Universität Münster

Mathematics Research Data Initiative - MaRDI4NFDI for PUNCH4NFDI

Frank Wübbeling (University of Münster)

living.knowledge

Universität Münster

Who we are

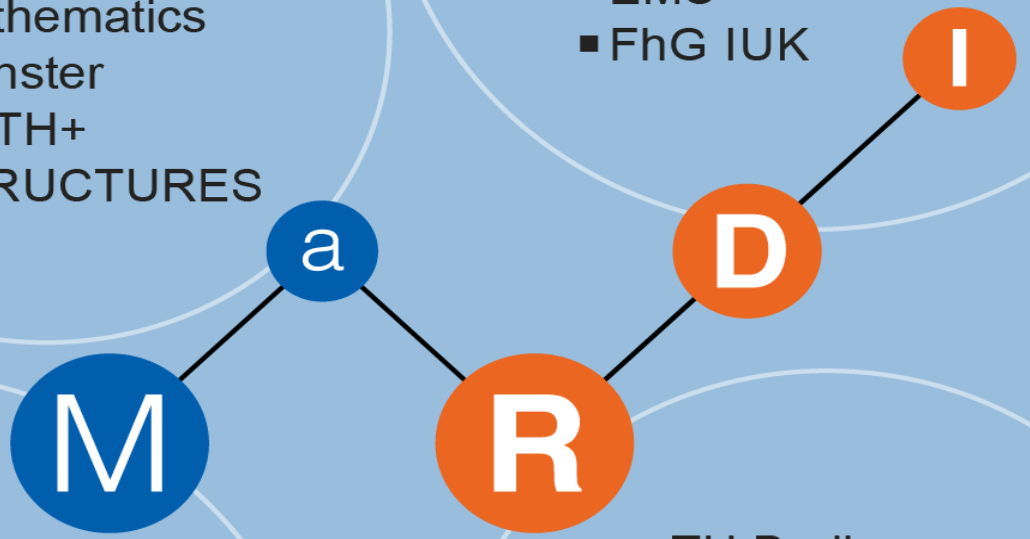
Mathematics
Cross-Disciplinary
Infrastructure
Providers

Clusters of Excellence

- SimTech
- Mathematics Münster
- MATH+
- STRUCTURES

Scientific Associations and Societies

- DMV
- GAMM
- GOR
- EMS
- FhG IUK



IMAGINARY
open mathematics

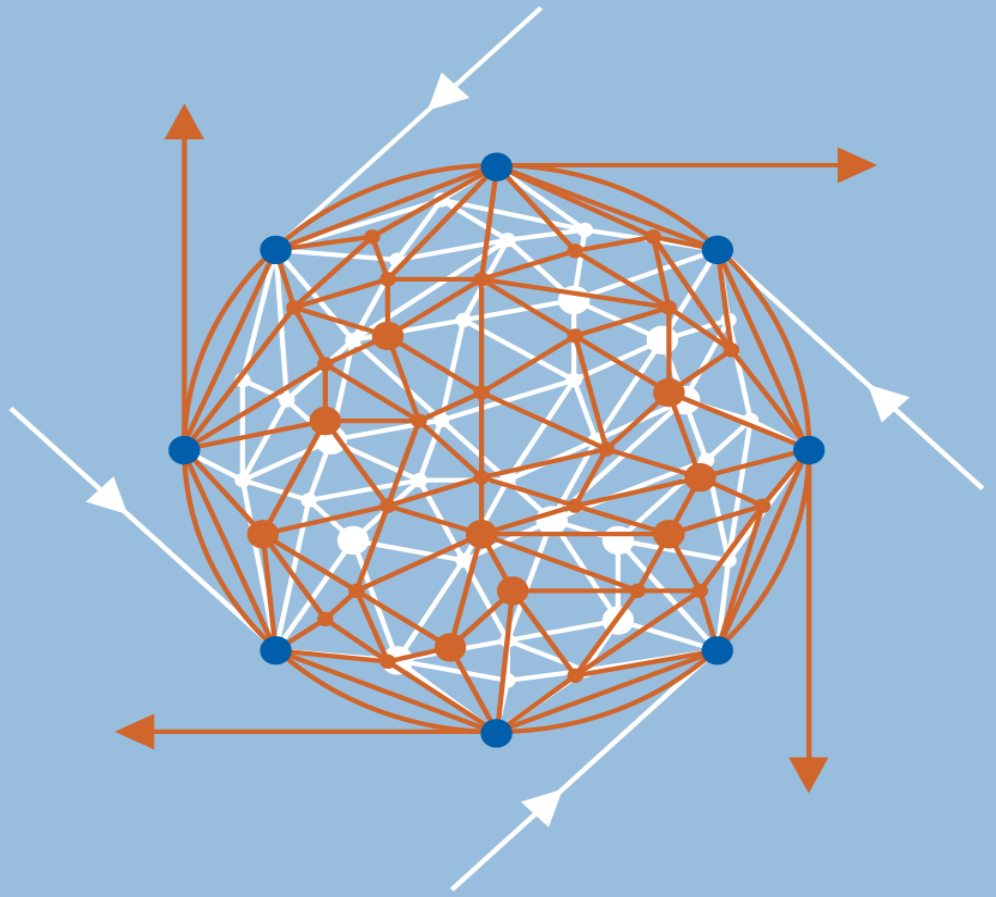
- **WIAS**
- ZIB
- MPI DCTS Magdeburg
- FIZ Karlsruhe
- MPI MIS Leipzig
- Fraunhofer ITWM
- MFO

- TU Berlin
- FU Berlin
- U Stuttgart
- LMU München
- TU München
- WWU Münster
- TU Kaiserslautern

Leibniz Network
Mathematical
Modeling and
Simulation

Computer Algebra

Inter-disciplinary



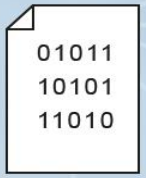
Scientific Computing

Statistics and Machine Learning

Math

$$a(u, v) = b(u)$$
$$a(u, u) \geq m\|u\|^2$$

Data

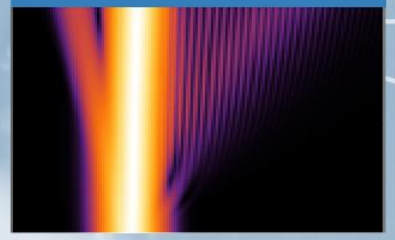


Models

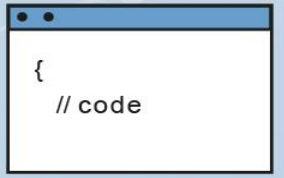
$$\partial_t u_1 = \nabla \cdot j_1(u_1, \nabla u_1$$
$$\partial_t u_2 = \nabla \cdot j_2(u_2, \nabla u_2$$

Documents

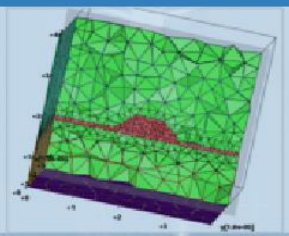
Visualisations

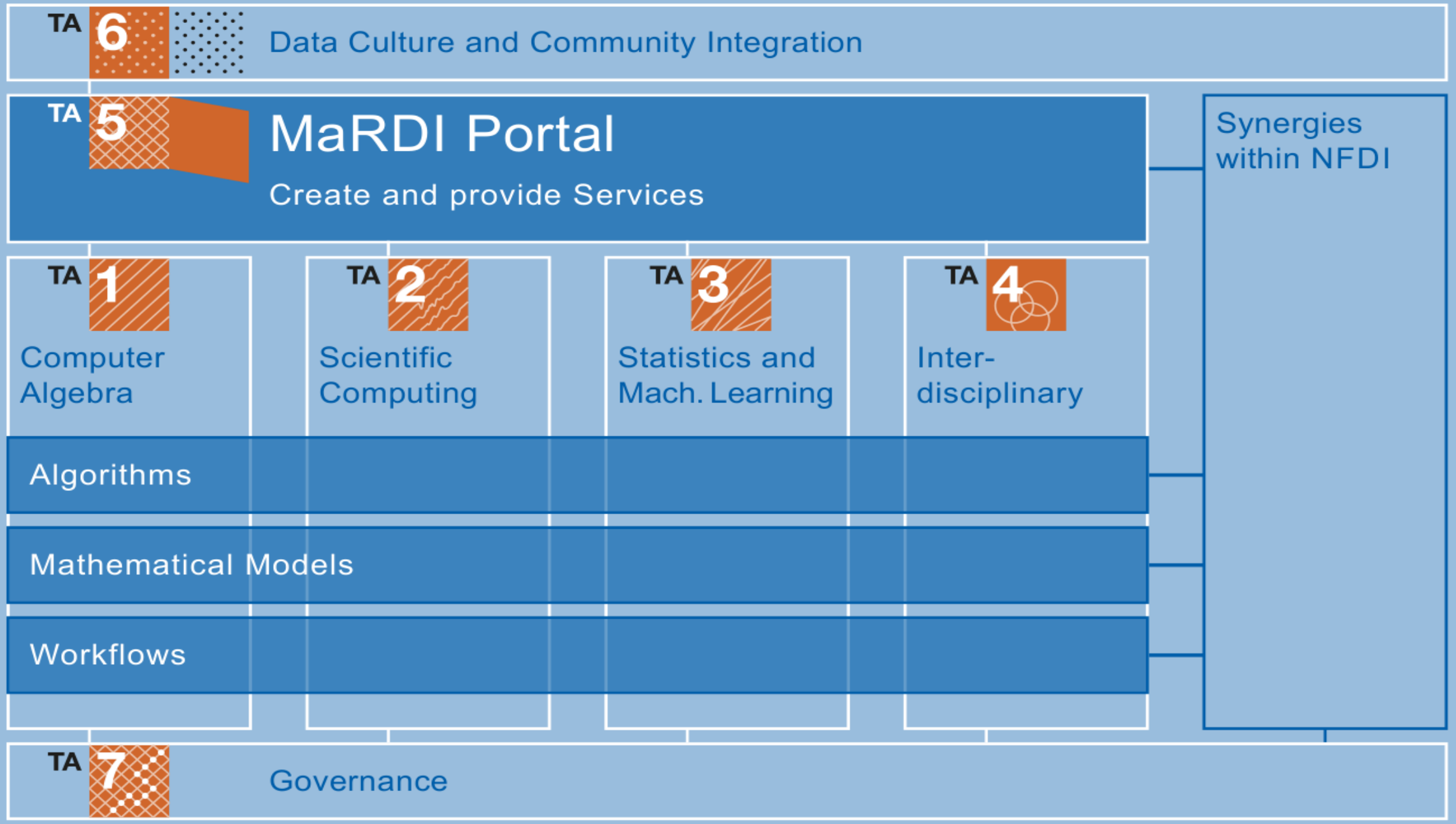


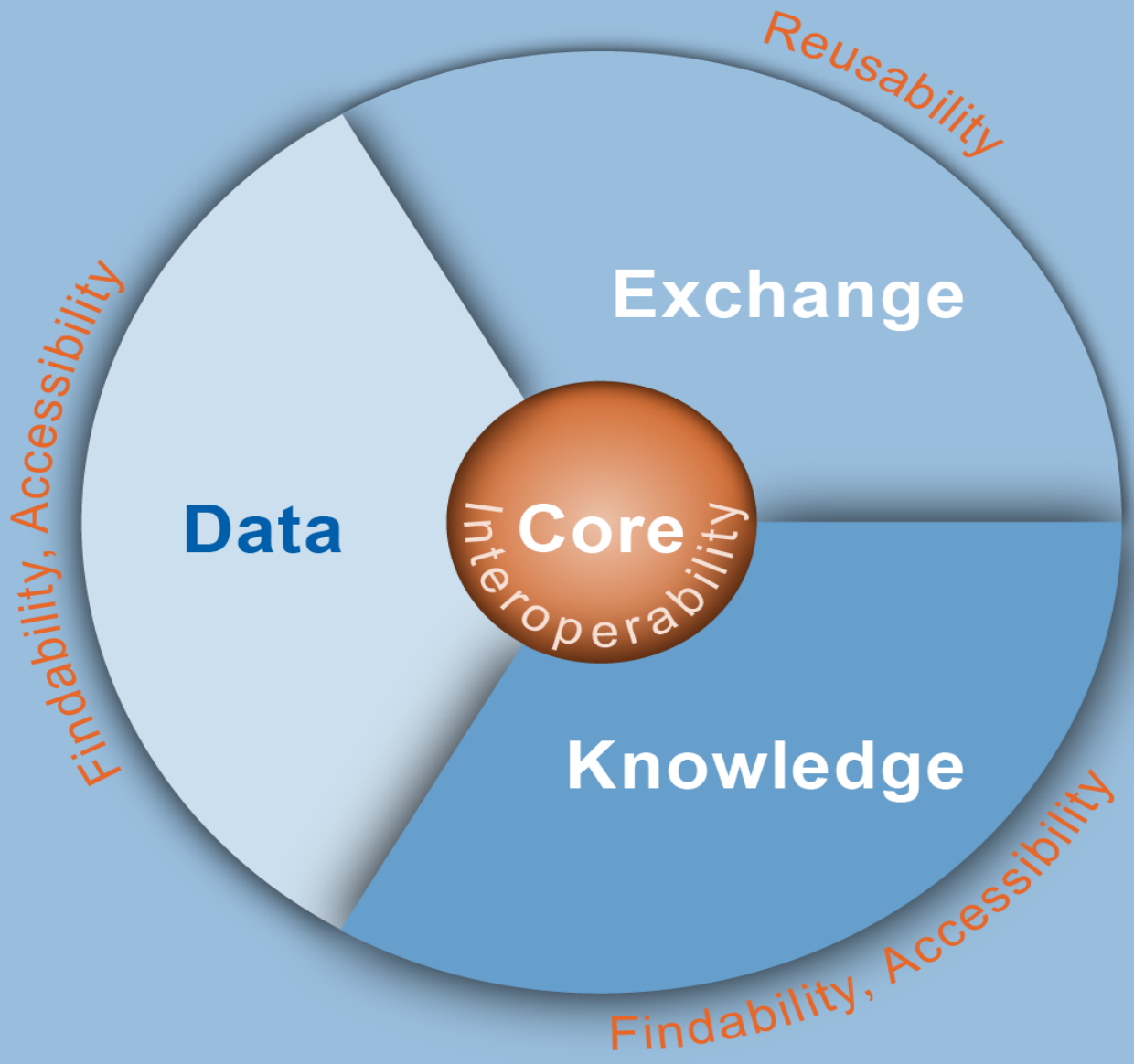
Software



3D Objects




TA 6
Data Culture and Community Integration
TA 5
MaRDI Portal
Create and provide Services
Synergies within NFDI
TA 1
Computer Algebra
TA 2
Scientific Computing
TA 3
Statistics and Mach. Learning
TA 4
Inter-disciplinary
Algorithms
Mathematical Models
Workflows
TA 7
Governance



Example: Linear Solver $Ax=b$

X1: Core

- Data structures for matrices and vectors
- Representation formats
- Exchange formats
- Application programming interfaces (APIs)

X2: Data

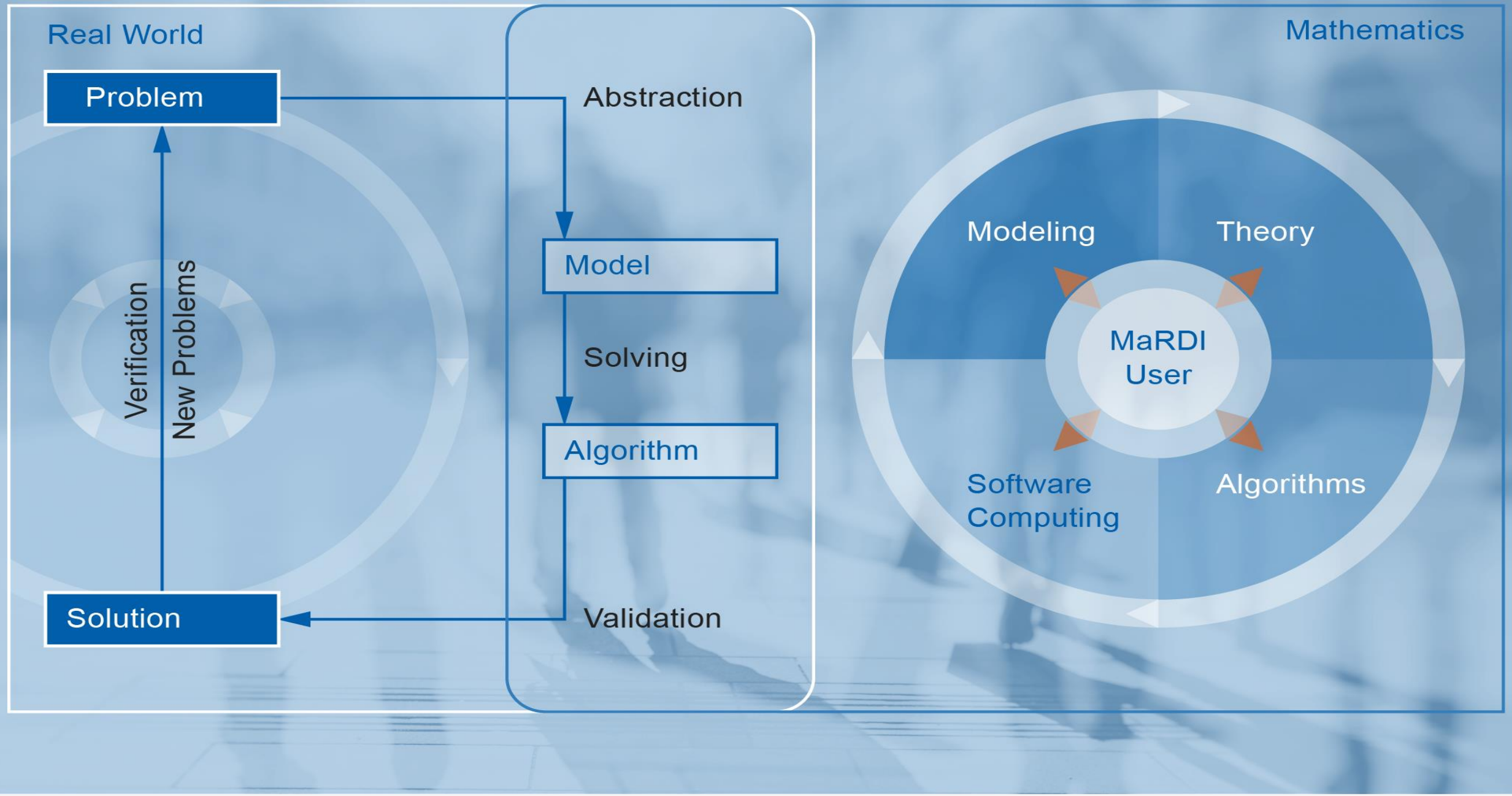
- Test cases (matrices, solutions)
- Matrix properties (meta-data)
- Persistent identifiers (PIDs)

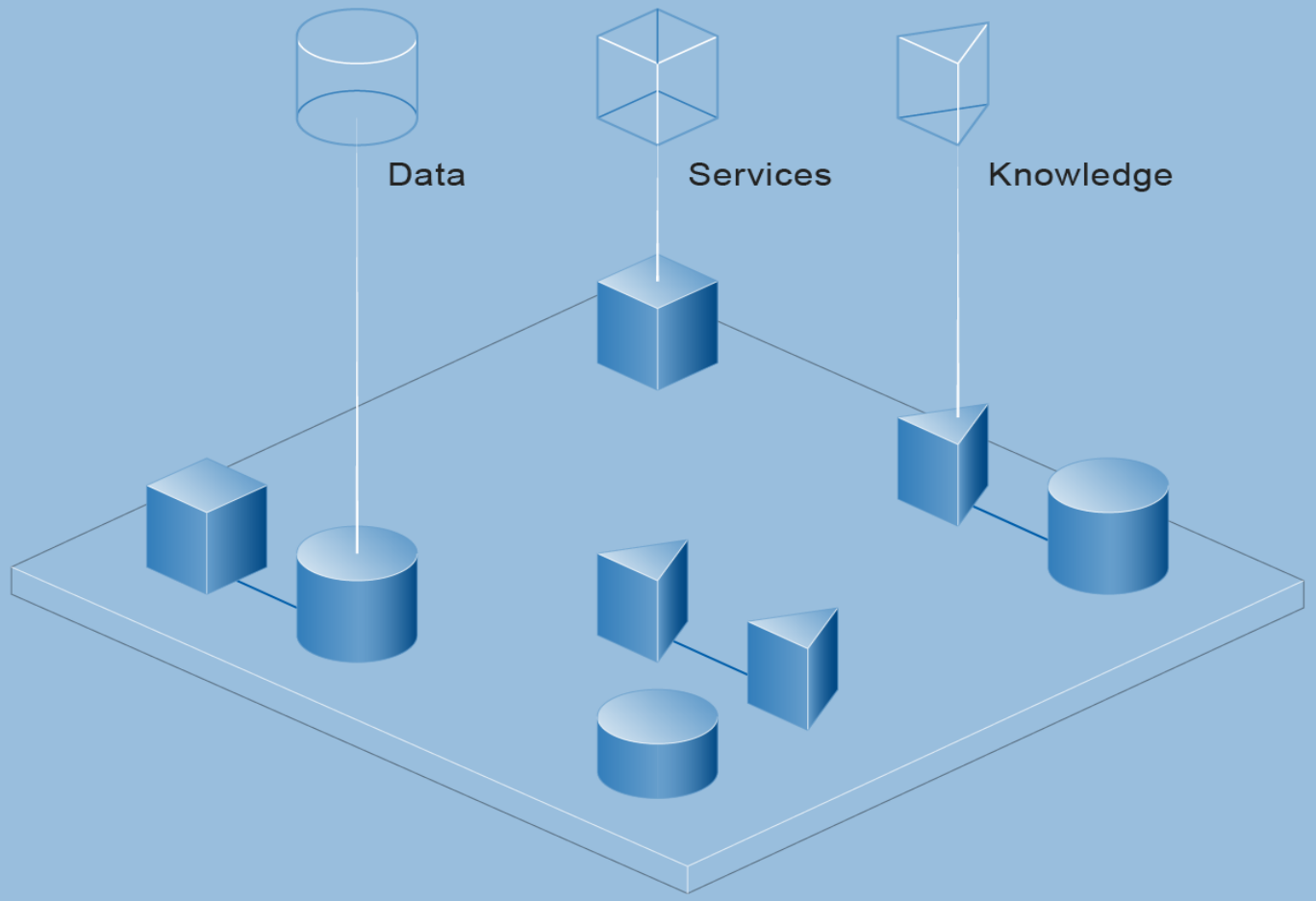
X3: Exchange

- Benchmark framework
- Pre-defined software environments
- Workflows, continuous-benchmarking

X4: Knowledge

- Ontology of linear problems and solvers
- Link to algorithms, publications software and test-data


Real World
Mathematics
Problem
Abstraction
Model
Solving
Algorithm
Verification
New Problems
Solution
Validation
Modeling
Theory
MaRDI
User
Software
Computing
Algorithms



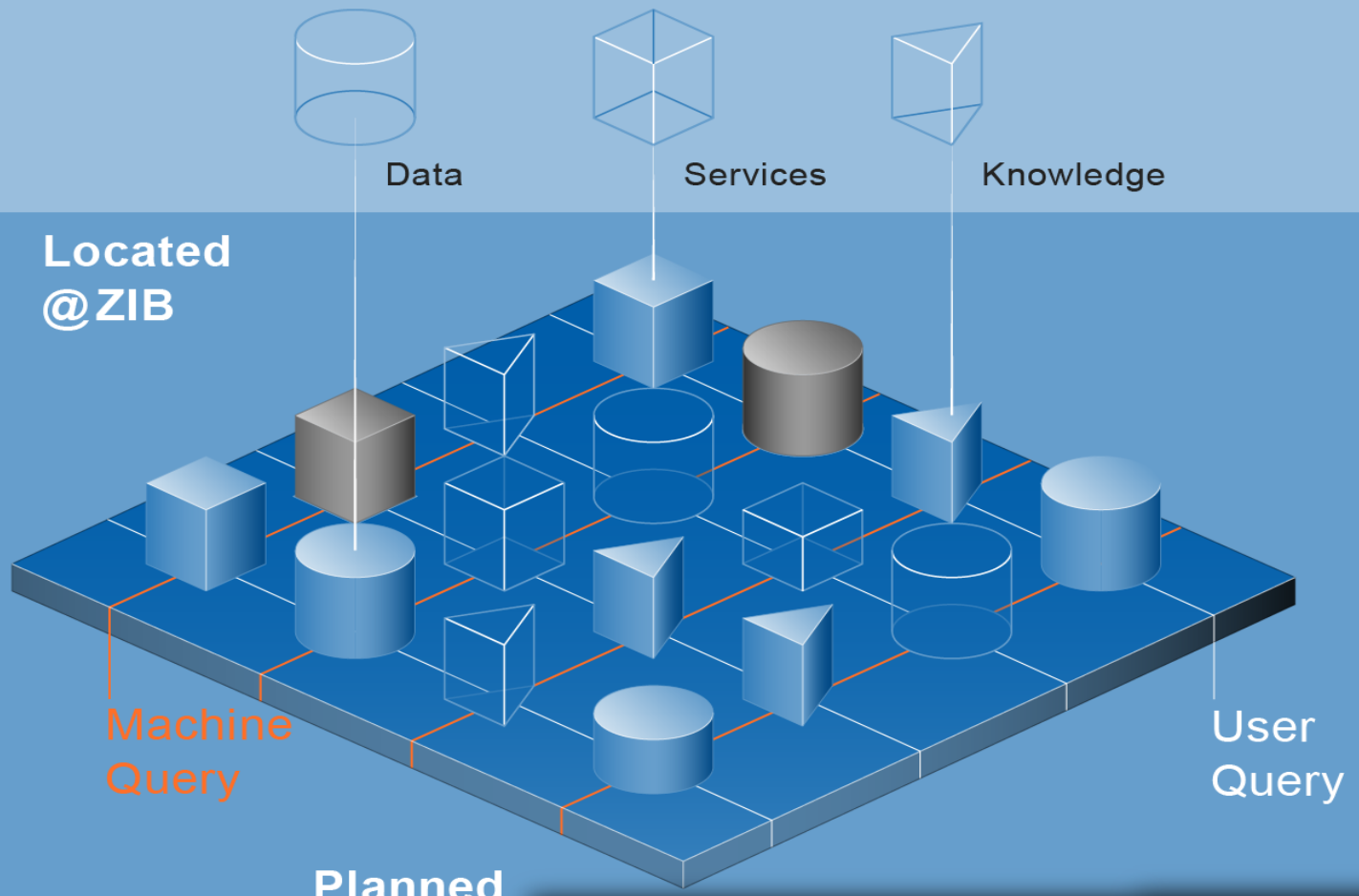
Examples for Services

- OpenML
- MORwiki
- polyDB & SmallGrp

Examples for Infrastructures

- zbMATH & swMATH
- Encyclopedia of Mathematics
- RADAR

Located @ Provider-Site



Located @ ZIB

External

- OpenAIRE
- Zenodo
- DLMF

Examples for Services

- OpenML
- MORwiki
- polyDB & SmallGrp

Examples for Infrastructures

- zbMATH & swMATH
- Encyclopedia of Mathematics
- RADAR

Planned Services:

Data

- Repositories for Computer Algebra
- Library of Benchmark Data
- Persistent Identifier Registry

Research

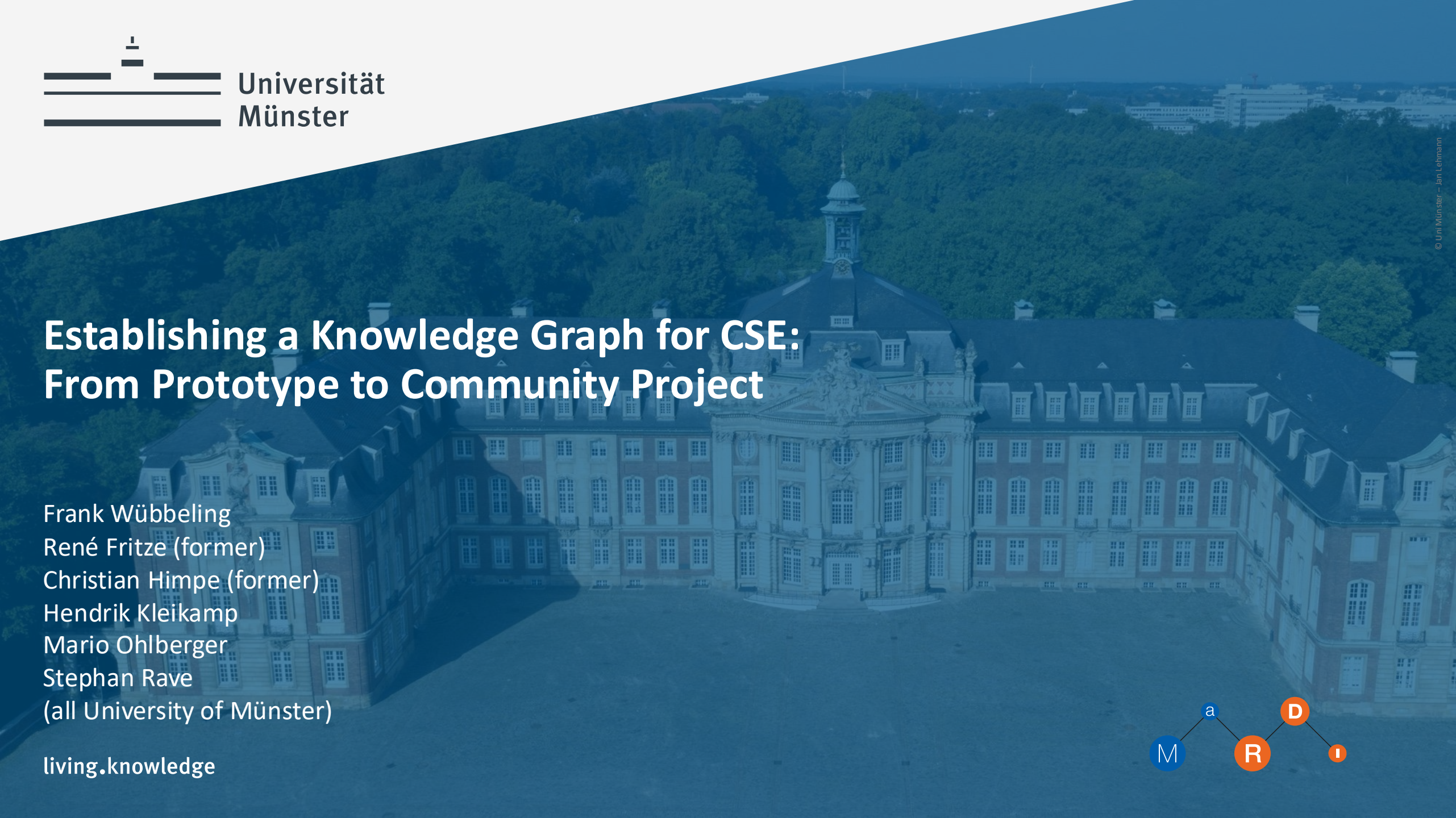
- Benchmark Framework
- Library of Statistical Analysis
- Workflow and Data Certification Service

Knowledge

- Knowledge Graph of Numerical Algorithms
- Knowledge Graph for Modeling and Simulation
- Terminology Service

Existing New Services

- Computer Algebra (OSCAR):
  - **FAIR file formats; MaRDI data packaging system; FAIR Computer Algebra Workflow guidelines**
- Scientific Computing:
  - **Algorithm, Mathematical Model Knowledge Graph; OpenInterfaces; Benchmarking and Workflow**
- Statistics/Machine Learning
  - **OpenML datasets and FAIR algorithms for data-driven approaches**
- Central
  - **The MaRDI portal (portal.mardi4nfdi.de)**
  - **Teaching Lecture Notes and Slides; Research Guidelines; Help Desk**



Universität Münster

Establishing a Knowledge Graph for CSE: From Prototype to Community Project

Frank Wübbeling
René Fritze (former)
Christian Himpe (former)
Hendrik Kleikamp
Mario Ohlberger
Stephan Rave
(all University of Münster)

living.knowledge

Why?

- Scientific computing focusses on algorithms. But:
  - **There is currently no way of systematically searching for data associated with an algorithm**
  - **Where ``data´´ is**
    - **Articles inventing, discussing, testing, analyzing, applying … it**
    - **Software implementing it**
    - **Benchmarks testing it**
    - **Problems they solve (the reverse is more interesting)**
    - **Required/exploited prerequisites (spd, sparse), properties (order, complexity)**

What?

- Knowledge graphs are a well-established technique for representing knowledge in a graph-structured data model. In a nutshell, they represent relations between objects (triples).
- Application to scientific computing
  - Algorithms solve problems
  - Posess mathematical properties
  - Implemented in software
  - Tested by benchmarks
  - Documented in publications

What‘s the benefit? Get answers to…

Starting as a practitioner from a (practical) problem:

- What are the available algorithm options for my problem? (Hint: Need link to Engineering graph)
- Are the prerequisites for an algorithm satisfied by my problem? Does it exploit its properties?
- Which implementations are available? Where do I find a common analysis?
- How can I document which algorithms I used for my problem?
- Need references for all.

Starting as an algorithm developer:

- Which methods am I competing against? (keeping up with developments in vivid fields ain‘t easy)
- Which properties distinguish my algorithm from others?
- Which benchmarks should I run? Which implementations should I use for competing algorithms?
- Where do I find problems that used competing algorithms?
- Need references for all.

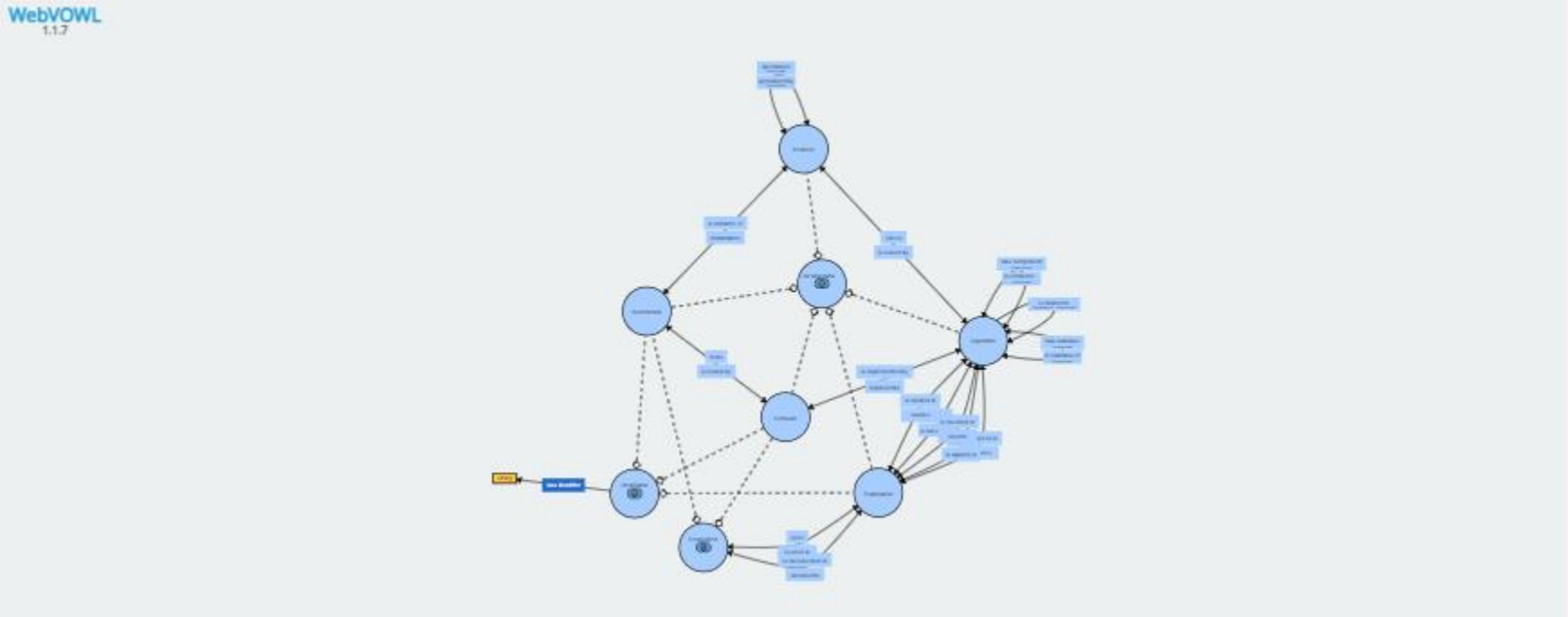
Which structure?

- The ontology formalizes this structure using predefined relations between objects.
- Which questions do you want answered?
  - **Which algorithms solve emission tomography problems?**
  - **Which publication invented Maximum Likelihood Expectation Maximization?**
  - **Which software implements filtered backprojection?**
  - **Which benchmark tests Ordered Subset Expectation Maximization?**

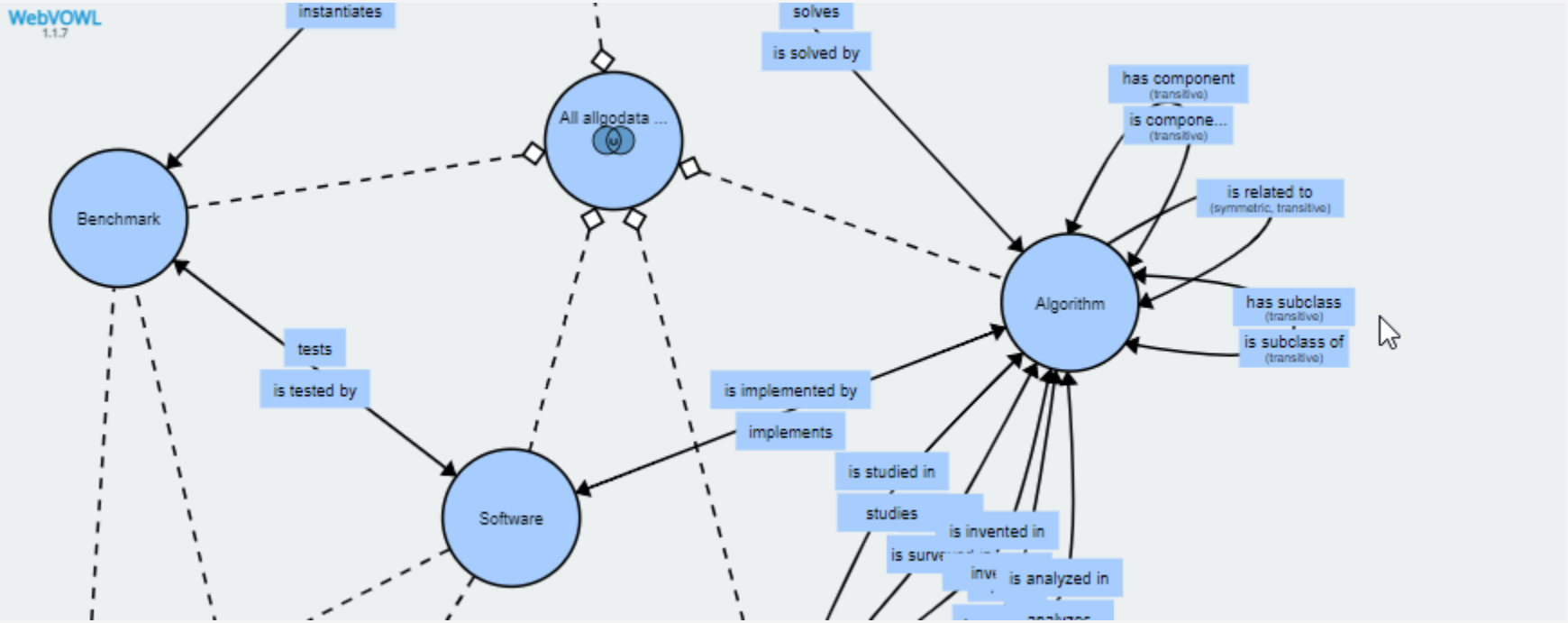
What does it finally consist of?

- Set of objects and relations using ontology web language (OWL) standards
  - **Objects: Problem, Algorithms, Software, Benchmark, Publication, Mathematical Property**
  - **Hierarchy: Problems, Algorithms, Software**
  - **Relations: implements, implementedBy, solves, solvedBy, analyzedIn…**
    (total 35 relations, but most of these with inverse, 20 actual relations)
- **Our Goal: Create a curated knowledge graph for scientific computing in this ontology.**

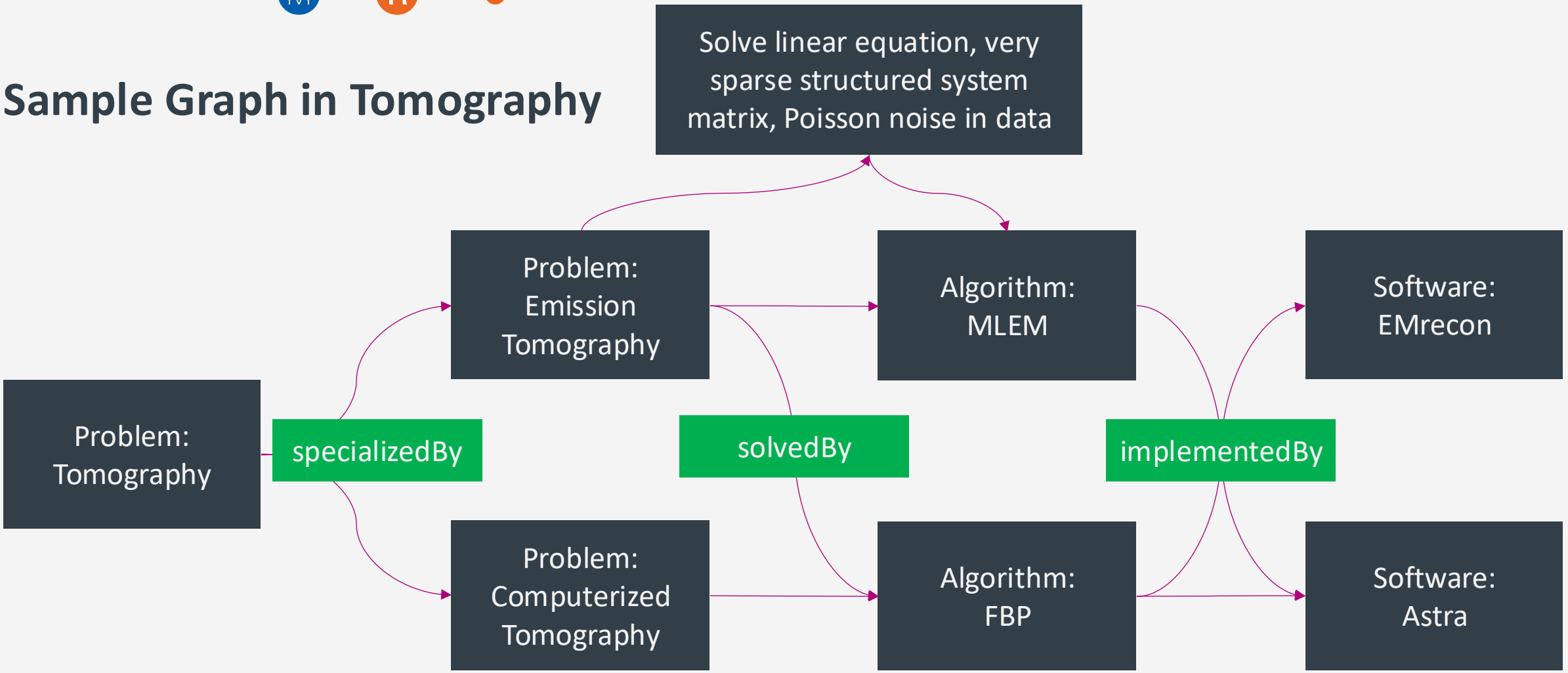
OWL Graph



OWL Graph


WebVOWL
1.1.7
instantiates
solves
is solved by
has component
(transitive)
is compone...
(transitive)
All allgodata ...
Benchmark
is related to
(symmetric, transitive)
Algorithm
has subclass
(transitive)
is subclass of
(transitive)
tests
is tested by
is implemented by
implements
is studied in
studies
is invented in
is analyzed in
Software

Sample Graph in Tomography



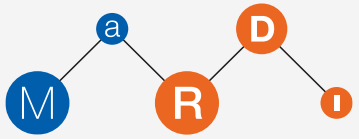
What is available?

- Sample knowledge graphs in two fields:
  - **Numerical Linear Algebra (basic)**
  - **Model order reduction:**
    22 problems/subproblems, 90 algorithms, 17 implementations,
    38 benchmark problems, 604 publications
- Sample open source implementation of query infrastructure and GUI
  - **Django on top of Jena-Fuseki with OWL reasoner in a container setup**
  - **Public available**

How can I use it?

- Raw graph in standard data formats (turtle, RDF, triples, …)
- SPARQL database access
- Standardized REST/API access
  ex: return all publications that analyze MLEM (as JSON)
  ex: convenience functions for resolution of DOIs etc.
- Send natural language requests through an LLM
- Sample GUI
  relies on the API (and has minimal functionality in itself)
- **Public on https://mathalgodb.mardi4nfdi.de**
- **Beta of unified graphs with TA4 on https://mtsr2024.m1.mardi.ovh/**

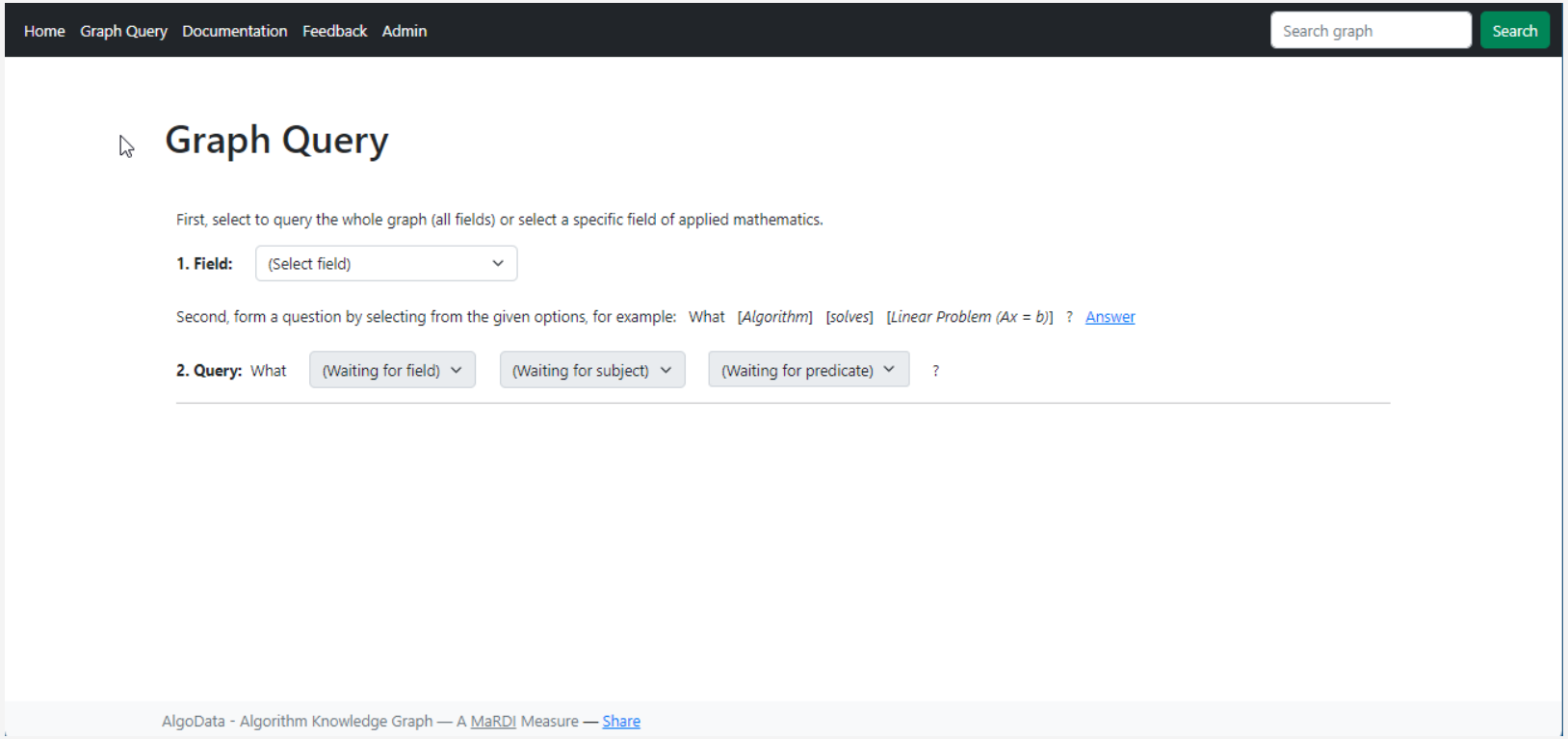
Universität Münster



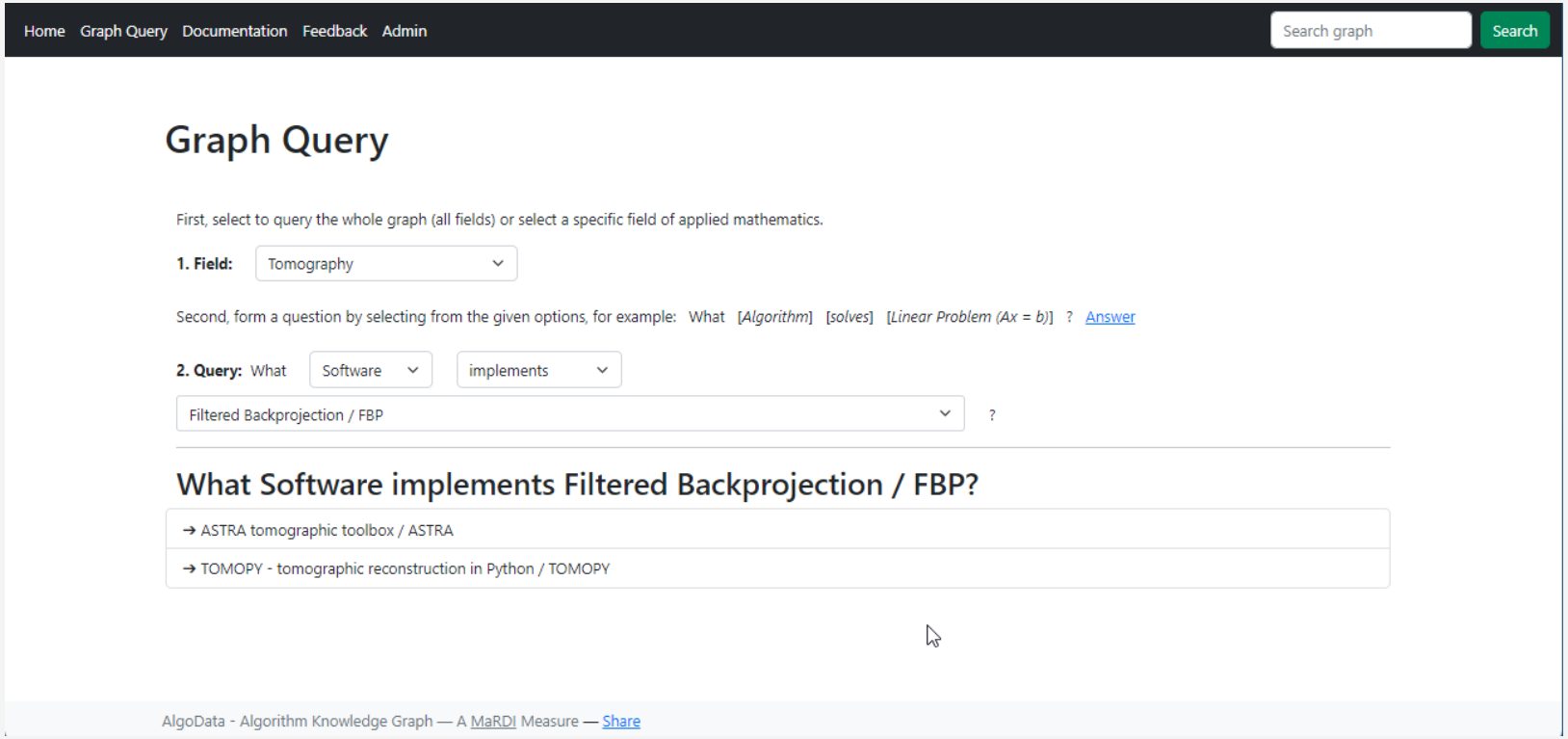
Processing natural language requests with LLMs

https://www.uni-muenster.de/FB10srvi/service/openai/query.php

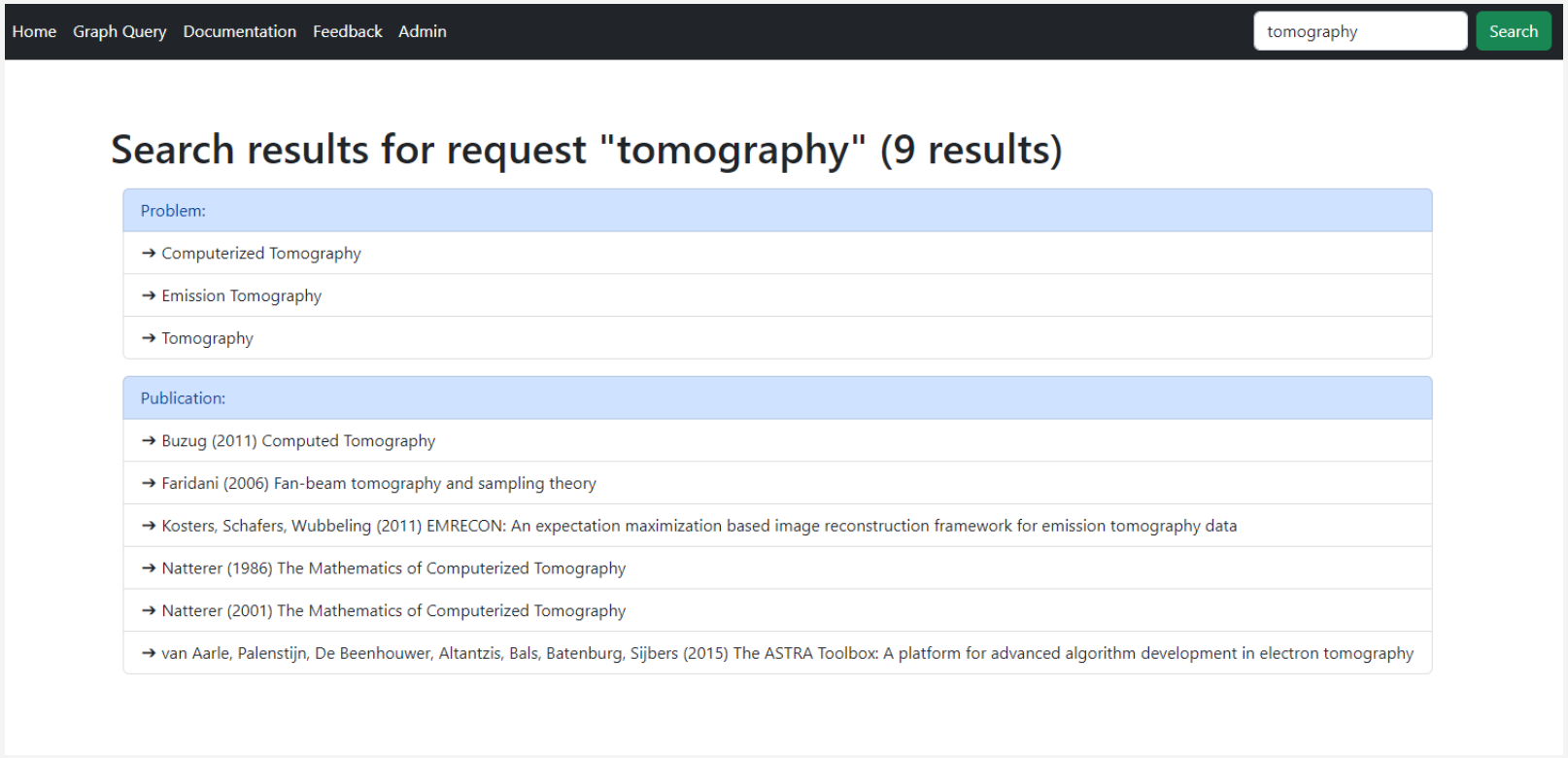
GUI


Home
Graph Query
Documentation
Feedback
Admin
Search graph
Search
Graph Query
First, select to query the whole graph (all fields) or select a specific field of applied mathematics.
1. Field:
(Select field)
Second, form a question by selecting from the given options, for example: What [Algorithm] [solves] [Linear Problem (Ax = b)] ? Answer
2. Query: What
(Waiting for field)
(Waiting for subject)
(Waiting for predicate)
?
AlgoData - Algorithm Knowledge Graph — A MaRDI Measure — Share

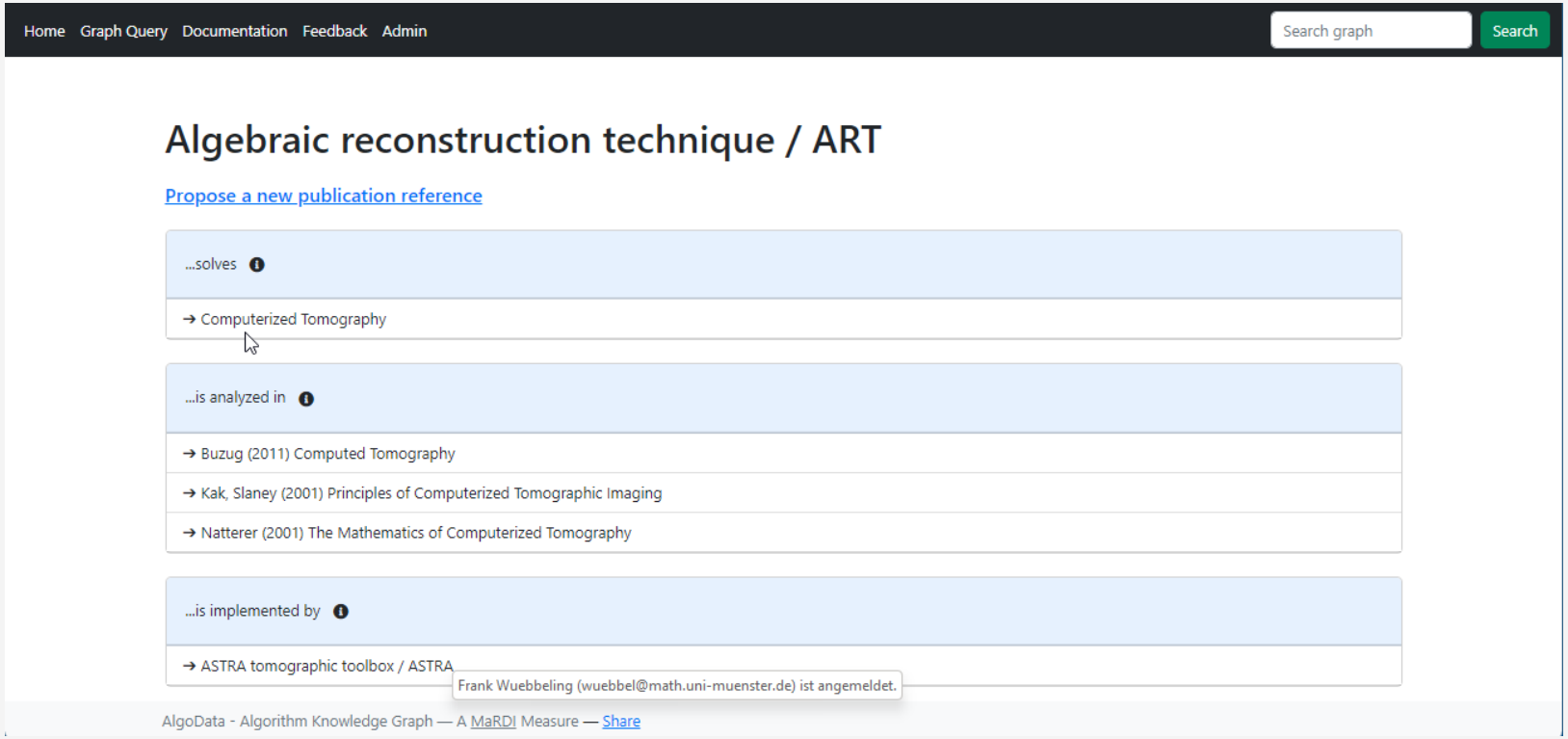
GUI: Guided Queries



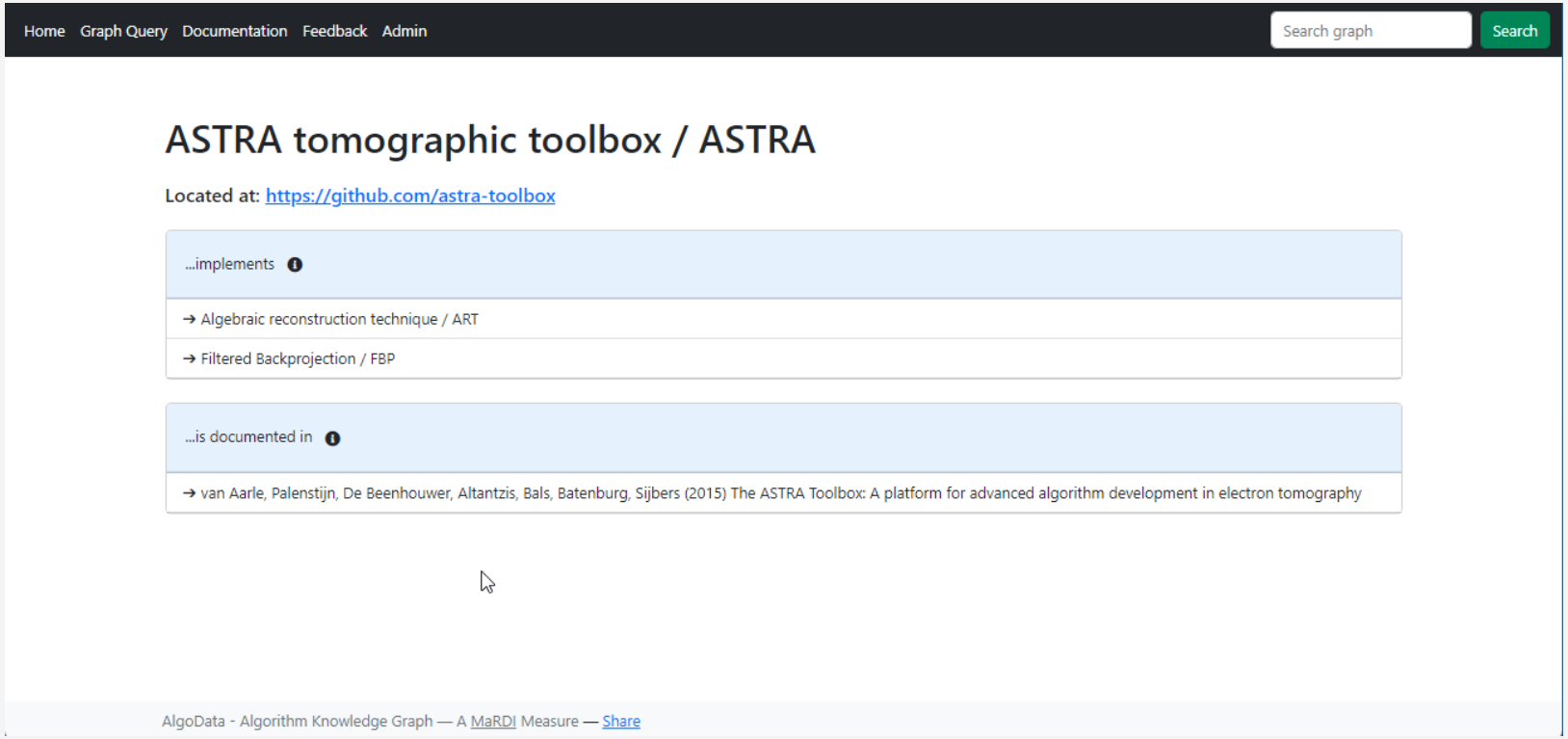
GUI: keyword search



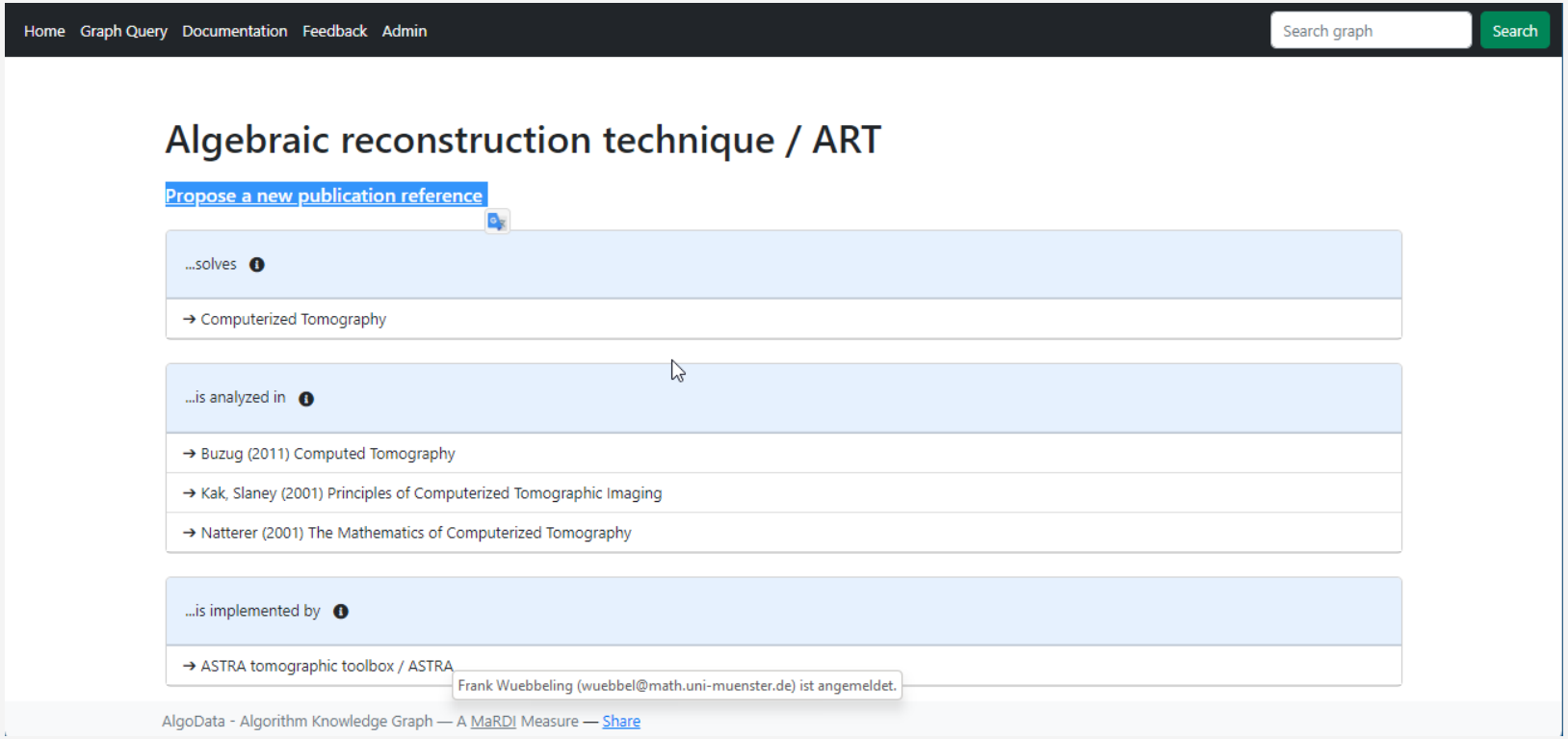
GUI result display



GUI result display



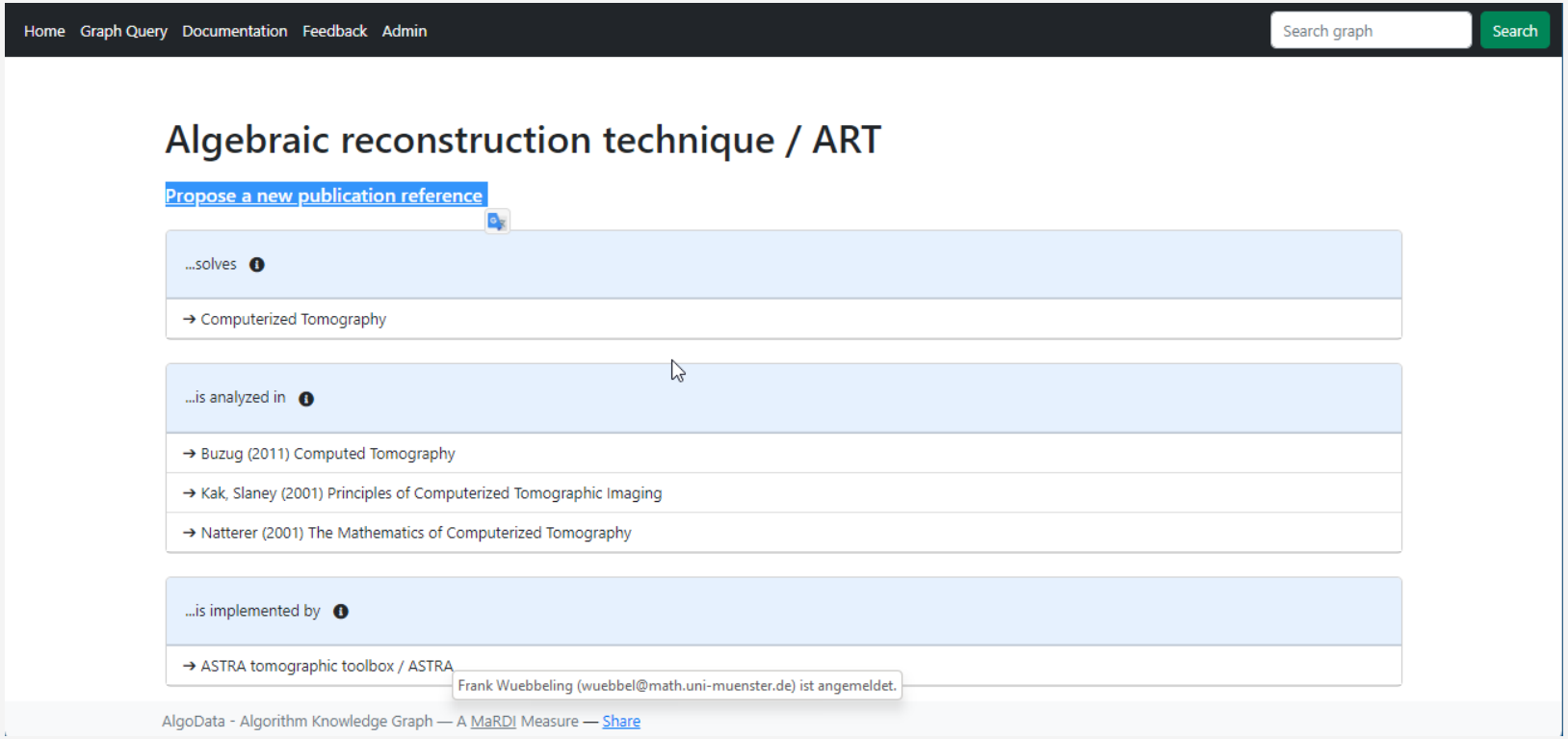
GUI: Invites to play around, discover, explore



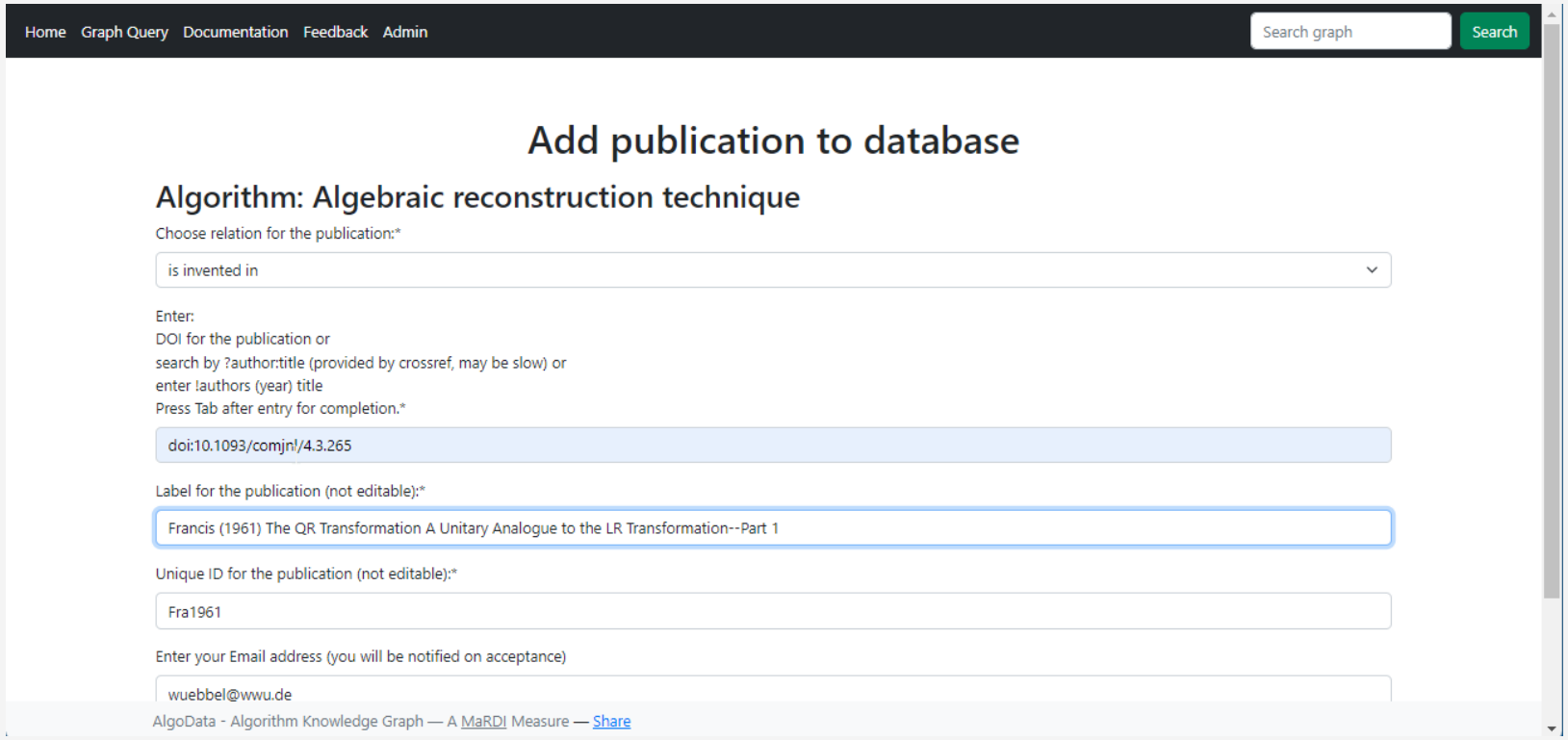
How can we overcome the hen and egg problem?

- The project will succeed only with support from the communities.
- It is vital that the work of adding pieces to the graph is as simple as possible
- Workflow:
  - **An initial graph for a field, formalizing the general structure (basic problems, algorithms, …) is created by editors from the community.**
  - **The graph is continually updated using proposals from the community, accepted or denied by the editors.**

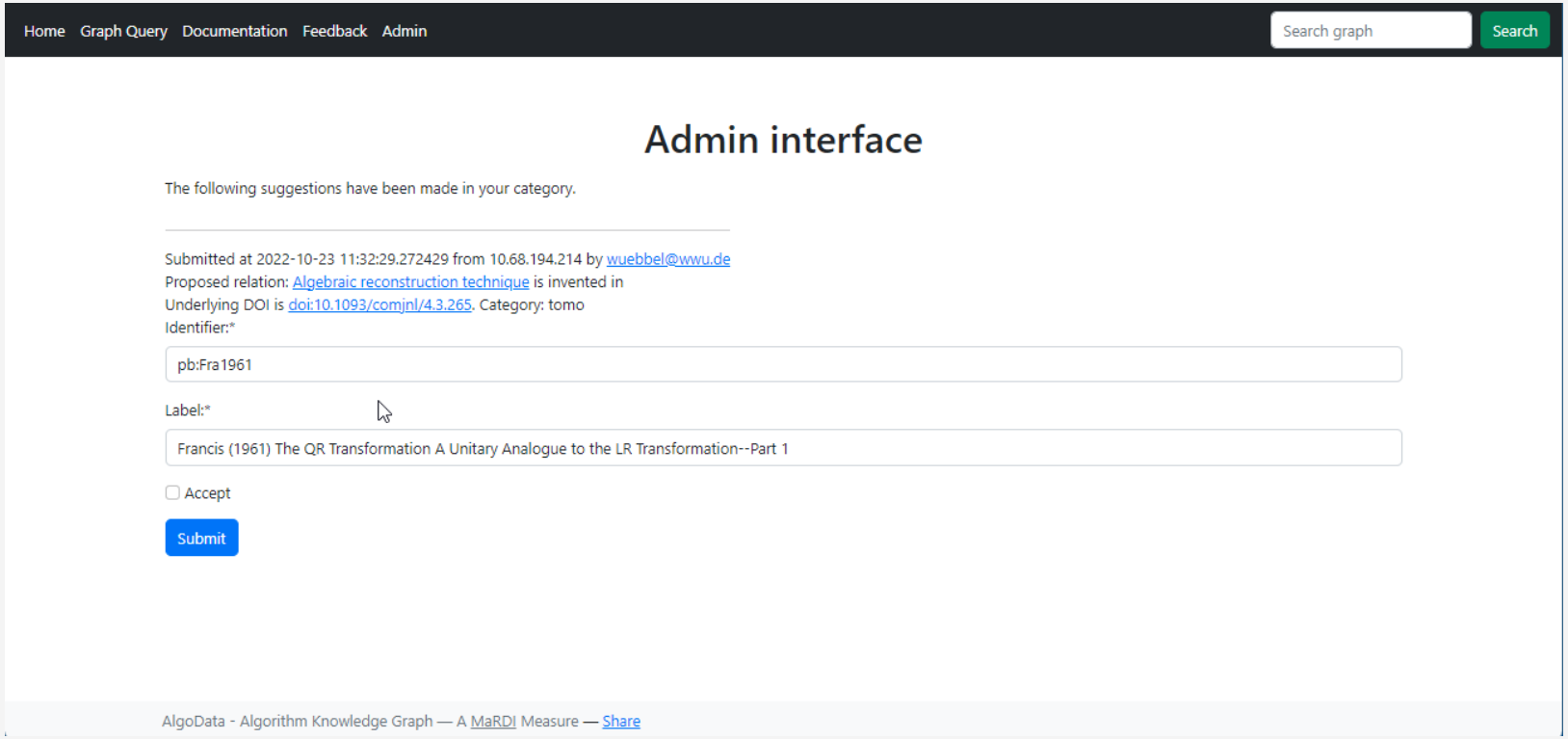
Contributing to the knowledge graph



Contributing to the knowledge graph



Checked by editorial board



Summary

- We defined a knowledge graph for Scientific Computing
- The knowledge graph supports scientists exploring fields
- Structure and data access are open, sample implementations are based on open source standards.
- Available access levels are raw data, database access, REST API, GUI
- Sample datasets have been constructed (with only MOR being fairly wide)
- Adding data to the graph and creating a new field is easy
- **Public on https://mathalgodb.mardi4nfdi.de/**
- **Public beta of unified graph with MathModDB on https://mtsr2024.m1.mardi.ovh/**

Outlook

- Integration into larger ontology for mathematics / engineering
- More fields
- Permanent production deployment
- Community building
- Board of experts in relevant fields