# Pragmatic semantic data management with CaosDB

Alexander Schlemmer, Ulrich Parlitz, Stefan Luther

Max Planck Institute for Dynamics and Self-Organization

2022-03-16

www.bmp.ds.mpg.de

Caosdb

an open scientific database

# Overview

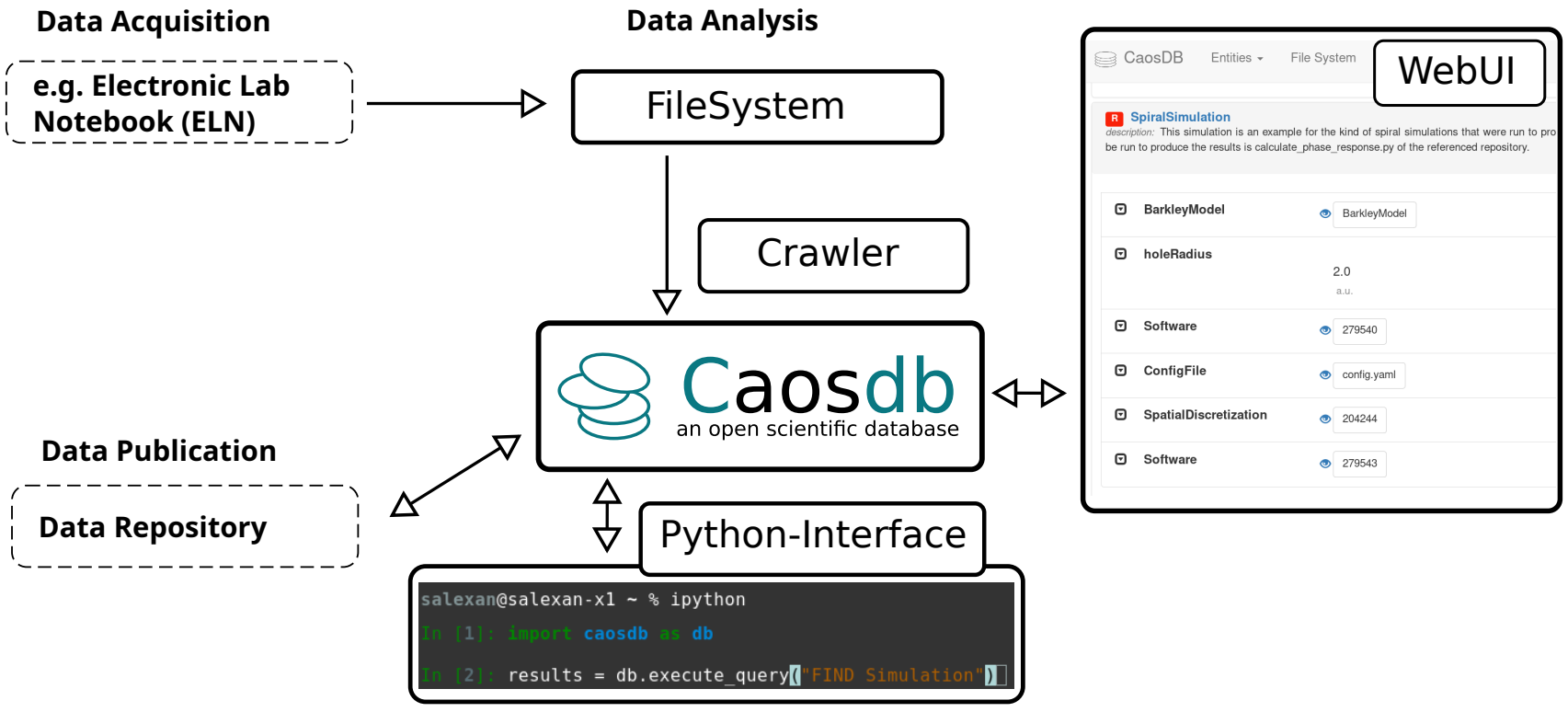
- CaosDB Overview
- Data Models
- CQL and Python API
- Crawler

# What is CaosDB?

- Semantic Research Data Management System (RDMS) Fitschen et.al., *Data* 2019, 10.3390/data4020083
- Developed at the Max Planck Institute for Dynamics and Self-Organization (Göttingen) since 2010
- Open Source project since 2018: gitlab.com/caosdb
- Commercial support available by IndiScale GmbH[1] (since 2019)
- Currently approximately 15 instances in very different scientific disciplines

[1] A.S. is a co-founder of IndiScale.

# CaosDB


Data Acquisition
e.g. Electronic Lab Notebook (ELN)
Data Analysis
FileSystem
Crawler
Caosdb
an open scientific database
Data Publication
Data Repository
Python-Interface
salexan@salexan-x1 ~ % ipython
In [1]: import caosdb as db
In [2]: results = db.execute_query("FIND Simulation")
CaosDB
Entities
File System
WebUI
SpiralSimulation
description: This simulation is an example for the kind of spiral simulations that were run to pro
be run to produce the results is calculate_phase_response.py of the referenced repository.
BarkleyModel
BarkleyModel
holeRadius
2.0
a.u.
Software
279540
ConfigFile
config.yaml
SpatialDiscretization
204244
Software
279543

# Aims of the CaosDB Project

- Flexibility: Data models frequently change in scientific environments
- Linking of arbitrary data types with arbitrary file sizes
- Management of raw data, algorithms, analyses and publications (c.f. Spreckelsen et.al., *Data* 2020, 10.3390/data5020043)
- Integration of decentrally organized data sources
- Integration of arbitrary data acquisition- and data analysis software
- Built on robust (open source) software and standards

# Common Misconceptions (1)

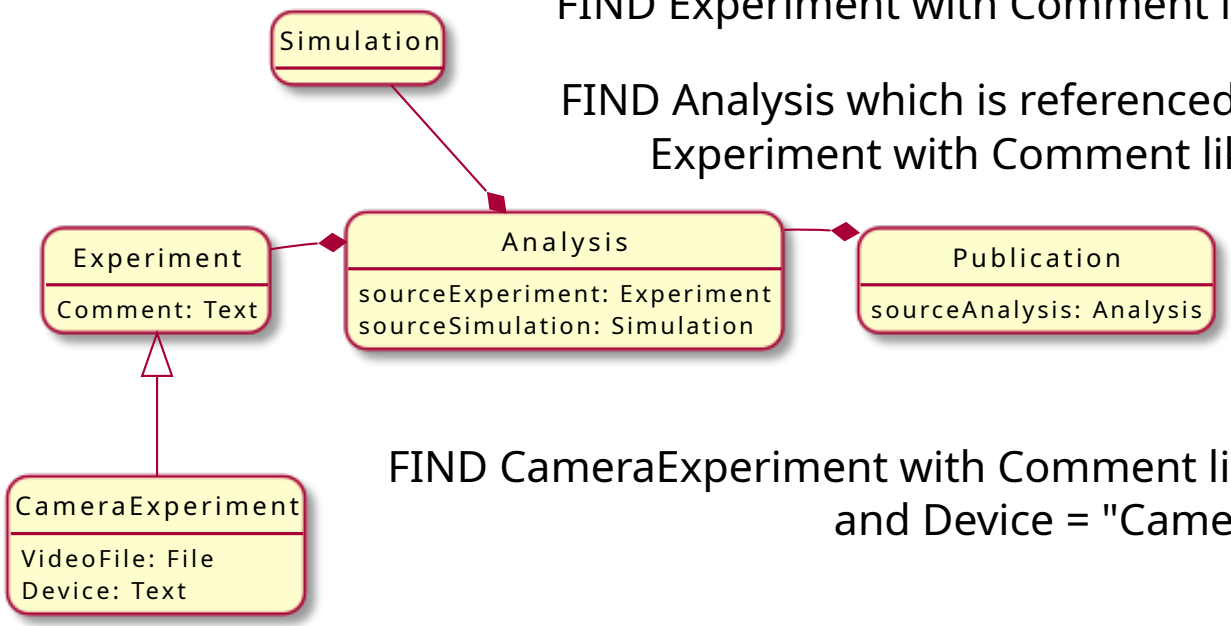
- "Just give me your data model, then I can design a database (schema) for your data..."
- User might not know the (complete) data model yet.
- Data model design and RDMS not completely independent.
- Database schema will probably be outdated upon completion.

# Possible Solution: Flexible Data Model

- Use a system that allows for (extensive) modifications of the data model.
- Start with a simple data model and add more details later.
- The data model will be improved iteratively.

# Flexible Semantic Data Modell

**Data Model**




**Query Language CQL**

FIND Experiment with Comment like "*successful*"

FIND Analysis which is referenced by
Experiment with Comment like "*successful*"

FIND CameraExperiment with Comment like "*successful*"
and Device = "Camera 1"

# 1st Practical Session: CaosDB and data models

*Live Presentation*

# CQL: CaosDB Query Language

*Why another language?*

- "There is SPARQL as a standard language."
- "Why implement a search language at all and not just create a user interface?"

# Recent use case

*Let's build a query!*

- First or last name starts with letter "M"
- Beginning of 20th century
- From the United Kingdom
- Female
- Writer

# SPARQL

```
select distinct ?item ?itemLabel ?fullName where {
    ?item wdt:P31 wd:Q5; wdt:P27 wd:Q145; wdt:P21 wd:Q6581072;
          wdt:P106 wd:Q36180; wdt:P569 ?birthday;
          wdt:P570 ?diedon; wdt:P734 [rdfs:label ?familyName];
          wdt:P735 [rdfs:label ?givenName].
  BIND(concat(?givenName, " ", ?familyName) as ?fullName)
  FILTER(?birthday > "1870-01-01"^^xsd:dateTime
         && ?diedon < "1950-01-01"^^xsd:dateTime)
  filter(lang(?familyName) = "en")
  filter(lang(?givenName) = "en")
  filter(regex(?givenName, "M.*") || regex(?familyName, "M.*"))
  SERVICE wikibase:label
  { bd:serviceParam wikibase:language "en" } }
```

# SPARQL

```
select distinct ?item ?itemLabel ?fullName where {
    ?item wdt:P31 wd:Q5;  # Any instance of a human.
          wdt:P27 wd:Q145; # United Kingdom
          wdt:P21 wd:Q6581072; # female
          wdt:P106 wd:Q36180; # writer
          wdt:P569 ?birthday;
          wdt:P570 ?diedon;
          wdt:P734 [rdfs:label ?familyName];
          wdt:P735 [rdfs:label ?givenName].
  BIND(concat(?givenName, " ", ?familyName) as ?fullName)
  FILTER(?birthday > "1870-01-01"^^xsd:dateTime
         && ?diedon < "1950-01-01"^^xsd:dateTime)
  filter(lang(?familyName) = "en")
  filter(lang(?givenName) = "en")
  filter(regex(?givenName, "M.*") || regex(?familyName, "M.*"))
  SERVICE wikibase:label
  { bd:serviceParam wikibase:language "en" } }
```

# CQL

```
FIND Woman with occupation = writer and
  (first_name like "M*" or last_name like "M*") and
  birthday > 1870 and died < 1950 and citizenship = UK
```

# Use cases for a high level query language

- Automatic data processing
- Complex searches
- Users who like terminals more than graphical user interfaces (There are more of them than you think!)

# 2nd Practical Session: Python API

*Live Presentation*

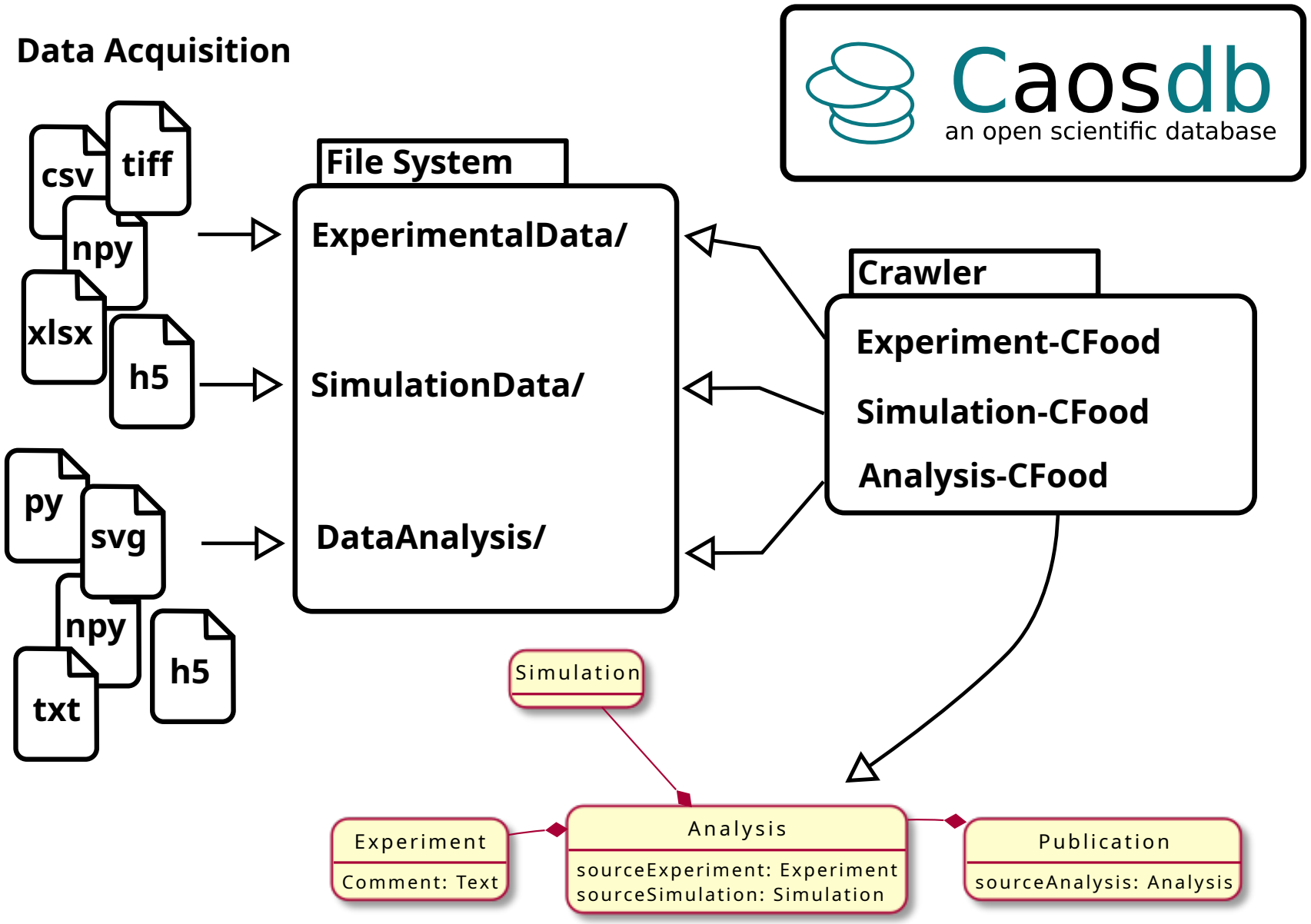
# Common Misconceptions (2)

- "Just migrate all your data into this system to obtain the perfect data management system..."
- Vendor-Lock-in effects?
- Different needs of users might require different solutions. Interoperability?
- What to do if some data does not fit?

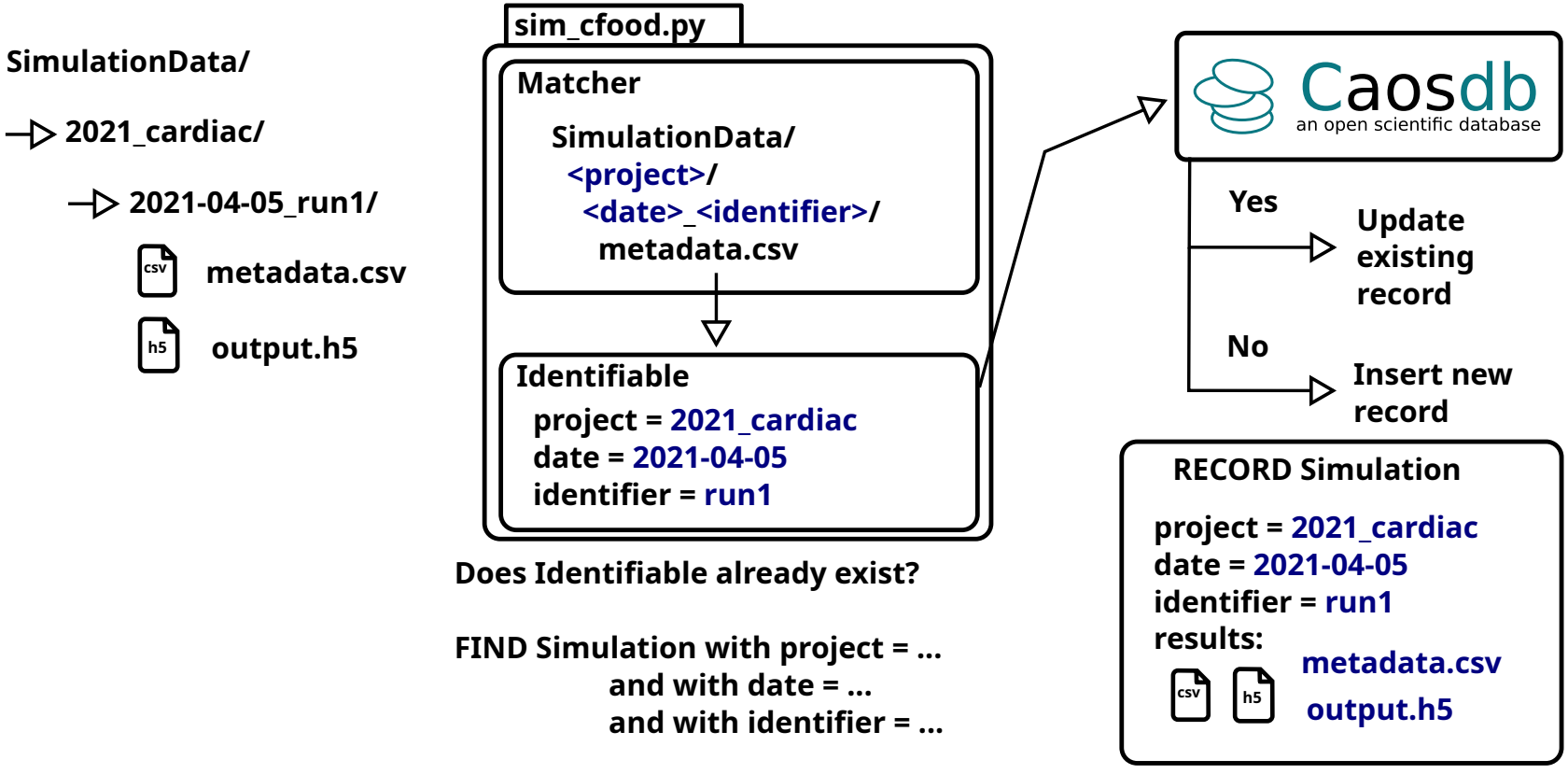
# Possible Solution: Data Migration / Data Crawler

- Don't migrate data, synchronize it!
- Update information in RDMS based on file system.
- RDMS and file system can be used simultaneously.
- Prevents lock-in
- (Additionally) allows for operation without RDMS

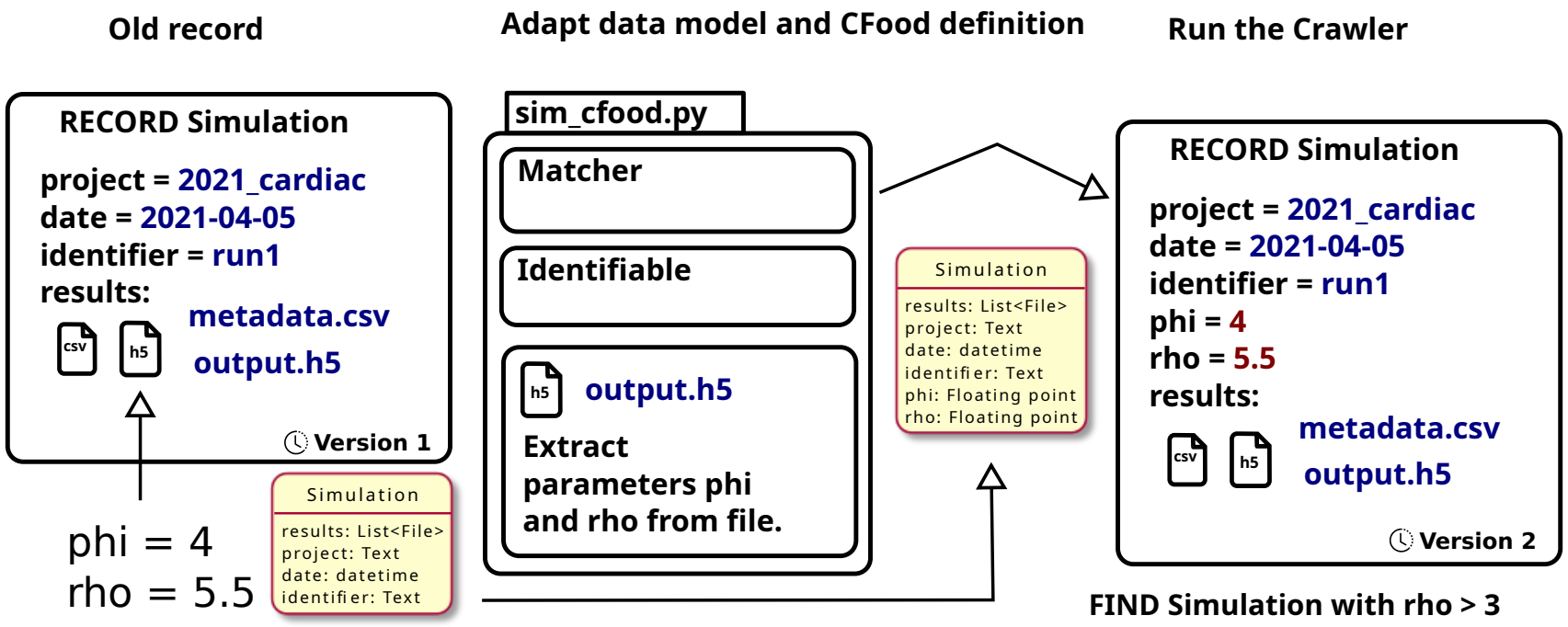
# The Crawler (1)



# The Crawler (2)



# The Crawler (3)



# Common Misconceptions (3)

- "Just let an AI/ML-system categorize and organize your data automatically..."
- Users need to adapt to the structure, that the AI came up with.
- Probably not enough training data for your specific use case.
- Users should be aware of their own data models.
- Might be too unreliable for automatic processing of structures.

# Possible Solution: Human readable crawler specification

- Describes data in sufficient detail
- Allows to synchronize data with RDMS
- Serves as data documentation

# 3rd Practical session: Crawler

*Live Presentation*

# Thank You!

*More information about CaosDB:*

- `http://www.bmp.ds.mpg.de/software/caosdb/`
- `https://gitlab.com/caosdb`
- `https://doi.org/10.3390/data4020083`
- `https://caosdb.org/`

www.bmp.ds.mpg.de

Caosdb

an open scientific database