Federation: Beyond the home organization.

Jerven Bolleman

Principal Software Engineer – Swiss-Prot group

(Con)Federation

20 institutional partners all over Switzerland

Legend — lines:

- IC 1 — Genève-Aéroport–Bern–Zürich HB–St.Gallen
- IC 2 — Zürich HB–Zug–Lugano
- IC 3 — Basel SBB–Zürich HB–Chur
- IC 4 — Zürich HB–Schaffhausen
- IC 5 — Genève-Aéroport/Lausanne–Biel/Bienne–Zürich HB(–St.Gallen)
- IC 6 — Basel SBB–Bern–Brig
- IC 8 — Brig–Bern–Zürich HB–Romanshorn
- IC 21 — Basel SBB–Luzern–Lugano
- IC 51 — Basel SBB–Biel/Bienne
- IC 61 — Basel SBB–Bern–Interlaken Ost
- IR 13 — Zürich HB–St.Gallen–Chur
- IR 15 — Genève-Aéroport–Lausanne–Bern–Luzern
- IR 16 — Bern–Olten–Brugg AG–Zürich HB
- IR 17 — Bern–Burgdorf–Olten(–Zürich HB)
- IR 20 — Basel SBB–Luzern–Erstfeld(–Airolo)
- IR 27 — Basel SBB–Olten–Luzern
- IR 36 — Basel SBB–Brugg AG–Zürich HB(–Zürich Flughafen)
- IR 37 — Basel SBB–Aarau–Zürich HB(–St.Gallen)
- IR 46 — Zürich HB–Zug–Erstfeld(–Airolo)
- IR 70 — Luzern–Zürich HB
- IR 75 — Luzern–Zürich HB–Konstanz
- IR 90 — Genève-Aéroport–Lausanne–Brig
- RE — Bern–Burgdorf–Olten
- RE — Erstfeld–Lugano(–Milano)
- RE — Zürich HB–Chur
- RE — Aarau–Zürich HB
- RE — Zürich HB–Bülach–Schaffhausen
- RE — Bern–Biel/Bienne
- RE — Genève–Lausanne–Vevey

Legend — symbols:

- IC, IR and RE
- International long-distance services
- RegioExpress and private railways (extract)
- Long-distance stop
- Long-distance stop at certain times
- Capital of the canton

Rail Service 0848 44 66 88 (CHF 0.08/min.)
sbb.ch/en

Cities / stations:

Schaffhausen, Kreuzlingen, Konstanz, Romanshorn, Weinfelden, Amriswil, Bülach, Frauenfeld, Rorschach, St.Margrethen, Heerbrugg, Winterthur, St.Gallen, Zürich Oerlikon, Zürich Flughafen, Wil, Uzwil, Flawil, Gossau SG, Herisau, Appenzell, Altstätten SG, Basel SBB, Rheinfelden, Stein-Säckingen, Frick, Brugg AG, Baden, Dietikon, Zürich Altstetten, Laufen, Muttenz, Liestal, Lenzburg, Sissach, Aarau, Thalwil, Zürich HB, Buchs SG, Delémont, Gelterkinden, Olten, Wädenswil, Siebnen-Wangen, Ziegelbrücke, Walenstadt, Moutier, Langenthal, Zofingen, Baar, Pfäffikon SZ, Glarus, Sargans, Bad Ragaz, Grenchen Nord, Solothurn, Sursee, Zug, Rotkreuz, Landquart, Grenchen Süd, Oensingen, Herzogenbuchsee, Arth-Goldau, Schwyz, Brunnen, Chur, Biel/Bienne, Lyss, Wynigen, Burgdorf, Luzern, Flüelen, Neuchâtel, Bern, Sarnen, Stans, Altdorf, Erstfeld, Göschenen, Yverdon-les-Bains, Fribourg/Freiburg, Thun, Interlaken Ost, Interlaken West, Airolo, Ambri-Piotta, Faido, Lavorgo, Bodio TI, Lausanne, Falézieux, Vevey, Romont, Spiez, Biasca, Castione-Arbedo, Montreux, Aigle, Bex, St-Maurice, Brig, Visp, Bellinzona, Giubiasco, Allaman, Rolle, Gland, Nyon, Sierre/Siders, Leuk, Sion, Lamone-Cademario, Coppet, Martigny, Lugano, Genève-Aéroport, Genève, Mendrisio, Chiasso, Morges, Renens VD

Fostering scientific collaboration

SIB collaboration network

Past or ongoing collaborations between SIB groups

254 publications in 2018

Fostering scientific collaboration

Federation inside the home
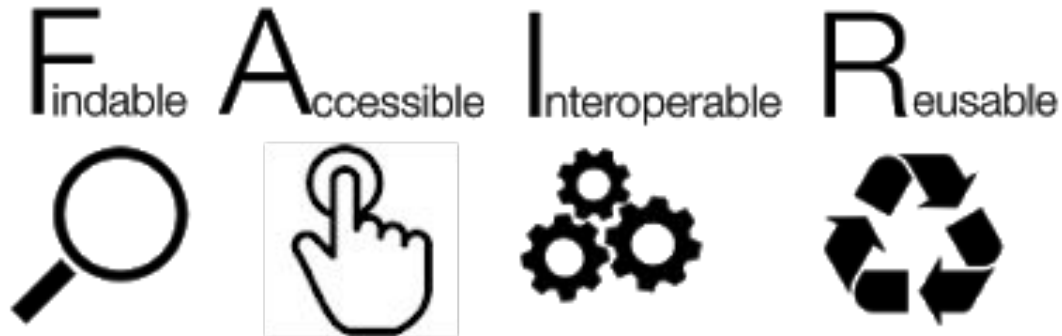

Publicly available SPARQL endpoints

More internal scale

Thousands to multi billions

Lots of different implementations

- RDF4j native
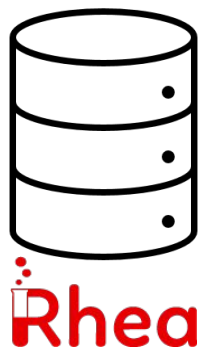- Virtuoso 7.2 Open Source
- GraphDB
- Blazegraph (to be replaced)
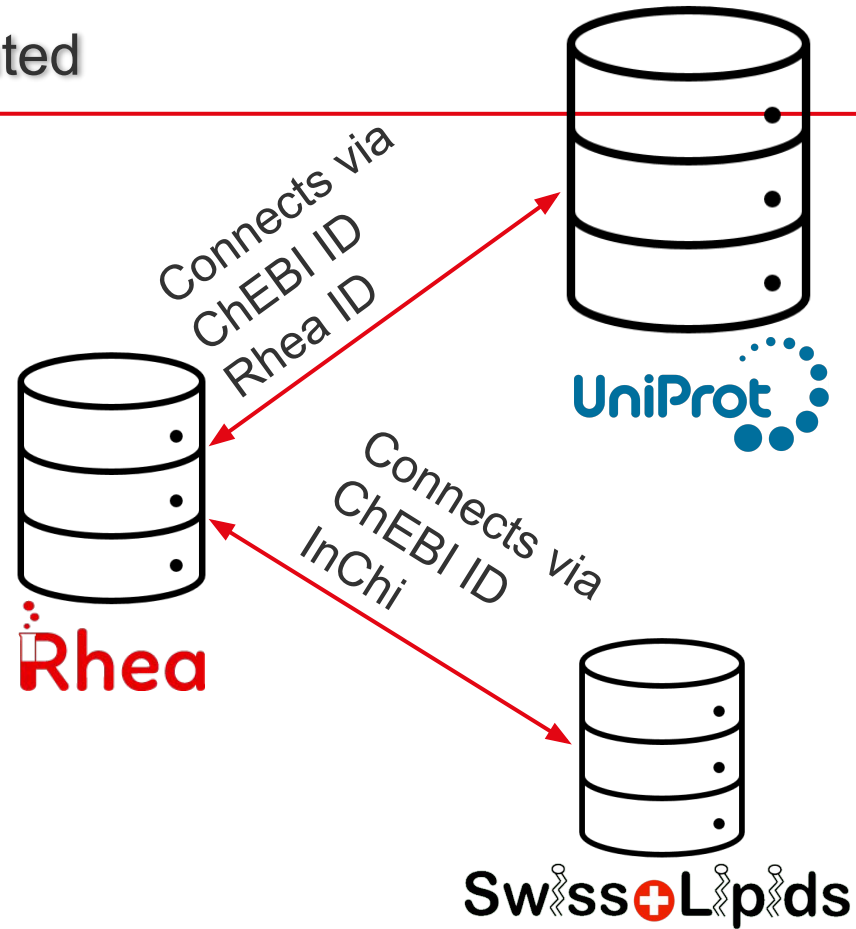
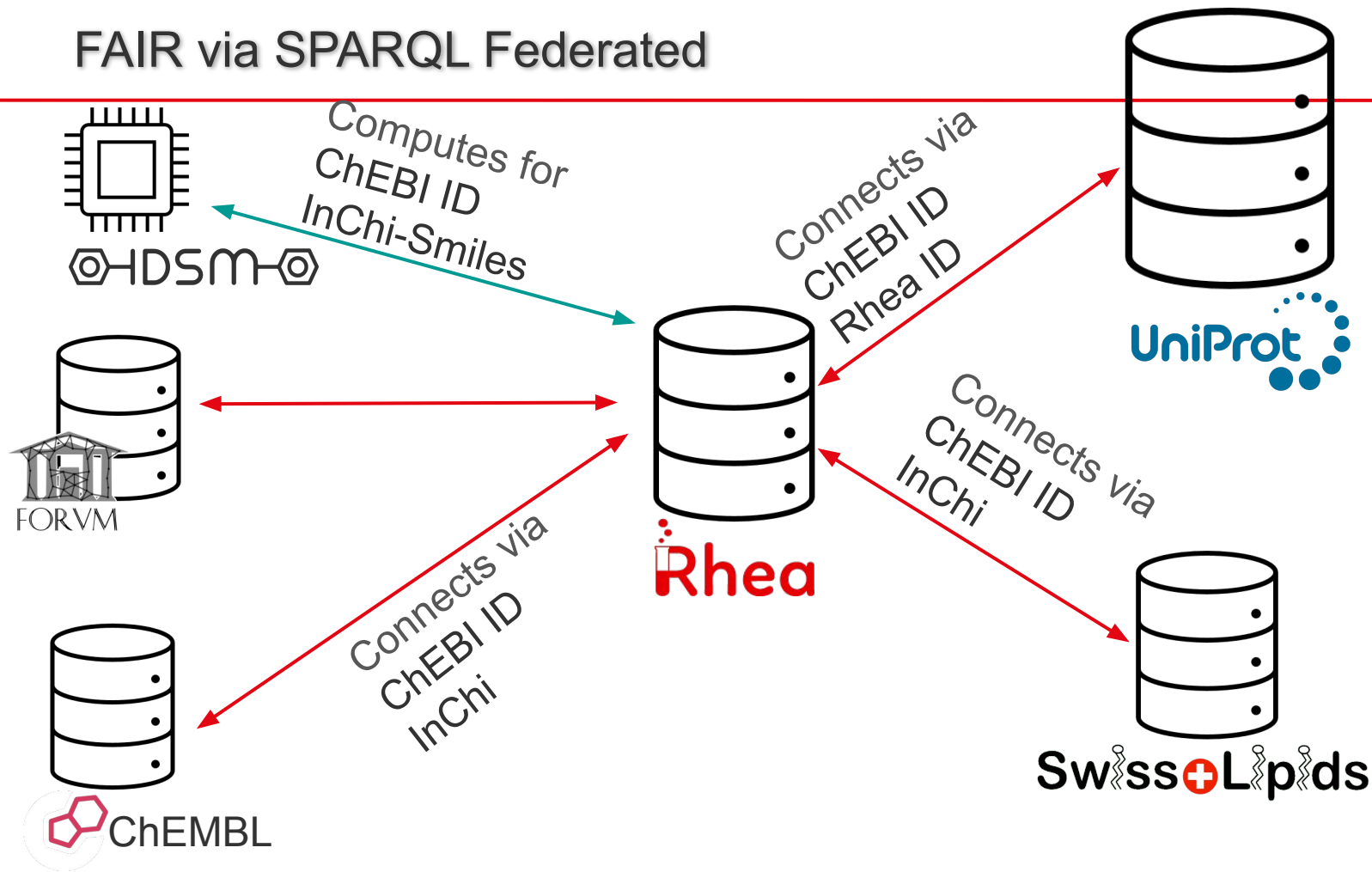# The FAIRest format of them all

F indable   A ccessible   I nteroperable   R eusable
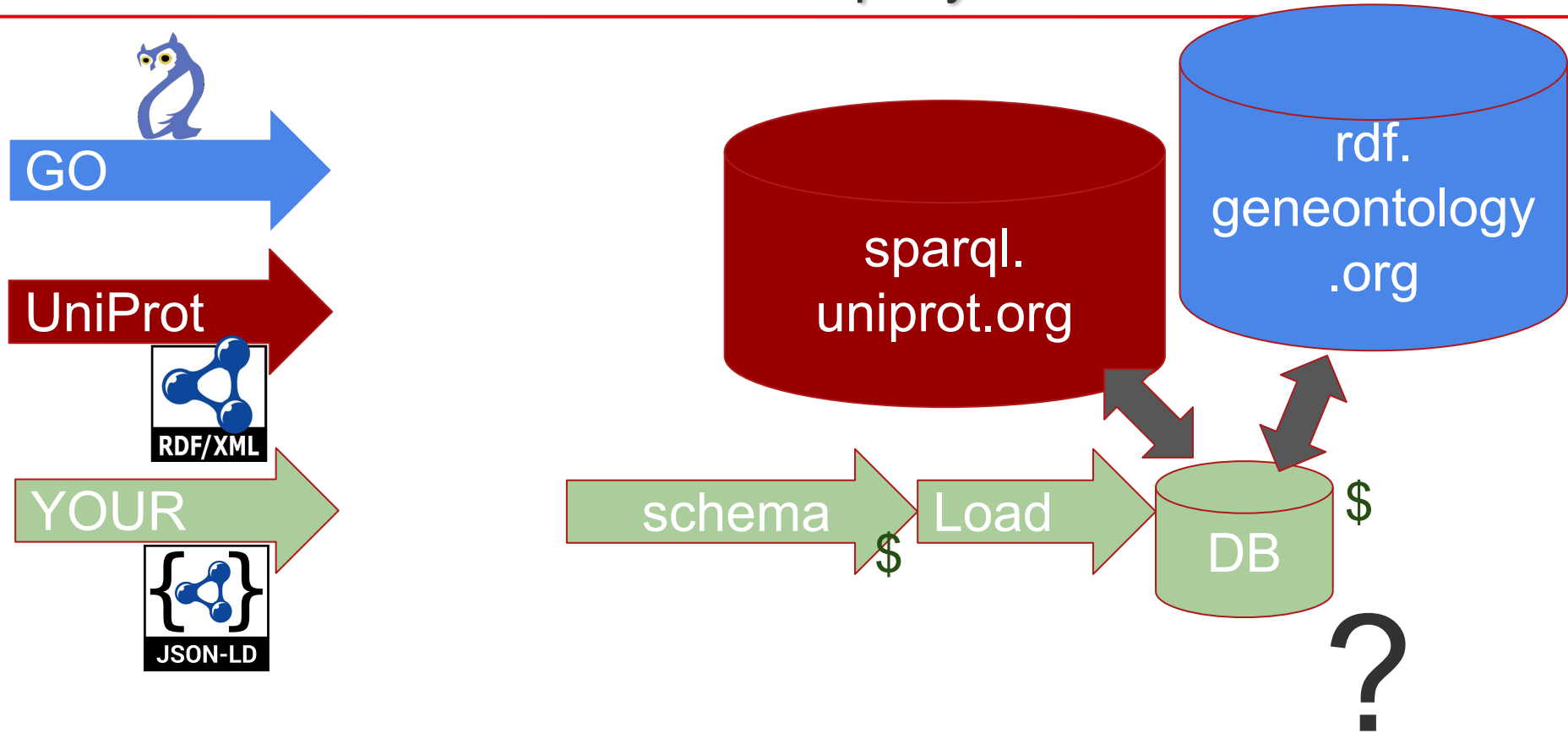
FAIR via SPARQL

Chemical reaction database

https://github.com/sib-swiss/sparql-training

FAIR via SPARQL Federated

Connects via ChEBI ID Rhea ID

Connects via ChEBI ID InChi

Rhea

UniProt

Swiss Lipids

FAIR via SPARQL Federated

Computes for
ChEBI ID
InChi-Smiles

Connects via
ChEBI ID
Rhea ID

Connects via
ChEBI ID
InChi

Connects via
ChEBI ID
InChi

HDSM

FORVM

ChEMBL

Rhea

UniProt

Swiss Lipids

Not possible with REST services

GO

UniProt

RDF/XML

YOUR

JSON-LD

Only SPARQL can connect databases outside the organization.

Because ID's are IRIs

rdf.
ontology
.org

schema $

Load

DB $

?
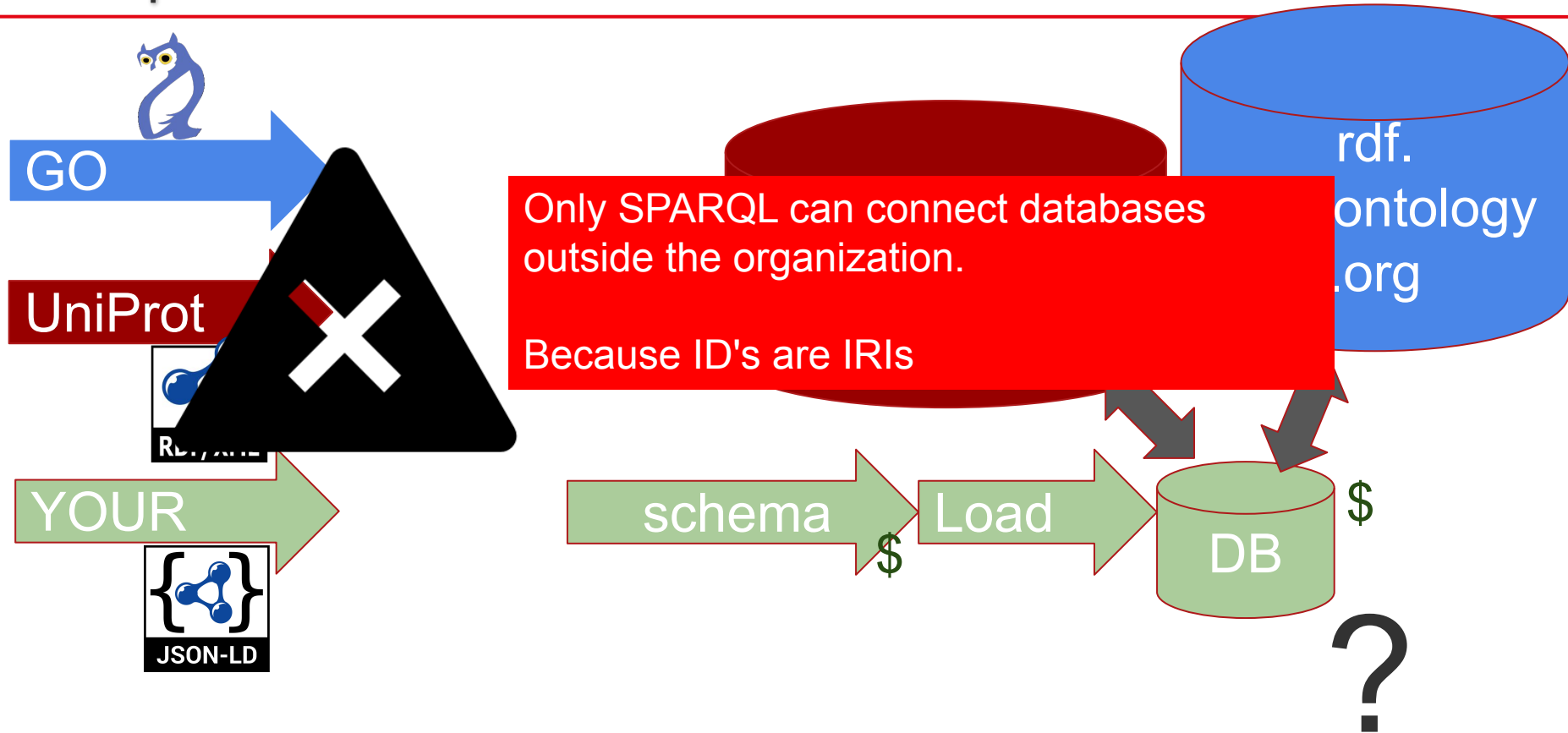
Ask DB 2 to be part of the solution Rhea

```
BASE <http://purl.uniprot.org/uniprot/>
PREFIX ...

SELECT
 ?protein ?rhea
WHERE {
  SERVICE <https://sparql.rhea-db.org/sparql> {
    ?rhea rdfs:subClassOf rh:Reaction ;
      rh:status rh:Approved ;
      rh:side/rh:contains/rh:compound/rh:chebi/rdfs:subClassOf+
CHEBI:26739 .
  }
  ?protein up:annotation/up:catalyticActivity/up:catalyzedReaction ?rhea .
}
```

Issue when federating

- Bugs add up
    - SPARQL 1.1 compliance
    - Downtime
- Optimizers are blind
    - Order
    - Distance
    - Endpoints change
        - Number and kind of triples
        - Implementation

Federation – UniProt to Japanese Medical Dictionary

```
SELECT
  ?protein ?englishLabelStr
WHERE {
  SERVICE <http://data.allie.dbcls.jp/sparql>{
    [] rdfs:label "アミロイド前駆体タンパク質"@ja ;
       rdfs:label ?englishLabel .
    FILTER(lang(?englishLabel) = "en")
  }
  BIND (STR(?englishLabel) AS ?englishLabelStr)
  ?protein
    a up:Protein ;
    (up:recommendedName|up:alternativeName) ?structuredName .
}
```

```
SELECT ?taxon ?ncbiTaxid ?eunisTaxon ?eunisname ?image
WHERE
{
    ?taxon a up:Taxon .
    ?taxon rdfs:subClassOf taxon:8835 .
    BIND(substr(str(?taxon), 35) AS ?ncbiTaxid)
    SERVICE <https://semantic.eea.europa.eu/sparql> {
        ?eunisTaxon a eunisSpecies:SpeciesSynonym ;
                eunisSpecies:binomialName ?eunisname ;
                eunisSpecies:sameSpeciesNCBI ?ncbiTaxid ;
                foaf:depiction ?image .
    }
}
```

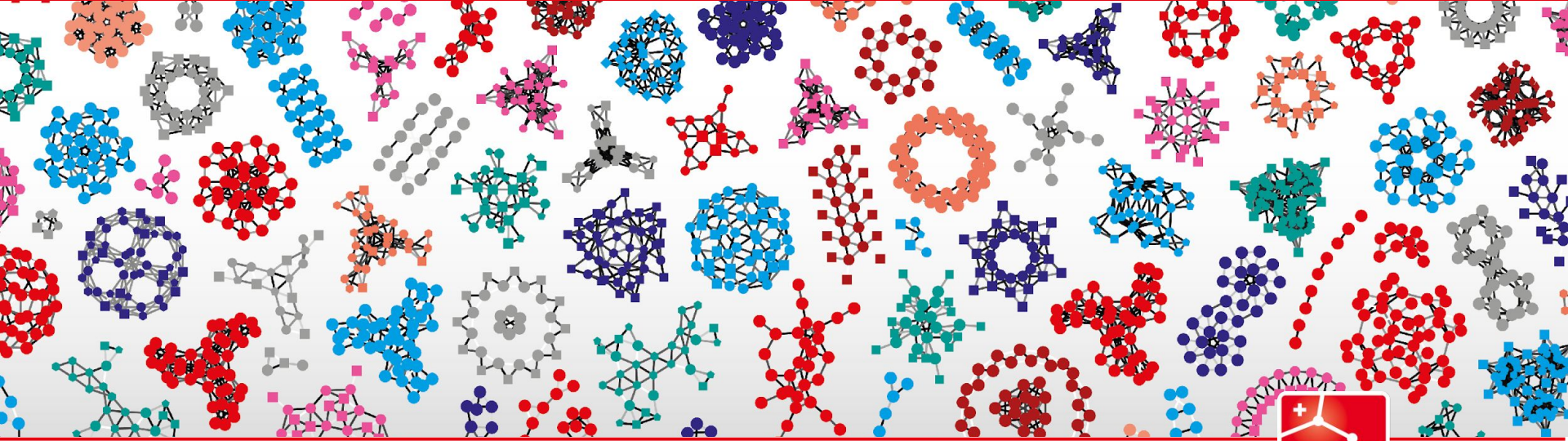# Federation – Rhea to Sachem (compute)

```
SELECT ?chebi ?name ?rhea
WHERE {
  SERVICE <https://idsm.elixir-czech.cz/sparql/endpoint/chebi>
  { ?ssc sachem:compound ?chebi ;
       sachem:score ?ligandSimilarityScore ;
       sachem:similaritySearch/sachem:query
"""CC1=C(CCC([O-])=O)C2=...=c(C)c(CCC([O-])=O)c1=C2""";
       sachem:cutoff "8e-1"^^xsd:double ;
       sachem:aromaticityMode sachem:aromaticityDetect ;
  }
  ?rhea
  rh:side/rh:contains/rh:compound/(rh:chebi|(rh:reactivePart/r
  h:chebi)|(rh:underlyingChebi/rh:chebi)) ?chebi .
  ?chebi up:name ?name .
  }
}
```

Federation – Query Against Programs

- Override 1 method
  - **RDF4j -** getStatementsInternal
  - **RDFlib -** triples
  - **Jena -** eval
  - **OxiGraph** - encoded_quads_for_pattern
  - **Ruby-RDF** - query-instance_method

Power of all SPARQL

Knowledge should be shared !