Technology neutral provenance storage & sharing

Dr. Frank Dressel
Institute of Software Methods for Product Virtualization
frank.dressel@dlr.de
https://www.dlr.de/sp

HMC conference 2022
Online

Background

Provenance is information about the origin of data. It is one of the requirements for fair data (see [1]). Essential for reviewable data.

Whenever data is used, provenance is needed together with the data to review it for the given usage

- Data formats for combining provenance & data exists (for example hdf5 [2], data packages [3])
- Cloud based services exists (see for example [4])
- Metadata management systems are in use (see various MMS or PDM)

Scenario 1: Data usage of after some years/decades (see for ex. aviation regulations for record keeping)

- Maintenance matters
- Technology & tools will change

Scenario 2: Data exchange between stakeholders without common infrastructure

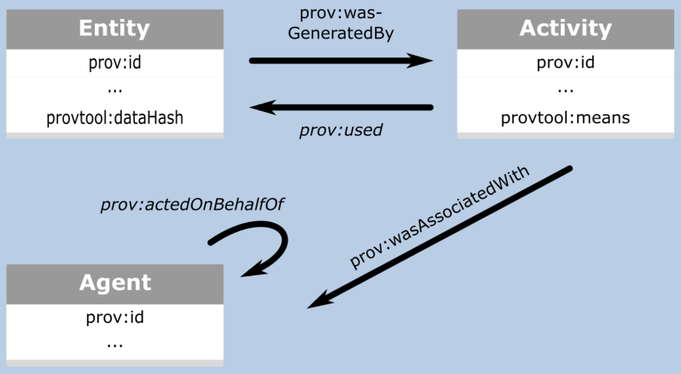
- No direct synchronization of data possible
- No direct synchronization wanted

Provenance model used

Model: Reduced W3C prov model [5]: Activities, agents, entities

Additional properties

- provtool: datahash, :means



Reduced povenance model base on the W3C prov model.

Approach

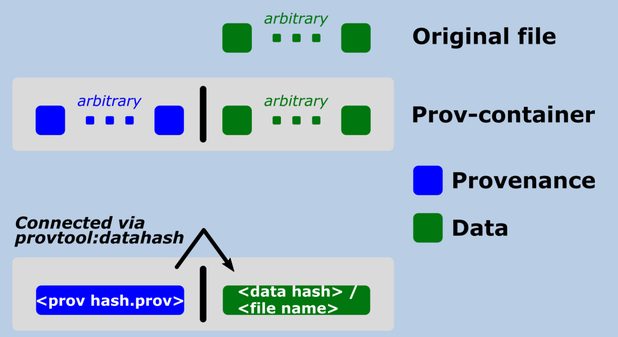
Assumption: Data & provenance exchange somehow possible, concrete technology unknown (time & space)

Scope:

- No additional services/tools needed (but support possible)
- Data as is
- Not necessarily local
- Usable by nearly any existing DMS
- Lookup and search out of scope
- Provenance entity id = provenance hash

Provenance container

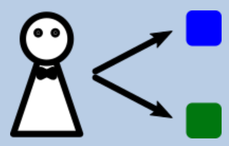
- Data kept as binary
- Provenance in utf-8 encoded json according to [6] and reduced model for current activity only
- datahash as reference from provenance to data
- Serialized independently with hash sum as id & name



Usage

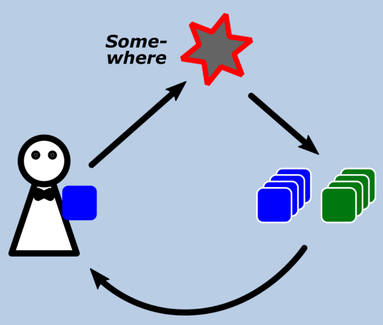
Production

- Agent generates data and provenance in text format



Usage

- User starts with provenance
- Queries used



Implications

Hash as id: Content addressable

- Storage location is ir
- Nearly any storage technology will do

Effective Immutability

- Modification detectable due to hash id
- Data generation using an existing container seals the previous container

Self describing within model

- Provenance container ≙ Entity

Arbitrary DMS possible

- Use already: Shepard [7, 8]
- File storage & indexing with lucene

Reference to primary sources

- Build & traverse provenance graph
- Additional systems may provide easy lookup

Exchange w/o technical debts

- Only existing formats and utf8-text
- Nearly any existing exchange format will do

Provenance first! *(see [4] for similar id-based approach)*

Con:

Two artefacts instead of one

Purely organizational effort to use it: Can start today

Deutsches Zentrum für Luft- und Raumfahrt

[1]: Wilkinson et al.: „The FAIR Guiding Principles for scientific data management and stewardship”, Scientific Data 3, 2016
[2]: Koziol, Robinson: HDF5, https://doi.org/10.11578/dc.20180330.1
[3]: Walsh, Pollock: https://specs.frictionlessdata.io//data-package/
[4]: Chard et al.: „I'll take that to go: Big data bags and minimal identifiers for exchange of large, complex datasets”, 2016 IEEE International Conference
[5]: Groth, Moreau: “PROV-Overview”, https://www.w3.org/TR/prov-overview/
[6]: Huynh et al.: „The PROV-JSON Serialization“, https://www.w3.org/Submission/prov-json/
[7]: Dressel et al.: „Common Source & Provenance at Virtual Product House: Integration with a Data Management System”, DLRK 2022
[8]: Haase et al.: „shepard - storage for heterogeneous product and research data”, DOI: 10.5281/zenodo.5091604