Provenance and **Reproducibility**. Feedbacks from the VAMDC experience

C.M. Zwölf, N. Moreau and VAMDC consortium

Provenance and reproducibility issues

Provenance and reproducibility are two facets of the same problem

- Provide somebody with all the elements and information about the production of a given result.

Hot topic!
With
several
unsolved
issues

Provenance and reproducibility issues

Provenance and reproducibility are two facets of the same problem

- Provide somebody with all the elements and information about the production of a given result.

Hot topic!
With
several
unsolved
issues

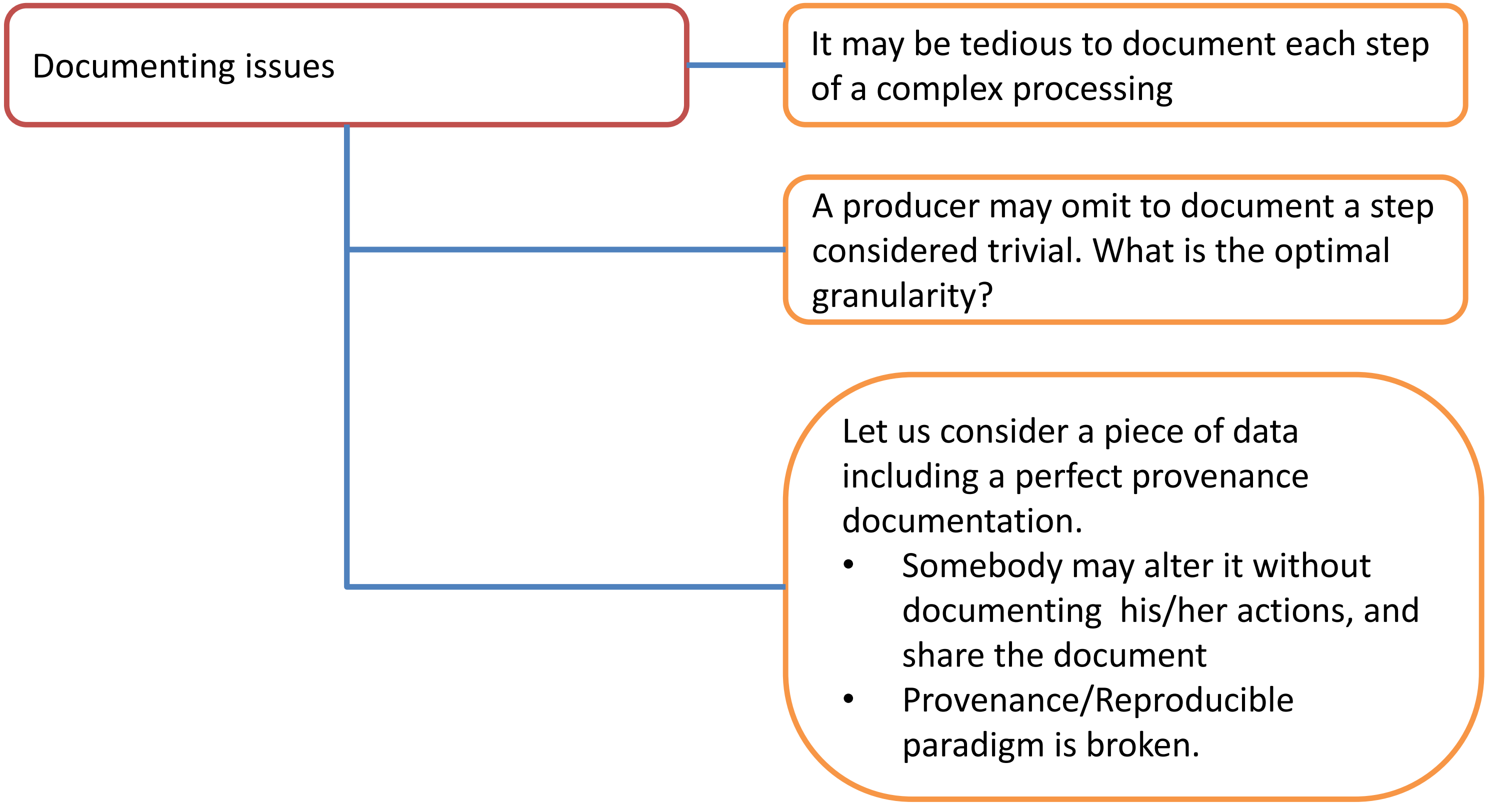
Documenting issues

Provenance and reproducibility issues

Provenance and reproducibility are two facets of the same problem

- Provide somebody with all the elements and information about the production of a given result.

Hot topic! With several unsolved issues



Documenting issues

It may be tedious to document each step of a complex processing

A producer may omit to document a step considered trivial. What is the optimal granularity?

Let us consider a piece of data including a perfect provenance documentation.

- Somebody may alter it without documenting his/her actions, and share the document
- Provenance/Reproducible paradigm is broken.

Provenance and reproducibility issues

Provenance and reproducibility are two facets of the same problem

- Provide somebody with all the elements and information about the production of a given result.

Hot topic!
With
several
unsolved
issues

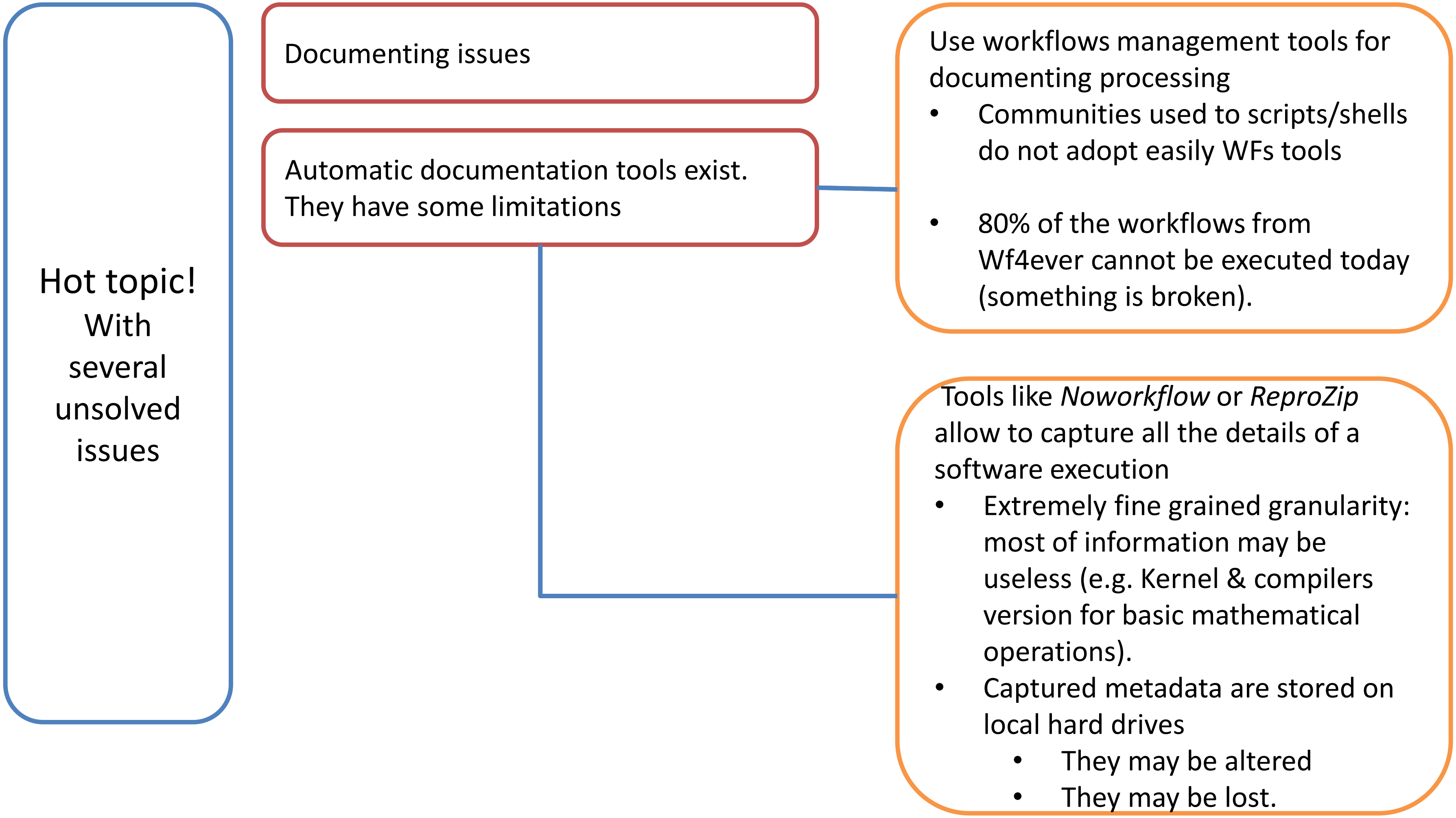
Documenting issues

Automatic documentation tools exist.
They have some limitations

Provenance and reproducibility issues

Provenance and reproducibility are two facets of the same problem

- Provide somebody with all the elements and information about the production of a given result.



Hot topic!
With
several
unsolved
issues

Documenting issues

Automatic documentation tools exist.
They have some limitations

Use workflows management tools for documenting processing

- Communities used to scripts/shells do not adopt easily WFs tools
- 80% of the workflows from Wf4ever cannot be executed today (something is broken).

Tools like *Noworkflow* or *ReproZip* allow to capture all the details of a software execution

- Extremely fine grained granularity: most of information may be useless (e.g. Kernel & compilers version for basic mathematical operations).
- Captured metadata are stored on local hard drives
  - They may be altered
  - They may be lost.

Provenance and reproducibility issues

Provenance and reproducibility are two facets of the same problem

- Provide somebody with all the elements and information about the production of a given result.

Hot topic!
With
several
unsolved
issues

Documenting issues

Automatic documentation tools exist.
They have some limitations

Metadata Format

Provenance and reproducibility issues

Provenance and reproducibility are two facets of the same problem

- Provide somebody with all the elements and information about the production of a given result.

Hot topic!
With
several
unsolved
issues

Documenting issues

Automatic documentation tools exist.
They have some limitations

Metadata Format

What is the format to adopt for metadata reporting?

- W3C / IVOA provenance
- OAIS
- ISO 19115 (geophysical core)
- Others?

Provenance and reproducibility issues

Provenance and reproducibility are two facets of the same problem

- Provide somebody with all the elements and information about the production of a given result.

Hot topic!
With
several
unsolved
issues

Documenting issues

Automatic documentation tools exist.
They have some limitations

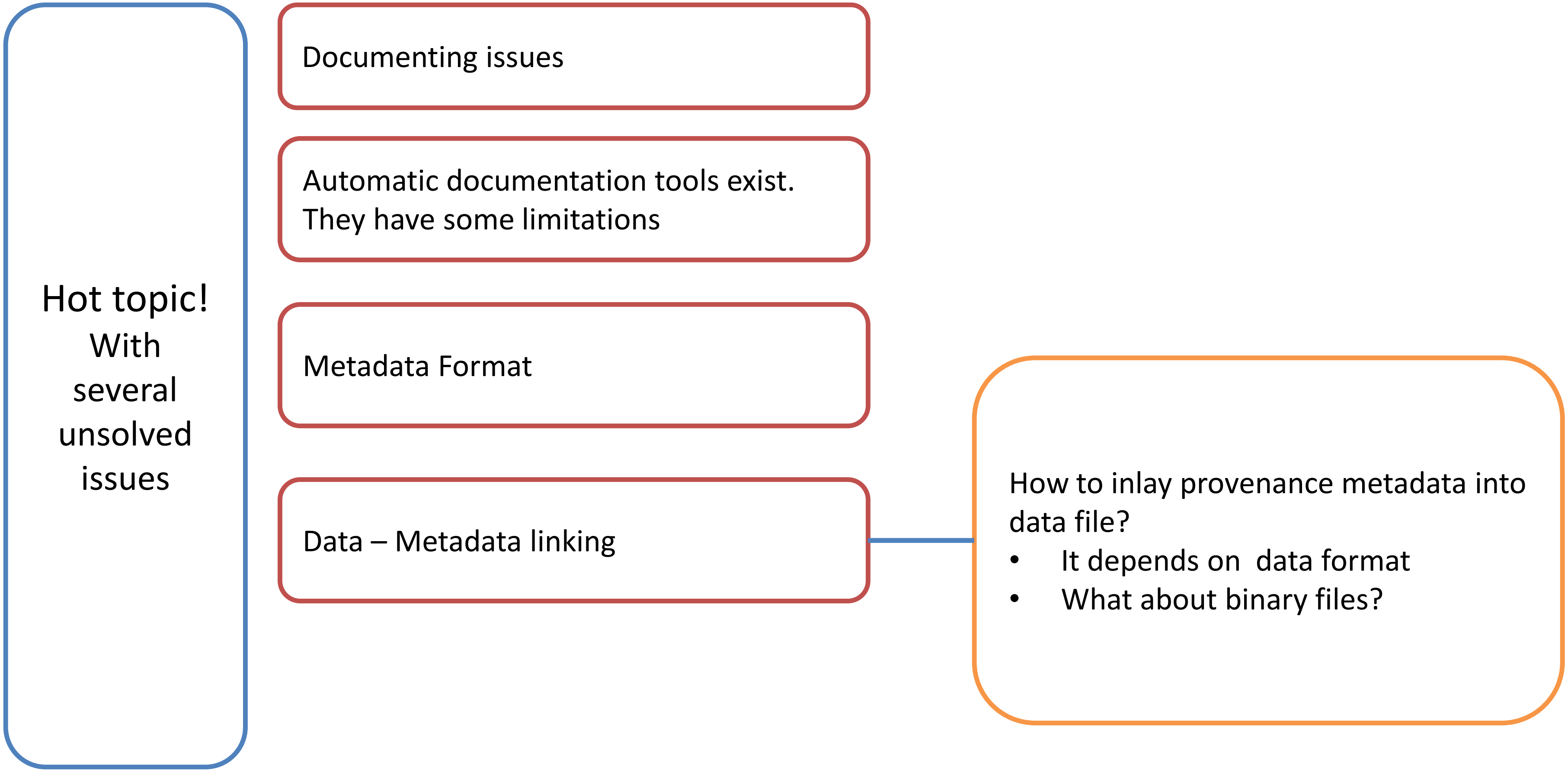
Metadata Format

Data – Metadata linking

Provenance and reproducibility issues

Provenance and reproducibility are two facets of the same problem

- Provide somebody with all the elements and information about the production of a given result.



Hot topic! With several unsolved issues

Documenting issues

Automatic documentation tools exist. They have some limitations

Metadata Format

Data – Metadata linking

How to inlay provenance metadata into data file?

- It depends on data format
- What about binary files?

Provenance and reproducibility issues

Provenance and reproducibility are two facets of the same problem

- Provide somebody with all the elements and information about the production of a given result.

Hot topic!
With
several
unsolved
issues

Documenting issues

Automatic documentation tools exist.
They have some limitations

Metadata Format

Data – Metadata linking

⚠ Provenance & Reproducibility

≠ Repeatability

Provenance and reproducibility issues

Provenance and reproducibility are two facets of the same problem

- Provide somebody with all the elements and information about the production of a given result.

Hot topic! With several unsolved issues

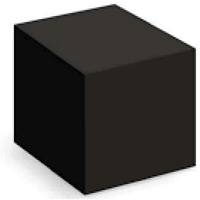
Documenting issues

Automatic documentation tools exist. They have some limitations

Metadata Format

Data – Metadata linking

⚠ Provenance & Reproducibility ≠ Repeatability



Virtual machines and containers (e.g. Docker) may be distributed for sharing processes:

- These tools work as black-boxes
- Result may be repeated, but
- With no information this is useless for our purposes.

Provenance and reproducibility issues

Provenance and reproducibility are two facets of the same problem

- Provide somebody with all the elements and information about the production of a given result.

Hot topic!
With
several
unsolved
issues

Documenting issues

Automatic documentation tools exist.
They have some limitations

Metadata Format

Data – Metadata linking

Provenance & Reproducibility

≠ Repeatability

Guidelines for solutions:
Provenance metadata

- should be automatically produced by tools with no effort for practitioners/operators.
- Should be stored online in reliable repositories (e.g. DSA certified).

Sketching a solution

We propose a solution built on two layers

Persistent unique resolvable identifiers (**PURI**)

A “smart” log service

Sketching a solution

We propose a solution built on two layers

Persistent unique resolvable identifiers (**PURI**)

A “smart” log service

Software element

- Function in a code
- Web service
- Whatever

Sketching a solution

We propose a solution built on two layers

Persistent unique resolvable identifiers (**PURI**)

A “smart” log service

Has a version

Takes parameter(s)

Receives input(s)

Produces output(s)

Runs on a given configuration

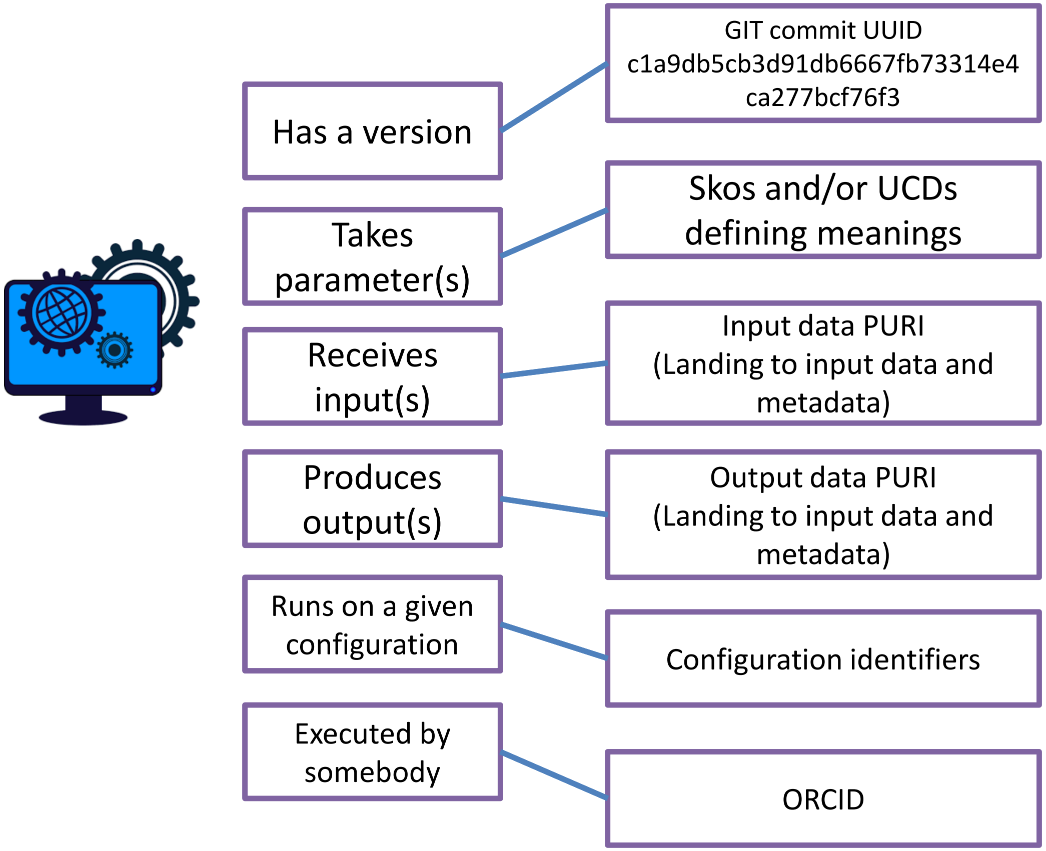
Executed by somebody

Sketching a solution

We propose a solution built on two layers

Persistent unique resolvable identifiers (**PURI**)

A “smart” log service



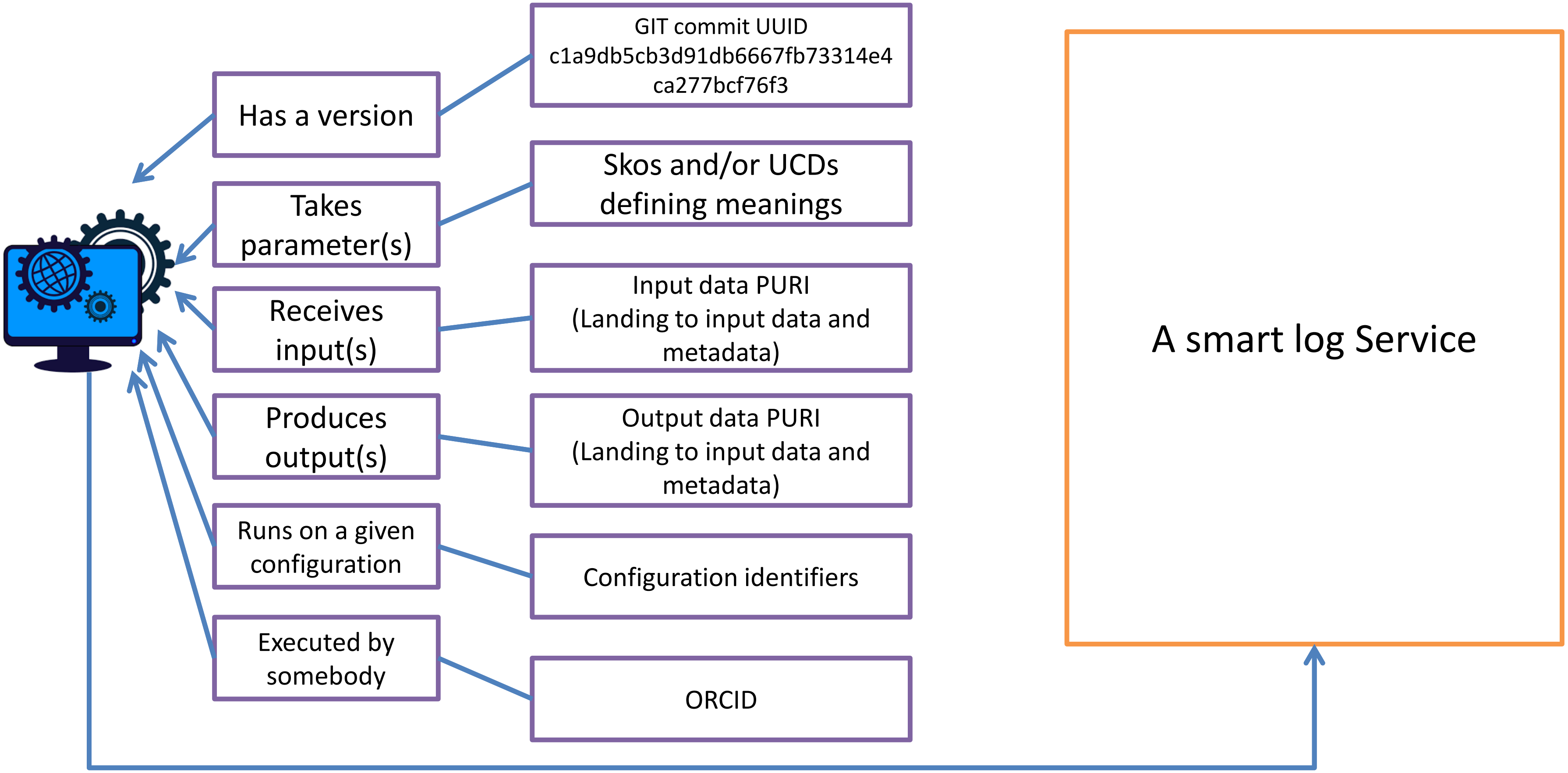
Has a version	GIT commit UUID c1a9db5cb3d91db6667fb73314e4 ca277bcf76f3
Takes parameter(s)	Skos and/or UCDs defining meanings
Receives input(s)	Input data PURI (Landing to input data and metadata)
Produces output(s)	Output data PURI (Landing to input data and metadata)
Runs on a given configuration	Configuration identifiers
Executed by somebody	ORCID

Sketching a solution

We propose a solution built on two layers

Persistent unique resolvable identifiers (**PURI**)

A “smart” log service



Has a version

GIT commit UUID
c1a9db5cb3d91db6667fb73314e4
ca277bcf76f3

Takes parameter(s)

Skos and/or UCDs defining meanings

Receives input(s)

Input data PURI
(Landing to input data and metadata)

Produces output(s)

Output data PURI
(Landing to input data and metadata)

Runs on a given configuration

Configuration identifiers

Executed by somebody

ORCID

A smart log Service

Provide the software element with the ability to log this information

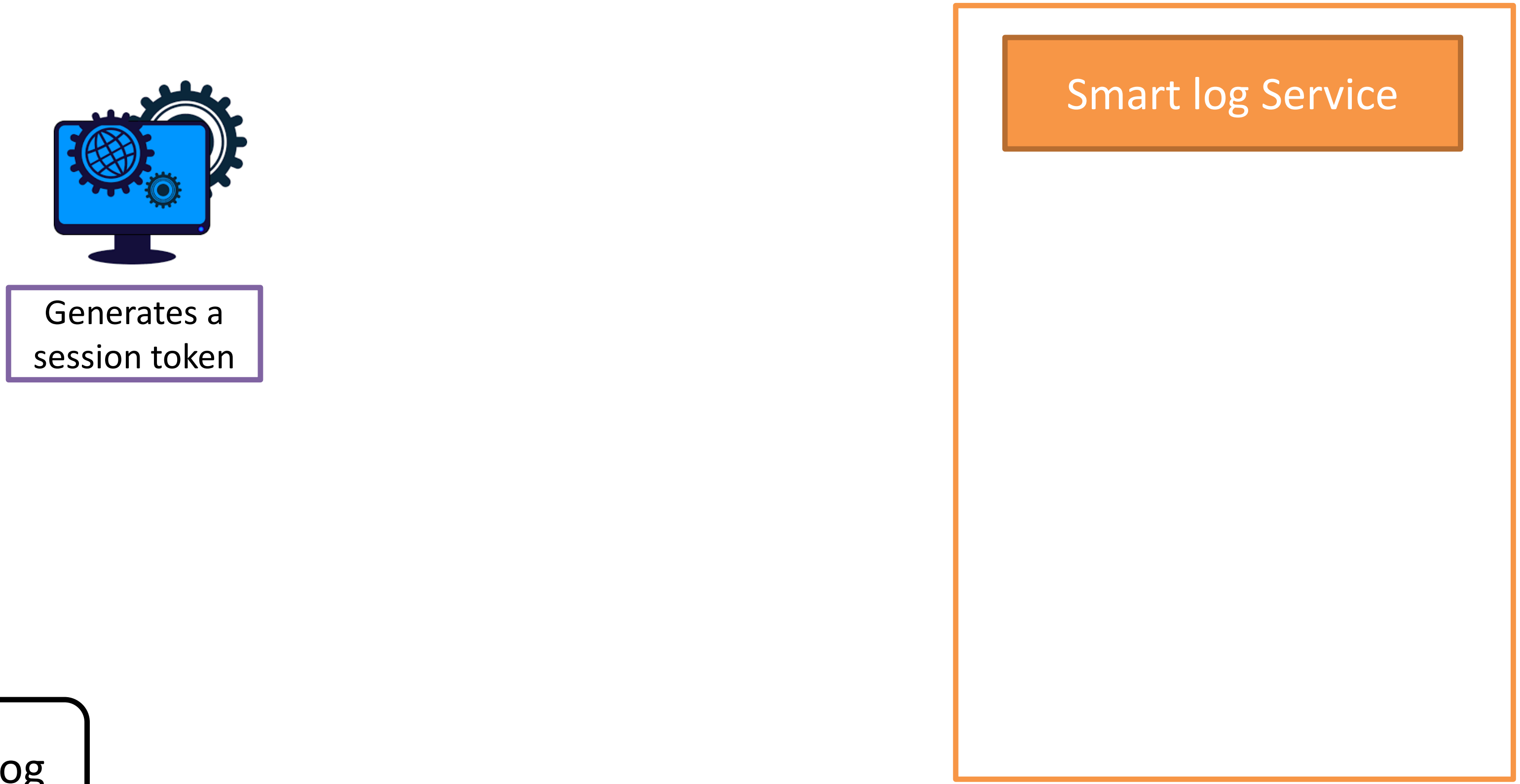
Sketching a solution

We propose a solution built on two layers

Persistent unique resolvable identifiers (**PURI**)

A “smart” log service

In order to work with distributed architectures, we focus on an asynchronous web oriented architecture



Generates a session token

Smart log Service

Phase 1 : log

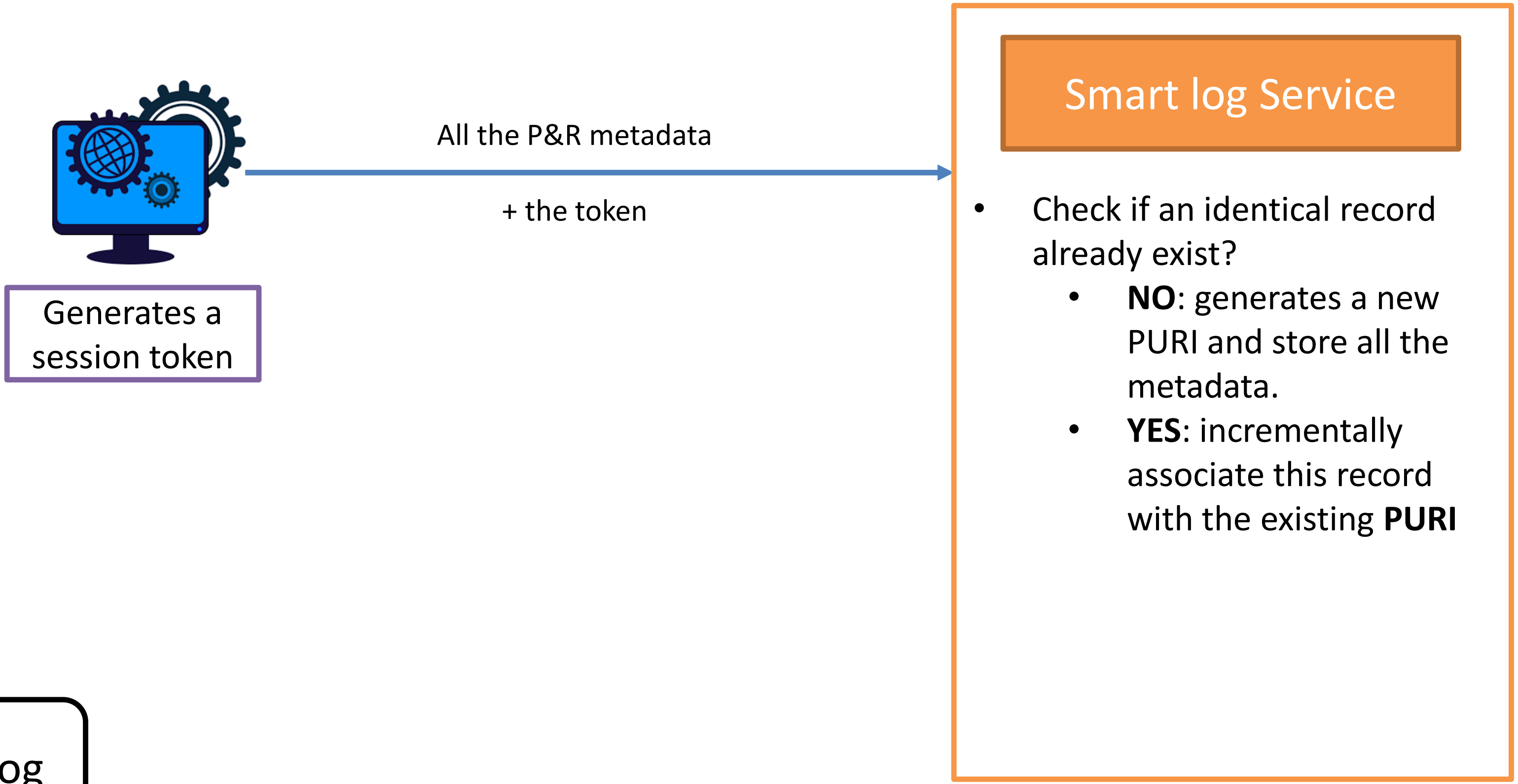
Sketching a solution

We propose a solution built on two layers

Persistent unique resolvable identifiers (**PURI**)

A “smart” log service

In order to work with distributed architectures, we focus on an asynchronous web oriented architecture



Generates a session token

All the P&R metadata

+ the token

Smart log Service

- Check if an identical record already exist?
  - **NO**: generates a new PURI and store all the metadata.
  - **YES**: incrementally associate this record with the existing **PURI**

Phase 1 : log

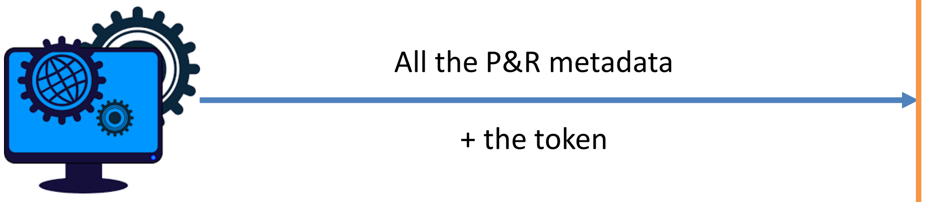
Sketching a solution

We propose a solution built on two layers

Persistent unique resolvable identifiers (**PURI**)

A “smart” log service

In order to work with distributed architectures, we focus on an asynchronous web oriented architecture



All the P&R metadata

+ the token

Generates a session token

Smart log Service

- Check if an identical record already exist?
  - **NO**: generates a new PURI and store all the metadata.
  - **YES**: incrementally associate this record with the existing **PURI**

The unicity criterion has to be defined ad hoc, according to the scientific context.

Phase 1 : log

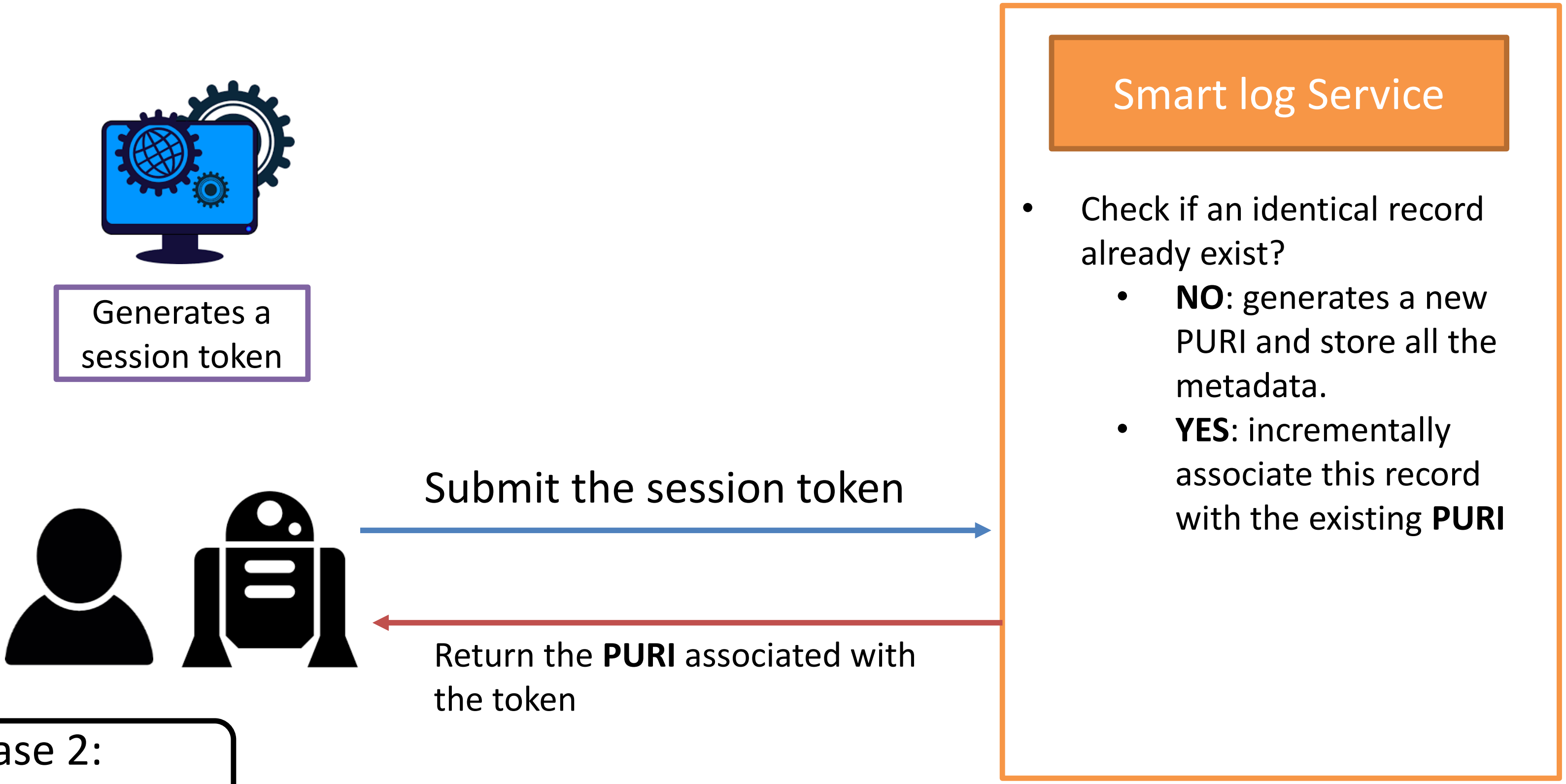
Sketching a solution

We propose a solution built on two layers

Persistent unique resolvable identifiers (**PURI**)

A “smart” log service

In order to work with distributed architectures, we focus on an asynchronous web oriented architecture



Generates a session token

Submit the session token

Return the **PURI** associated with the token

Smart log Service

- Check if an identical record already exist?
  - **NO**: generates a new PURI and store all the metadata.
  - **YES**: incrementally associate this record with the existing **PURI**

Phase 2: association

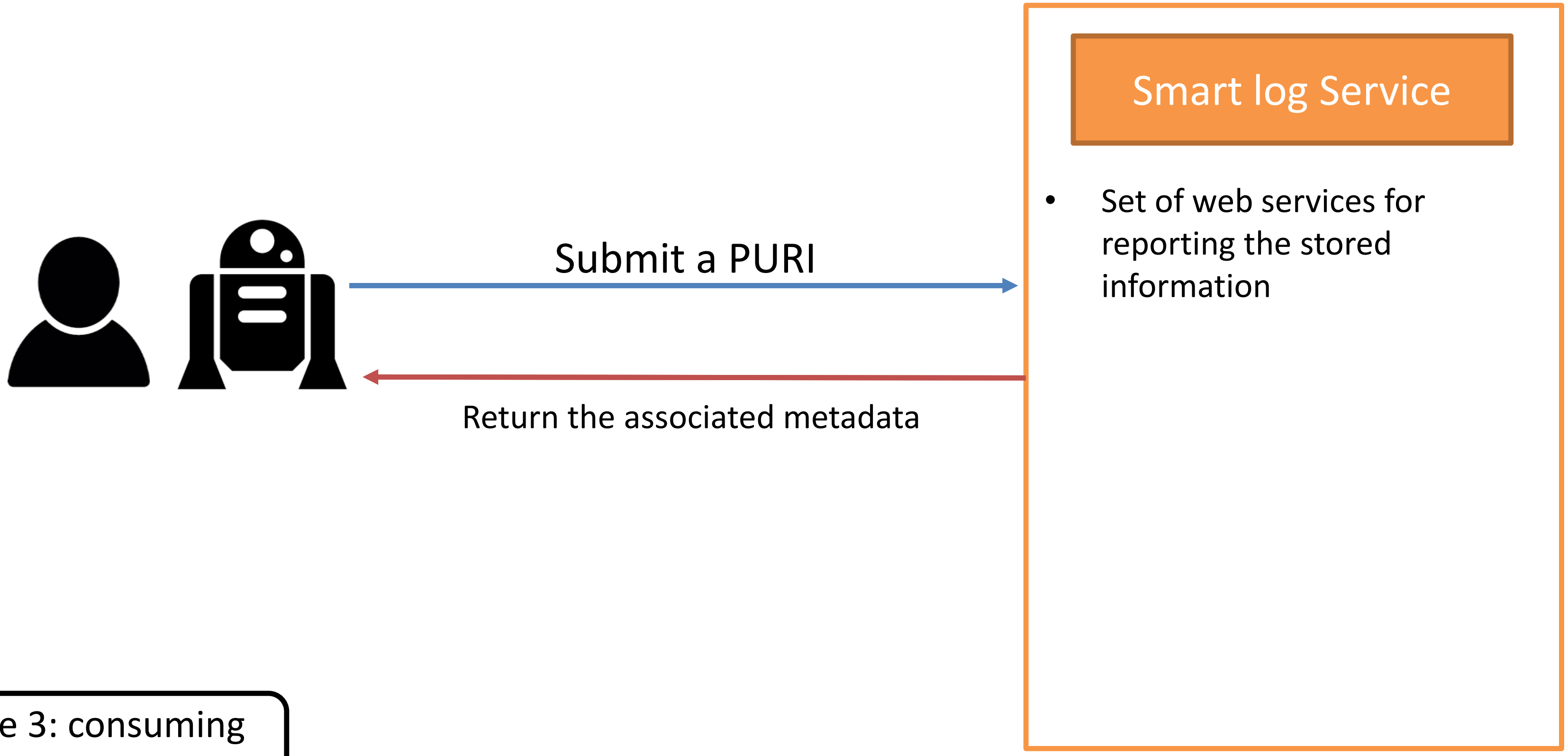
Sketching a solution

We propose a solution built on two layers

Persistent unique resolvable identifiers (**PURI**)

A “smart” log service

In order to work with distributed architectures, we focus on an asynchronous web oriented architecture



Submit a PURI

Return the associated metadata

Smart log Service

- Set of web services for reporting the stored information

Phase 3: consuming the PURI

Benefits of the proposed architecture

- Log information is "*just a http invocation*"
  - It is very easy to include it in software to track
  - Will not slow down processing

- Metadata will be centrally stored in ad hoc repositories. Precious metadata won't be lost in computer crashes.

Modular and agile approach

Several metadata formats may be supported for reporting the stored information (it is easy to include new one without broking anything!!):
http://my-provenance.org/metadataFormat1/66d6b32b-dab7-4e27-af6b-44e8e62f4fdf
http://my-provenance.org/metadataFormat2/66d6b32b-dab7-4e27-af6b-44e8e62f4fdf

- Full data genealogy may be achieved by following iteratively the PURIs discovered by resolving a starting given PURI.
- Curation is simplified: one has just to check that all the PURI linked from a point are not corrupted.
- Further metadata (e.g. sematic tagging, link to similar results) may be added a-posteriori by associating them to an already existing PURI.

A working example

- We implemented the described architecture for building the VAMDC Query Store
  - ➢ Following the Research Data Alliance recommendation on citation of dynamic data
  - ➢ The tracked software elements are
    - ➢ The VAMDC clients
    - ➢ The VAMDC federated databases (which are TAP services)

- The software developed is generic and may be easily adapted to other cases.
  - We are happy to help people in adapting our approach to their cases

Live Demonstration at https://youtu.be/kDDWFpi22cU