Adamant: A JSON-Based Metadata Editor for Researchers

Ihda Chaerony Siffa, Marjan Stankov, and Markus M. Becker
23 March, 2023

Department of Plasma Modelling & Data Science, Leibniz Institute for Plasma Science and Technology (INP)

Leibniz Gemeinschaft

FROM IDEA TO PROTOTYPE

Outline

- **Basics**
  - FAIR data principles
  - Metadata
  - JSON Schema
  - Electronic lab. notebook (eLabFTW)
- **Adamant**
  - Architecture
  - Main features
- **Live Demo**
  - Setting up Adamant
  - RDM workflows with Adamant
  - Other workflows
- **Outlook**

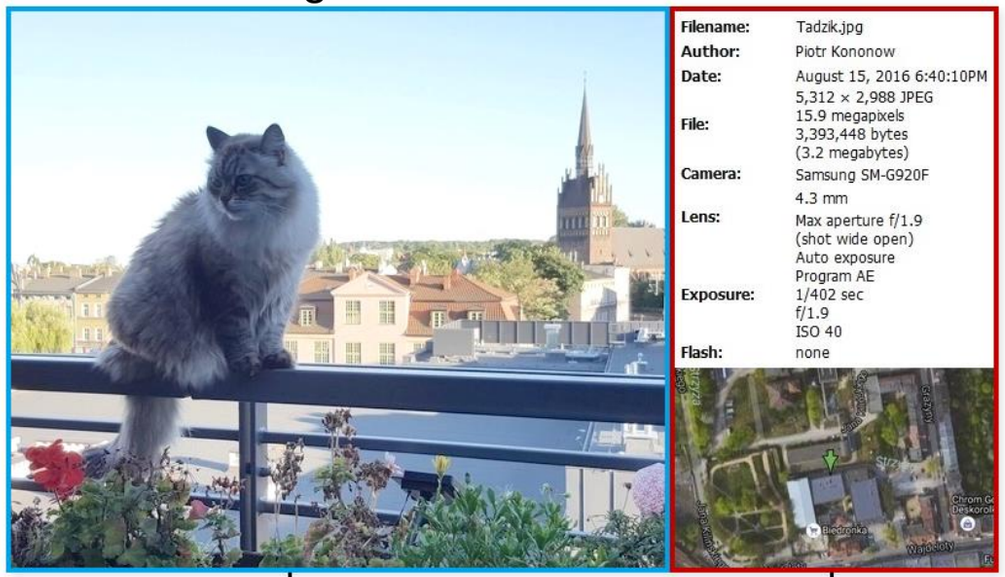
Basics
FAIR data principles

Seminal paper of M. Wilkinson *et al.* 2016 *Sci Data* **3,** 160018, https://doi.org/10.1038/sdata.2016.18
The FAIR Guiding Principles for scientific data management and stewardship

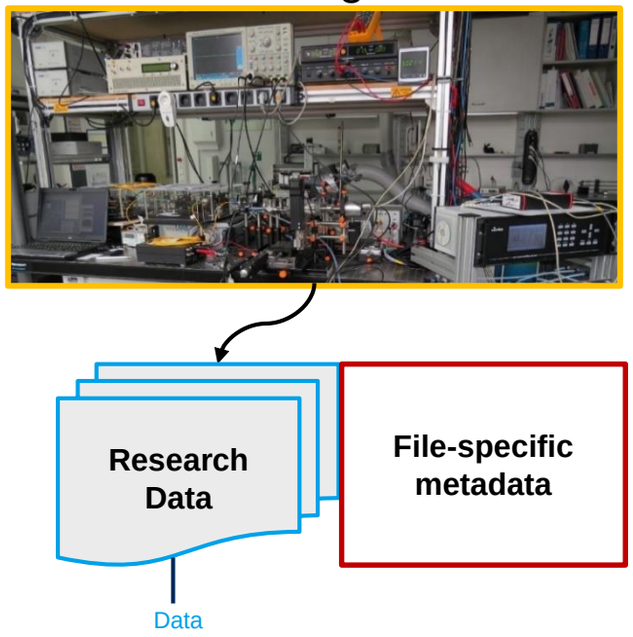
Findable	**A**ccessible	**I**nteroperable	**R**eusable
▪ Identifiable data items ▪ Persistent ▪ Searchable	▪ Retrievable by standardized communication protocols ▪ Authentication where necessary ▪ Access to metadata, even if data is not accessible	▪ Formal, accessible representation of data ▪ Qualified references to other items	▪ Rich and standard-compliant metadata ▪ Proper licensing ▪ Provenance

Basics
What is metadata?

General setting



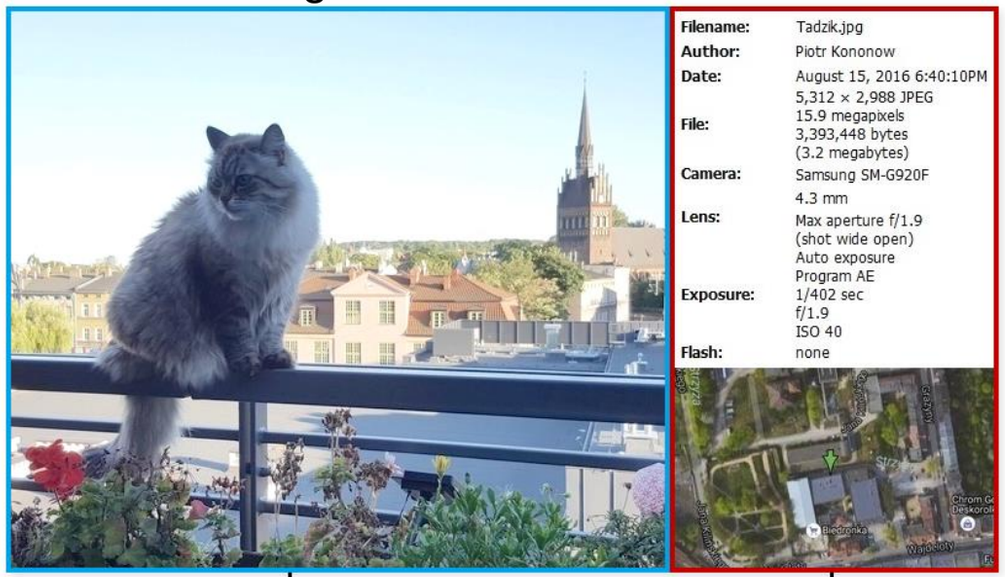
Research setting



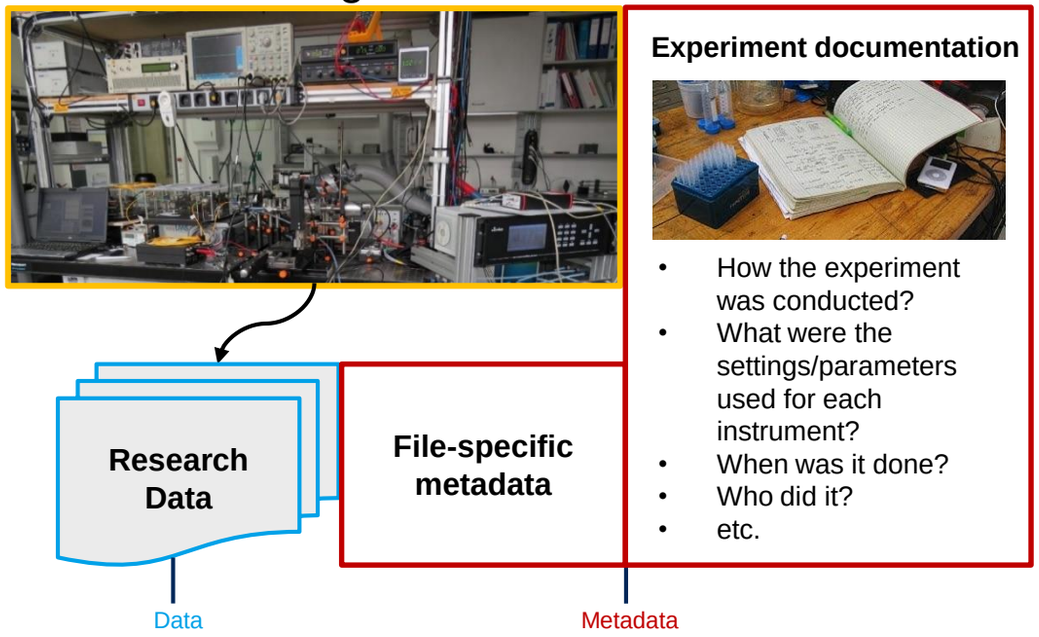
https://dataedo.com/kb/data-glossary/what-is-metadata

Basics
What is metadata?

General setting



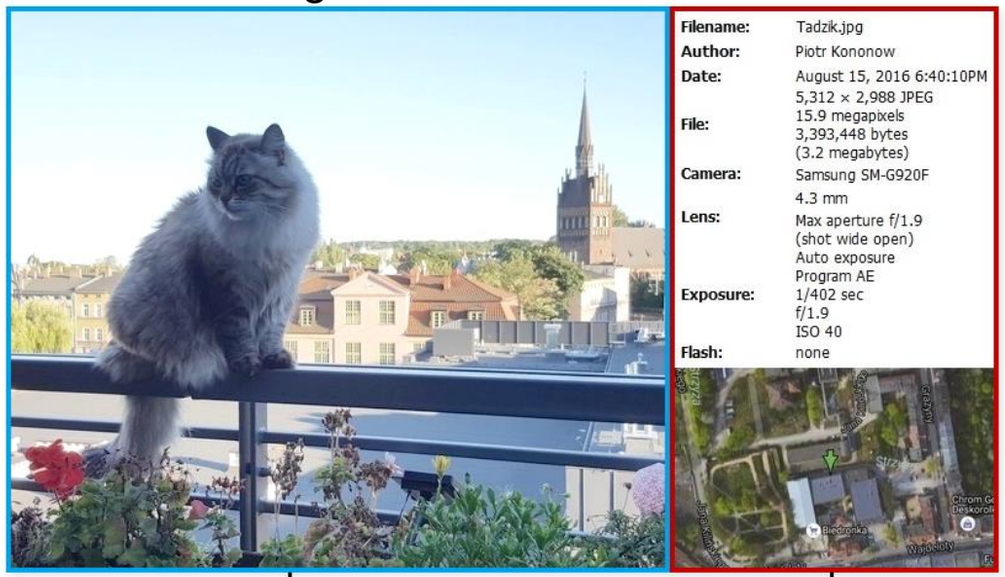
Research setting



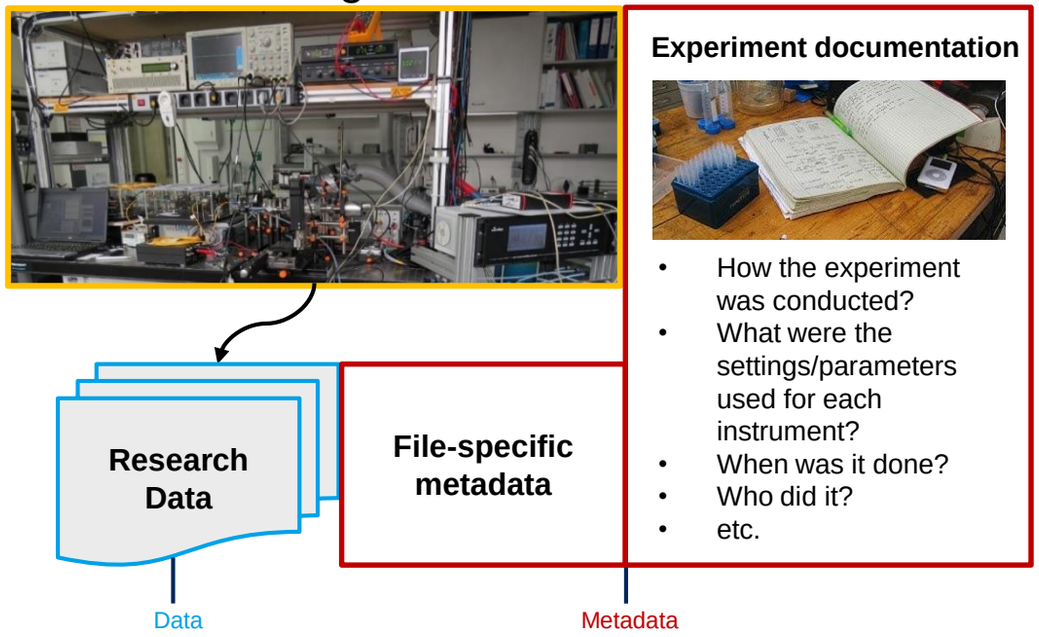
https://dataedo.com/kb/data-glossary/what-is-metadata

Basics
What is metadata?

General setting



Research setting



Metadata is simply data about data!

https://dataedo.com/kb/data-glossary/what-is-metadata

Basics
JSON Schema

- From the main site:

  *“**JSON Schema** is a declarative language that allows you to **annotate** and **validate** JSON documents.” – https://json-schema.org/*

Basics
JSON Schema – examples and validation

```
{
  "type": "string",
  "minLength": 2,
  "maxLength": 3
}
```

"A" ✗

"AB" ✓

"ABC" ✓

"ABCD" ✗

```
{ "type": "number" }
```

42 ✓

-1 ✓

Simple floating point number:

5.0 ✓

Exponential notation also works:

2.99792458e8 ✓

Numbers as strings are rejected:

"42" ✗

```
{
  "type": "object",
  "properties": {
    "number": { "type": "number" },
    "street_name": { "type": "string" },
    "street_type": { "enum": ["Street", "Avenue", "Boulevard"] }
  }
}
```

```
{ "number": 1600, "street_name": "Pennsylvania", "street_type": "Avenue" }
```
✓

If we provide the number in the wrong type, it is invalid:

```
{ "number": "1600", "street_name": "Pennsylvania", "street_type": "Avenue" }
```
✗

By default, leaving out properties is valid. See Required Properties.

```
{ "number": 1600, "street_name": "Pennsylvania" }
```
✓

By extension, even an empty object is valid:

```
{ }
```
✓

By default, providing additional properties is valid:

```
{ "number": 1600, "street_name": "Pennsylvania", "street_type": "Avenue", "direction": "NW" }
```
✓

JS package implementing JSON Schema validation:

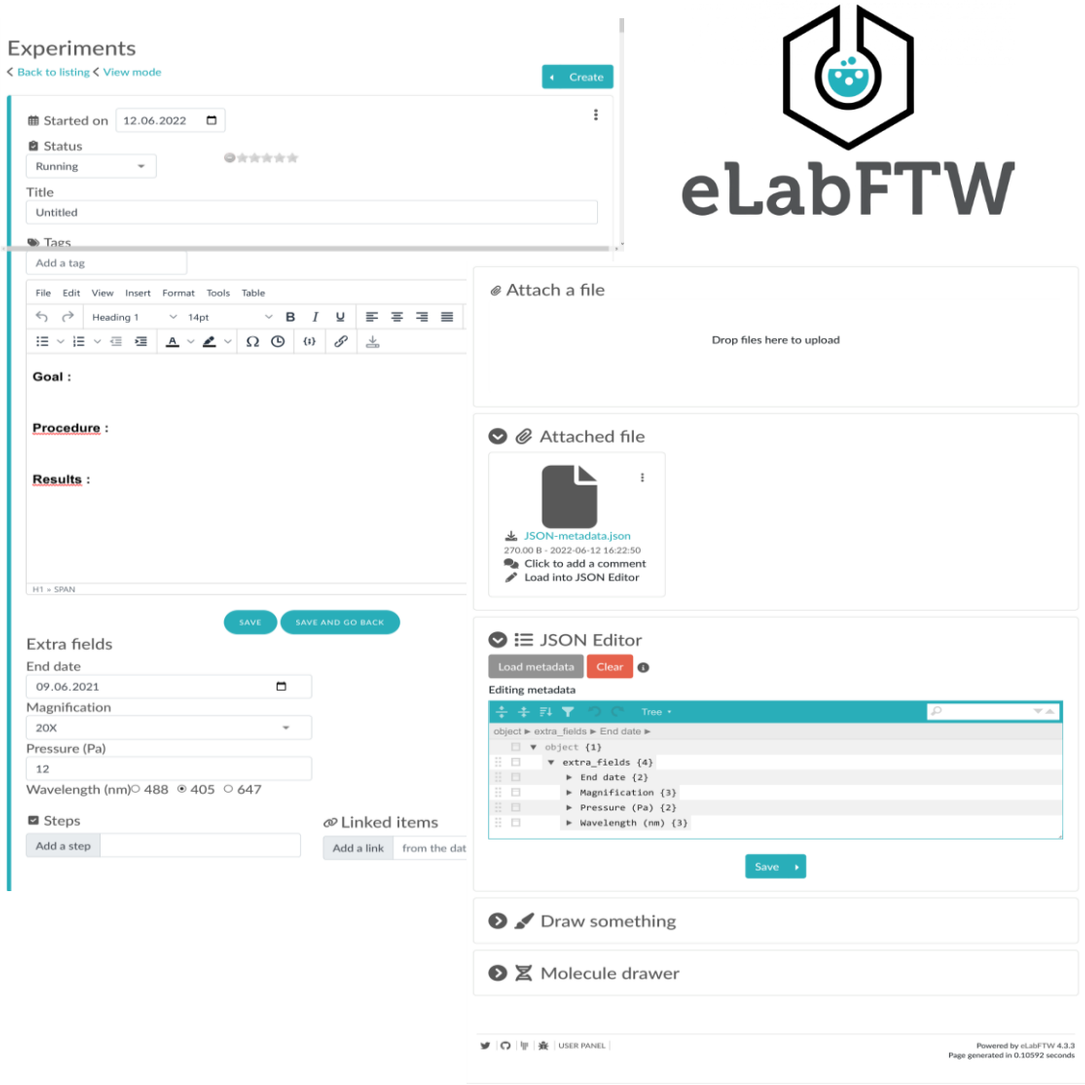
Ajv JSON schema validator

https://json-schema.org, https://ajv.js.org/

Basics

Electronic lab. notebook – ElabFTW: an open-source general purpose ELN

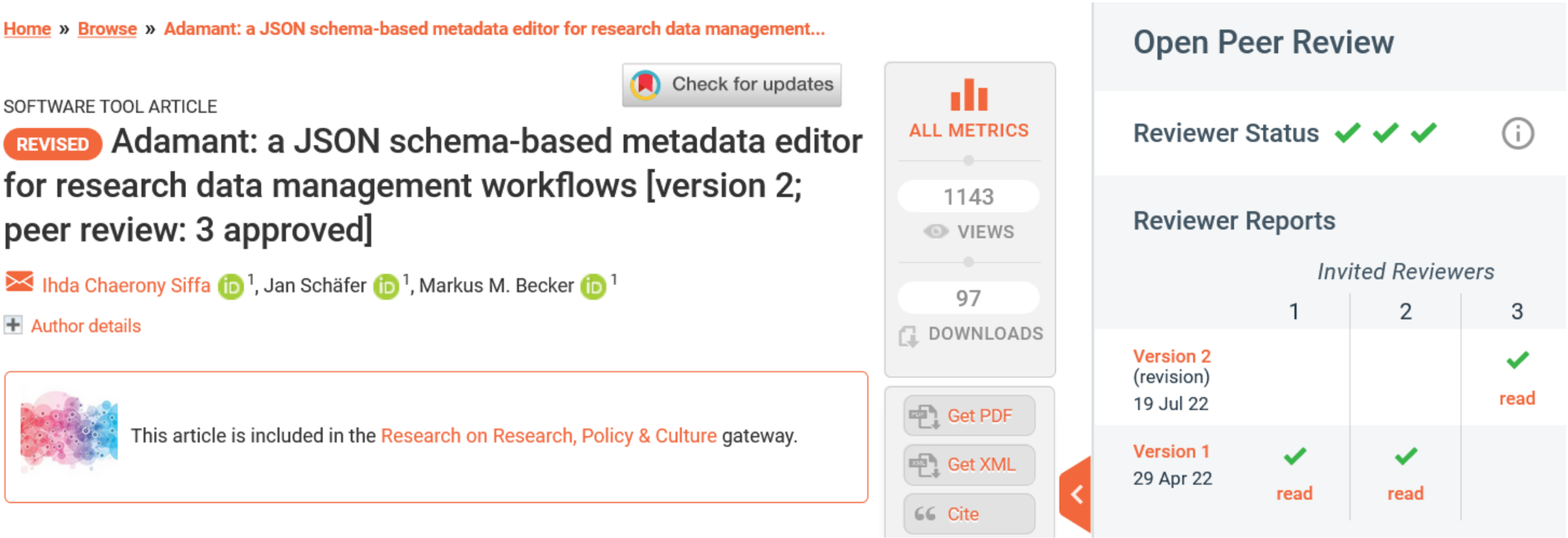
- **It's free and open source**
- **You can self-host it**
- **Flexible and generic, not only for notebook purposes**
  - Experiment templates, easy to create and to tailor
  - Item database
  - Scheduler
  - And many other features



CARPi et al, (2017), eLabFTW: An open source laboratory notebook for research labs, Journal of Open Source Software, 2(12), 146,

Adamant

Technical paper (open peer review)



Home » Browse » Adamant: a JSON schema-based metadata editor for research data management...

Check for updates

SOFTWARE TOOL ARTICLE

REVISED **Adamant: a JSON schema-based metadata editor for research data management workflows [version 2; peer review: 3 approved]**

Ihda Chaerony Siffa [1], Jan Schäfer [1], Markus M. Becker [1]

Author details

This article is included in the Research on Research, Policy & Culture gateway.

ALL METRICS

1143 VIEWS

97 DOWNLOADS

Get PDF

Get XML

Cite

Open Peer Review

Reviewer Status ✓ ✓ ✓

Reviewer Reports

	Invited Reviewers		
	1	2	3
Version 2 (revision) 19 Jul 22			✓ read
Version 1 29 Apr 22	✓ read	✓ read	

GitHub repository: https://github.com/plasma-mds/adamant

Live playground: https://plasma-mds.github.io/adamant/

Chaerony Siffa I, Schäfer J and Becker MM. Adamant: a JSON schema-based metadata editor for research data management workflows [version 2; peer review: 3 approved]. *F1000Research* 2022, **11**:475

Adamant
Architecture

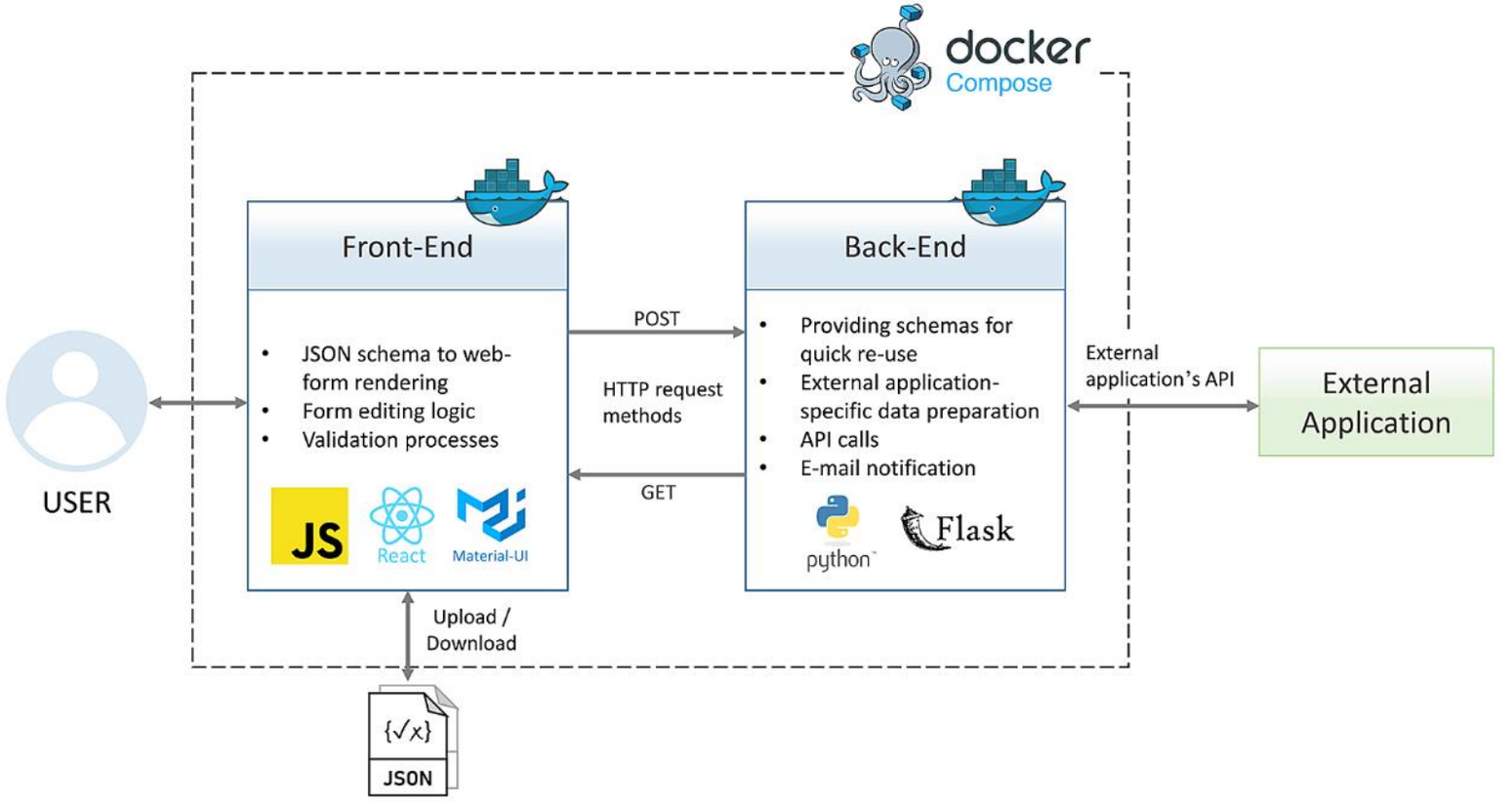


Figure 1. Overview of Adamant's software architecture. API, Application Programming Interface; JSON, JavaScript Object Notation; HTTP, Hypertext Transfer Protocol.

Adamant

Main features – rendering of editable user-friendly web-form

Input: JSON Schema file

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "id": "http://scanning-electron-microscopy",
    "title": "Scanning Electron Microscopy (SEM)",
    "description": "A schema to describe a Scanning Electron Microscopy used in an
        experiment (demo schema)",
    "type": "object",
    "required":["DeviceModel","SEMParameters"],
    "properties": {
        "DeviceModel":{
            "title": "Model of SEM Device",
            "description": "SEM device model used in the experiment",
            "type":"string"
        },
        "SEMParameters":{
            "title": "SEM Parameters",
            "description": "SEM parameters used in the experiment",
            "type":"object",
            "properties": {
                "AccelerationVoltage":{
                    "title": "Acceleration Voltage [kV]",
                    "description": "Voltage applied to accelerate the electrons",
                    "type": "number"
                },
                "WorkingDistance":{
                    "title": "Working Distance [mm]",
                    "description": "Distance from the lens to the sample/specimen",
                    "type": "number"
                },
                "ProbeCurrent":{
                    "title": "Probe Current [nA]",
                    "description": "Electrical current or electron beam focused on the
                        sample/specimen",
                    "type": "number"
                }
            }
        }
    }
}
```

Listing 1. Example of a draft-4 JSON schema containing typical schema-specific keywords presented in blue with their values presented in black, and field element keywords presented in red. JSON, JavaScript Object Notation.

Output: rendered editable web-form

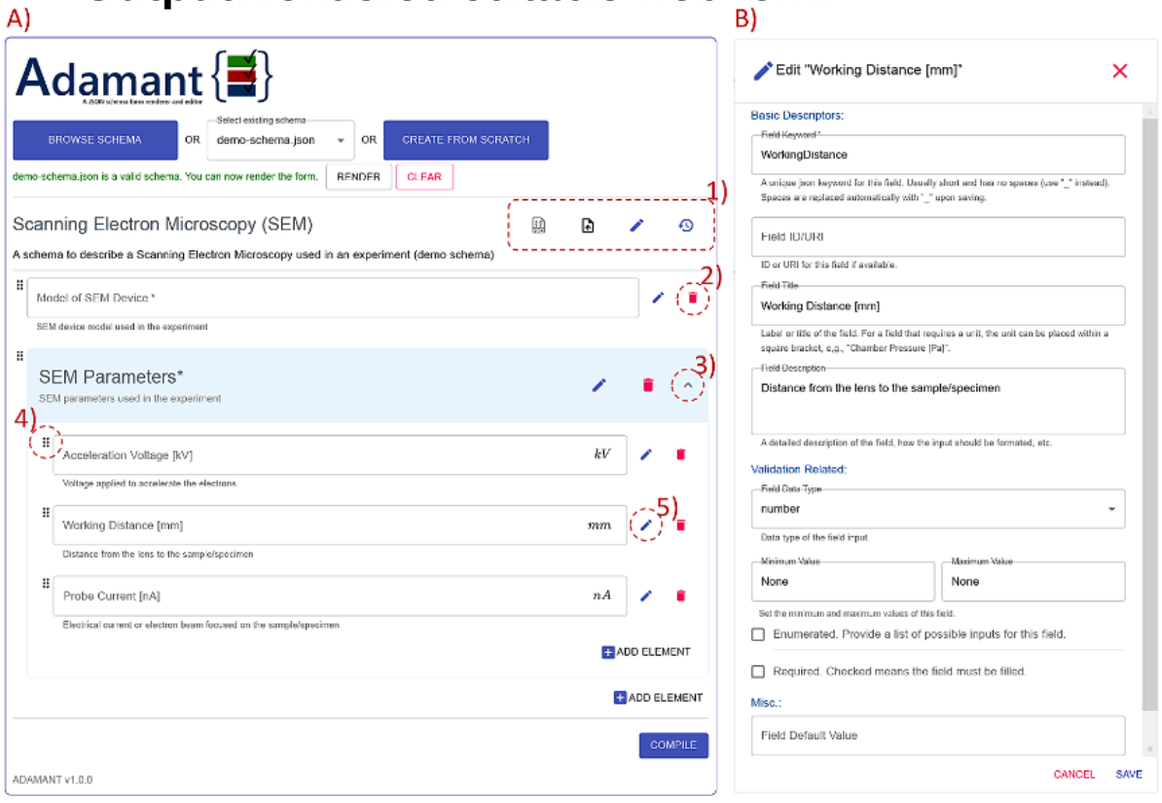


Figure 3. Overview of the Adamant UI with a rendered web-form based on the schema in Listing 1 as an example. (A) Main corpus of the UI; (1) from left to right: JSON schema viewer, auto-populate form, edit schema description, revert all changes; (2) remove form field; (3) collapse or expand the field container; (4) field drag handle; (5) edit field description and (B) field editing panel (as a pop-up on top of the main UI) triggered by clicking (5) the edit button. UI, user interface; JSON, JavaScript Object Notation.

Adamant

Main features – rendering of editable user-friendly web-form

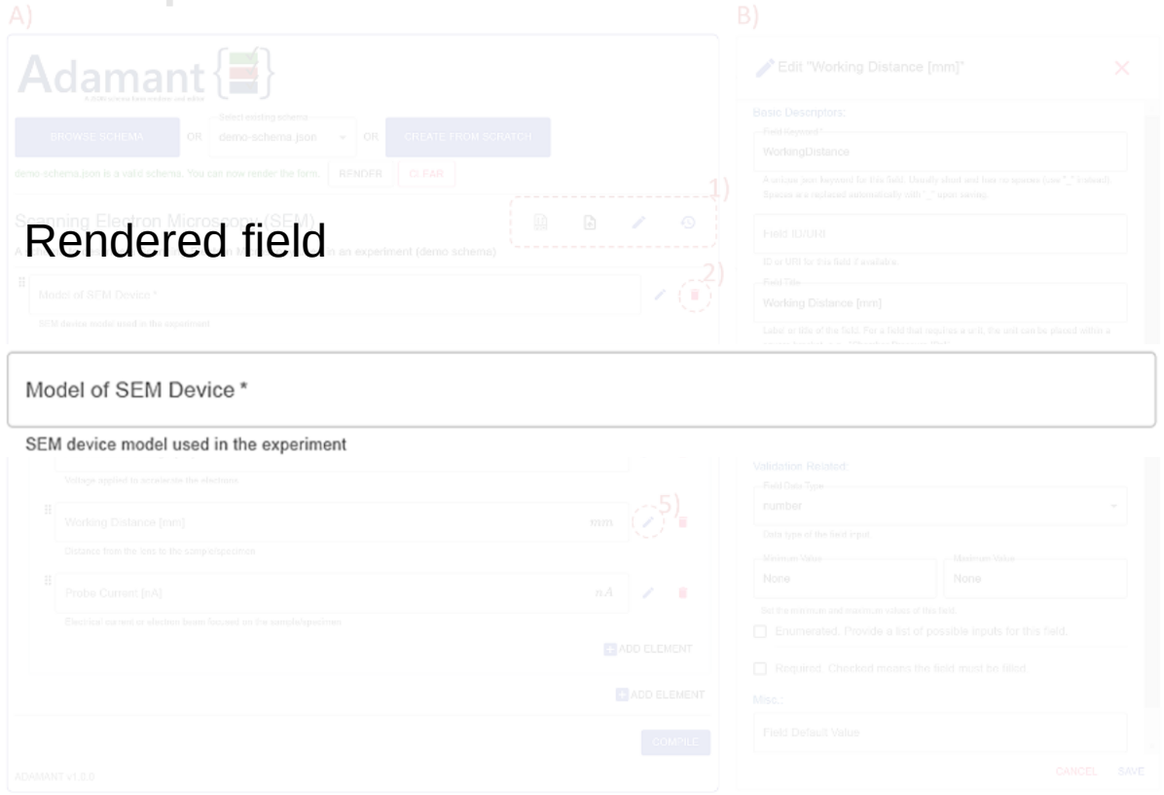
Input: JSON Schema file

Form field specification in the schema

```
"DeviceModel":{
    "title": "Model of SEM Device",
    "description": "SEM device model used in the experiment",
    "type":"string"
},
```

Output: rendered editable web-form

Rendered field

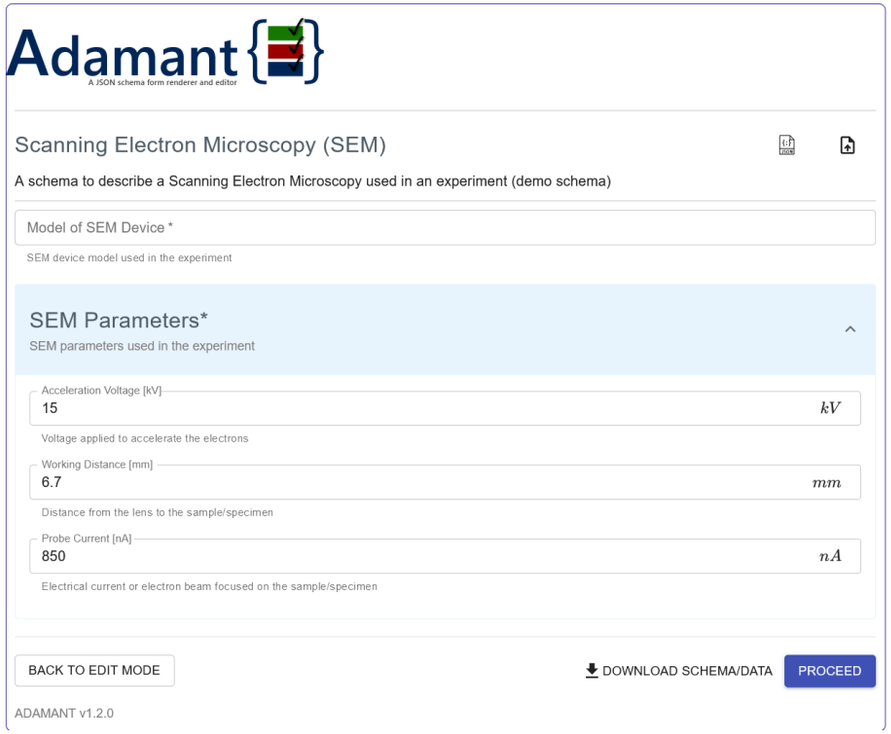


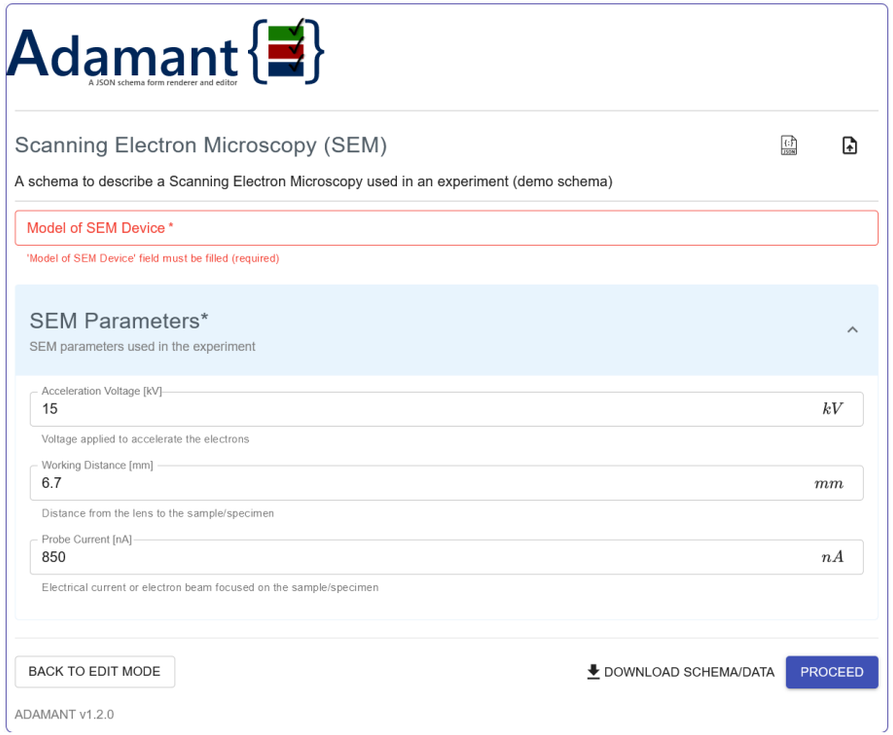
Listing 1. **Example of a draft-4 JSON schema containing typical schema-specific keywords presented in blue with their values presented in black, and field element keywords presented in red.** JSON, JavaScript Object Notation.

Figure 3. **Overview of the Adamant UI with a rendered web-form based on the schema in Listing 1 as an example.** (A) Main corpus of the UI; (1) from left to right: JSON schema viewer, auto-populate form, edit schema description, revert all changes; (2) remove form field; (3) collapse or expand the field container; (4) field drag handle; (5) edit field description and (B) field editing panel (as a pop-up on top of the main UI) triggered by clicking (5) the edit button. UI, user interface; JSON, JavaScript Object Notation.

Adamant
Main features – validation of input data

- Type checking and visual feedback

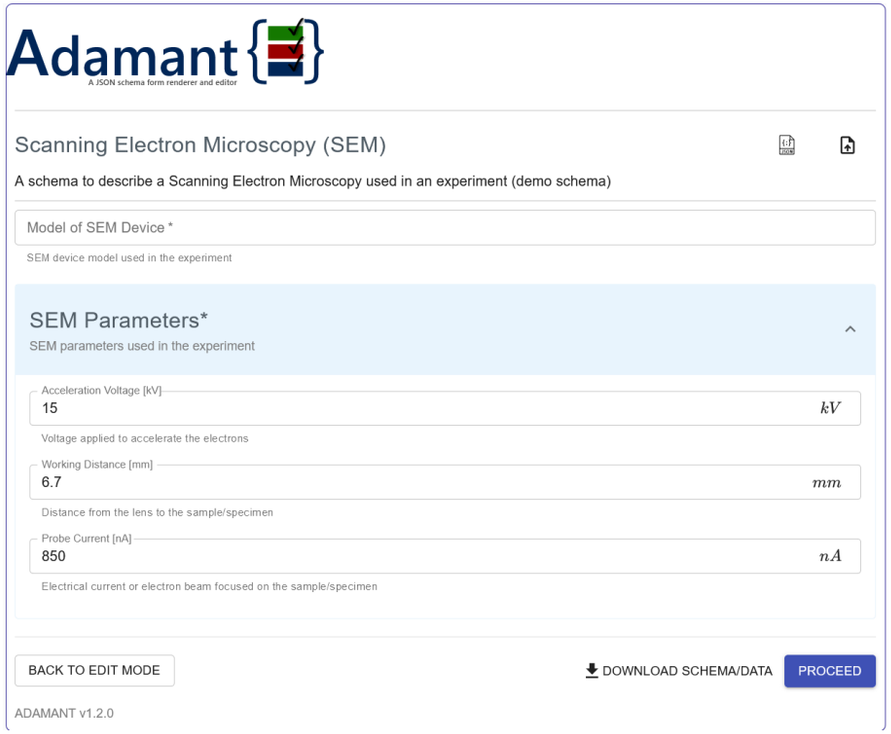


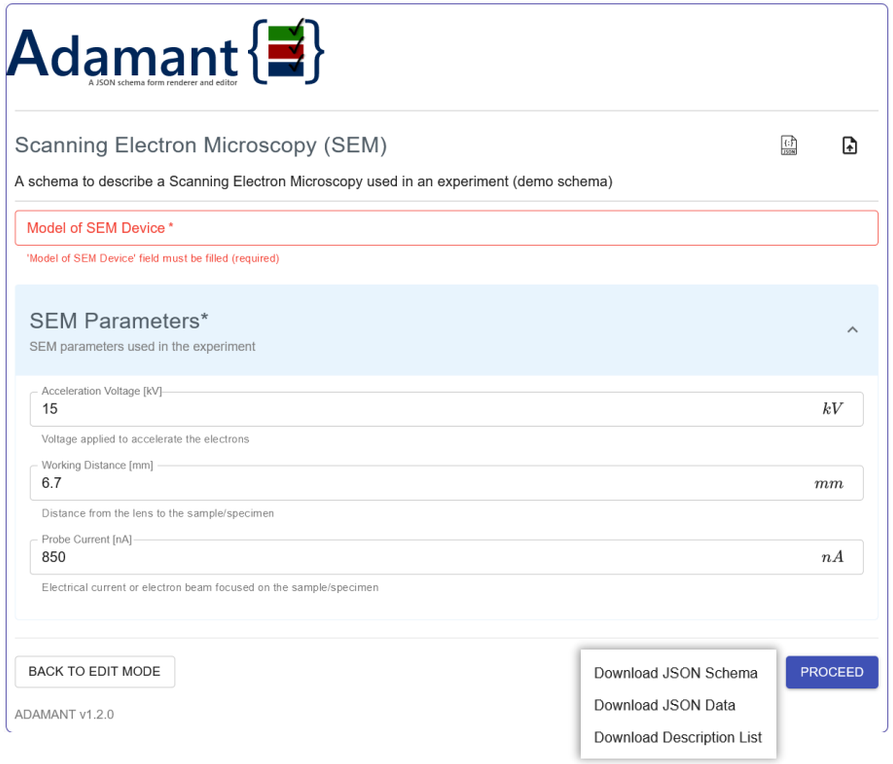


Adamant

Main features – downloadable schema and JSON data

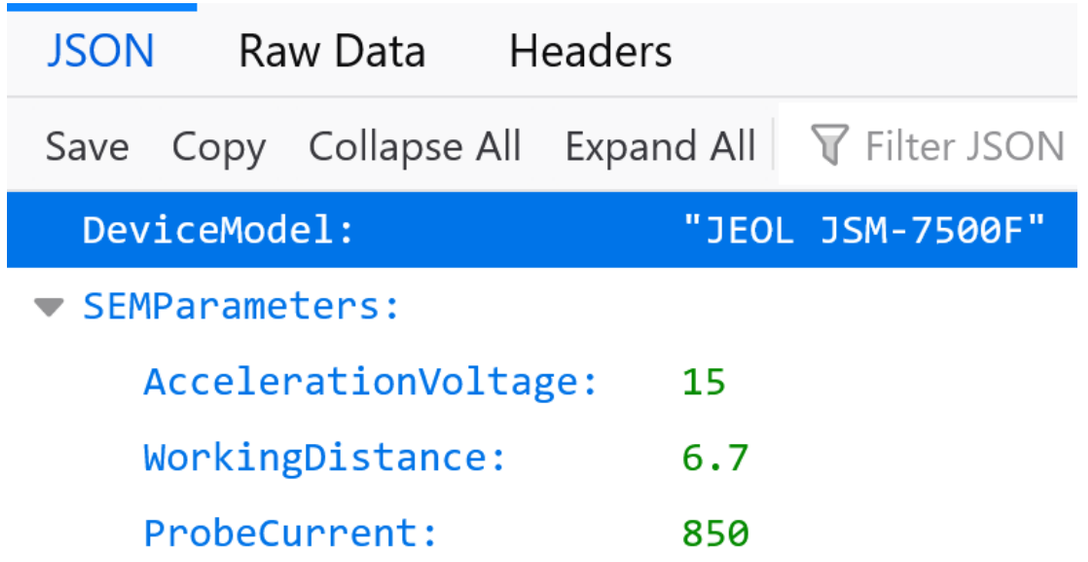
- Type checking and visual feedback

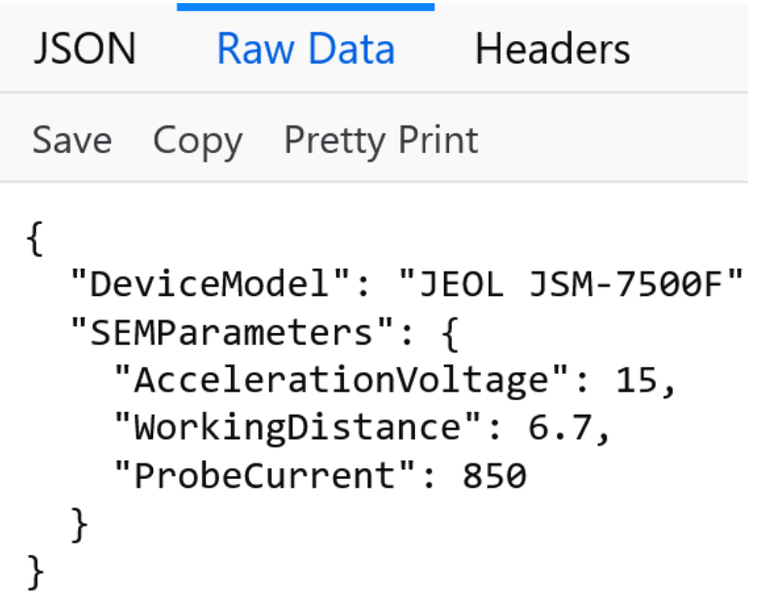




Adamant
Main features – downloadable schema and JSON data

- Downloaded JSON data collected / metadata





Live Demo

Setting up Adamant for development and deployment

GitHub repository: https://github.com/plasma-mds/adamant

Development

Setting up Adamant on a local machine for development:

- `$ git clone https://github.com/csihda/adamant.git` —clone the repository
- `$ cd adamant` —go to adamant project directory
- `adamant$ npm install` —install the dependencies for the client-side
- `adamant$ cd backend` —go to backend directory
- `adamant/backend$ python -m venv venv` —create a python virtual environment
- `adamant/backend$ ./venv/Scripts/activate` —activate the virtual environment
- `adamant/backend$ pip install -r requirements.txt` —install the dependencies for the back-end
- `adamant/backend$ cd ..` —go back to adamant project directory
- `adamant$ yarn start-api` —start the back-end
- `adamant$ yarn start` —on a new terminal, in the adamant project directory, start the front-end

By default, Adamant is accessible at `http://localhost:3000`.

Deployment

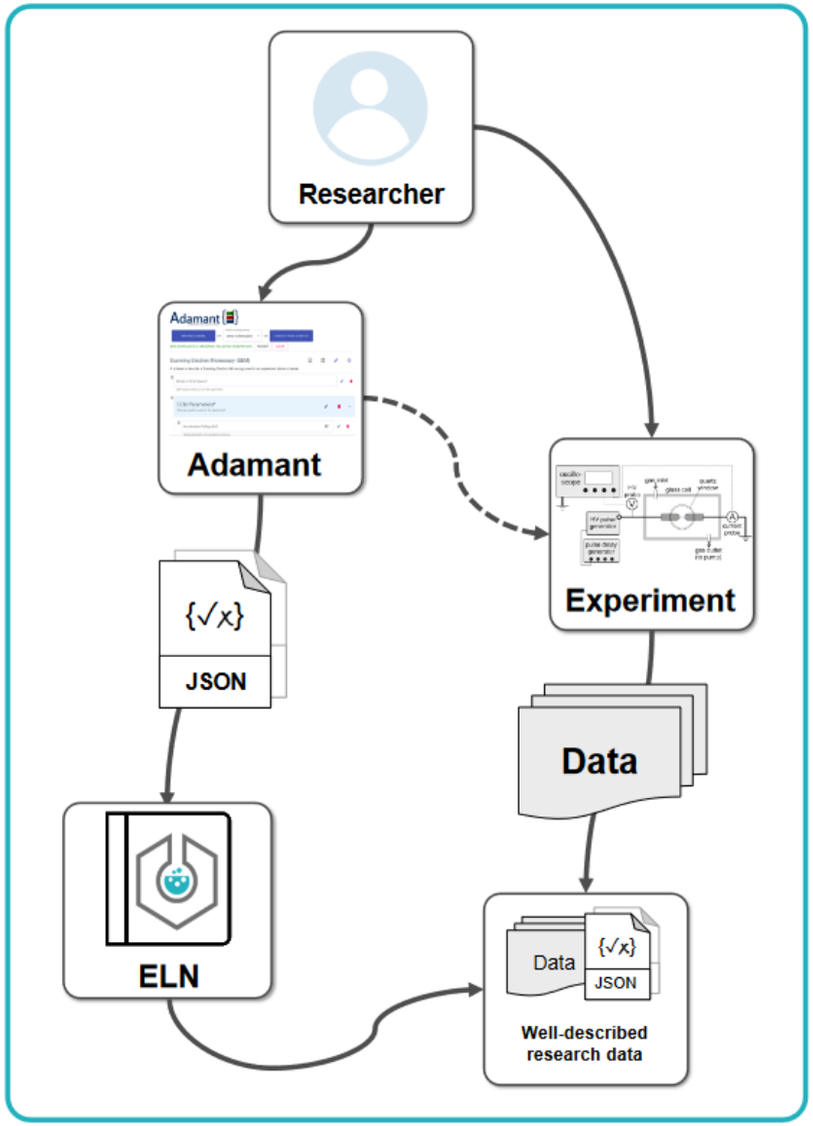
We recommend deploying Adamant with docker-compose, which can be done with ease:

- `$ git clone https://github.com/csihda/adamant.git` —clone the repository
- `$ cd adamant` —go to adamant project directory
- `adamant$ docker-compose build` —build the docker images for both back-end and front-end
- `adamant$ docker-compose up -d` —start both client and server containers, i.e., the whole system

By default, the deployed system can be accessed at `http://localhost:3000`.

Live Demo

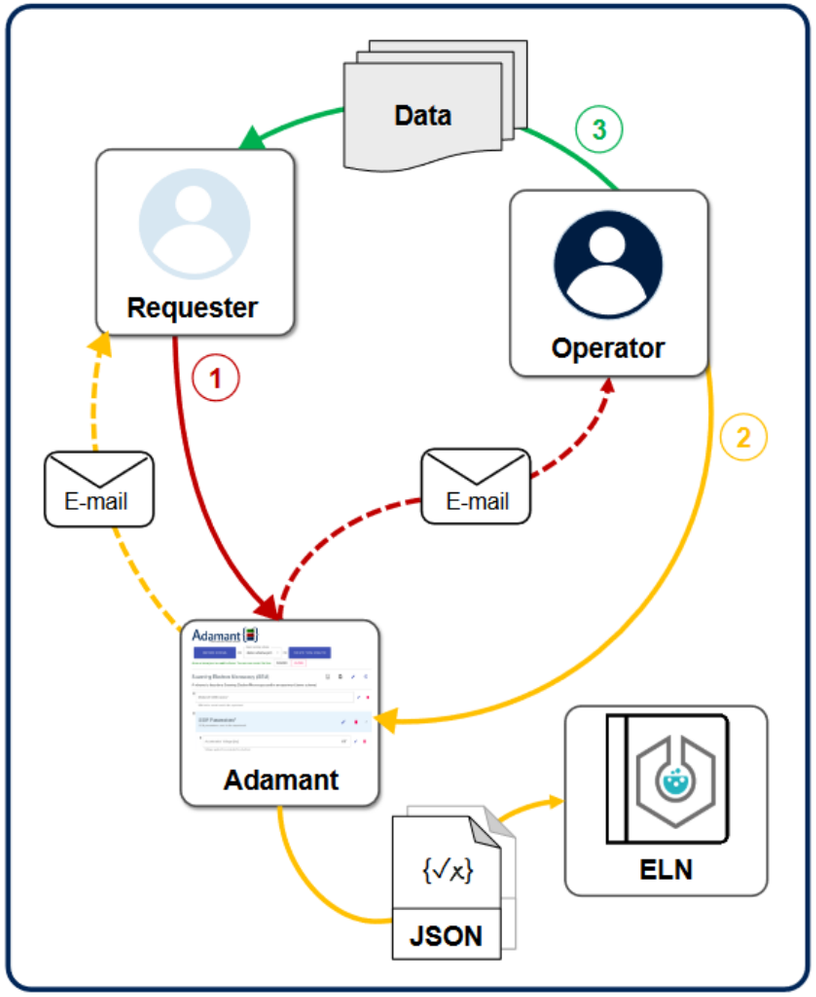
Generic schema-based metadata collection



- Browse, select, or create experiment schema and render the web-form
- Fill in relevant information regarding the experiment
- Submit to store experiment metadata and the used schema in the ELN system
- Experiment metadata can be re-used for dataset publication and other automated processes

Live Demo

Job request of scientific instrument operation



❑ **Motivation**:

Only a handful of researchers can operate a certain advanced scientific instrument. The instrument has to be more accessible to researchers with different skillsets to improve research outputs. A more automated workflow to request an instrument operation is required to make life easier.

❑ **Workflow**:

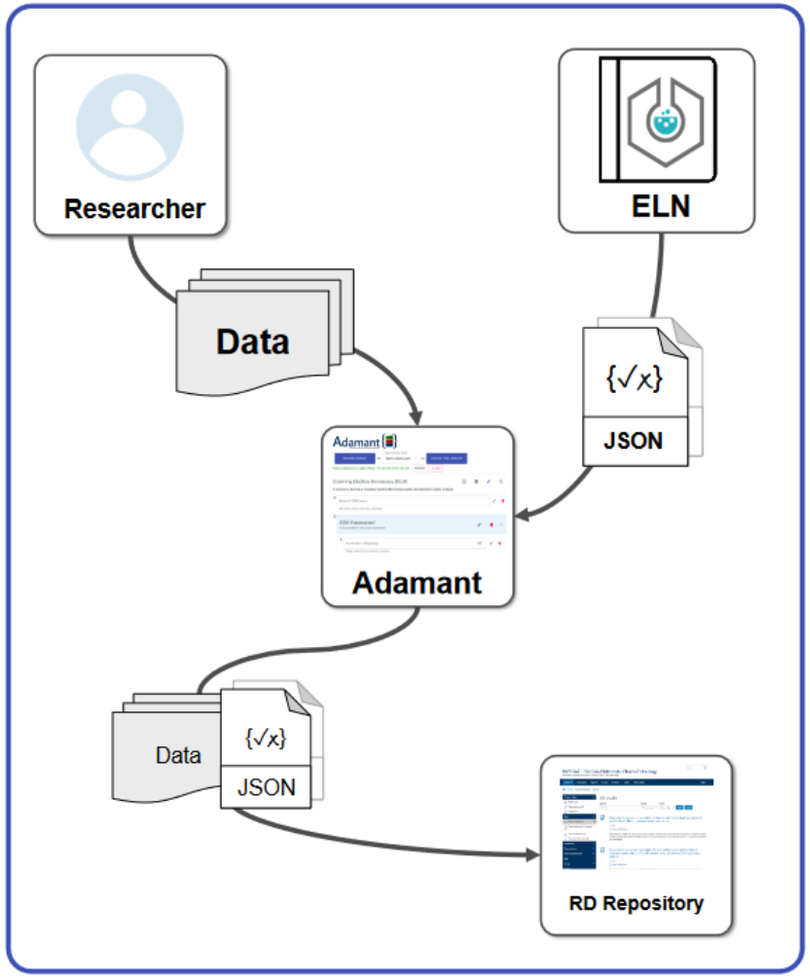
(1) Job request initiation by the requester (researcher who is not an expert on the instrument);

(2) Processing of the request by the operator (expert on the instrument), experiment metadata is stored in ELN;

(3) Handing of research data (results) from operator to requester

Other workflows

Dataset preparation and publication (under preparation)



- Gather and load the data into Adamant
- Re-use experiment metadata stored in the ELN system
  to describe the dataset
- Adjust or extend the metadata as needed
- Submit the prepared dataset to a research data repository system (e.g., https://www.inptdat.de for datasets related to plasma science and technology)

Other workflows

Compilation of input configurations for plasma simulations (under preparation)

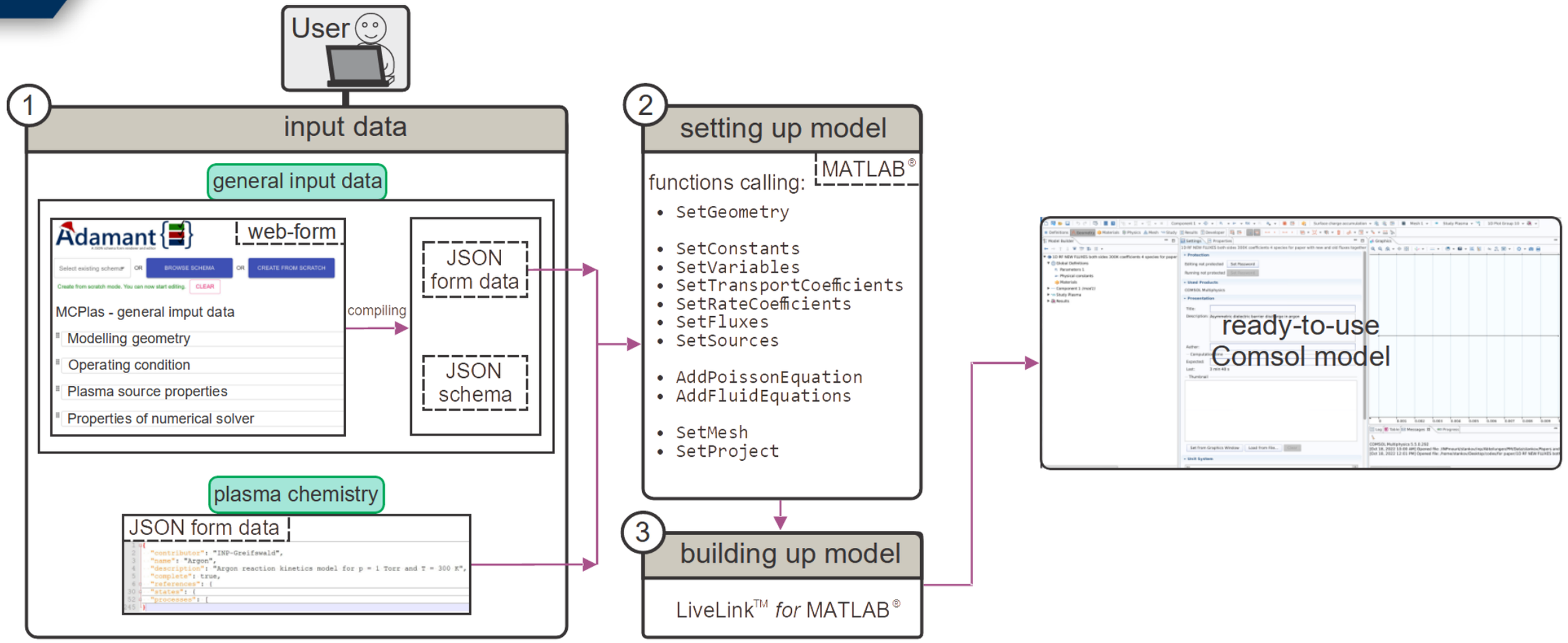
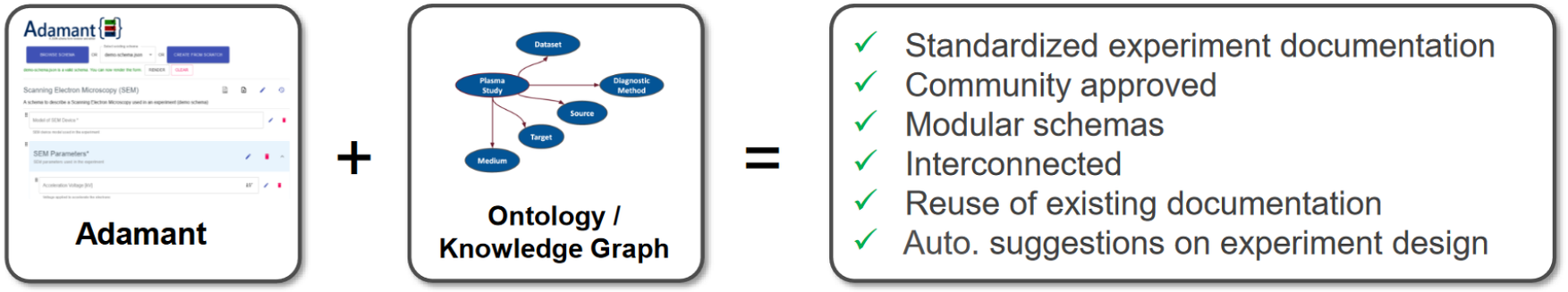


Figure credit: Marjan Stankov

Outlook
Further feature developments

- More JSON Schema keyword implementations
- Better UI and UX
- Ontology / Knowledge graph integration



- Many more!

Acknowledgement

RDM@INP

From left to right:

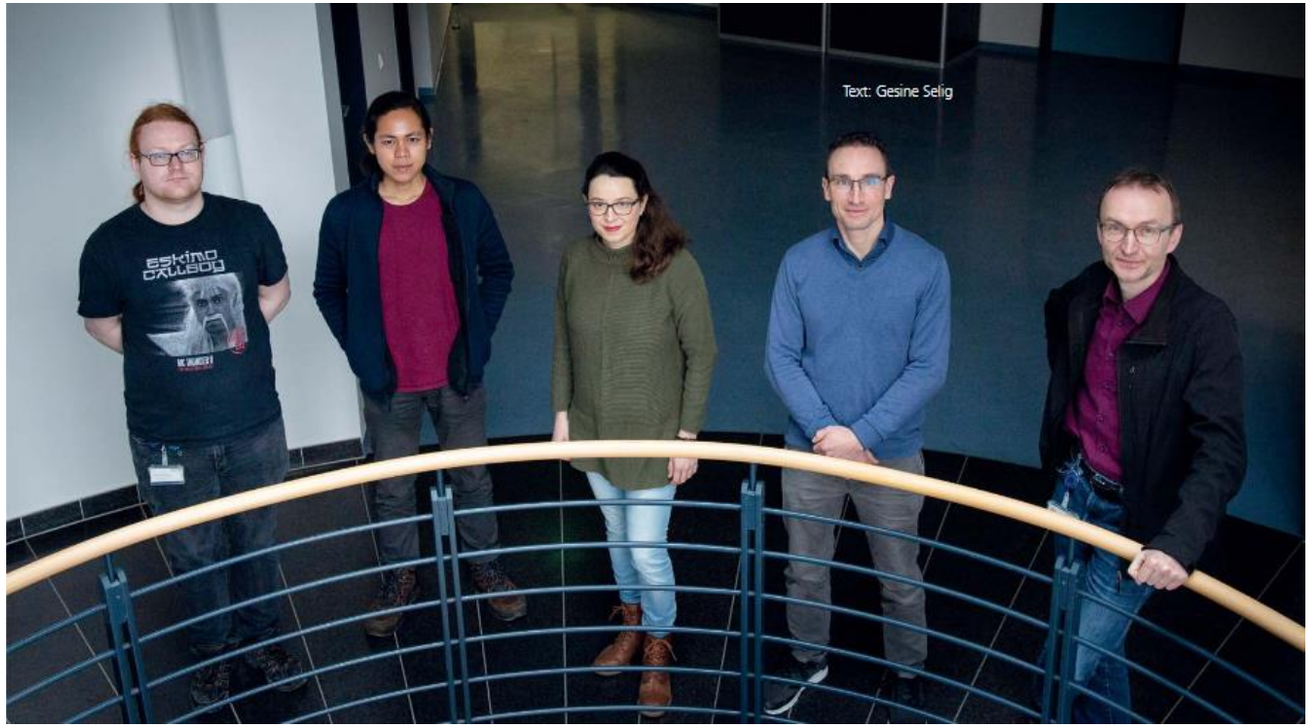
Nick Plathe

Ihda Chaerony Siffa

Laura Vilardell Scholten

Markus M. Becker

Steffen Franke (Alumnus)



BMBF Project:

QPTDat
Quality | Plasma Technology | Data

SPONSORED BY THE
Federal Ministry of Education and Research

The German Federal Ministry of Education and Research (BMBF) funded this work under the grant mark 16QK03A

Contact

Leibniz Institute for Plasma Science and Technology (INP)

Address: Felix-Hausdorff-Str. 2, 17489 Greifswald

Phone: +49 - 3834 - 554 300, Fax: +49 - 3834 - 554 301

E-Mail: welcome@inp-greifswald.de, Web: www.leibniz-inp.de