



# Managing Discipline-Specific Metadata Within an Integrated Research Data Management System

ICEIS 2019:  
03. - 05.05.2019

Marius Politze, Sarah Bensberg, Matthias S. Müller  
RWTH Aachen University

# Outline

---

- The Framework:  
Research Data Management at RWTH Aachen University
- Requirements Engineering  
Supported Business Processes and Requirements
- Putting Ontologies into Practice  
Implementation & Prototype Application
- Results  
First Evaluation and Conclusion

# Timeline Coordinated Research Data Management at RWTH Aachen University

since 2015:

Project introducing research data management (RDM),  
Cooperation of University Library, IT Center and  
Department Research & Career

2016: coordinated workshops and

2016: Consulting for researchers and research groups

2017: simpleArchive & Matadata Manager:

Archival and description of research data

2018: Git ~2.5 TB & ObjectStore (current 500 TB, aim 5 PB in late 2019):

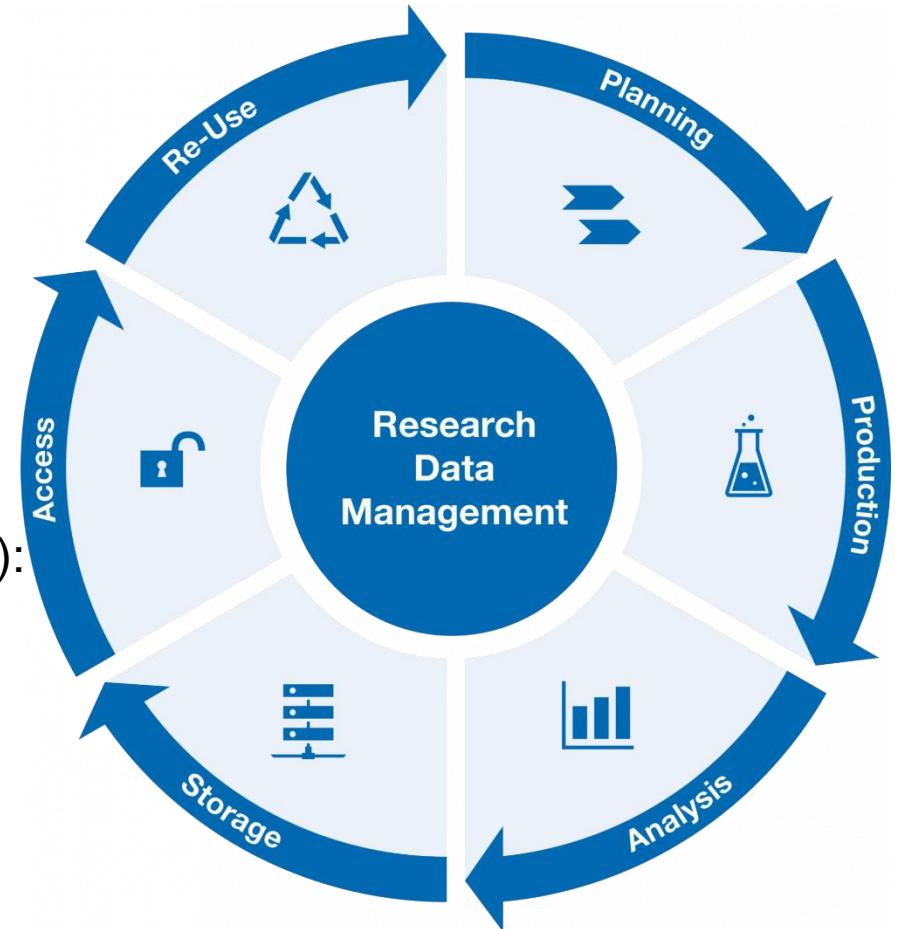
Storage infrastructures for Data Management

2016: Combination of private and public cloud services

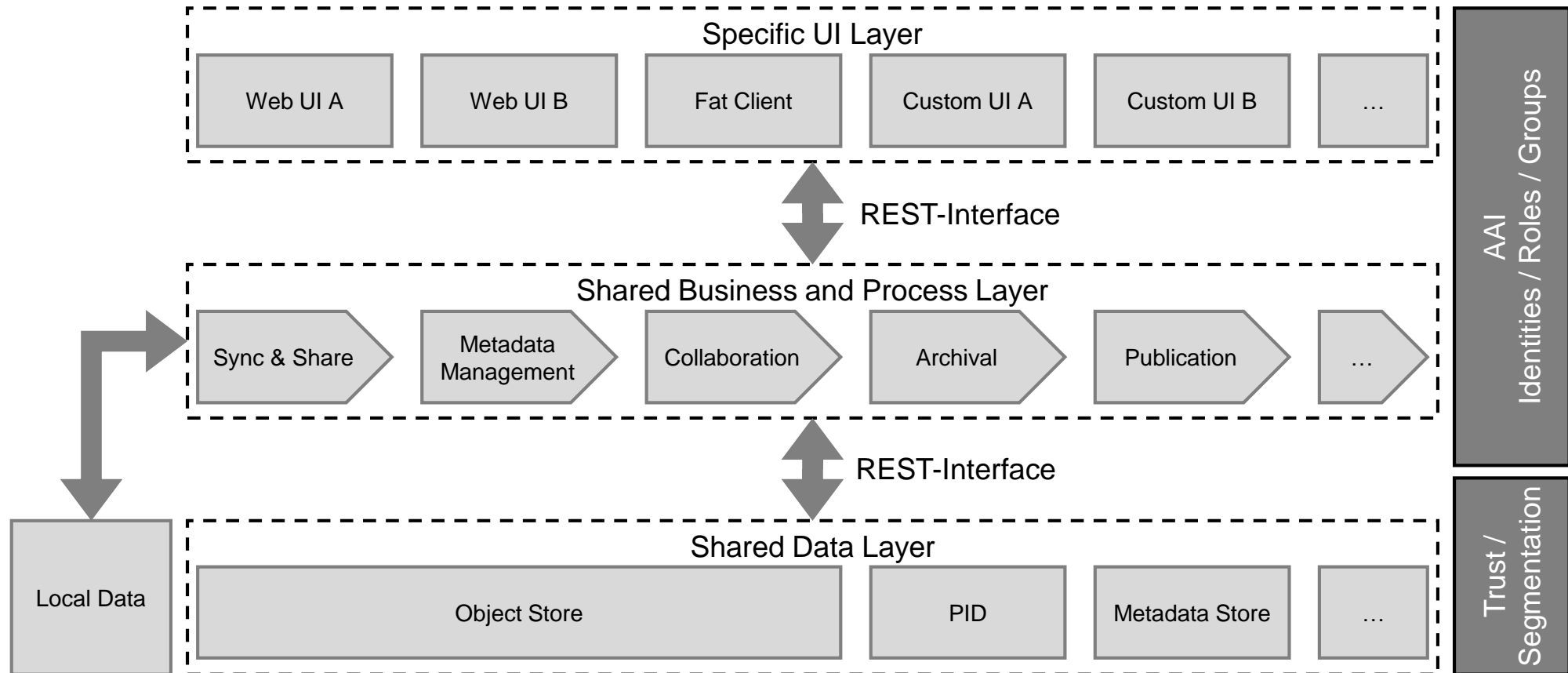
2017: Cooperation with FZ Jülich

2017: Cooperation with TU Darmstadt

2018: Service Provider within national NFDI4Ing Initiative



# Integrated Research Data Management System



## Example “Research Process”

Scientific Instrument



Attached Computer



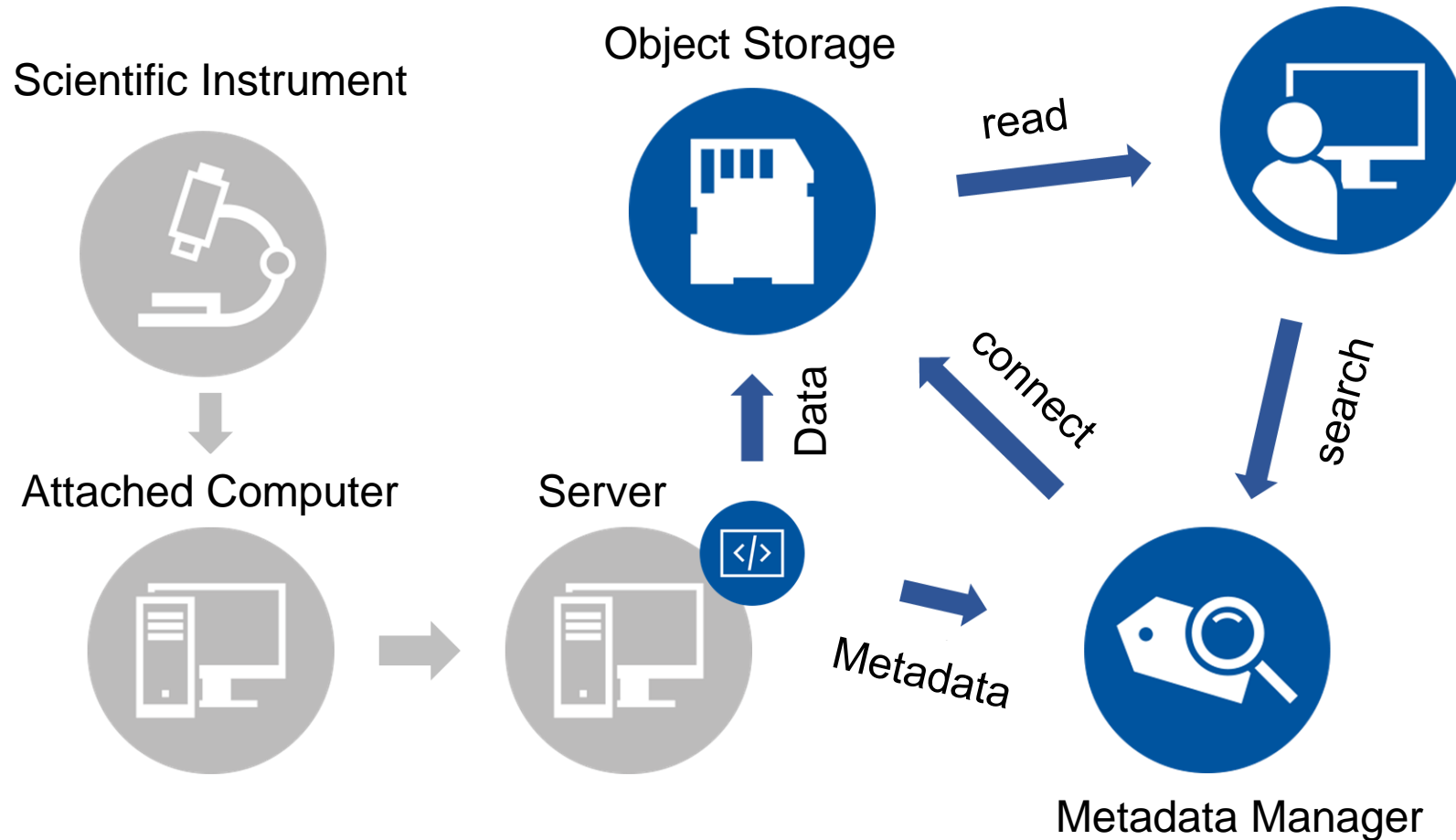
Server



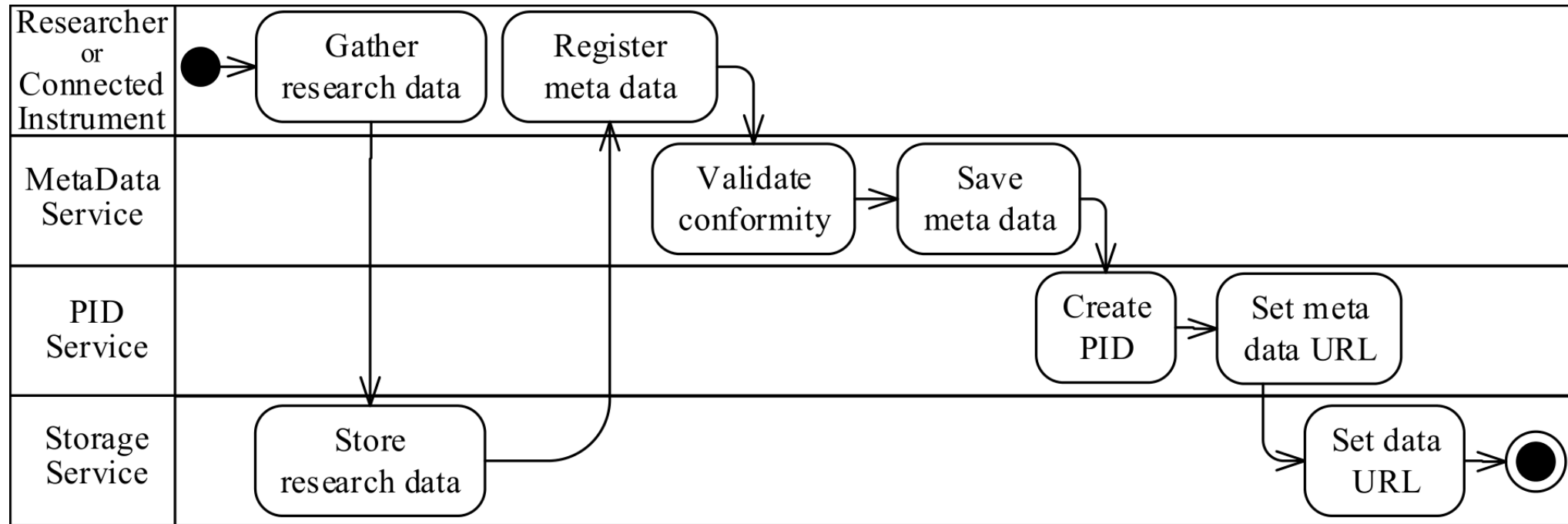
Windows Share



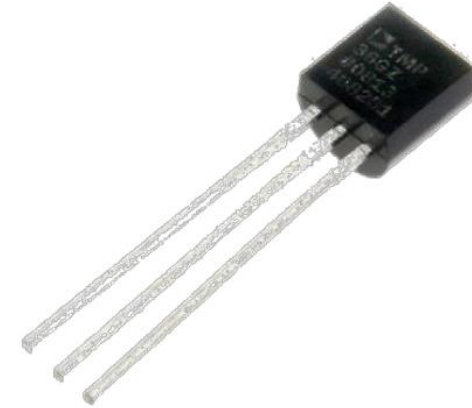
# Digitally Enhanced “Research Process”



# Formalized “Research Process”



## Industry 4.0 Example Semantic Representation of Sensor Data



```
myd:m123245      rdf:type          i40:SensorMeasurement .
myd:m123245      rdf:hasValue     "27.9"^^i40:DegreeCelsius .
myd:m123245      i40:hasMeasureTime "2016-03-24T12:38:54:12Z"^^xsd:DateTime .
myd:m123245      i40:fromSensor   myd:Sensor123 .
...
# ^ subject      ^ predicate      ^ object
```





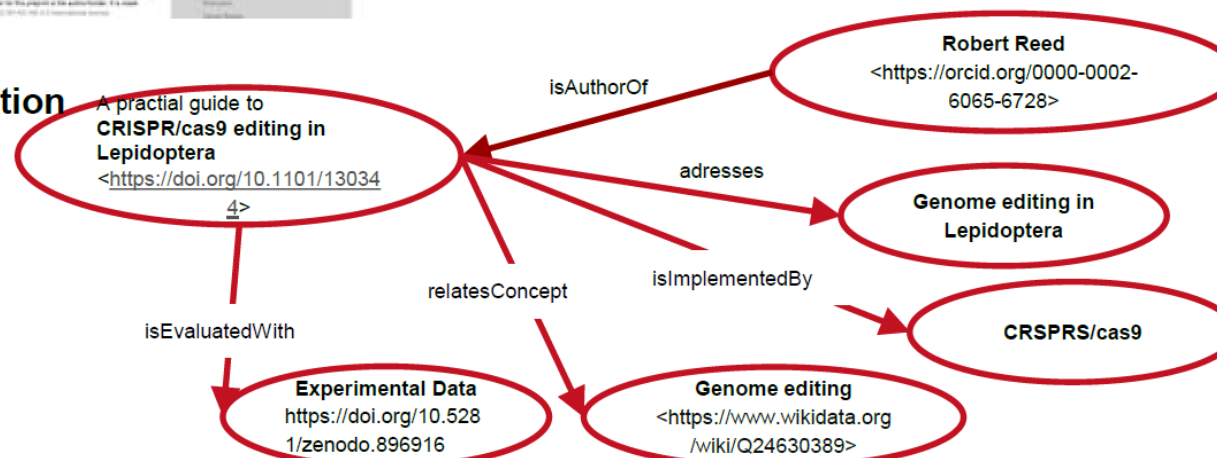
## 1. Original Publication



## 2. Graph Curation Form

Author	Robert Reed
Research Problem	Genome editing in Lepidoptera
Methods	CRISPR/cas9
related Concepts	Lepidoptera; Genome editing; CRSIPR
Experimental Data	<a href="https://doi.org/10.5281/zenodo.896916">https://doi.org/10.5281/zenodo.896916</a>

## 3. Graph representation

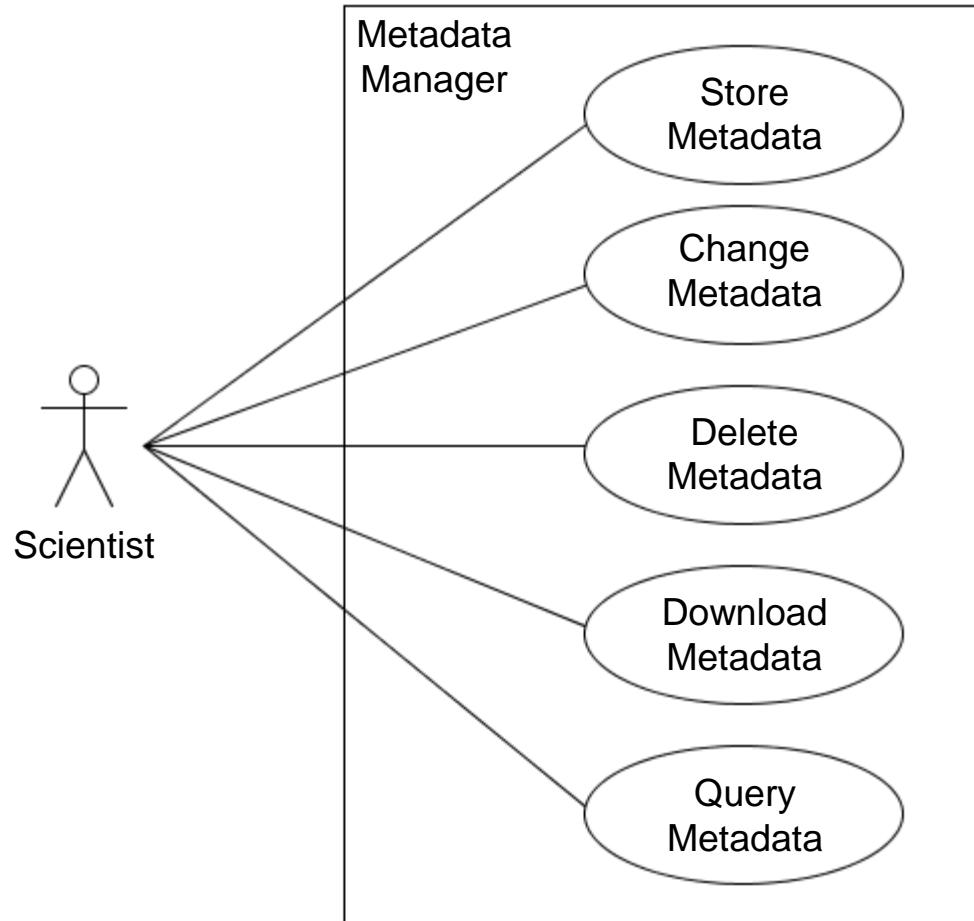


Seite 57

Slide by Sören Auer: Towards an Open Research Knowledge Graph, <https://www.slideshare.net/soeren1611/towards-an-open-research-knowledge-graph>

- 
- Problem statement:
    - Management of heterogeneous research data is becoming more important on institutional level
    - Description of data within a knowledge graph seems like a feasible solution
    - Authoring semantic information is not feasible for most researchers
    - Information needs to be recorded when data is produced otherwise knowledge may be lost
  
  - Goal
    - Build an application that allows researchers to ingest their data into the research knowledge graph
    - Allow flexible application profiles to create a single knowledge graph for the whole university

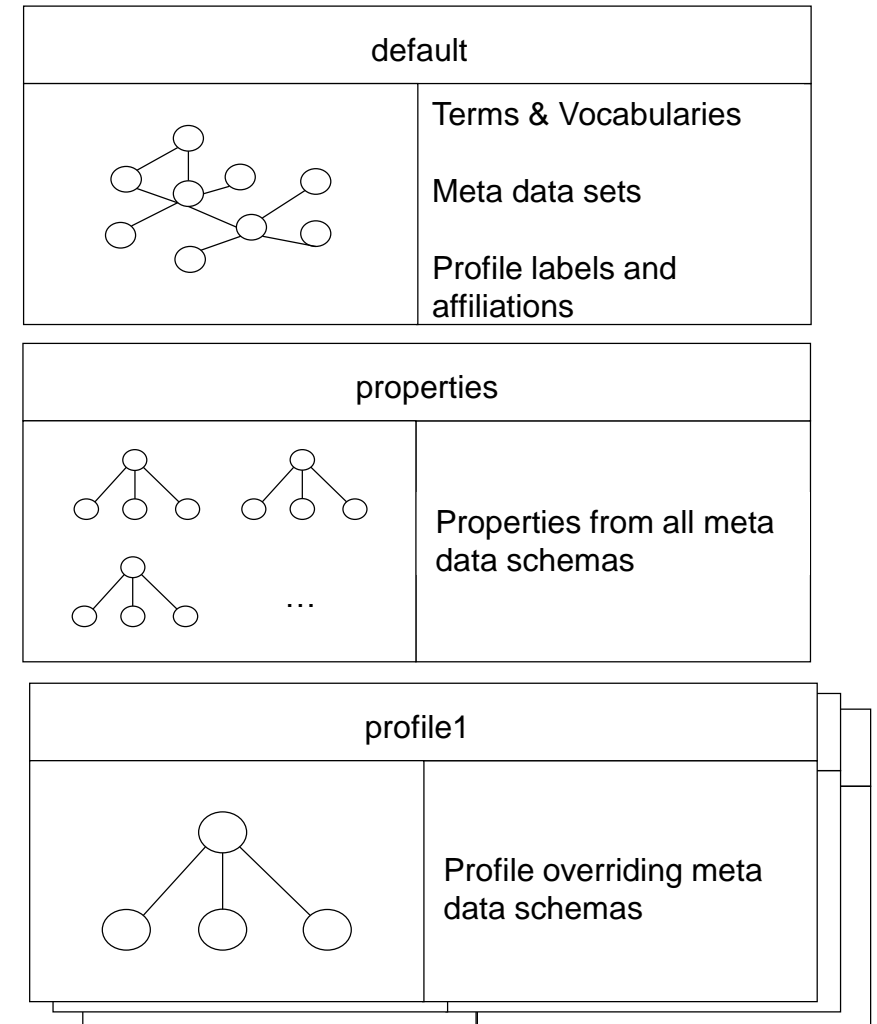
# Considered Use Cases



- Functional Requirements
  - F1: retrieve available profiles
  - F2: save new meta data set
  - F3: metadata visibility
  - F4: show all own meta data sets
  - F5: edit stored meta data set
  - F6: query stored meta data
  - F7: suggestions for vocabulary ranges
  - F8: render meta data form based on a profile
- Non-Functional Requirements
  - N1: internationalization
  - N2: compatibility with DCAT
  - N3: Dublin Core as cross discipline standard

# Semantic Data Model: A Path to the Scientific Knowledge Graph

- Using Virtuoso quad-store (Graph, Subject, Predicate, Object)
  - Represent data artifacts by PID
  - Record meta data as triples having the PID as a subject
  - Create a high level “Research Knowledge Graph”
- Separate management of “user generated” and “provided” data
  - Multiple disconnected graphs with different purposes
  - Default:
    - All Terms, Vocabularies, Meta Data, ...
    - Default target for storage and search
  - Properties:
    - Includes all Properties from all application profiles
  - Profile1 ... ProfileN
    - Application profile specific overrides



# Prototype Application and Webservice – Rendering Form Based on Application Profile

## Chemical Experiment

Lab Technician\*

Description\*

Subject Area

Solute

Solvent

RESET

SAVE

RDF Range	HTML5 Type
rdfs:Literal	text
xml:dateTime	date
md:metadataVisibility	radio
None	Text
Other	select

```
dc:creator
· · a owl:AnnotationProperty ;
· · md:calculatedValue "{ME}";
· · md:position 1;
· · rdfs:label "Lab Technician"@en ;
· · rdfs:range rdfs:Literal .
```

```
dc:title
· · a owl:AnnotationProperty ;
· · md:position 2;
· · rdfs:label "Description"@en .
```

```
dc:subject
· · a owl:AnnotationProperty ;
· · rdfs:range <http://udcdata.info/029653> ;
· · md:position 3;
· · rdfs:label "Subject Area"@en .
```

```
:solute
· · rdfs:subPropertyOf csmd:samplertype_molecularFormula ;
· · a owl:AnnotationProperty ;
· · md:position 4;
· · rdfs:label "Solute"@en .
```

```
:solvent
· · rdfs:subPropertyOf csmd:samplertype_molecularFormula ;
· · a owl:AnnotationProperty ;
· · md:position 5;
· · rdfs:label "Solvent"@en .
```

# Prototype Application and Webservice – Storing Meta Data and Translate to Linked Data

## Chemical Experiment

Lab Technician\*

Description\*

Subject Area

Solute

Solvent

RESET

SAVE

```
POST /metadata/profileN/20.11102/1d53500-75f7-475e-9128-825da4d90664
{
  ····"Description": "Solving salt in water",
  ····"Lab Technician": "John Doe",
  ····"Subject Area": "http://udcdata.info/030042",
  ····"Solute": "NaCl",
  ····"Solvent": "H2O"
}
```

```
SELECT ?s WHERE {
  ····GRAPH <profileN> {
  ······?s rdf:label ?label .
  ······FILTER REGEX (STR(?label), "Value", "i") .
  ····}
}
```

```
http://hdl.handle.net/20.11102/1d53500-75f7-475e-9128-825da4d90664
dc:title "Solving salt in water"@en
dc:creator "Solving salt in water"@en
dc:subject http://udcdata.info/030042
profileN:solute "NaCl"@en
profileN:solvent "H2O"@en
```

# Evaluation

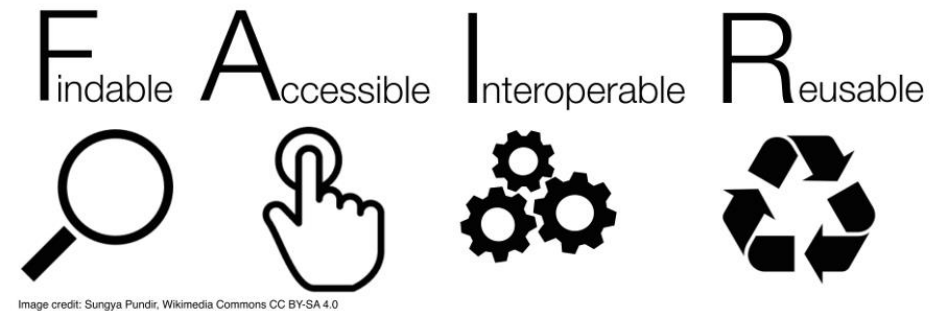
- Automated Software Tests
- Mapping with DCAT
  - Catalog
    - Public
    - Internal for each Affiliation
    - Private for each Researcher
  - CatalogRecord
    - Meta data from DC is compatible
    - Partially automatic acquisition of values (affiliation, user)
  - Dataset
    - (Meta) data set recorded by the software
  - Distribution
    - PID and additional fields like URL of the data artifact

Area	Coverage	
SPARQL	95.92%	47/49
RDFWrapperSchema	100.00%	68/68
RDFWrapperMetadata	91.32%	442/484
API	85.50%	171/200
MetadataSchema	94.44%	34/36
Metadata	92.65%	189/204
<b>Total</b>	<b>91.35%</b>	<b>951/1041</b>

# Conclusion

---

- PID references to data artifacts help identification across participating decentralized systems
- Minimal compliance of application profiles fosters FAIR guiding principles
- Application was launched with pilot users and is now introduces to other use cases in chemistry, electrical engineering, material science and combustion engineering
- API is understandable and can be operated by researchers themselves





**Thank you for your attention**

**Vielen Dank für Ihre Aufmerksamkeit**