Patrick Kuckertz (p.kuckertz@fz-juelich.de)
and Working Group around Task Area ELLEN

# An Approach to Increasing the Reuse of Scientific Software

NFDI4Ing Conference 26./27.10.2022

# The Reuse of Software is Central to Research Efficiency & Scientific Exchange

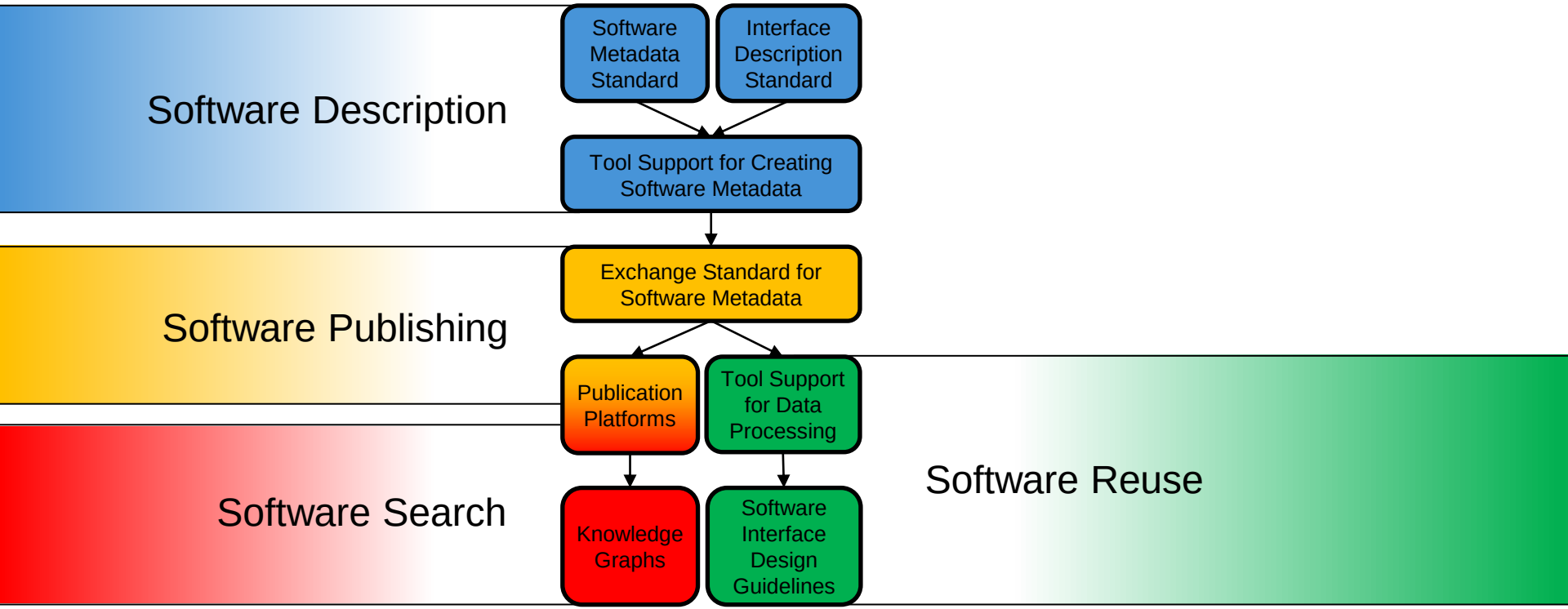
## Why is it important?

- It enables the reproduction and validation of results
- It enables the understanding and comparison of methods and approaches
- It reduces redundant software developments that must be individually documented, maintained and further developed
- It increases software quality, since larger developer and user communities translate into more expertise and larger work forces

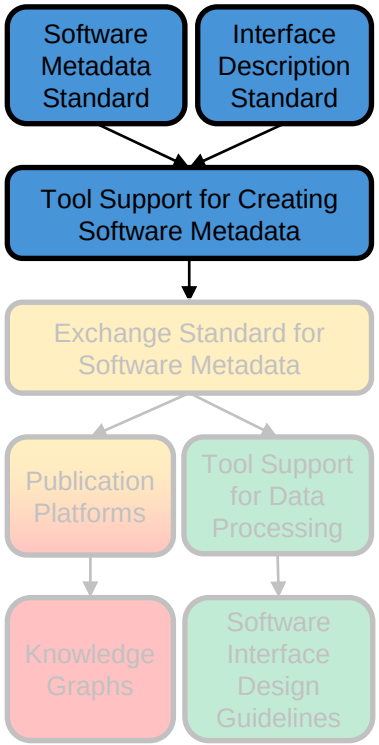
## What are current problems?

- Relevant software is hard to find
- Software is published without license, metadata and or insufficient documentation
  - Documentation standards often do not exist or are not adhered to
  - Documentation is only rarely available in a machine-actionable form
- The provision of this information is often associated with a high level of effort

# Facilitating Four Areas of the Software Life Cycle



# Facilitating Software Description




## What metadata do we need?

- **Detailed interface description** in addition to **general software metadata** such as author, license and date created
- Metadata must be semantically enriched to be **machine-actionable** (that is, using ontology terms instead of free text)

## Building on existing standards

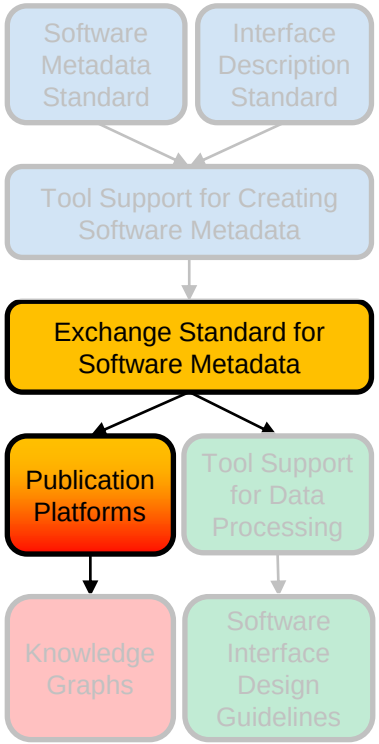
- CodeMeta[1] for general software metadata
- OpenAPI[2] as interface description standard

## The documentation effort must be low!

- **Tool support** for creating metadata
- Metadata annotations are partially **embedded in the source code** supporting code documentation which must be done anyway

[1]https://codemeta.github.io/ [2]https://www.openapis.org/

# Facilitating Software Publishing

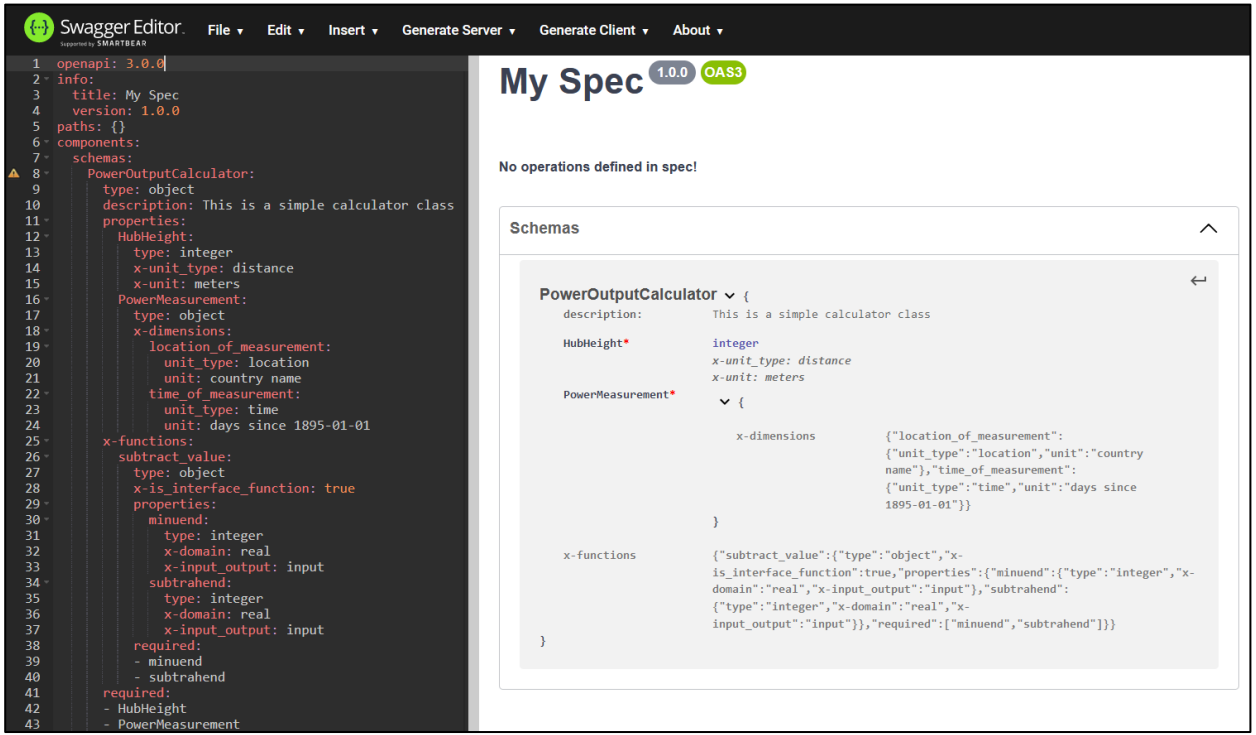



**Describing software with metadata only once enables many applications.**

- Once software is described with metadata, a metadata exchange file is generated
- Upload to many different software publication platforms:
  a. Documentation publication platforms (e.g., ReadTheDocs[1])
  b. Software repository platforms for sharing source code (e.g., GitHub[2])
  c. Software publication platforms for package management (e.g., PyPi[3])
  d. Software registry platforms for finding software (e.g., Open Energy Platform[4])
- Both, using standards and fostering machine-actionability increase the likelihood of widespread dissemination

[1]https://readthedocs.org/ [2]https://github.com/ [3]https://pypi.org/ [4]https://openenergy-platform.org/

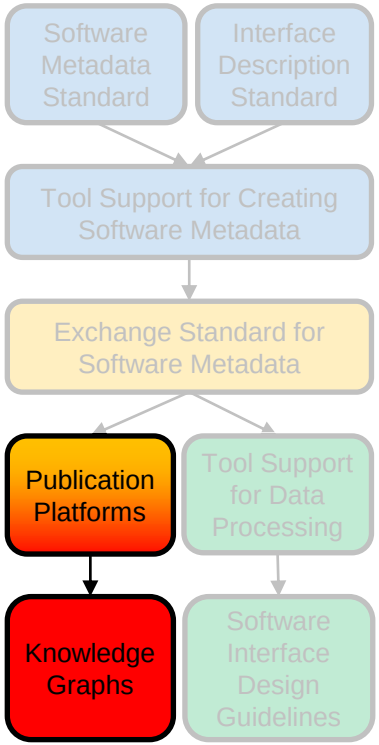
```python
@xattr(xattr={
      "HubHeight" : [("unit_type", "distance"), ("unit", "meters")],
      "PowerMeasurement" : (
         "dimensions",
         {
             "location_of_measurement" : {
                     "unit_type": "location",
                     "unit": "country name"
                 },
             "time_of_measurement" : {
                     "unit_type": "time",
                     "unit": "days since 1895-01-01"
                 }
         }
         ),
      "subtract_value" : [
             ("is_interface_function", True),
             {
                 "minuend" : [
                     ("domain", "real"),
                     ("input_output", "input")
                     ],
                 "subtrahend" : [
                     ("domain", "real"),
                     ("input_output", "input")
                     ],
             }
         ]
      })
class PowerOutputCalculator:
    """This is a simple calculator class"""
    HubHeight: int
    PowerMeasurement: pd.DataFrame

    def subtract_value(self, minuend : int, subtrahend : int):
        difference = minuend - subtrahend
        return difference
```




Interface annotation in source code → Resulting metadata exchange file → Generated documentation website[1]

[1]Utilization of the Swagger Editor: https://editor.swagger.io/

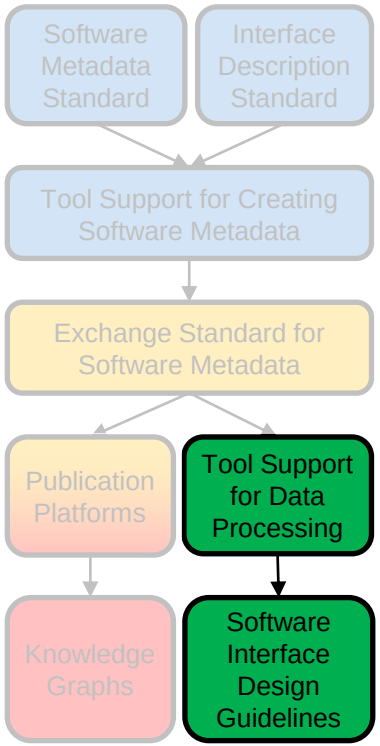
# Facilitating Software Search




- Detailed and machine-actionable software metadata improve software search
  - Different research artifacts are linked together via their metadata[1]
  - Implementation of semantic search methods that can be used across artifacts (e.g., search for existing data or software which produces the requested data)
- Crawling and generating software metadata from publication platforms for software not adhering to this approach

[1]E.g. in the Open Research Knowledge Graph: https://orkg.org/

# Facilitating Software Reuse




- Focus on innovative downstream applications
  - Automated data validation against software interfaces
  - Automated data conversion for matching software interfaces
  - Automated compilation of software and data to software workflows
- Development of interface design guidelines

# Thank you!

Comments and questions are welcome.

Patrick Kuckertz[1]
(p.kuckertz@fz-juelich.de)
Jan Göpfert[1]
Oliver Karras[2]
David Neuroth[1]
Julian Schönau[1]
Rodrigo Pueblas[1]
Stephan Ferenz[3]
Felix Engel[2]
Noah Pflugradt[1]
Jann M. Weinand[1]
Leander Kotzur[1]
Astrid Nieße[3]
Sören Auer[2]
Detlef Stolten[1,4]

1 JÜLICH Forschungszentrum | IEK-3 TECHNO-ECONOMIC SYSTEMS ANALYSIS

2 TIB LEIBNIZ-INFORMATIONSZENTRUM TECHNIK UND NATURWISSENSCHAFTEN UNIVERSITÄTSBIBLIOTHEK

3 CARL VON OSSIETZKY universität OLDENBURG

4 RWTH RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN





NFDi4ing

*https://nfdi4ing.de/archetypes/ellen/*

*Funded and supported by the German Federal Government, the German State Governments, the Joint Science Conference (GWK), and the German Research Foundation (DFG) - project number: 442146713.*
*Supported by the Lower Saxony Ministry of Science and Culture within the Lower Saxony "Vorab" of the Volkswagen Foundation under Grant 11-76251-13-3/19–ZN3488 (ZLE), and by the Center for Digital Innovation (ZDIN).*
*Supported by the Helmholtz Association under the program "Energy System Design".*