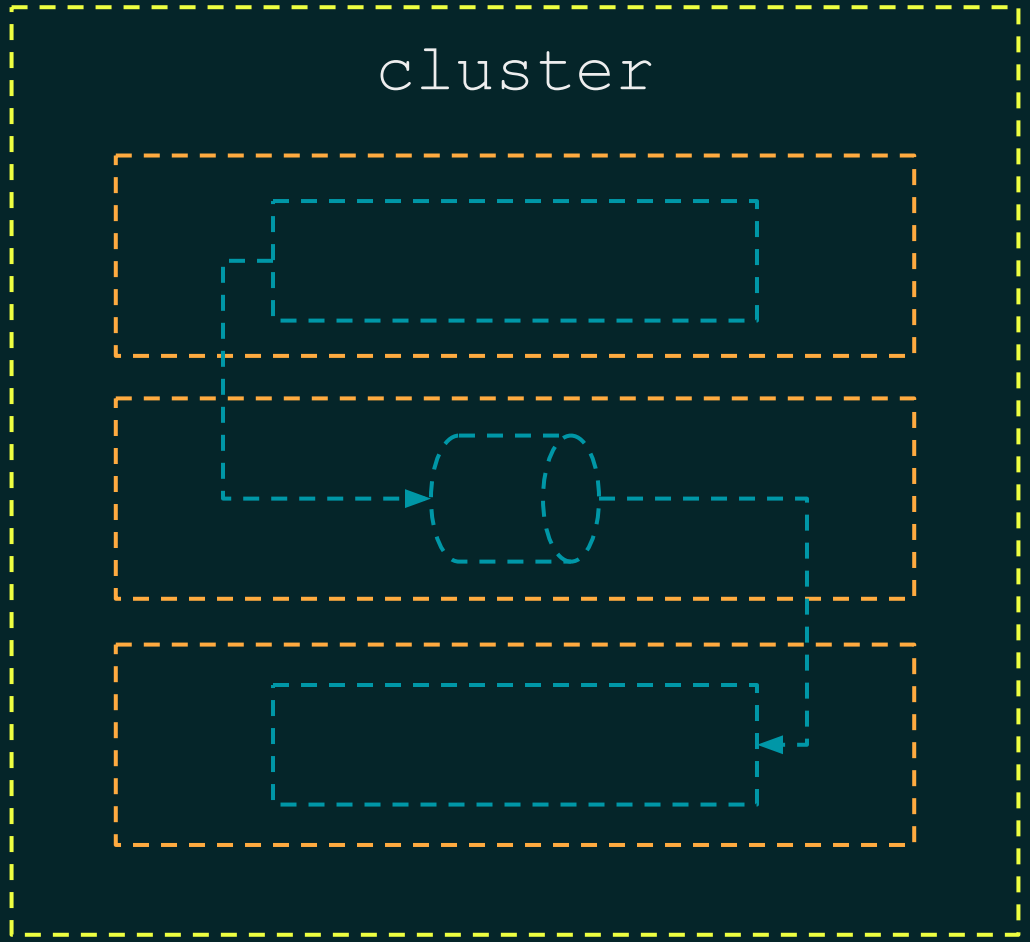
README.md

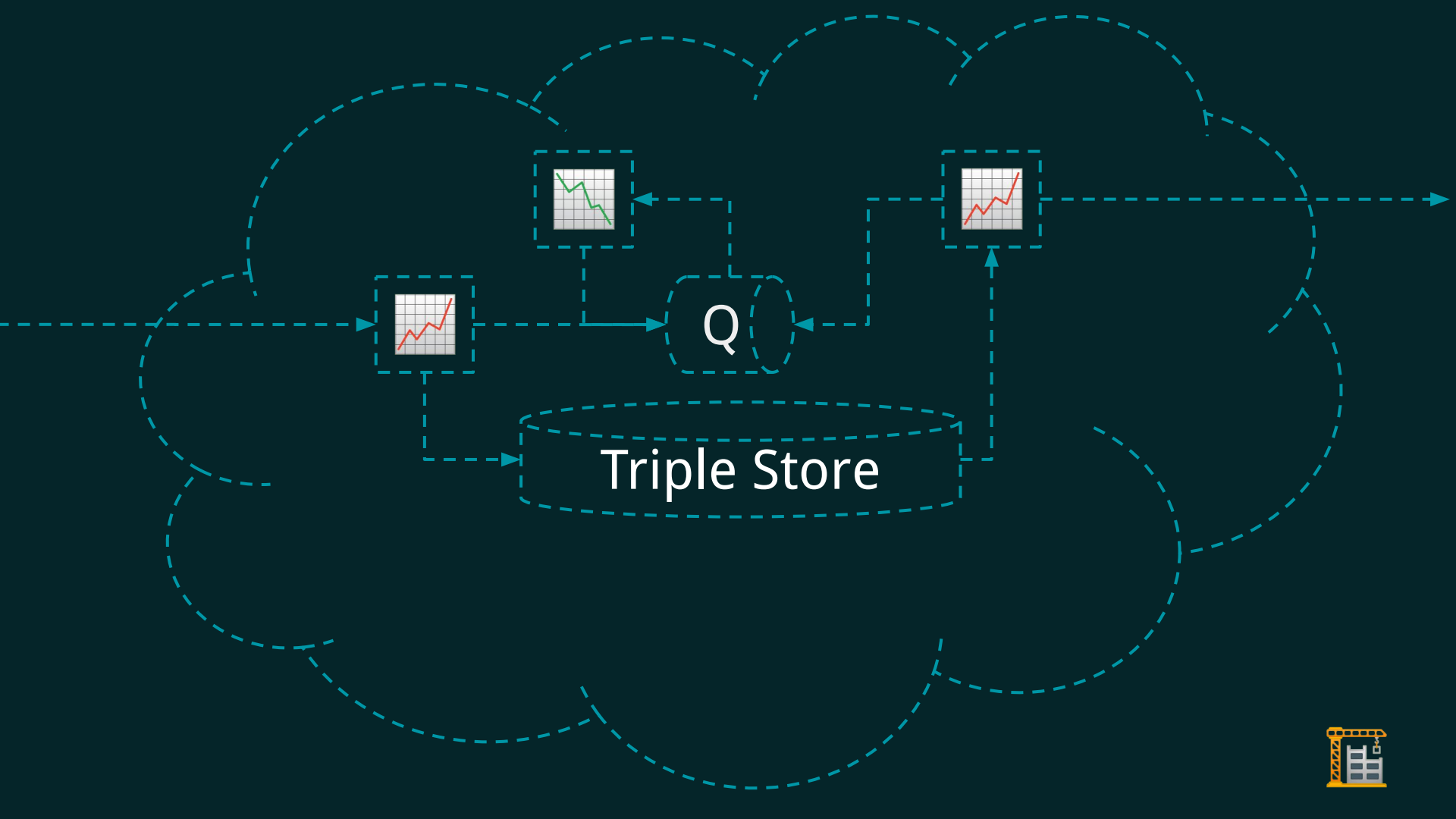
```python
from rdf_transformations import BaseComposition
from rdf_transformations.dublin_core.parts import Date, PhysicalObject, Identifier, Title
from sparql_constructor import TemplateContext


class Example(BaseComposition):

    def __init__(self, pid):
        with TemplateContext() as context:
            physical_object = context.create(PhysicalObject)
            physical_object.union(
                context.create(Date, source='PhysicalObject', target='?Subject'),
                context.create(Identifier, source='?PhysicalObject', target='?Subject'),
                context.create(Title, source='?PhysicalObject', target='?Subject')
            )
            physical_object.set_variable('?Subject', f'<{pid}>')

        super().__init__('Dublin Core', physical_object)


example = Example('https://id.rijksmuseum.nl/2005216')
print(example.sparql)
```

Scaling


container
component


container
component
Q
component


cluster


Q
Triple Store

What did we learn?

- 💖 We love standards
- 👌 Tiny apps are great
- 🗂️ More models, more flexibility
- 🕰️ Optimise later

This page intentionally left blank

Šī lapa ar nolūku ir atstāta tukša

c.dijkshoorn@rijksmuseum.nl

e.anderson@rijksmuseum.nl