
OXFORDIA

A Federated Data Ecosystem

Jackson Morgan · Solid Symposium 2026

Rare Is Not Rare

1 in 50,000

incidence of nemaline myopathy

the disease motivating this work

7,000+

rare diseases worldwide

each individually rare

300M

people affected globally

≈ 4% of the world population

No single institution has enough patients. Progress depends on pooling data across borders.

Pooling This Data Requires Considerations:



Jurisdictions

Different countries, different laws (HIPAA, GDPR, ...)



Privacy

Patient consent and re-identification risk



Harmonization

Each site stores data in its own schema and vocabulary



Scalability

Queries must run across many sites without central copies

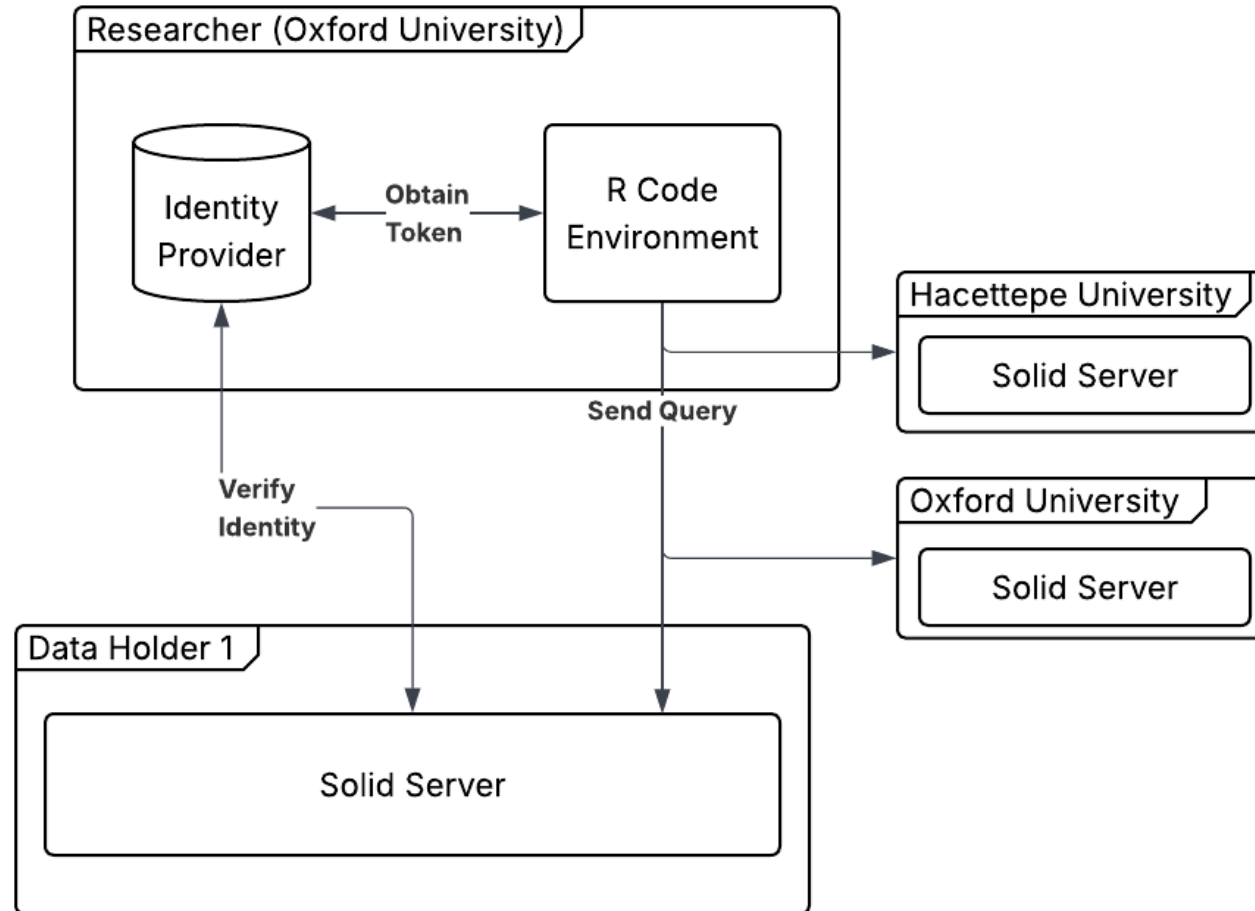


Ease of Use

Researchers and admins need tools they can actually deploy

OXFORDIA is built to address all five.

General Architecture



Three Roles in the System

0 1

Researchers

- Query data across many sites
- Get aggregate answers without raw records

0 2

Sites

- Stay compliant with local rules
- Share data without losing control

0 3

Admins

- Easy to deploy and operate
- Easy to upload data

REQUIREMENT 1

R Solid Auth Library

```
library(solidauthr)
library(httr2)

session <- solid_session(
  issuer = "https://solid-idp.university.edu/",
  client_id = Sys.getenv("SOLID_CLIENT_ID"),
  client_secret = Sys.getenv("SOLID_CLIENT_SECRET")
)

resp <- session$get("https://some-pod.example.org/private/data.ttl")
httr2::resp_body_string(resp)

session$put(
  "https://some-pod.example.org/private/new.ttl",
  body = "@prefix dct: <http://purl.org/dc/terms/> . <> dct:title \"Example\" .",
  content_type = "text/turtle"
)
```

```
install.packages("solidauthr")
```

```
github.com/SetMeld/solidauthr
```

REQUIREMENT 2

Easy Deploy

Universities should be able to spin up a compliant Pod with only a few one commands.

```
curl -LO https://github.com/OXFORDIA-project/OXFORDIA-node/releases
sudo apt install ./oxfordia-pod_<version>_amd64.deb
sudo vim /etc/default/oxfordia-pod
sudo systemctl enable --now oxfordia-pod
sudo journalctl -u oxfordia-pod -f
```

REQUIREMENT 3

Easy Query

Set up Query Targets

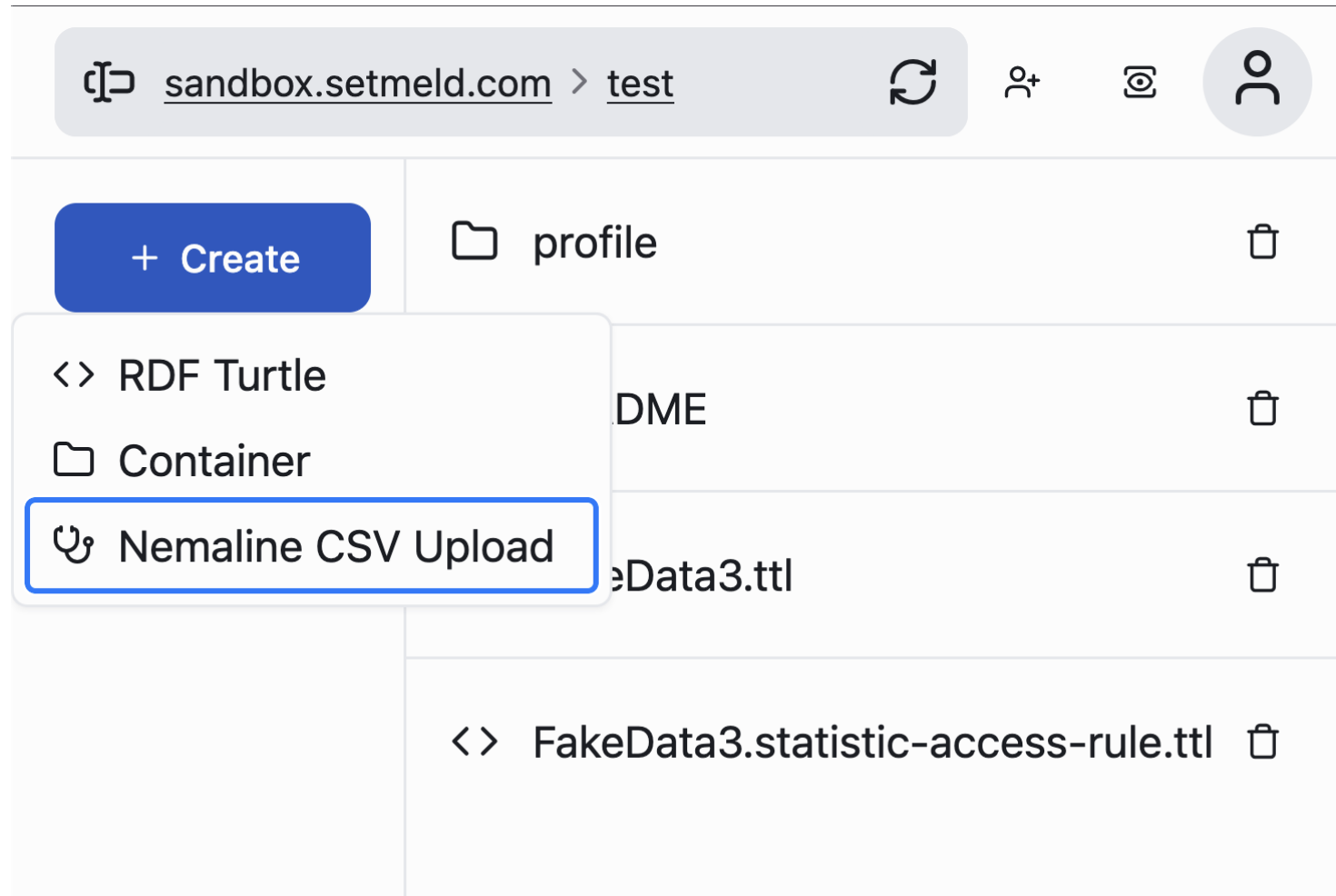
```
targets <- oxfordiar::ox_targets(  
  oxfordiar::ox_target(  
    name = Sys.getenv("OX_TARGET_1_SERVER_NAME"),  
    resource_uri = Sys.getenv("OX_TARGET_1_RESOURCE_URI")  
  ),  
  oxfordiar::ox_target(  
    name = Sys.getenv("OX_TARGET_2_SERVER_NAME"),  
    resource_uri = Sys.getenv("OX_TARGET_2_RESOURCE_URI")  
  ),  
  oxfordiar::ox_target(  
    name = Sys.getenv("OX_TARGET_3_SERVER_NAME"),  
    resource_uri = Sys.getenv("OX_TARGET_3_RESOURCE_URI")  
  )  
)
```

Perform Query

```
mean_result <- oxfordiar.stat.mean::ox_mean(  
  shortcut = oxfordiar.data.nemaline::ox_nemaline_shortcut(Sys.getenv(  
    "OX_MEAN_SHORTCUT"  
  )),  
  targets = targets,  
  auth = auth  
)
```

REQUIREMENT 4

Easy Data Sharing



THE NOVEL PROBLEM

Aggregate without Access

How do we let researchers ask aggregate questions of the data, **without granting full access?**

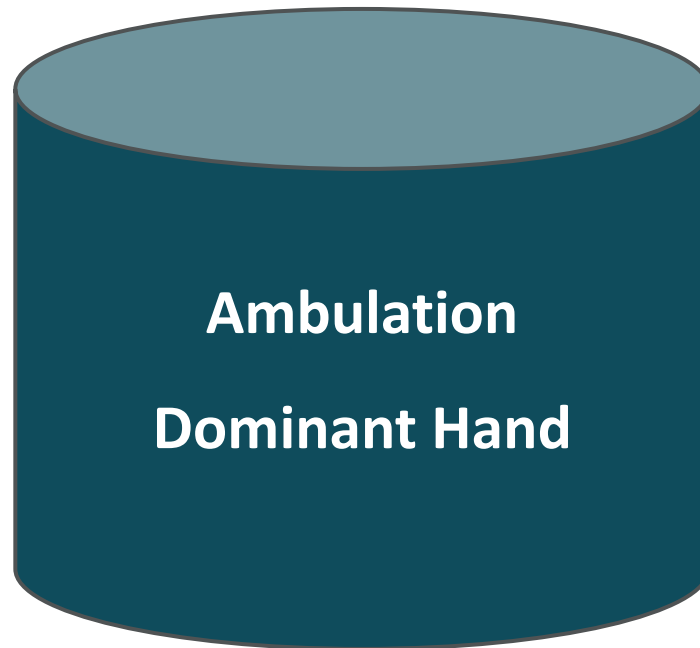
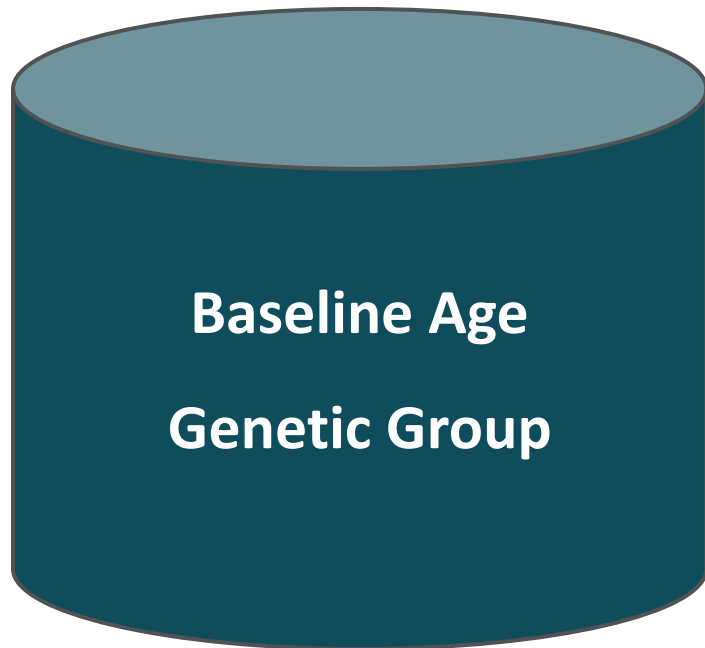
EXAMPLE

“Give me the mean baseline age, but not the individual values.”

ALTERNATIVES CONSIDERED

Why WAC and ACP fall short

✓ They can control what kind of data researchers have access to



✗ But they cannot control the kind of aggregations.

OUR APPROACH

Statistic Access Rule

The screenshot shows a web browser window with the URL `sandbox.setmeld.com > test > FakeData3.statistic-access-rule.ttl`. The page title is "Statistic Access Rule for FakeData3.ttl" and the data type is "nemaline". A "Save Changes" button is visible in the top right. The main content area is divided into two sections: "Allowed Agents" and "1. mean".

Allowed Agents

Only agent WebIDs listed here can run statistic queries for this resource.

- Add Agent**
- Remove**

1. mean **Remove**

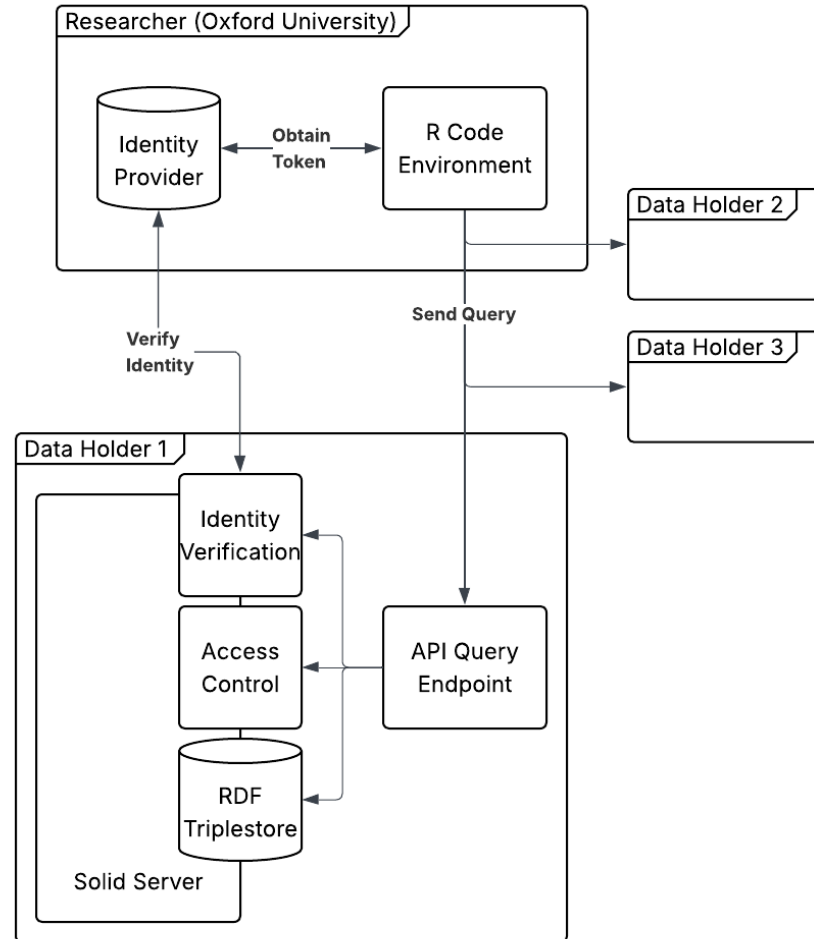
Graph Path

- BaselineAge**
- Advanced**

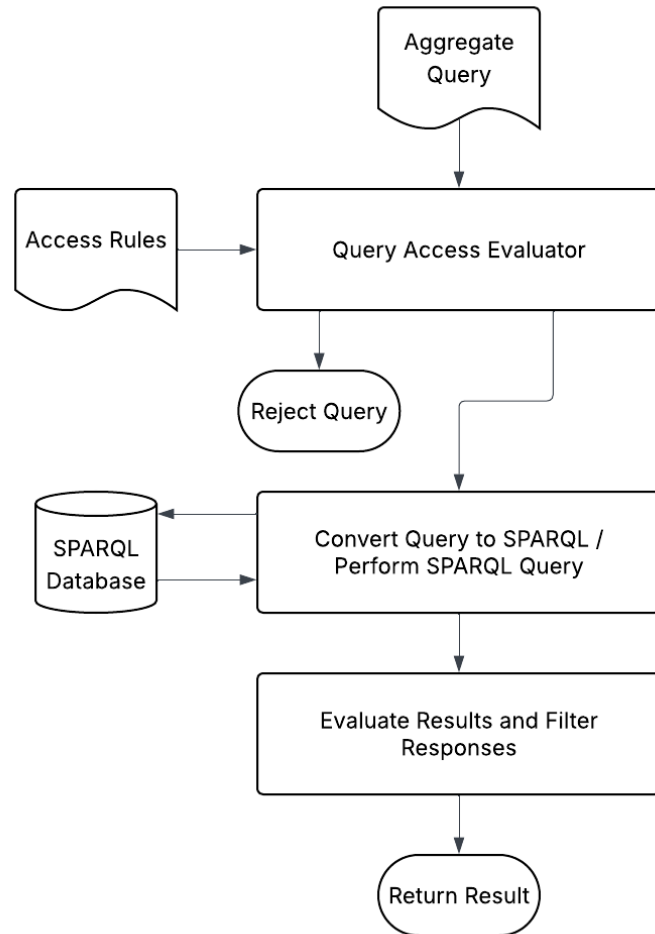
Min Count

ARCHITECTURE

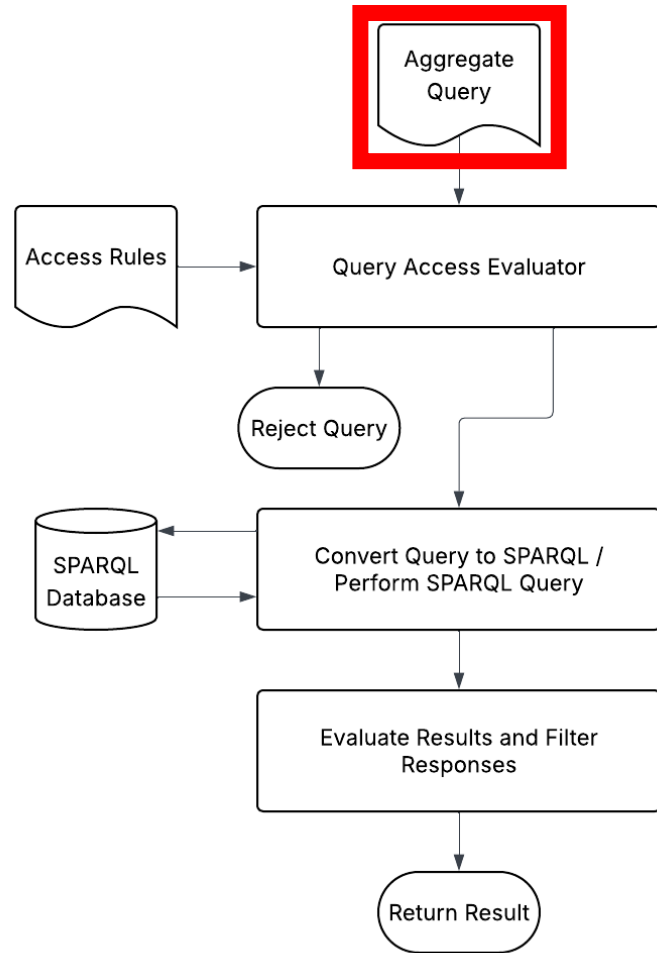
Updated Architecture



Updated Architecture

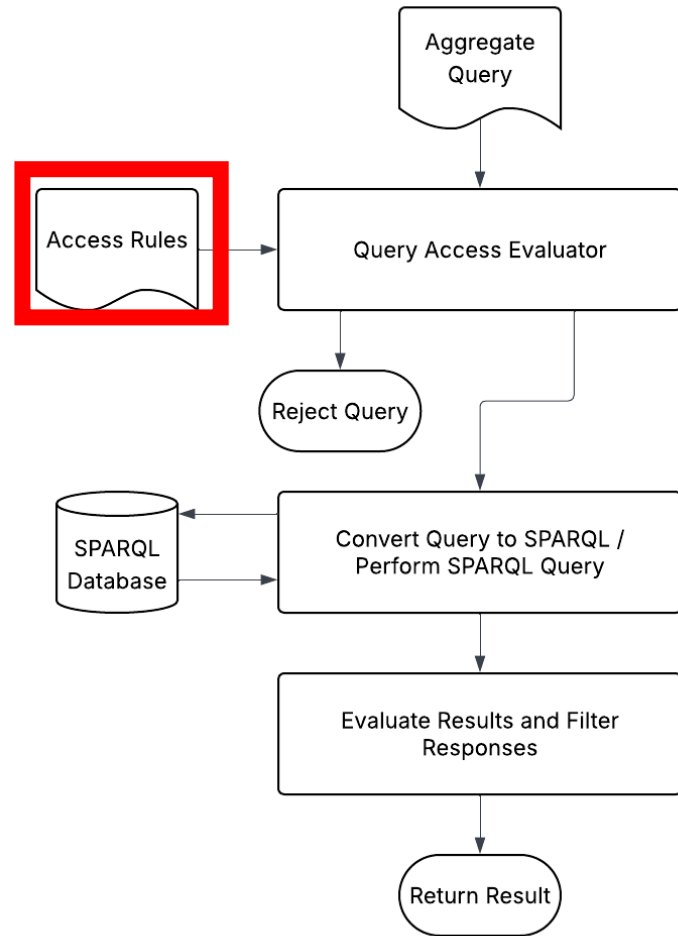


Updated Architecture



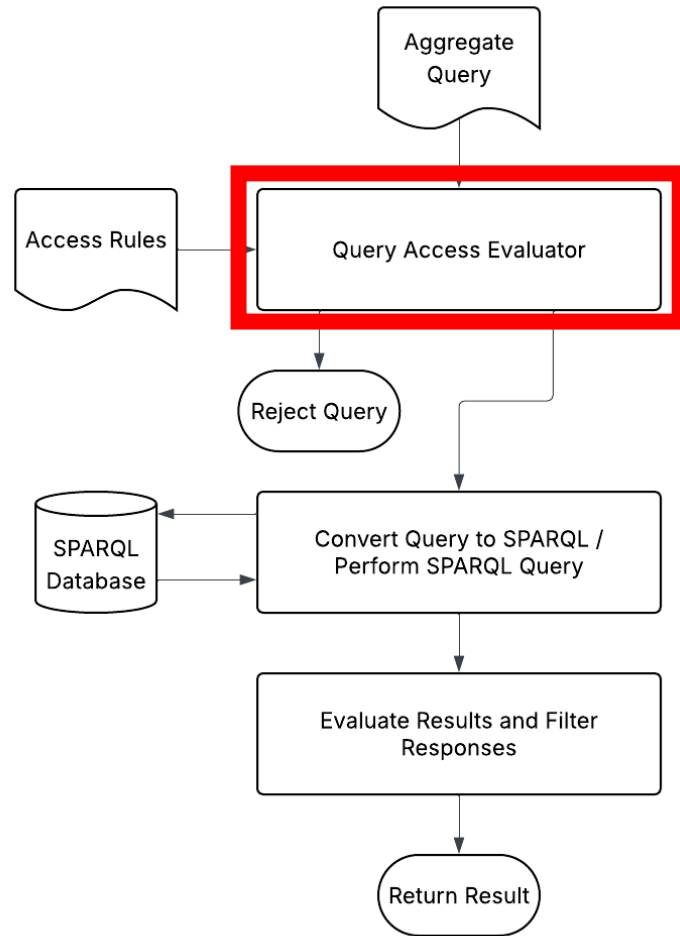
```
{
  resourceUri: 'http://localhost:3000/admin/FakeData2.ttl',
  graphPath: {
    '@id': '#graph-path-dfavcytw',
    name: 'BaselineAge',
    start: { '@id': '#graph-node-filter-ggtfwk3a', predicates: [ /* ... */ ] },
    steps: [ /* ... */ ]
  }
}
```

Updated Architecture



The screenshot shows a user interface for a query named "1. mean". At the top right is a "Remove" button. Below the query name is a "Graph Path" section with two buttons: "BaselineAge" and "Advanced". Underneath is a "Min Count" section with a text input field containing the value "1". A trash icon is visible on the right side of the interface.

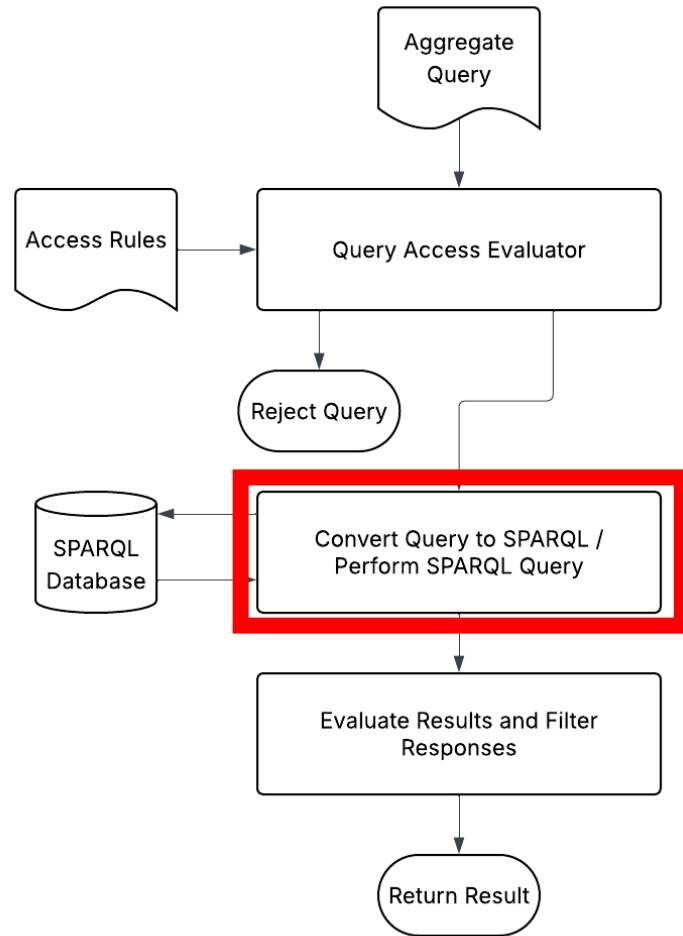
Updated Architecture



```
{
  resourceUri: 'http://localhost:3000/admin/FakeData2.ttl',
  graphPath: {
    '@id': '#graph-path-dfavcytw',
    name: 'BaselineAge',
    start: { '@id': '#graph-node-filter-ggtfwk3a', predicates: [ /* ... */ ] },
    steps: [ /* ... */ ]
  }
}
```

The screenshot shows a web interface for configuring a query. The query is named '1. mean' and has a 'Remove' button. The configuration includes a 'Graph Path' section with two buttons: 'BaselineAge' (selected) and 'Advanced'. Below this is a 'Min Count' section with a text input field containing the value '1'.

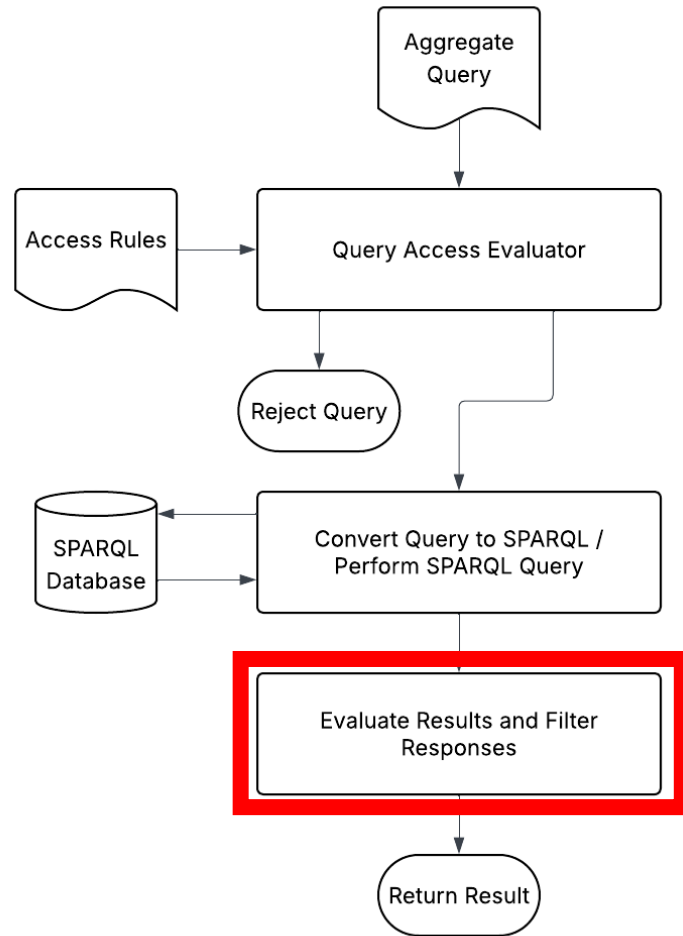
Updated Architecture



```
SELECT (AVG(?node4) AS ?mean) (COUNT(?node4) AS ?count)

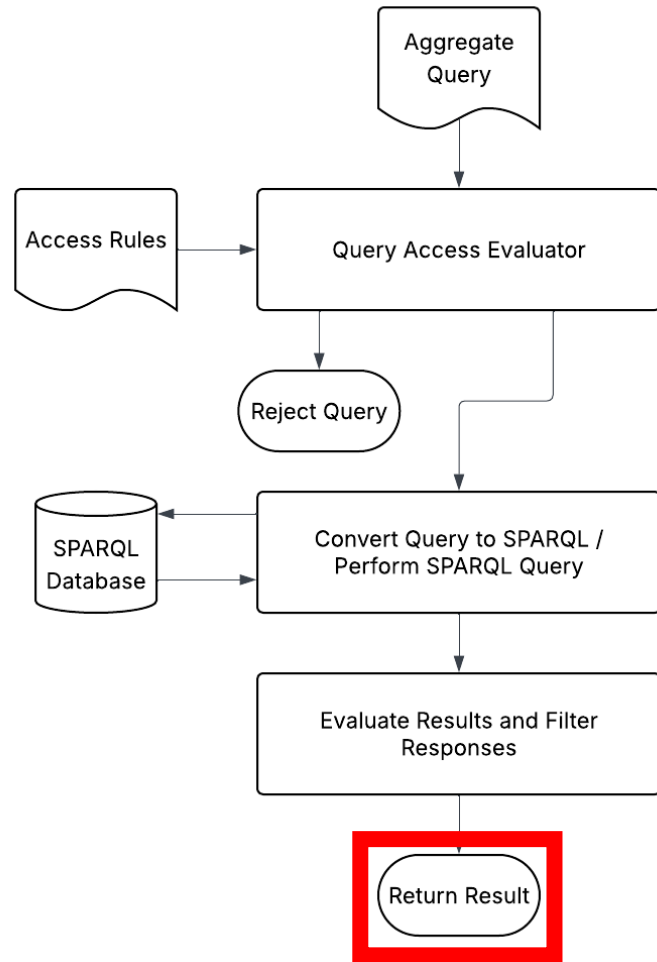
WHERE {
  GRAPH <http://localhost:3000/admin/FakeData2.ttl> {
    ?node0 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> ?some1 .
    VALUES ?some1 { <https://w3id.org/semanticarts/ns/ontology/gist/Person> }
    ?node0 <https://w3id.org/semanticarts/ns/ontology/gist/hasMagnitude> ?node2 .
    ?node2 <https://w3id.org/semanticarts/ns/ontology/gist/hasAspect> ?some3 .
    VALUES ?some3 { <https://w3id.org/semanticarts/ns/ontology/gist/Aspect_Age> }
    ?node2 <https://w3id.org/semanticarts/ns/ontology/gist/numericValue> ?node4 .
    FILTER(isNumeric(?node4))
  }
}
```

Updated Architecture



The screenshot shows a user interface for configuring a query. The title is '1. mean' with a 'Remove' button. Under 'Graph Path', there are two buttons: 'BaselineAge' and 'Advanced'. Under 'Min Count', there is a text input field containing the number '1'. A large orange arrow points to the '1' in the 'Min Count' field.

Putting It Together



```
{
  "mean": 20.2975,
  "count": 60
}
```

```
{
  "mean": 23.1138,
  "count": 40
}
```

```
{
  "mean": 21.9572,
  "count": 80
}
```



$$(20.2975 * 60 + 23.1138 * 40 + 21.9572 * 80) / (60 + 40 + 80) =$$

21.63098888889

Tradeoffs

✓ Pro

Fine-grained control over where/when computation runs, to adapt to privacy needs:

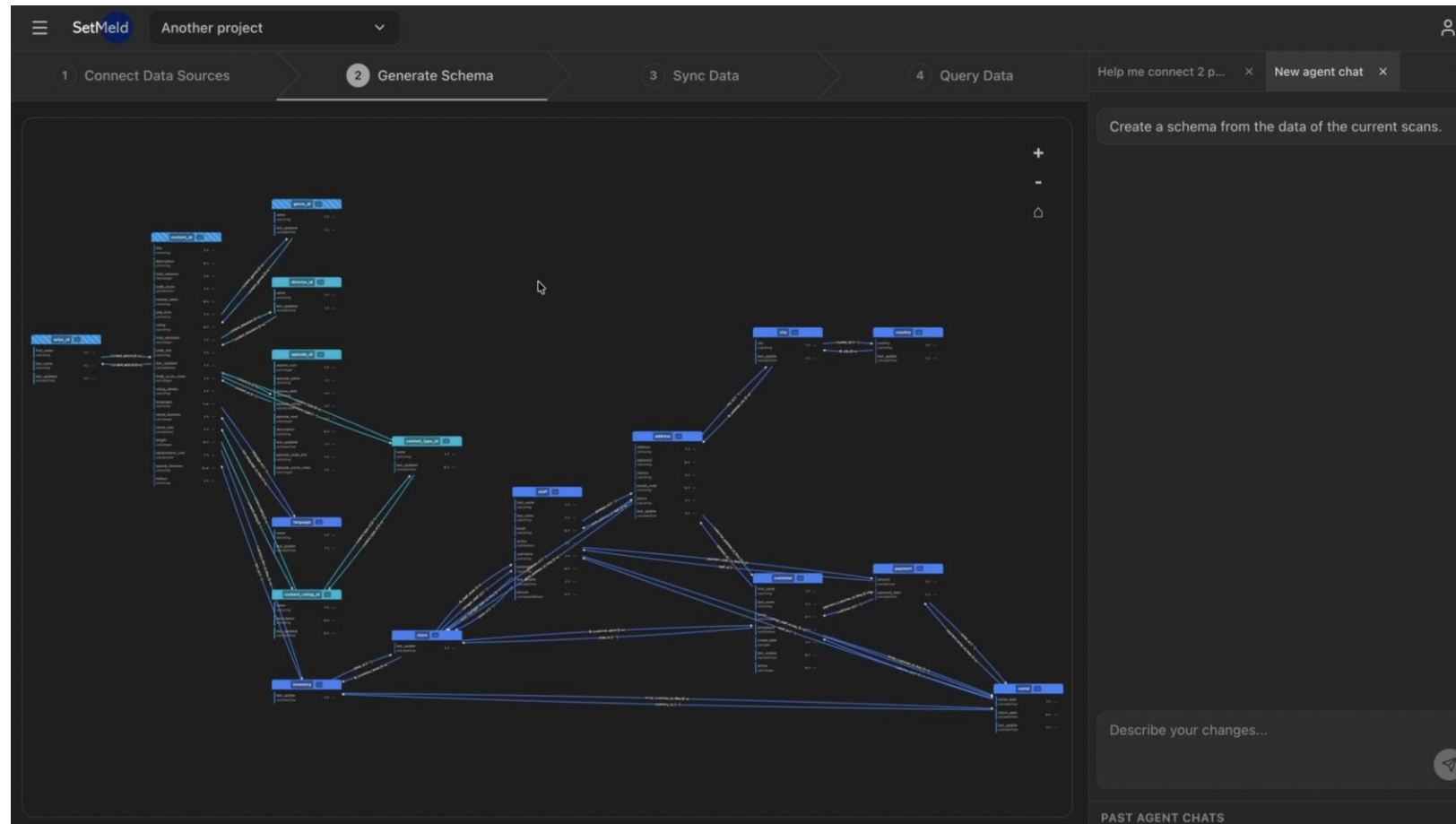
- On individual servers (before query)
- On individual servers (after query)
- On the client

X Con

New plugins must be installed for each new type of query.

THE NOVEL PROBLEM

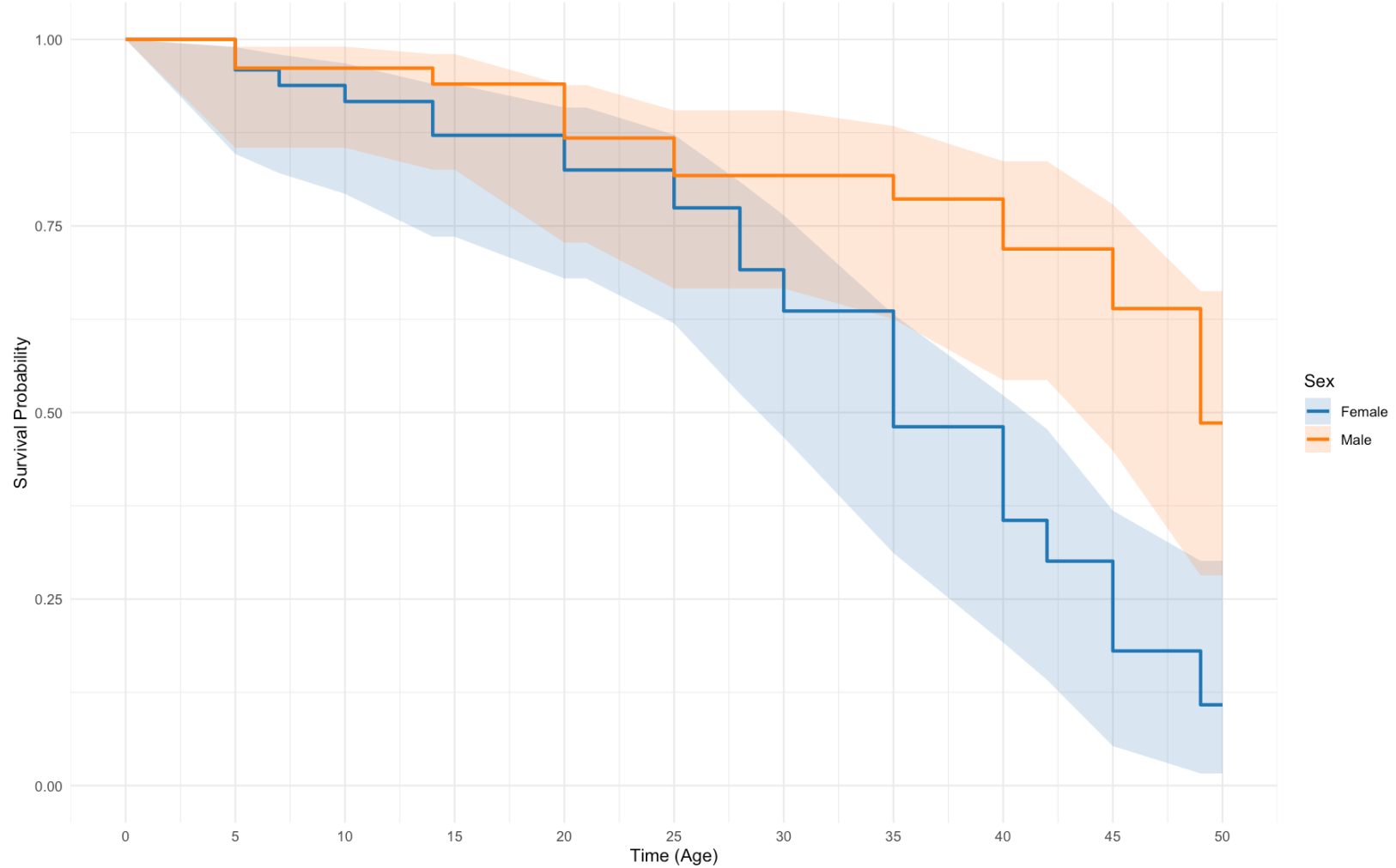
Integrating with Legacy Datasets



PRACTICAL USE CASE

Kaplan–Meier Analysis

Federated Kaplan-Meier Survival Curve
Aggregated from 6 data pods | Grouped by: Sex | Resolution: 5



Thank You

Questions welcome.

Jackson Morgan · Solid Symposium 2026