A RESTful JSON-LD Architecture for Unraveling Hidden References to Research Data

Konstantin Baierer, Philipp Zumstein
Mannheim University Library
SWIB15, 2015-11-24

Overview

- Context (data citations), Problem description
- Project InFoLiS: Overview
- Technical Architecture
- Demo

InFoLiS-Project (Integration of research data and literature)

gesis Leibniz Institute for the Social Sciences

UNIVERSITY OF MANNHEIM

HOCHSCHULE DER MEDIEN

UB MANNHEIM

Funded by the DFG Deutsche Forschungsgemeinschaft 2nd (funding) phase

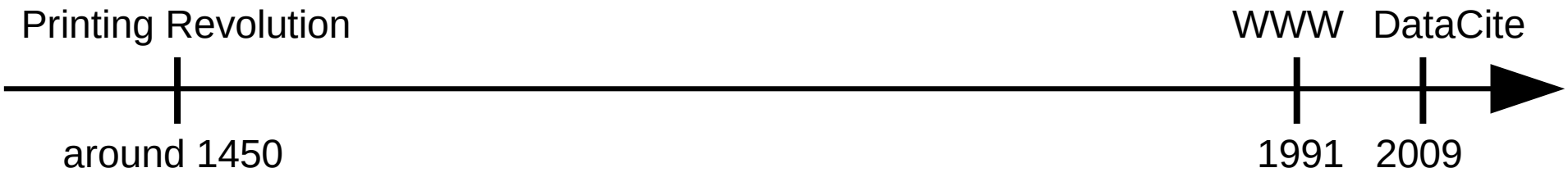
Data Citation

- **Research data** = raw data, intermediate results in the research process
  - Your own research data
  - Research data from a data provider
  - Data from official statistics
  - Research data from your colleague
- **Citation** = formal structured reference to another scholarly work
- **Data Citation** = formal structured reference to research data

Début of Data Citation

When was the first structured data citation used in a publication?

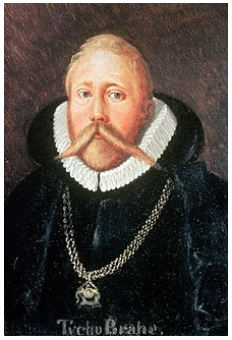
Maybe around the year **2000**?
(send your suggestion to @infolis_project)



When was the first unstructured reference to research data used in a publication?

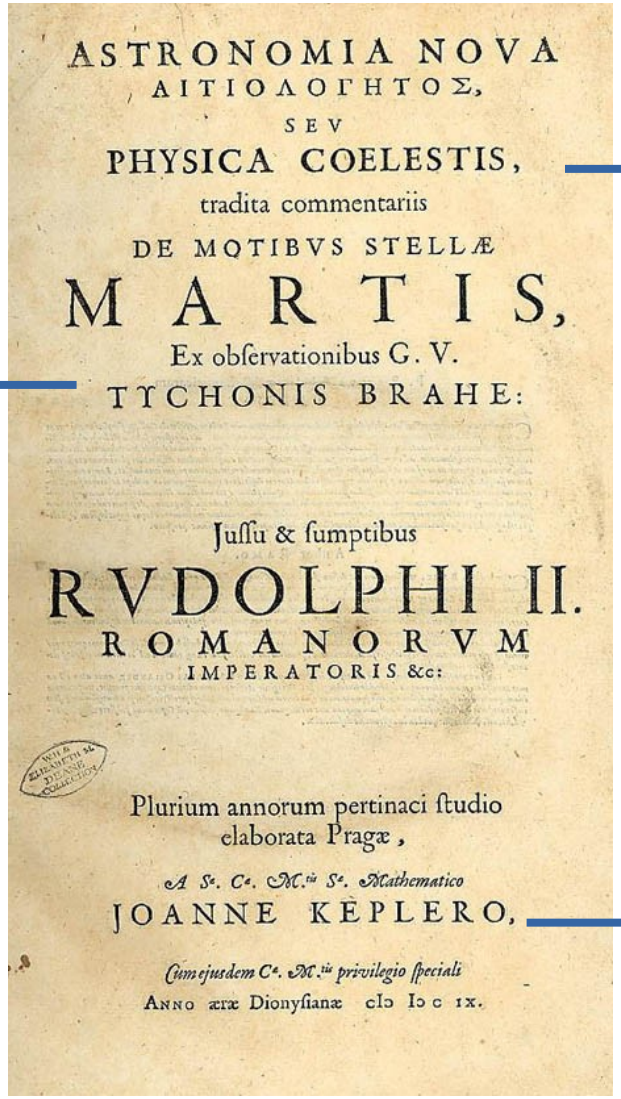
1609 or before (proof follows ...)

First Unstructured “Data Citation”



Tycho de Brahe
(1546-1601)

cites data from



title

“New Astronomy, Based upon Causes, or Celestial Physics, Treated by Means of Commentaries on the Motions of the Star Mars, from the Observations of Tycho Brahe”

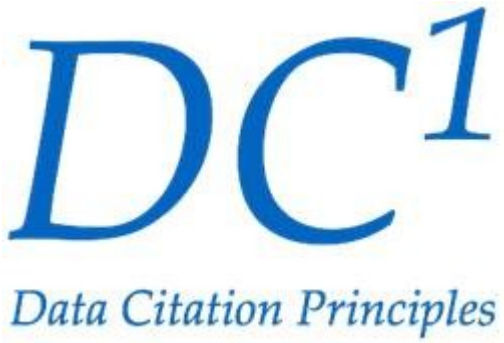
author

Johannes Kepler
(1571-1630)

Kepler (1609): Astronomia nova

Data Citations Principles

- Joint Declaration of Data Citation Principles:
  1. Importance
  2. Credit and Attribution
  3. Evidence
  4. Unique Identification
  5. Access
  6. Persistence
  7. Specificity and Verifiability
  8. Interoperability and Flexibility



- Currently 100 institutional supporters (39 data centers, 17 publishers, 26 societies and others)

Data Citations Format

Suggested Format by DataCite

creator (publication year): title.
version. publisher. resource type.
identifier

Rattinger, Hans; Roßteutscher, Sigrid; Schmitt-Beck, Rüdiger; Weßels, Bernhard (2012): Wahlkampf-Panel (GLES 2009). Version: 3.0.0. GESIS Datenarchiv. Dataset. doi:10.4232/1.11131

Data citation guidelines are included in APA style, NLM*, CMoS*, American Sociological Review, The American Economic Review, …
(*) at handles databases

But in practice...

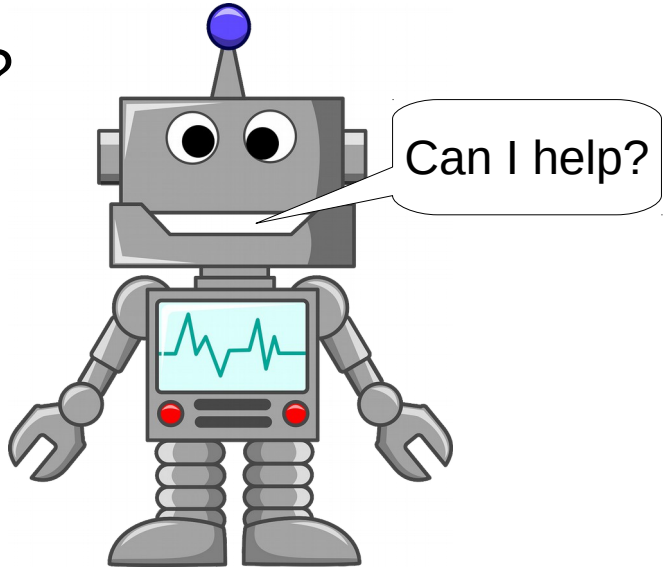
- Table 1: Population forecast for Germany depending on age cohorts – proportion in percent. Data base: 10th Population Forecast of the Federal Statistical Office.

- It already refers the IGLU study, according to which the ten-years-olds in Germany in a international comparison of reading literacy perform significantly better than the fifteen-years-olds.

- For this purpose, data from the Socio-Economic Panel (SOEP) of the years 1990 and 2003 are used and for both periods, the impact factors are estimated using linear regression models.

Processing Steps

- Detect data citations in running (full)text
- Resolve and normalize data citations
  - IGLU = Internationale Grundschul-Lese-Untersuchung
  - SOEP = Socio-Economic Panel
    = Sozio-oekonomische Panel
    = Sozioökonomische Panel
- Uniquely identify data citations
  - IGLU 2001, IGLU 2006 oder IGLU 2011?
- Find the cited research data
  - url
  - location

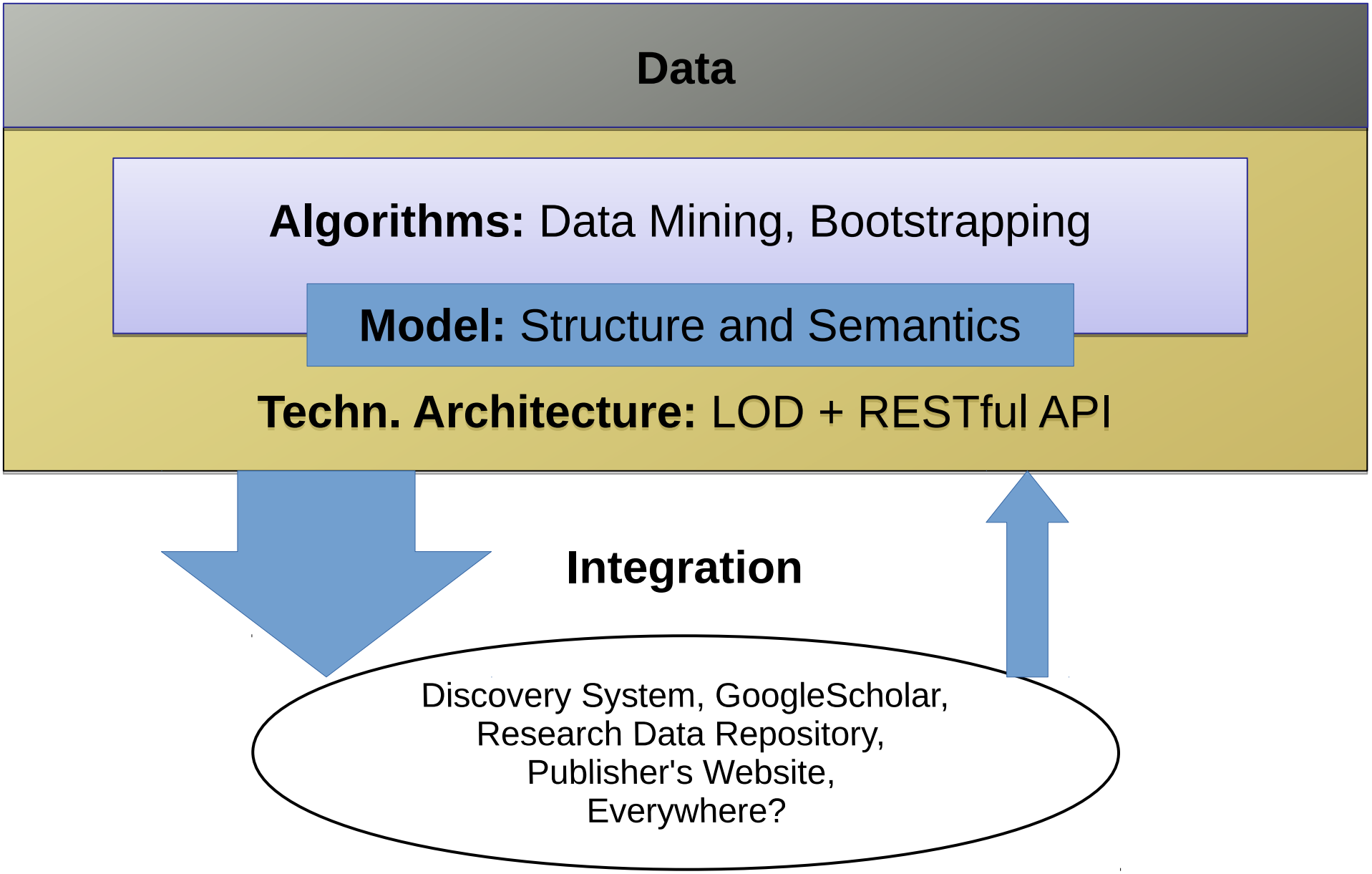


InFoLiS Project

Automating these processing steps,
i.e. automatically unraveling
hidden references (in running text) to research data
into structured data citations with URIs

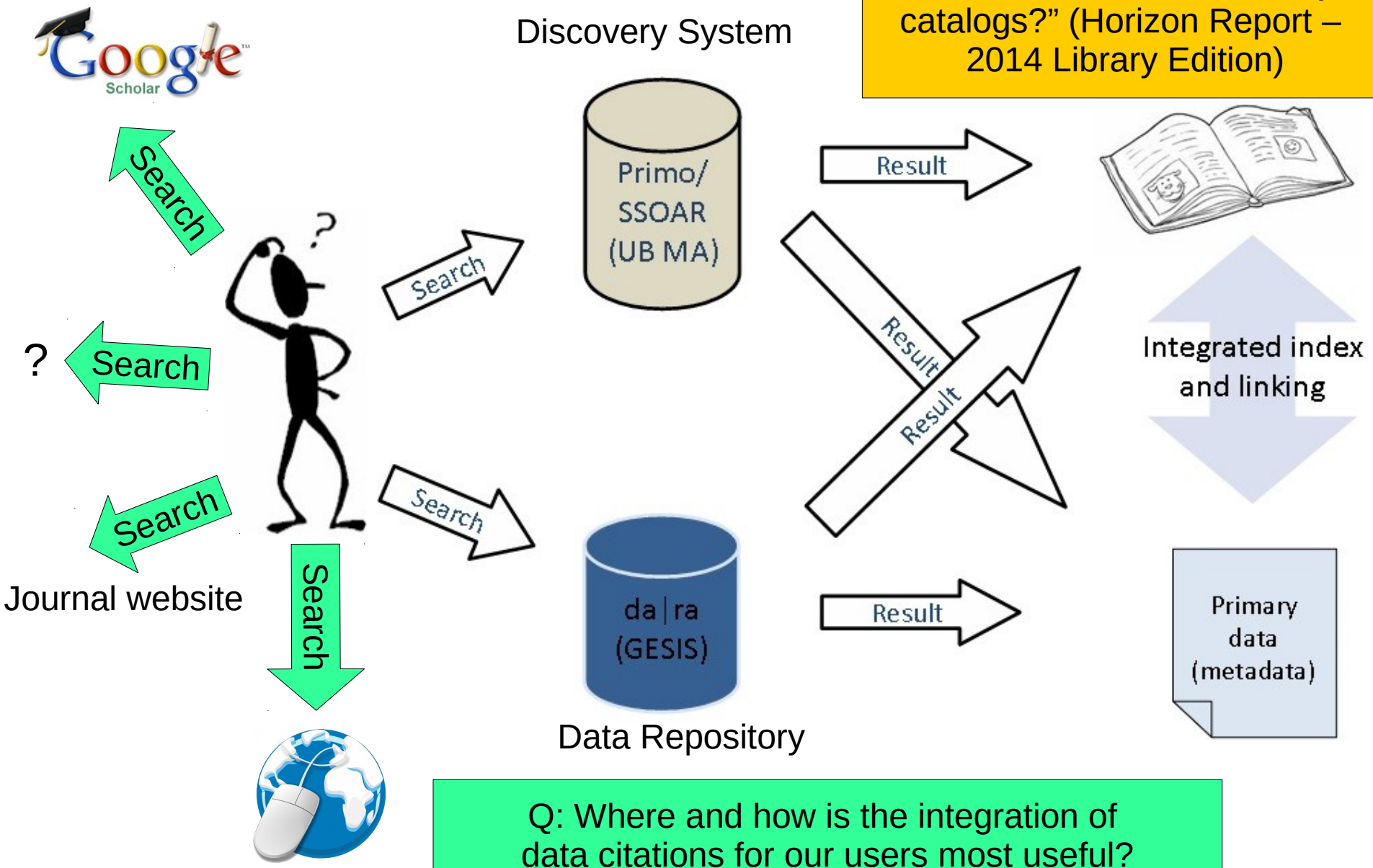
Flexible and long-term sustainable infrastructure

InFoLiS Project – more in depth



Integration

Q: “How to best incorporate data connections into library catalogs?” (Horizon Report – 2014 Library Edition)



Q: Where and how is the integration of data citations for our users most useful?

Different Agents want different data

#LD Linked Data Agent

text/turtle
application/rdf+xml
...

API Explorer

application/schema+json

Internal API

Text Extraction
Pattern Learning
Reference Extraction
Link Generation
File Storage

application/json

Public API

JSON-LD ↔ RDF
REST API
Simple HTTP API
Resource Storage

RD / OA Repository

OAI/PMH ?

RSS/Atom ?

Publisher

application/json

Bulk CLI Tool

application/json

Browser Plugin

application/ld+json

RDF Explorer

API Usability over Semantic Depth

Easy to maintain

Easy to consume

Possible to understand

RESTful(ish)

Protocol-independent

Serialization-independent

Easy to impement in code

JSON

Native Ordered Lists

High Performance

Deterministic structure

Main Operations in InFoLiS

Bootstrapping

Speed > Semantics

Text Extraction

Speed > Semantics

Pattern Application

Speed > Semantics

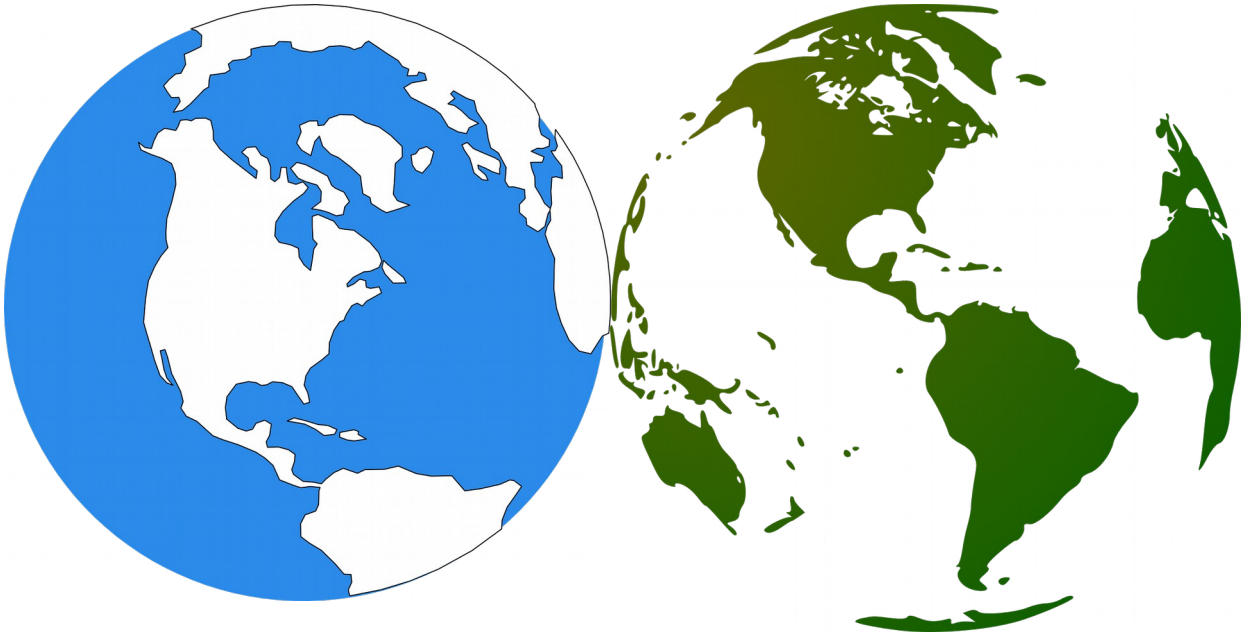
Dataset Resolution

Semantics > Speed

Deep modelling has its merit!

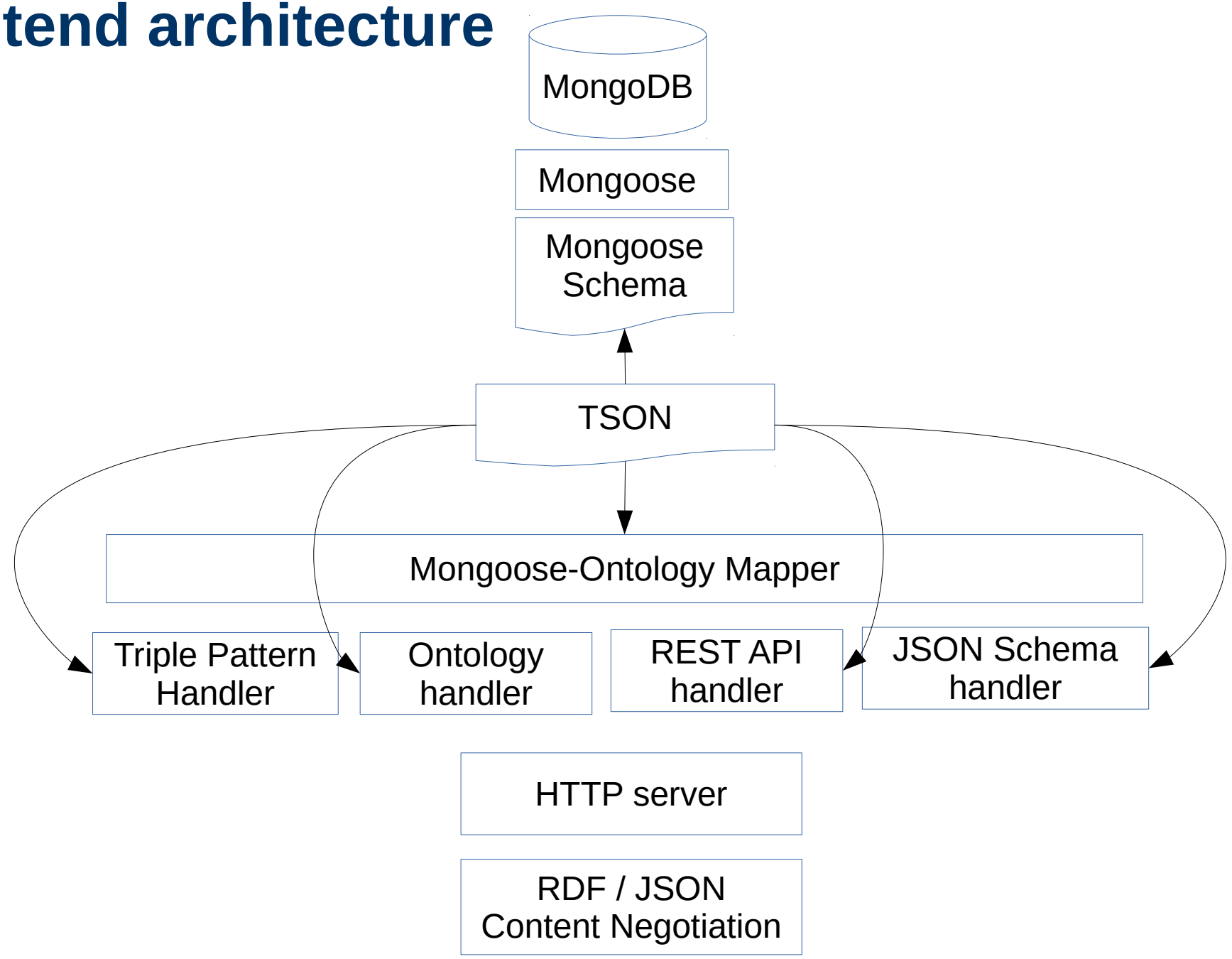
- Modelling Dataset granularity
  - Single issue of annual dataset?
  - Single panel of multi-faceted survey?
- Modelling Dataset reference vagueness
  - “As the results of **our study** indicate ...”
  - “According to **page 15** of the **DERP** panel …”
- Bibliometric Analyses
  - Spanning a graph of publications, datasets, people …
- Provenance Mining
  - Which patterns are found in different learn sets?
  - **Text A** sameAs **Text B** ⇨ **PDF A** textEquals **PDF B**

How to get the best out of both worlds?



Deep Modelling + KISS

Frontend architecture



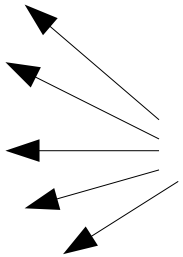
Extract from TSON-file

```
Execution
    @context
      dc:description "The concrete execution of an Algorithm."
      rdfs:subClassOf
          @id schema:Action
      dcterms:source
          @id <https://github.com/infolis/infoLink/blob/master/src/main/java/io/github/infolis/model/Execution.java>
```

RDF Class infolis:Execution

```
algorithm
    @context
        dc:description "Java class of the algorithm which is supposed to be executed within this execution."
        dcterms:source
          @id <https://github.com/infolis/infoLink/blob/master/src/main/java/io/github/infolis/model/Execution.jav
    required true
    index: true
    type: String
    enum: [
        'io.github.infolis.algorithm.ApplyPatternAndResolve'
        '[...]'
        'io.github.infolis.algorithm.Resolver'
    ]
```

RDF Property infolis:algorithm



Database schema

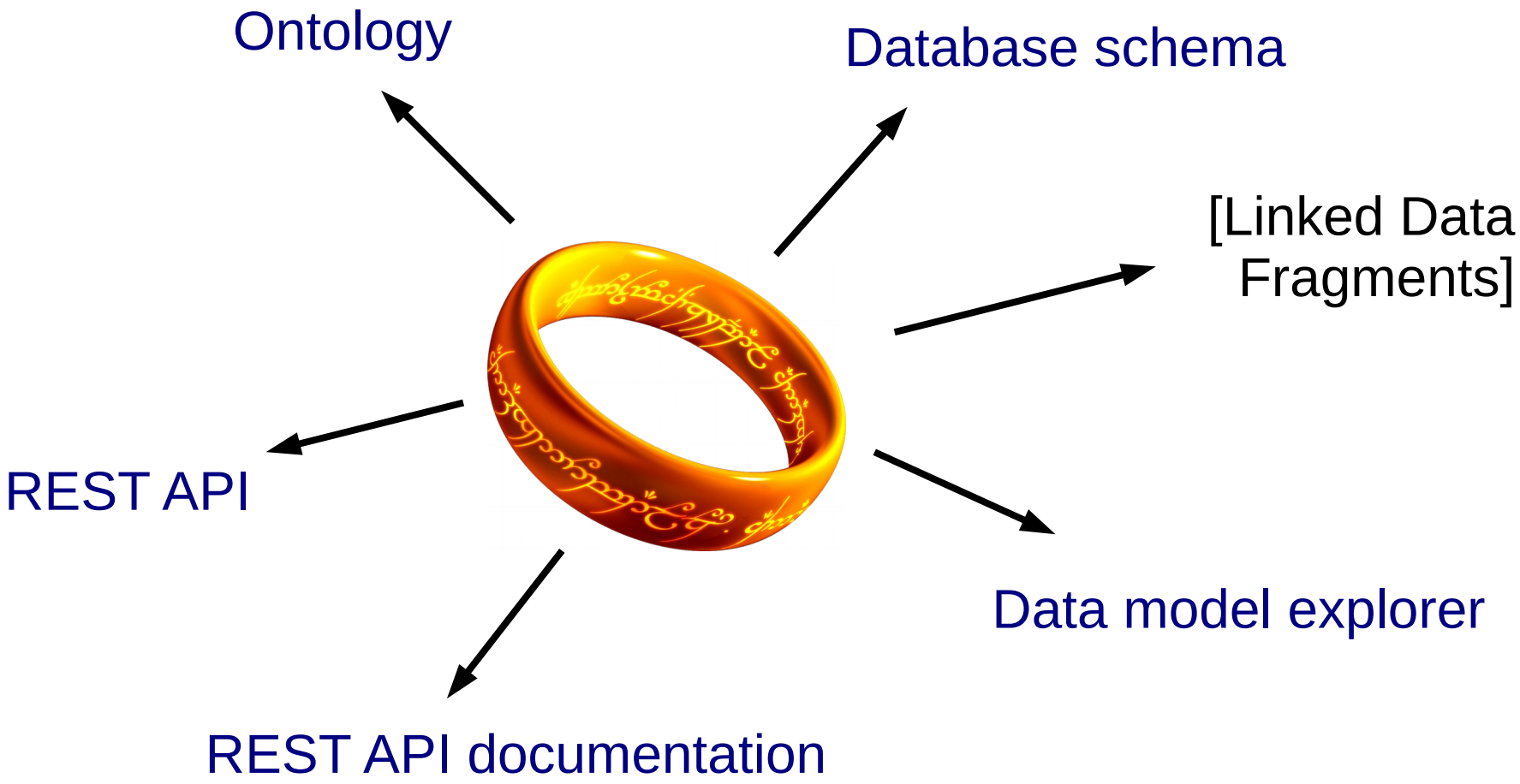
```
log
    @context
        dc:description "Log messages of this execution."
    type: ArrayOfStrings
    hideFromSwagger: true
```

RDF Property infolis:log

for Presentation

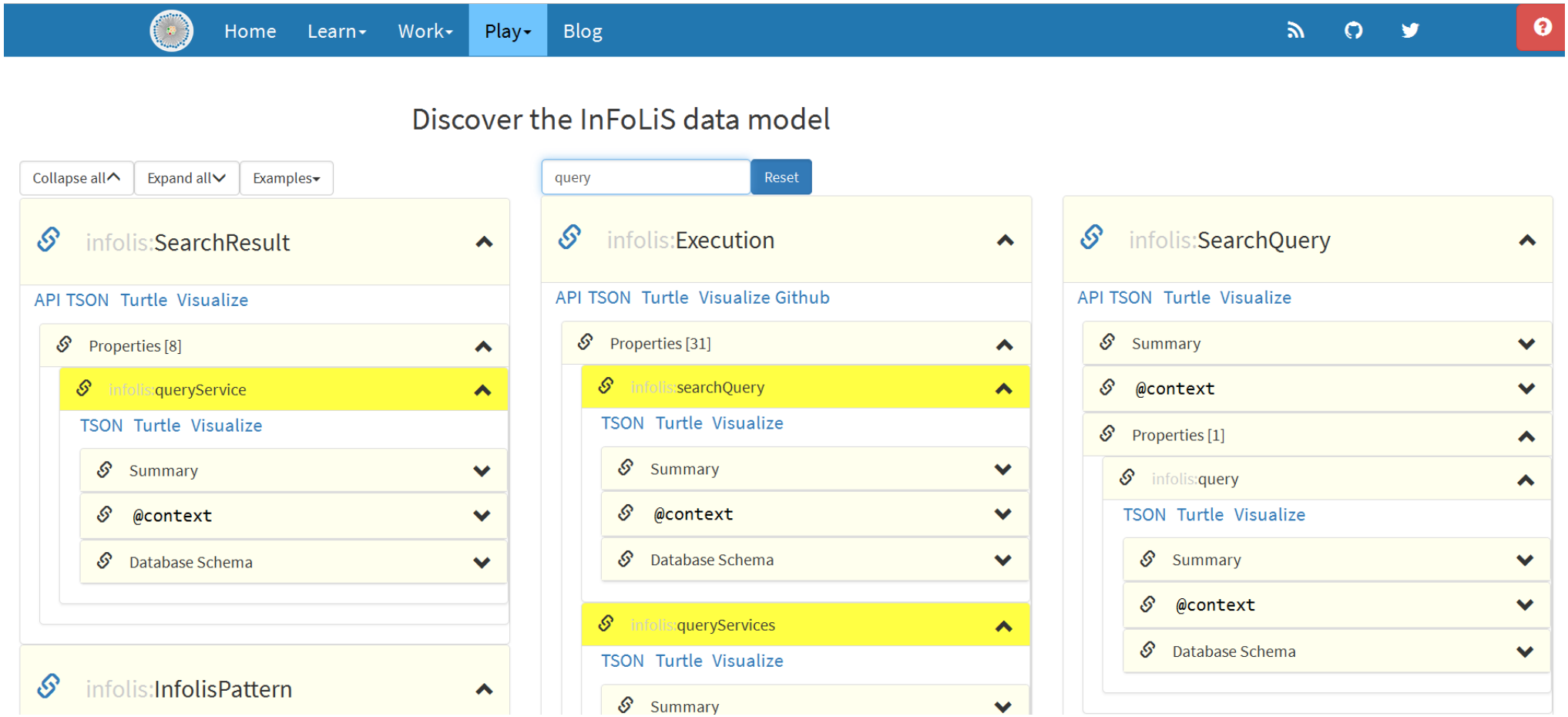
TSON = Turtleson = json-ld + json-schema in Turtle + CoffeeScript

One schema to rule them all



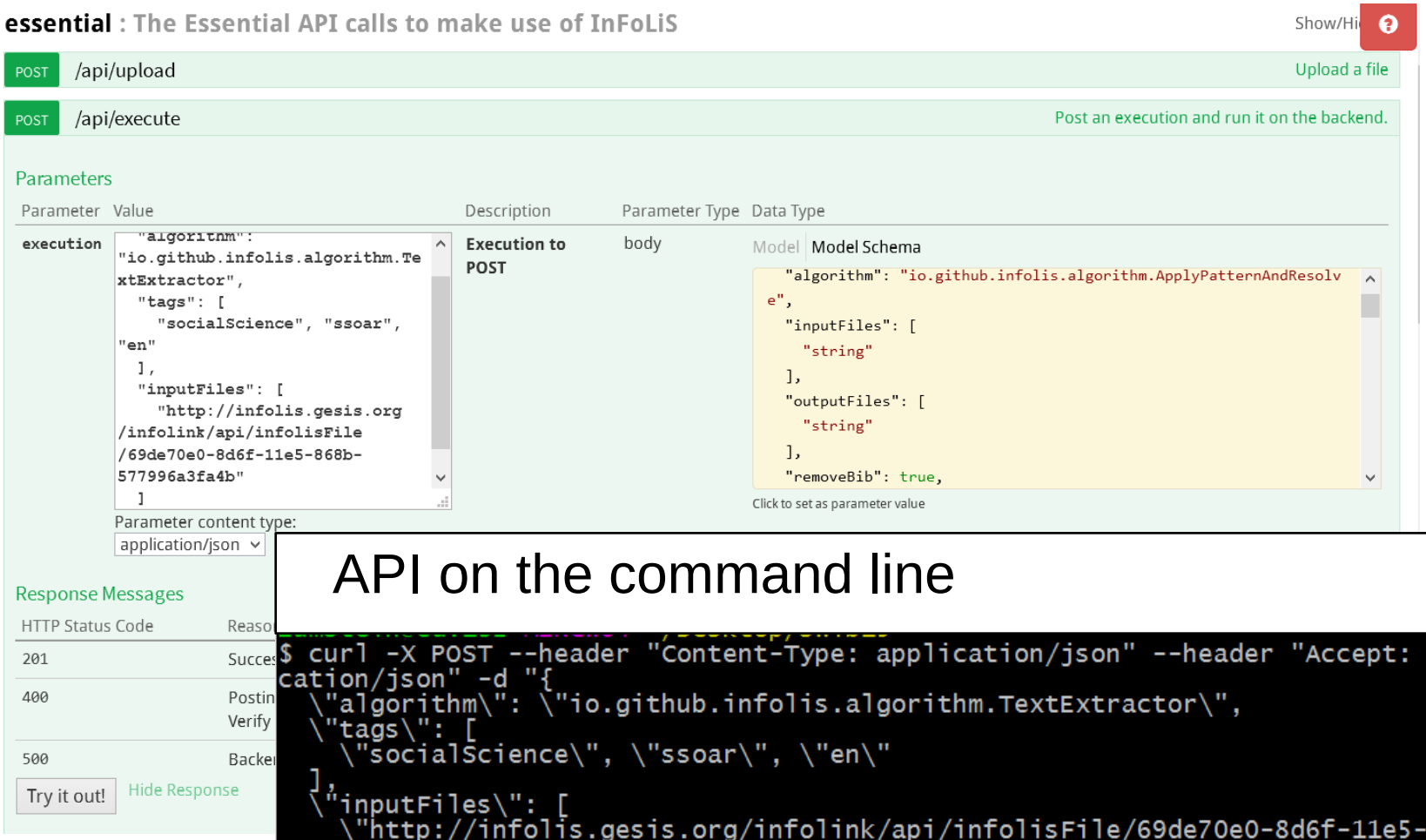
Demonstration

Discover the InFoLiS data model



Demonstration

API: graphical interface



API on the command line

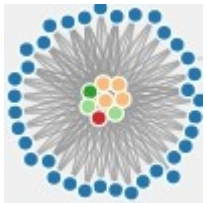
```
$ curl -X POST --header "Content-Type: application/json" --header "Accept: appli
cation/json" -d "{
  \"algorithm\": \"io.github.infolis.algorithm.TextExtractor\",
  \"tags\": [
    \"socialScience\", \"ssoar\", \"en\"
  ],
  \"inputFiles\": [
    \"http://infolis.gesis.org/infolink/api/infolisFile/69de70e0-8d6f-11e5-868b-
577996a3fa4b\"
  ]
}" "http://infolis.gesis.org/infolink/api/execute"
```

Thank you for your attention!

Questions?

Keep in touch:
{baierer, zumstein}@bib.uni-mannheim.de
Twitter: @infolis_project

Homepage:
(Info, API, Tools, …
...it's in rapid development)
http://infolis.github.io/



All InFoLiS Software is Open Source:
http://github.com/infolis