Reproducible Science with Python

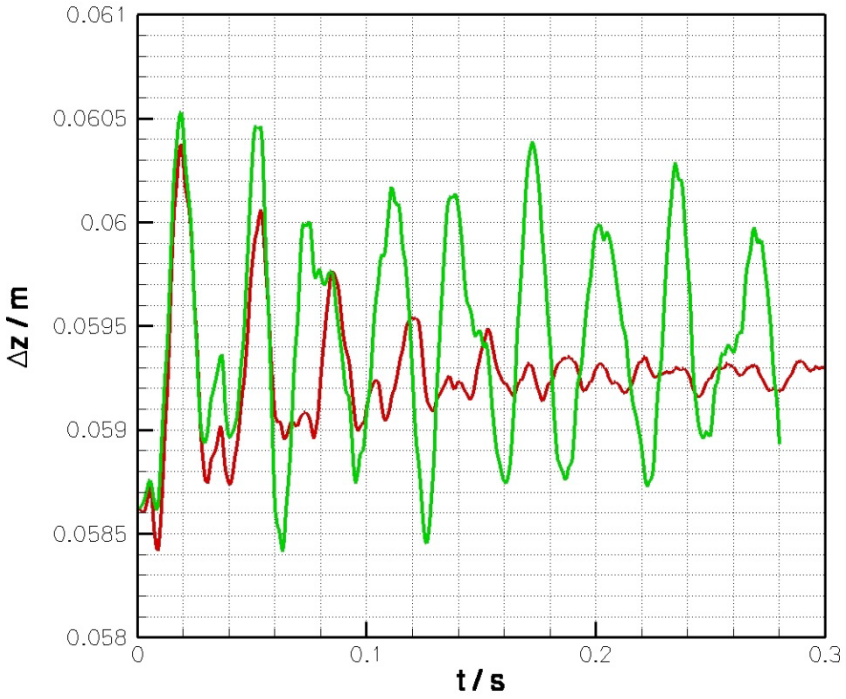
Andreas Schreiber

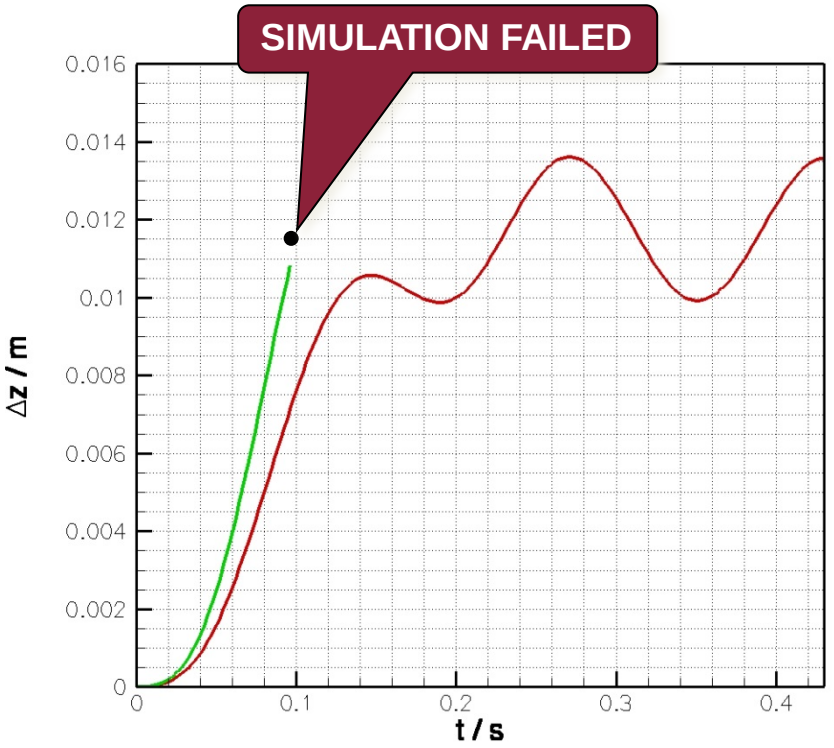
Department for Intelligent and Distributed Systems

German Aerospace Center (DLR), Cologne/Berlin

Science

Simulations, experiments, data analytics, …

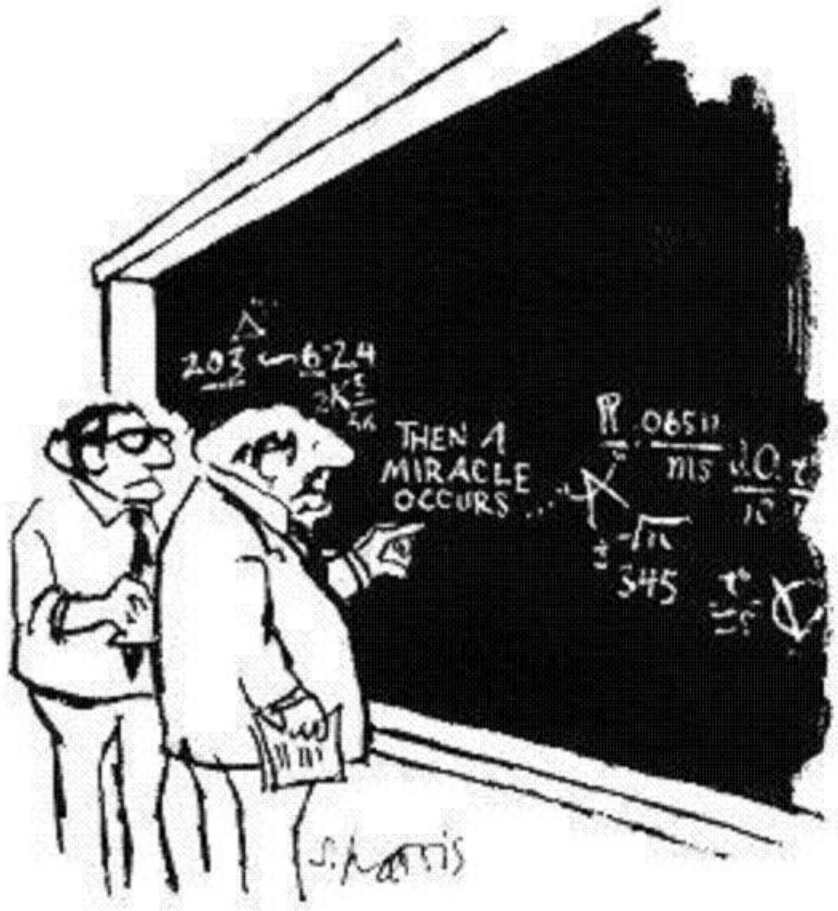




Reproducible Science

Reproducing results relies on

- Open Source codes
- Code reviews
- Code repositories
- Publications with code
- Computational environment captured (Docker etc.)
- Workflows
- Open Data formats
- Data management
- (Electronics) laboratory notebooks
- Provenance



"I think you should be more explicit here in step two."

Provenance

Provenance is information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness.

PROV W3C Working Group
https://www.w3.org/TR/prov-overview

PROV

W3C Provenance Working Group:
`https://www.w3.org/2011/prov`

PROV

- The goal of PROV is *to enable the wide publication and interchange of provenance* on the Web and other information systems
- PROV enables one *to represent and interchange provenance information using widely available formats* such as RDF and XML

Overview of PROV

Key Concepts

Entities

- Physical, digital, conceptual, or other kinds of things
- For example, documents, web sites, graphics, or data sets

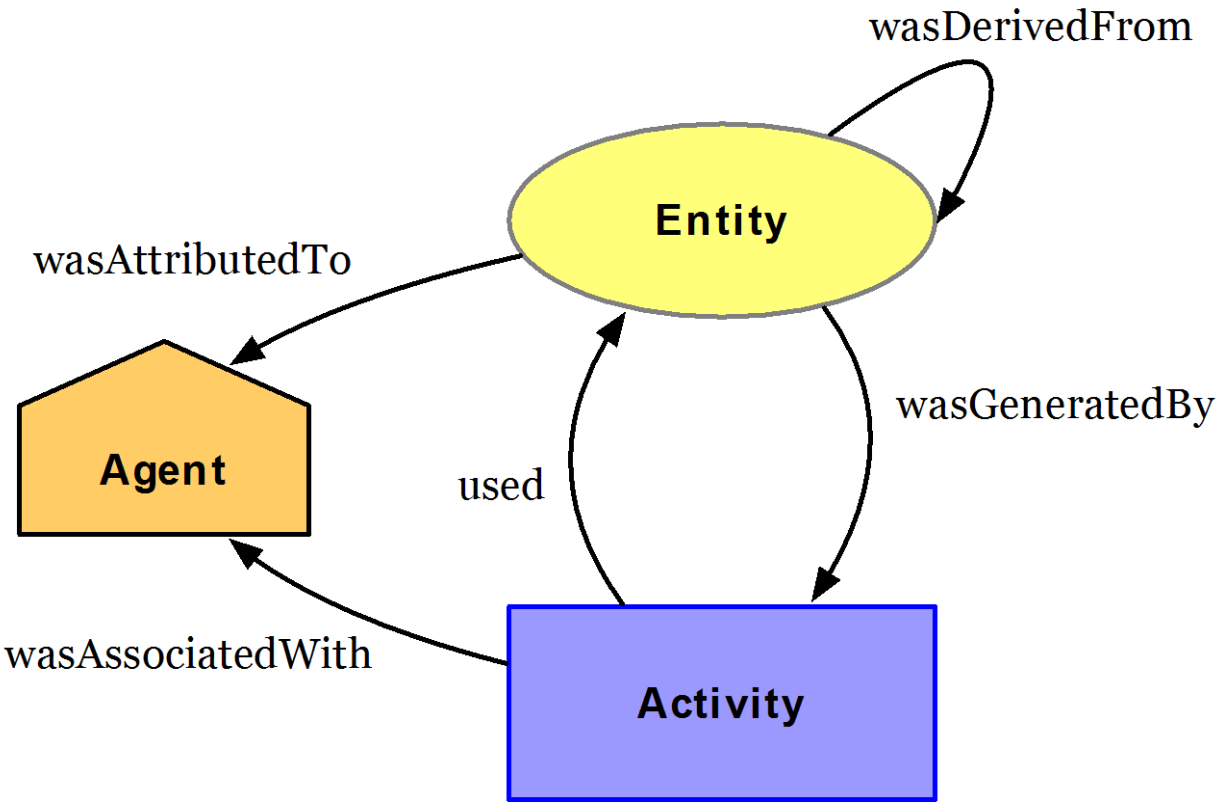
Activities

- Activities *generate* new entities or make *use* of existing entities
- Activities could be actions or processes

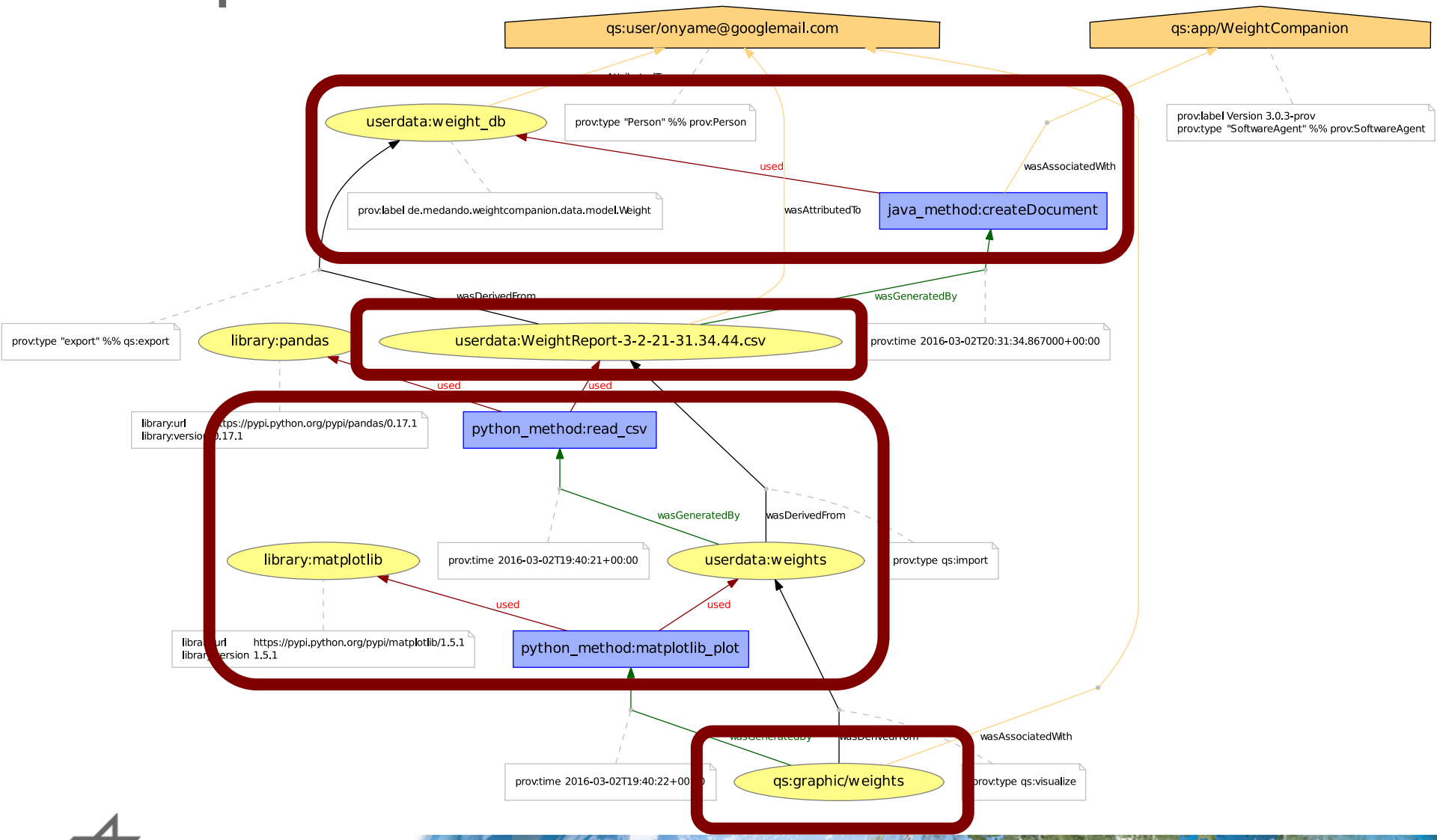
Agents

- Agents takes a role in an activity and have the responsibility for the activity
- For example, persons, pieces of software, or organizations

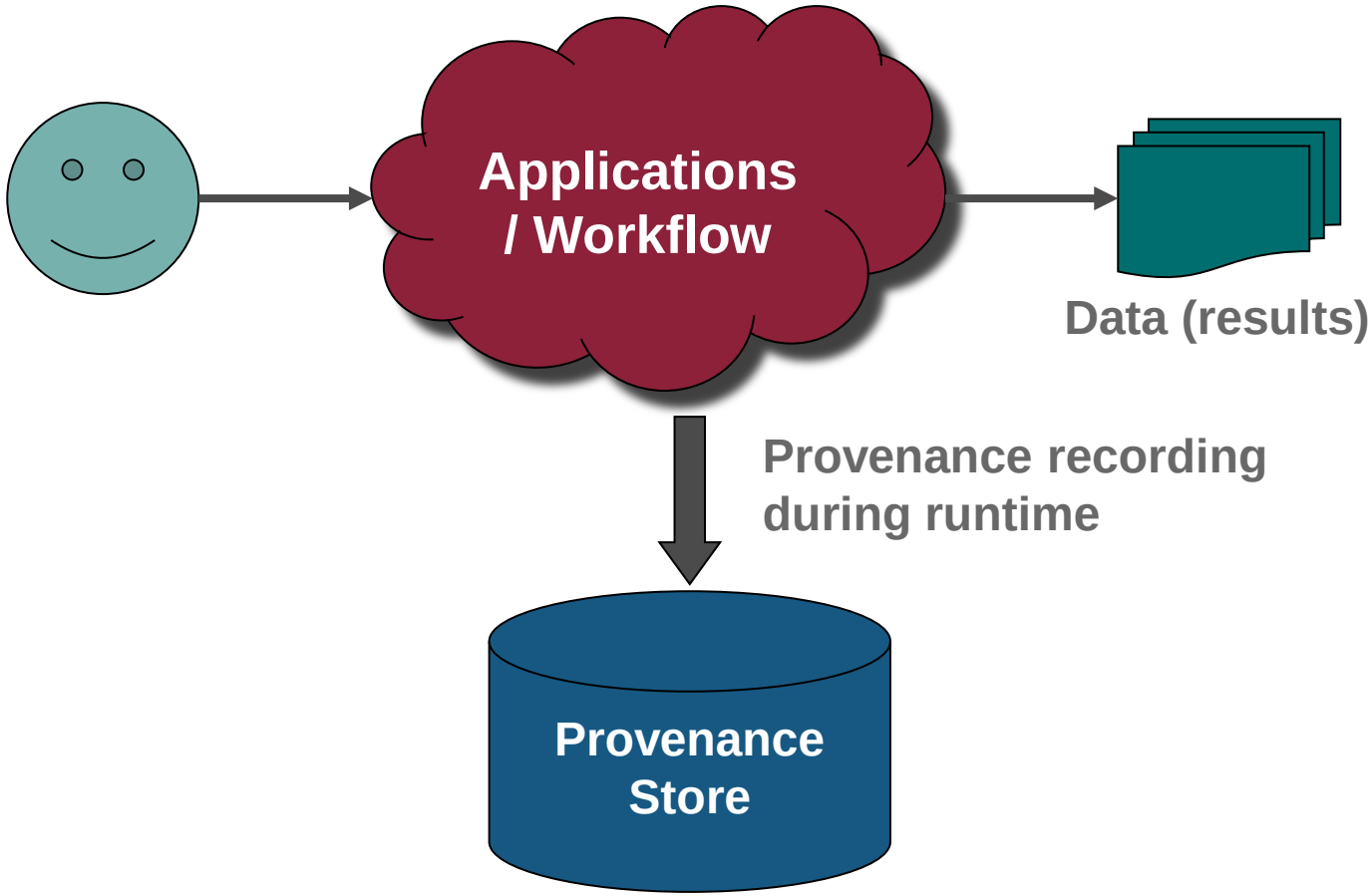
PROV Data Model



Example Provenance



Storing Provenance



Storing and Retrieving Provenance

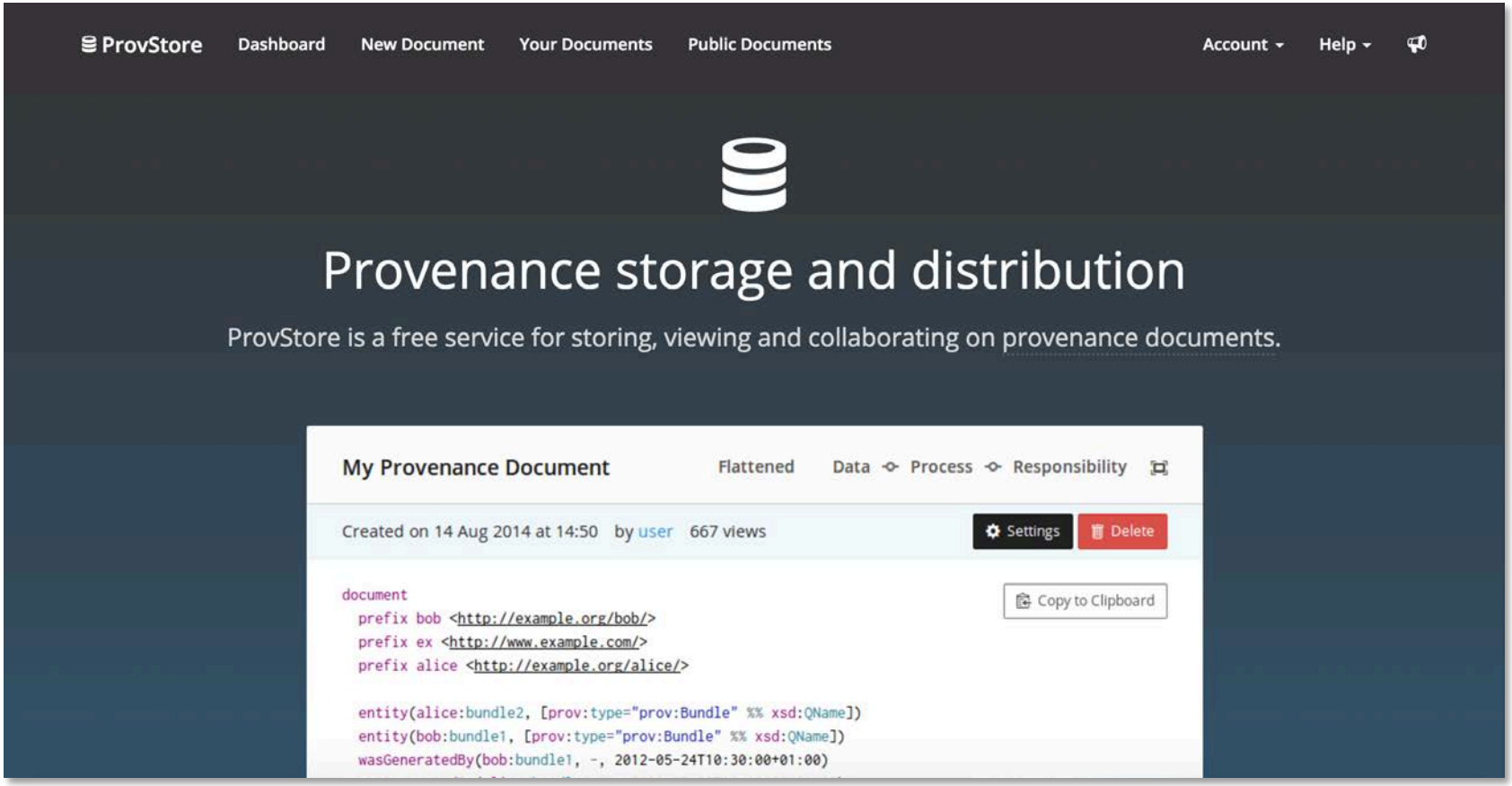
Some Storage Technologies

- Relational databases and SQL
- XML and Xpath
- RDF and SPARQL
- Graph databases and Gremlin/Cypher

Services

- REST APIs
- ProvStore (University of Southampton)

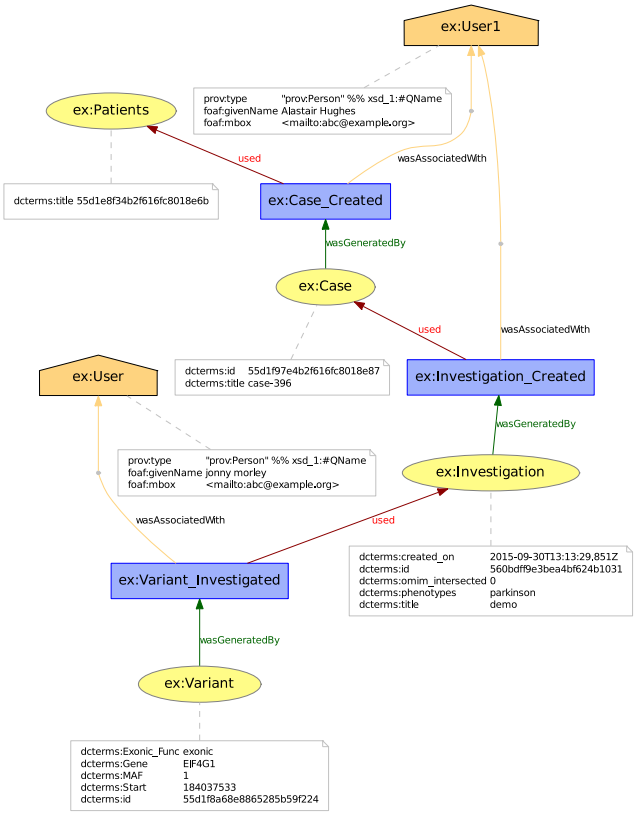
ProvStore

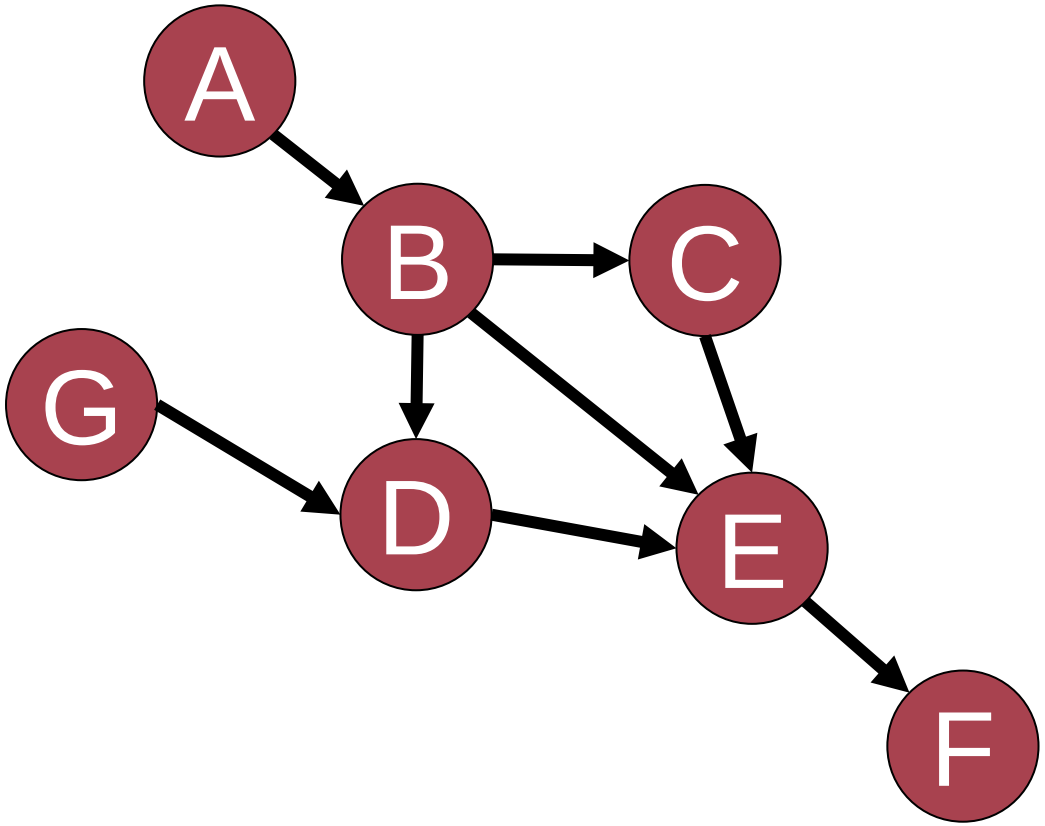


https://provenance.ecs.soton.ac.uk/store/

Graphs

Provenance is a directed acyclic graph (DAG)





Graph Databases

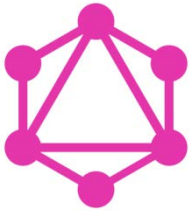
Naturally, graph databases are a good technology for storing (Provenance) graphs

Many graph databases are available

- Neo4J
- Titan
- ArangoDB
- ...

Query languages

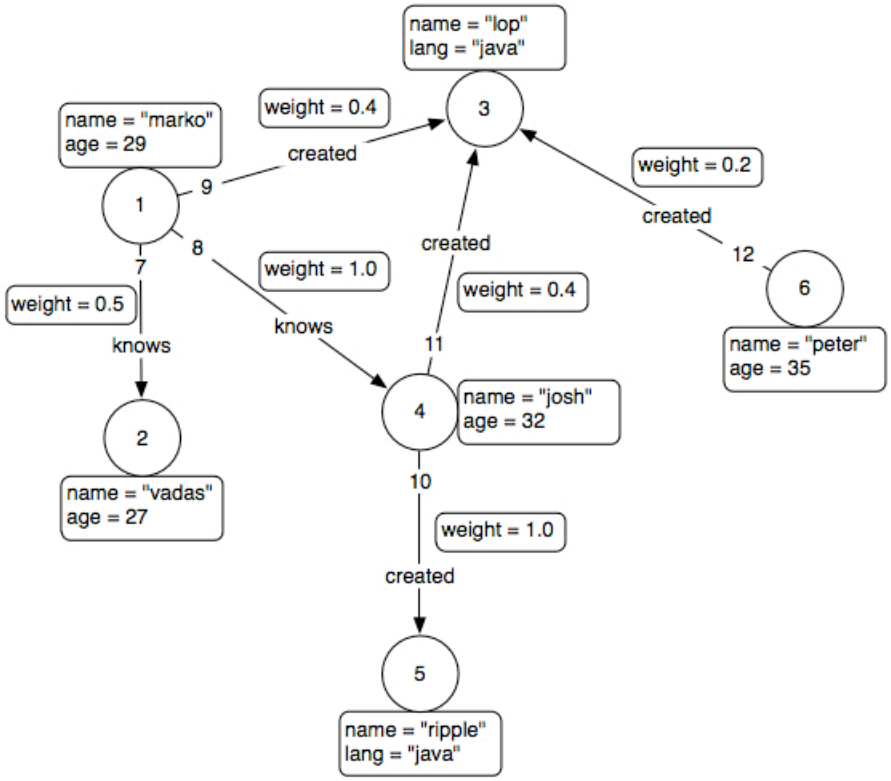
- Cypher
- Gremlin (TinkerPop)
- GraphQL



Neo4j

- Open-Source
- Implemented in Java
- Stores *property graphs* (key-value-based, directed)

`http://neo4j.com`



Gathering Provenance

Depends on your application (tools, languages, etc.)

Libraries for Python

- prov
- provneo4j
- NoWorkflow
- …

Tools

- Git2PROV
- Prov-sty for LaTeX
- …

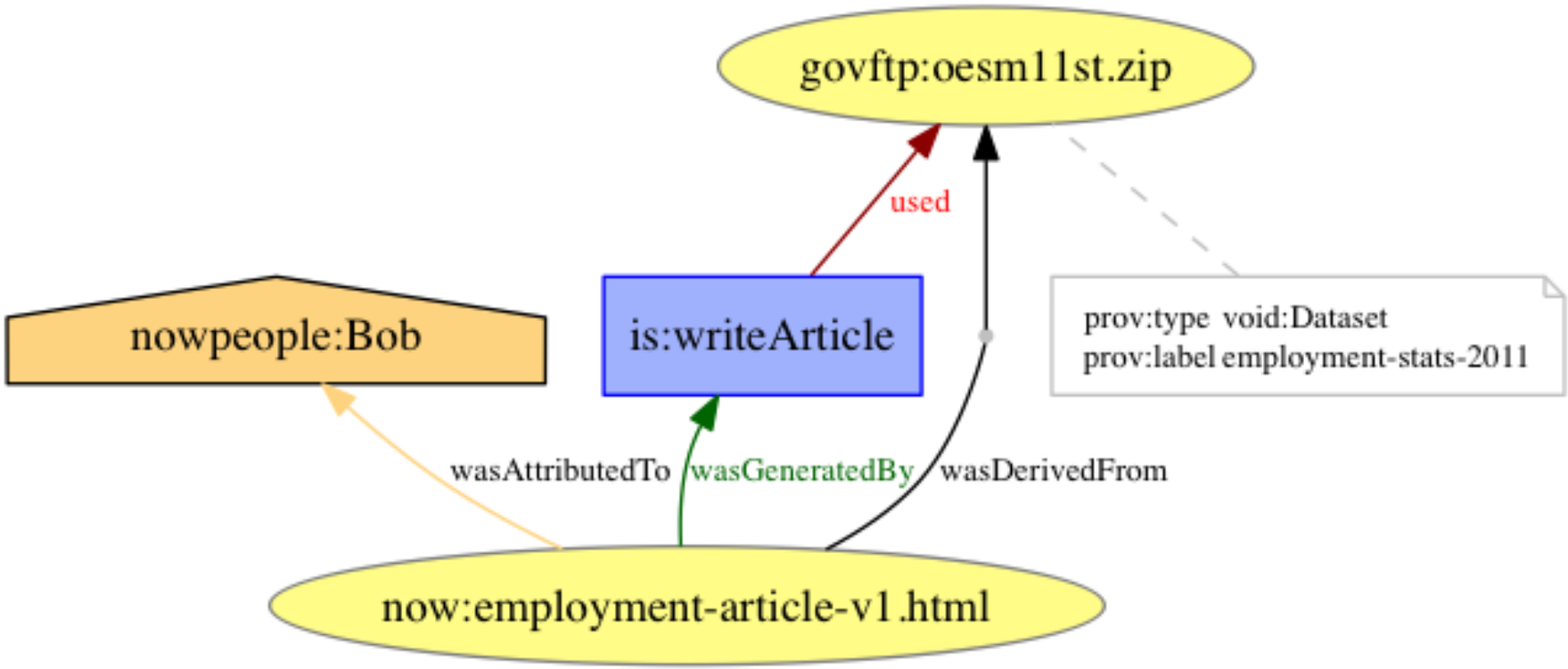
Python Library prov
https://github.com/trungdong/prov

```
from prov.model import ProvDocument
# Create a new provenance document
d1 = ProvDocument()
# Entity: now:employment-article-v1.html
e1 = d1.entity('now:employment-article-v1.html')
# Agent: nowpeople:Bob
d1.agent('nowpeople:Bob')
# Attributing the article to the agent
d1.wasAttributedTo(e1, 'nowpeople:Bob')
d1.entity('govftp:oesm11st.zip',
          {'prov:label': 'employment-stats-2011',
           'prov:type': 'void:Dataset'})
d1.wasDerivedFrom('now:employment-article-v1.html',
                  'govftp:oesm11st.zip')
# Adding an activity
d1.activity('is:writeArticle')
d1.used('is:writeArticle', 'govftp:oesm11st.zip')
d1.wasGeneratedBy('now:employment-article-v1.html', 'is:writeArticle')
```

DLR

Python Library prov

`https://github.com/trungdong/prov`



Example

http://localhost:8888/notebooks/WeightCompanion_PROV.ipynb

provneo4j – Storing PROV Documents in Neo4j

https://github.com/DLR-SC/provneo4j

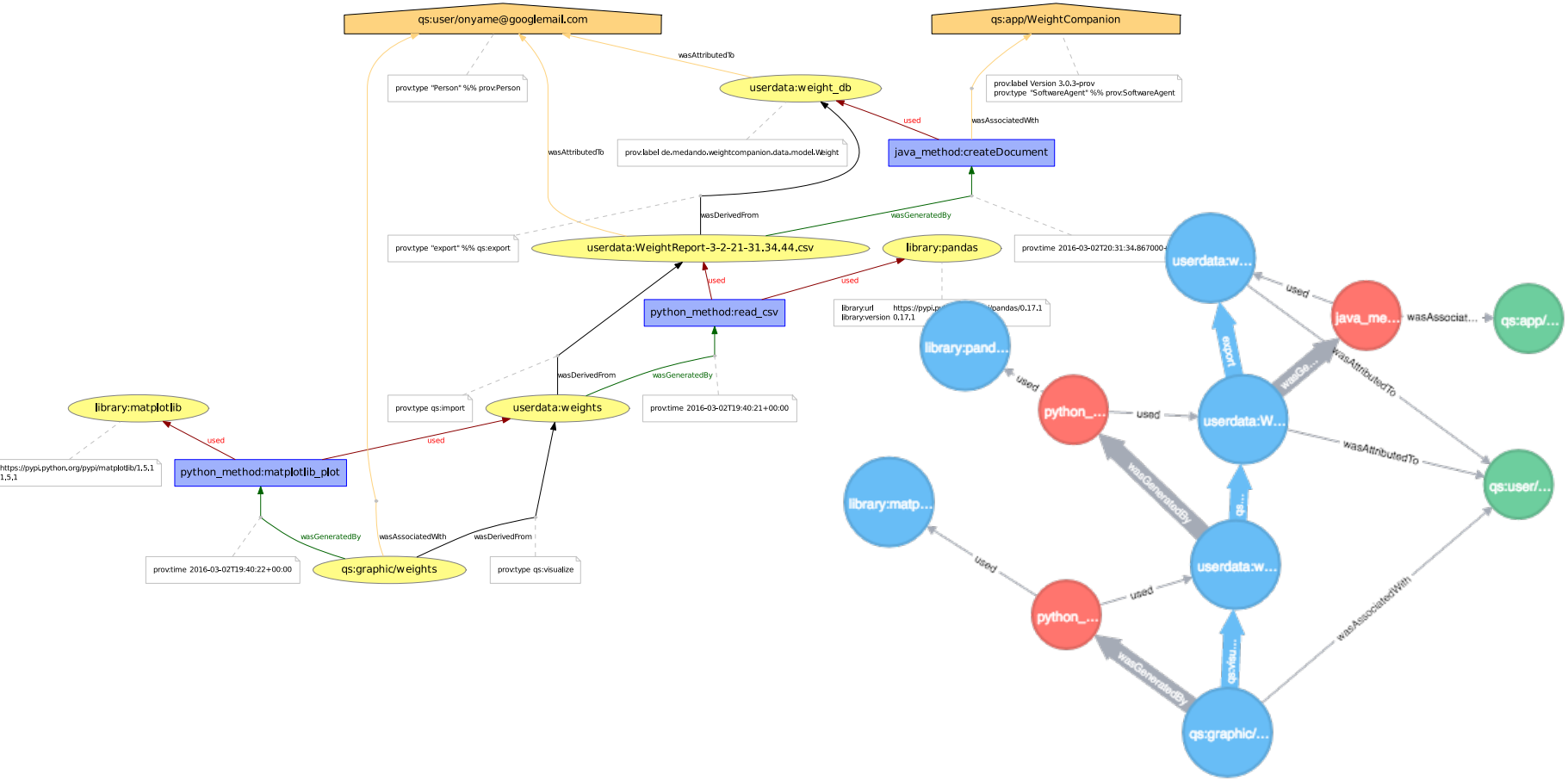
```
import provneo4j.api

provneo4j_api = provneo4j.api.Api(
        base_url="http://localhost:7474/db/data",
        username="neo4j", password="python")

provneo4j_api.document.create(prov_doc, name="MyProv")
```

provneo4j – Storing PROV Documents in Neo4j

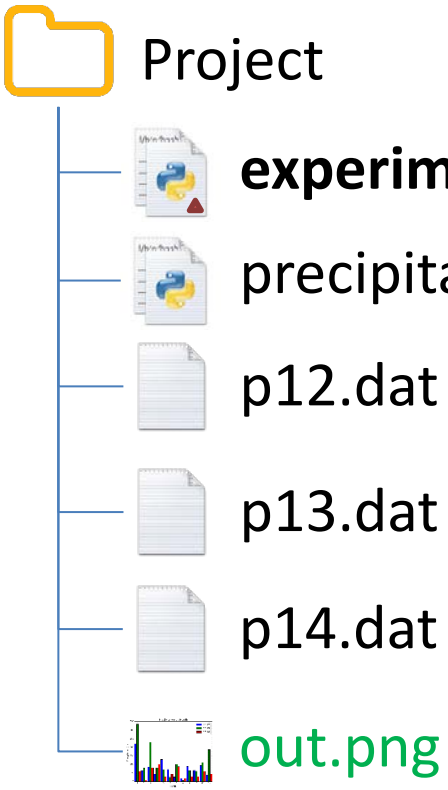
https://github.com/DLR-SC/provneo4j

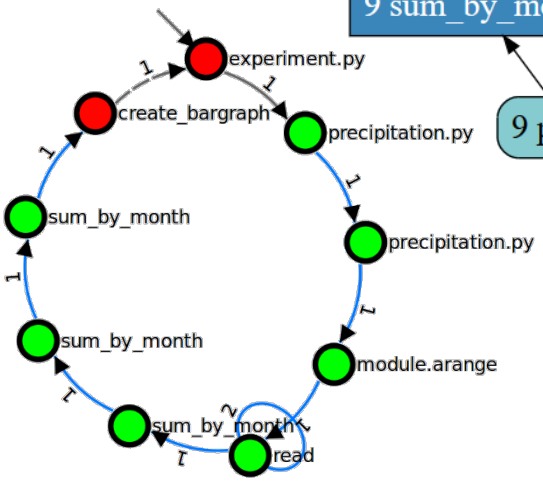


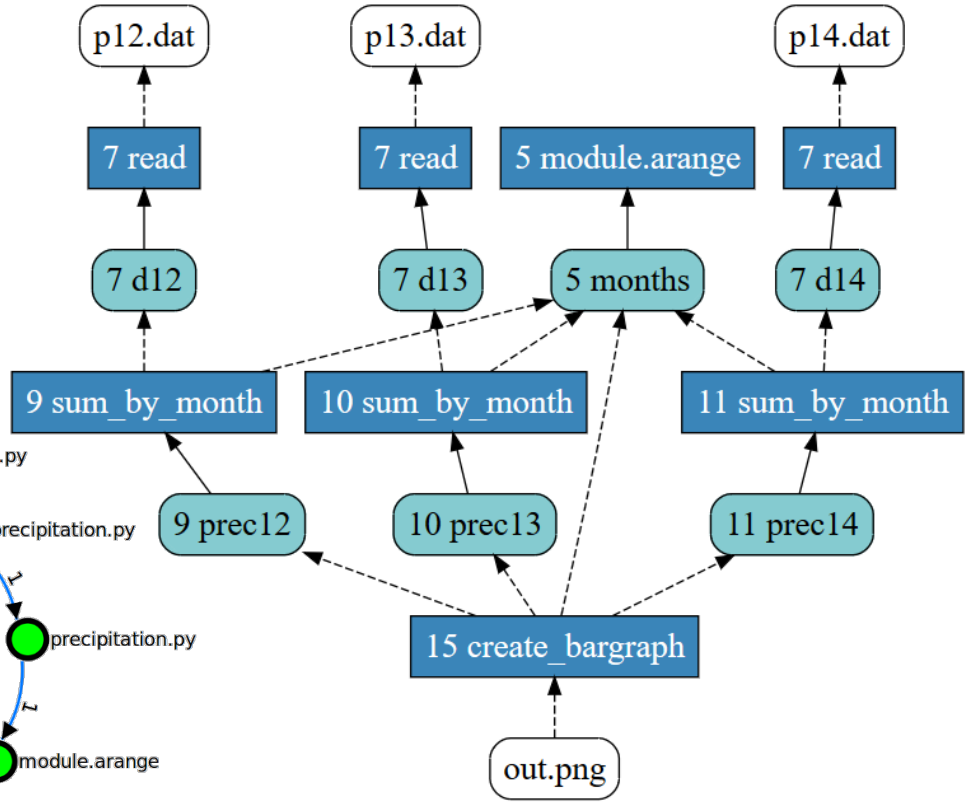
NoWorkflow – Provenance of Scripts

https://github.com/gems-uff/noworkflow

```
$ now run -e Tracker experiment.py
```

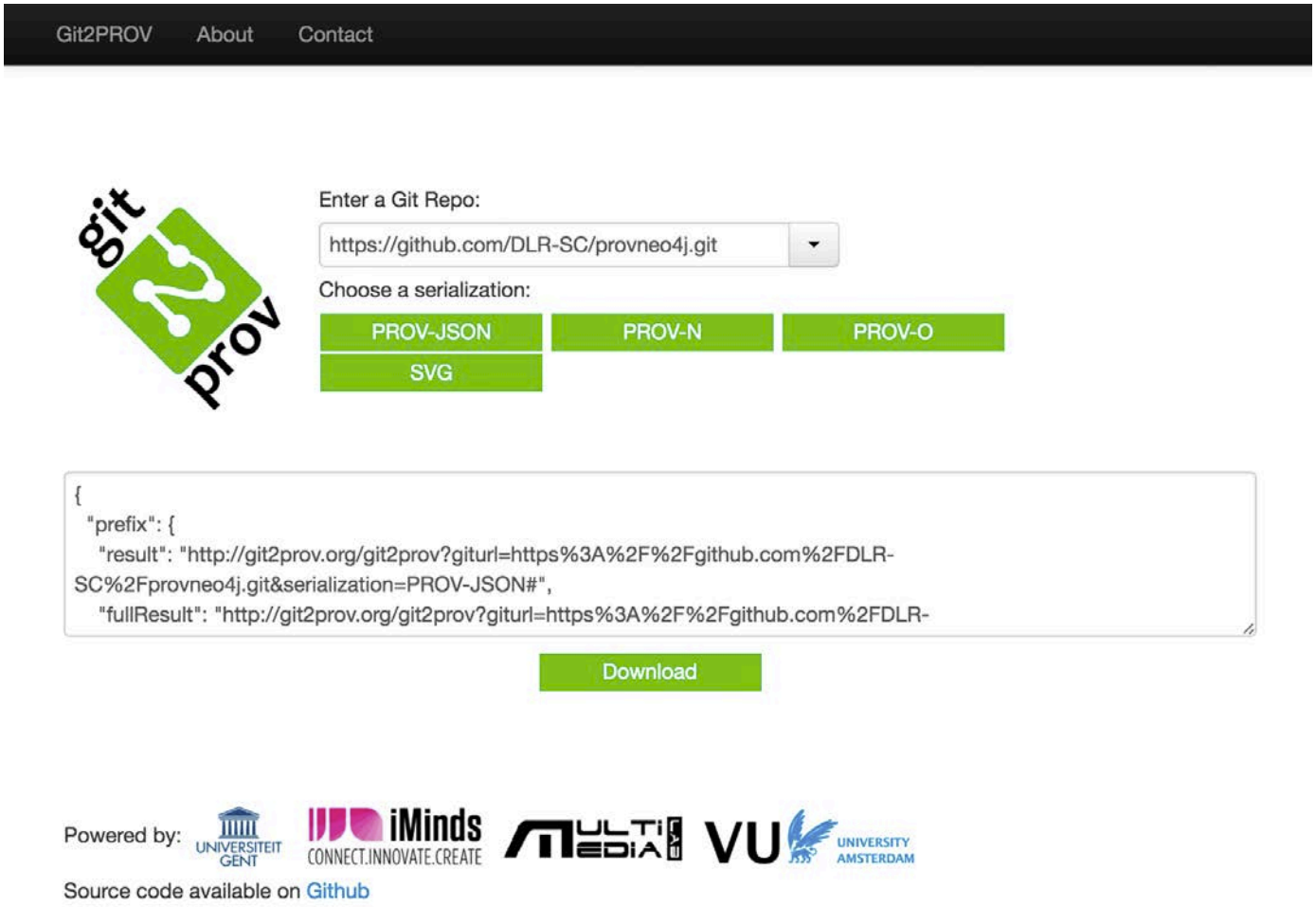






Git2PROV

http://git2prov.org



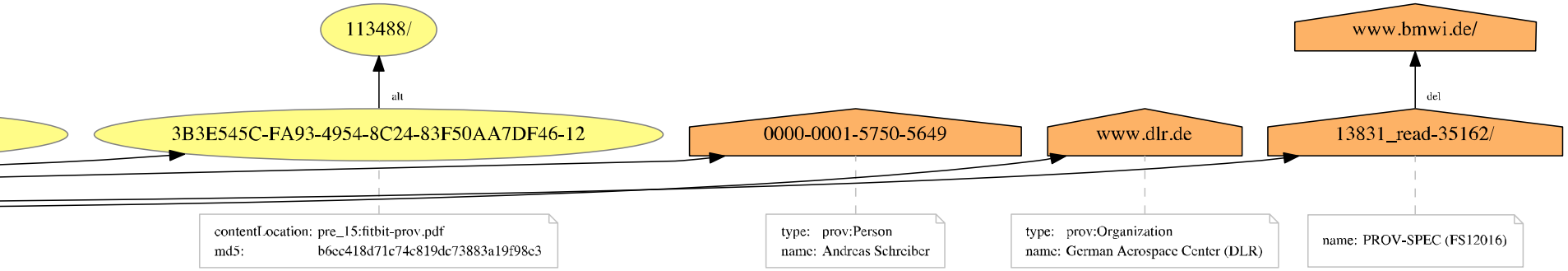

result:user-Michael-Meinel
provlabel Michael Meinel
result:file-README-md_commit-fd02c68c3d2a685744dc1a7316b51026a801dc7f_gen
result:file-src-F2x-template-_pinvoke-cs-t_commit-57225209c2a8170ddaeb55c3a82c99a2904218e6_commit-682813465b1216e4badd8ddae9080c9bacf546f0_
wasAssociatedWith
result:file-src-F2x-grammar-__init__-py
result:file-src-F2x-__init__-py
provrole author, committer
result:file-src-F2x-template-__init__-py
wasDerivedFrom
wasAttributedTo
specializationOf
provtype authorship
result:file-src-F2x-grammar-__init__-py-commit-ebf20319d30acaaa847a6745b9d5b3ca9edf42d73
provlabel src/F2x/grammar/__init__.py
provlabel src/F2x/__init__.py
provlabel src/F2x/template/__init__.py
result:file-src-F2x-grammar-fortran-g_commit-e7e122a25ab398dc4fa3dc64bb67b1db51f644e8
provtime 2016-07-18T13:43:07+00:00
provtime 2015-12-31T10:17:07+00:00
result:file-README-md_commit-0777b106655bc7d6ff4213e339dc48e718a8eff9_gen
result:file-README-md_commit-e7e122a25ab398dc4fa3dc64bb67b1db51f644e8_commit-0777b106655bc7d6ff4213e339dc48e718a8eff9_use
result:file-src-F2x-main-py_commit-7055d5f34
wasGeneratedBy
result:file-src-F2x-main-py_commit-ebf20319d30acaaa847a6745b9d5b3ca9edf42d73
provtime 2016-04-21T09:19:55+00:00
wasEndedBy
provtime 2016-07-20T13:41:26+00:00
result:file-src-F2x-template-_bindc-f90-t_commit-0777b106655bc7d6ff4213e339dc48e718a8eff9_commit-7b8cfbb73066370feef91d40bfa397fbd6f745f4_use
result:file-src-F2x-template-_bindc-f90-t_commit-7b8cfbb73066370feef91d40bfa397fbd6f745f4_gen
provlabel Finished first version of C#-Template
result:file-src-F2x-main-py_commit-baa85471e0ab8b5cb42461f28463c3b7dd8c4901
result:file-src-F2x-template-_bindc-f90-t_commit-7b8cfbb73066370feef91d40bfa397fbd6f745f4
result:file-src-F2x-main-py_commit-e993670727f3d3a3739e08259276d1810429e5ef_gen
provtime 2016-05-03T12:39:22+00:00
result:file-src-F2x-grammar-fortran-g_commit-7b8cfbb73066370feef91d40bfa397fbd6f745f4_commit-b0706322cce9f5b3bc14abaa77
result:file-src-F2x-main-py_commit-e993670727f3d3a3739e08259276d1810429e5ef

prov-sty for LaTeX

https://github.com/prov-suite/prov-sty

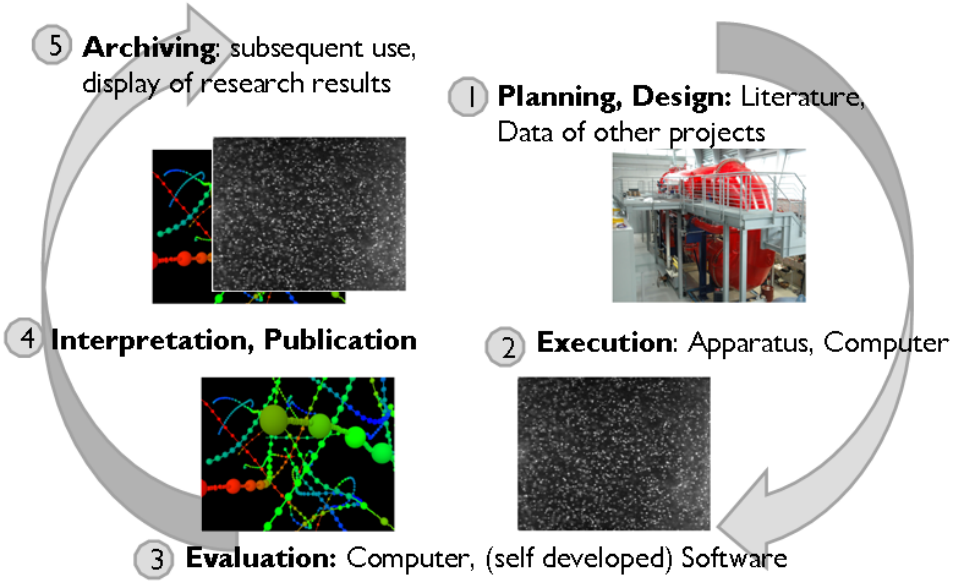
LaTeX Annotations

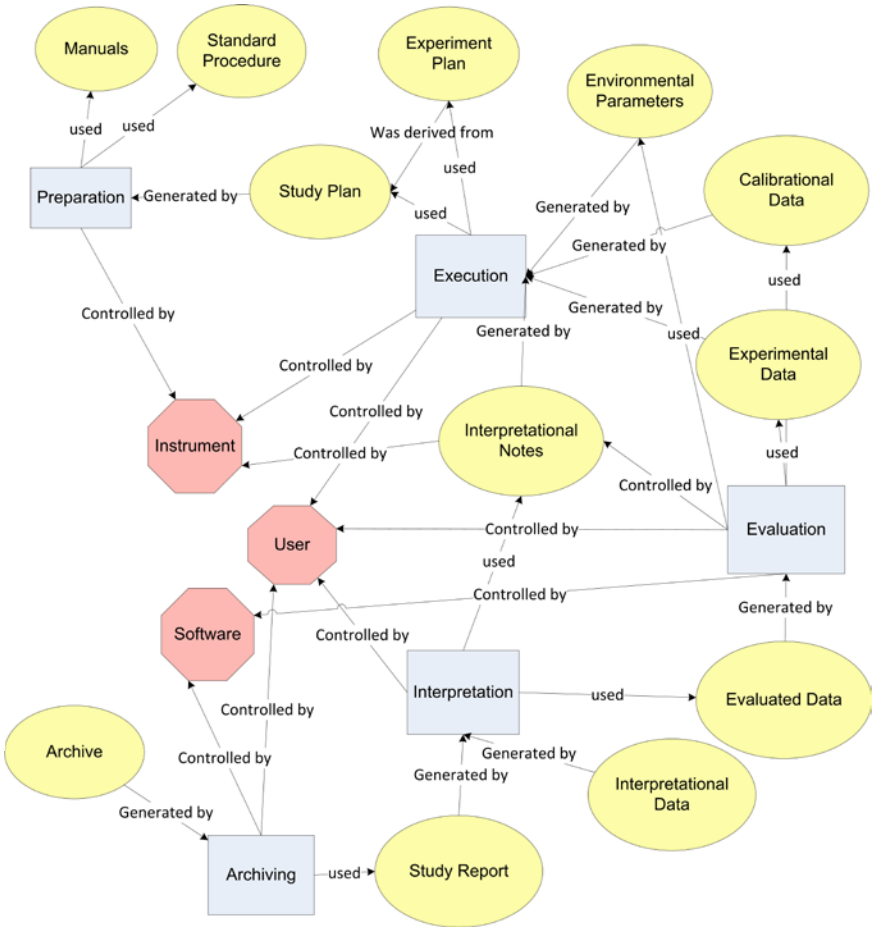
```
\begin{document}
\provAuthor{Andreas Schreiber}{http://orcid.org/0000-0001-5750-5649}
\provOrganization{German Aerospace Center (DLR)}{http://www.dlr.de}
\provTitle{A Provenance Model for Quantified Self Data}
\provProject
  {PROV-SPEC (FS12016)}
  {http://www.dlr.de/sc/desktopdefault.aspx/tabid-8073/}
  {http://www.bmwi.de/}
```



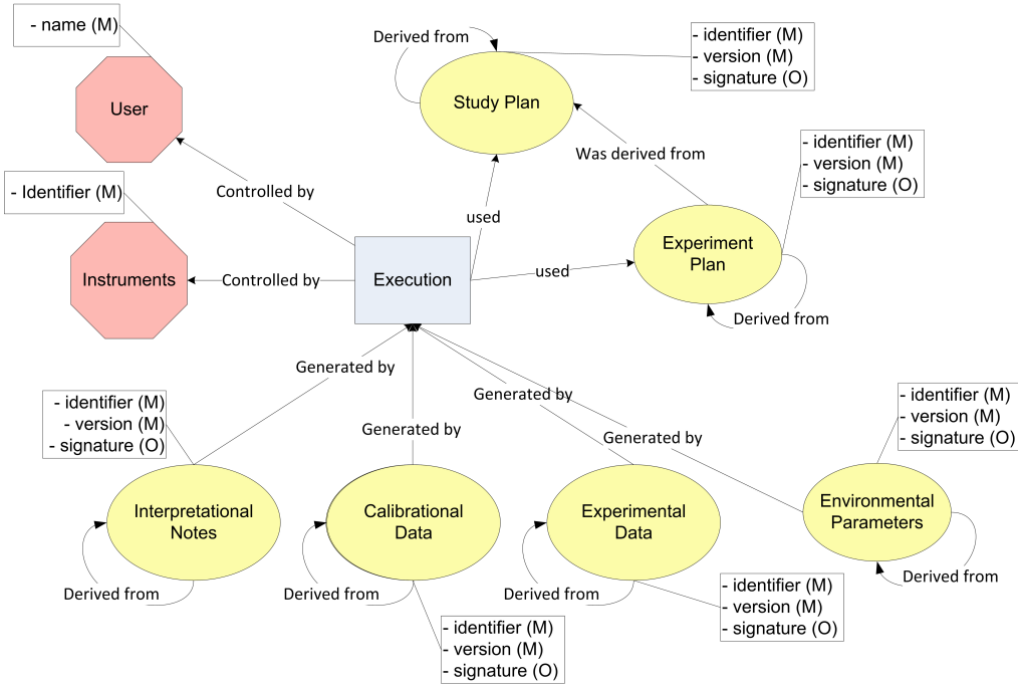
DLR

Electronic Laboratory Notebook



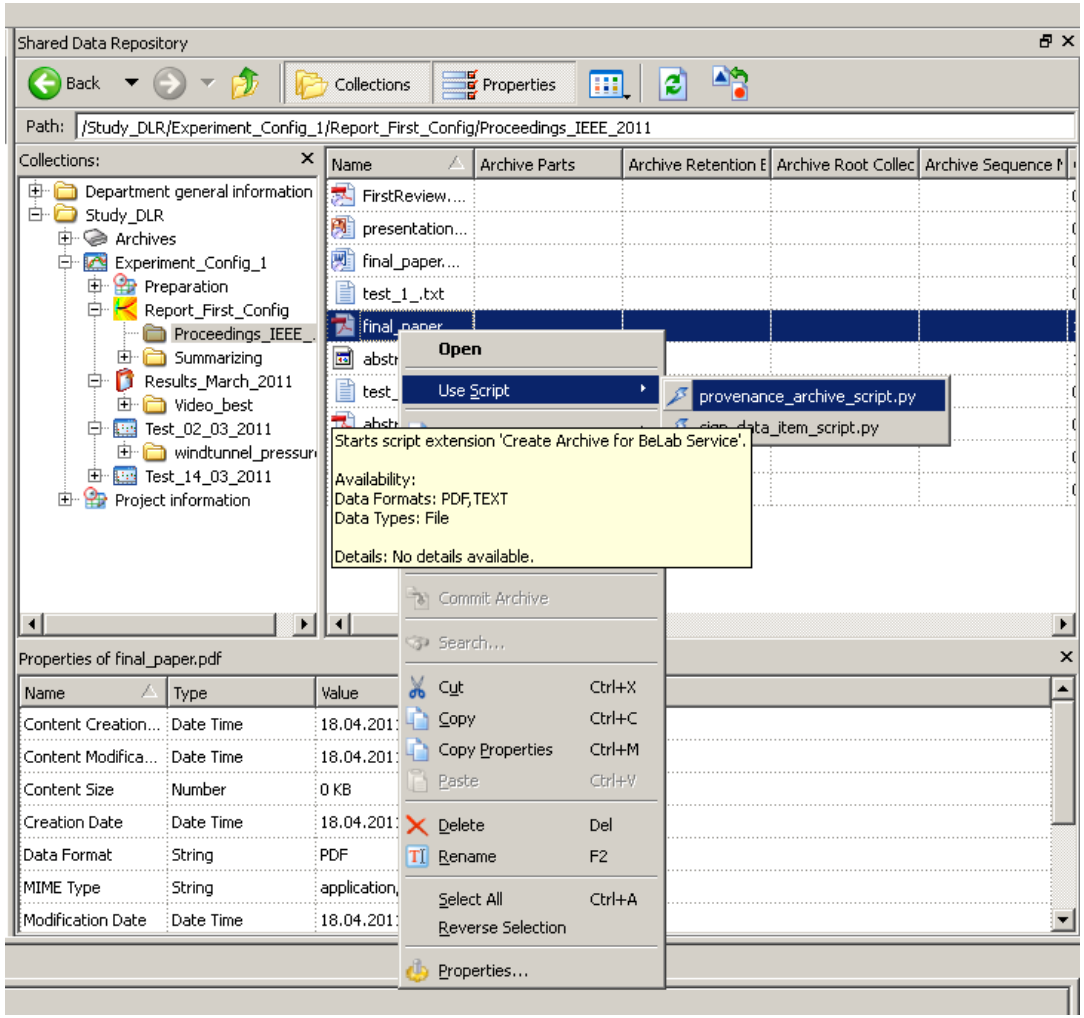


Query „*Who worked on experiment X?*“




```
$experiment = g.key($_g, 'identifier', X)
$user = $experiment/inE/inV/outE[@label = controlled_by]
```

Practical Use Case – Archive all Data of a Paper



https://github.com/DLR-SC/DataFinder

Current Research and Development

New PROV Library for Python in development

- https://github.com/DLR-SC/prov-db-connector
- Connectors for Neo4j implemented, ArangoDB planned
- APIs in REST, ZeroMQ, MQTT

Trusted Provenance

- Storing Provenance in Blockchains

Provenance for people

- New approaches for visualization
- For example, *PROV Comics*

Thank You!

Questions?

Andreas.Schreiber@dlr.de
www.DLR.de/sc | @onyame