Data Discoverability and Persistent Identifiers

EUDAT Summer School, Herkalion, 2017

Christine Staiger and Sofiane Bendoukha
SURFsara and DKRZ
EUDAT Summer School, Heraklion, 2017

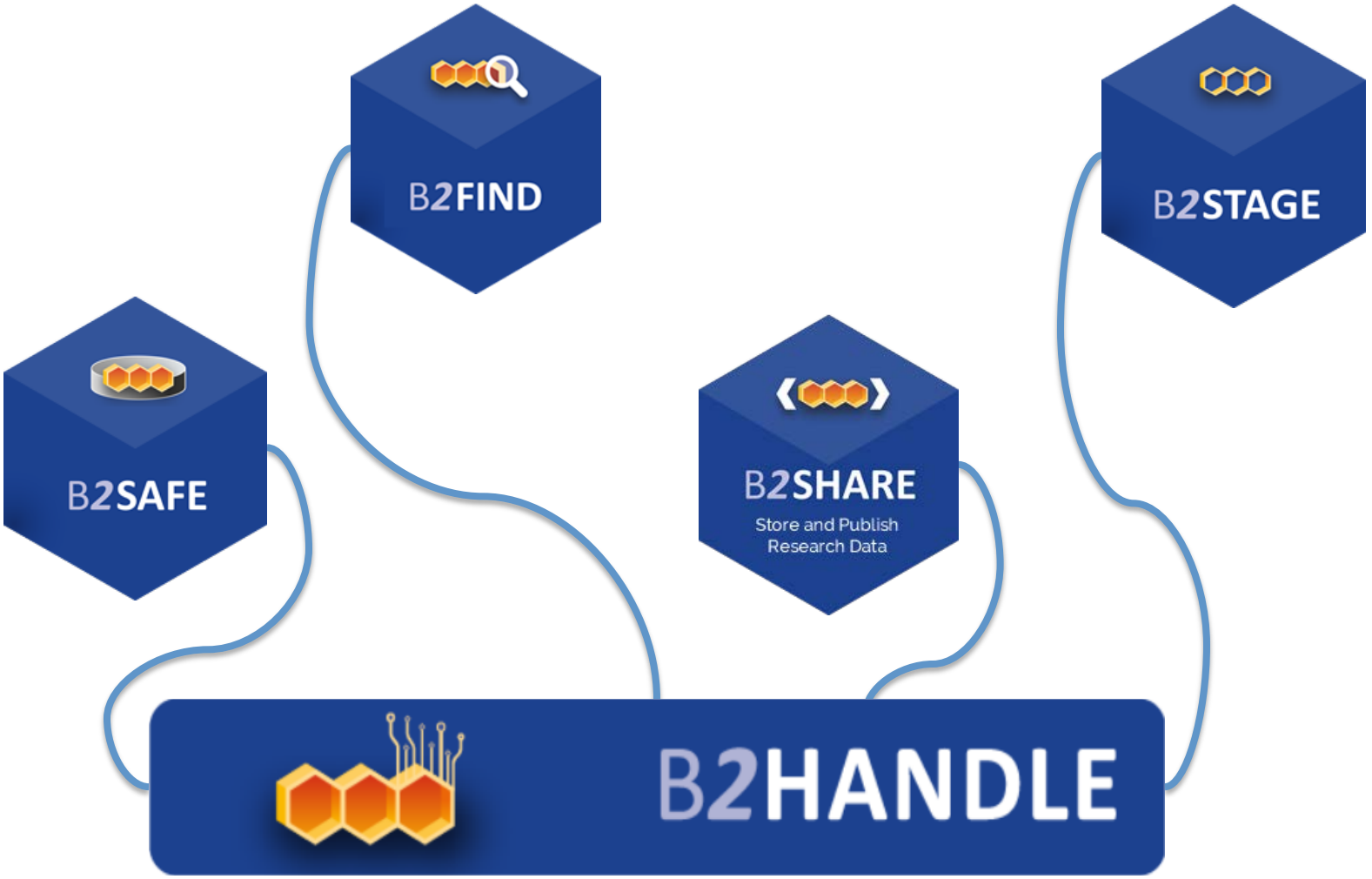
EUDAT receives funding from the European Union's Horizon 2020 programme - DG CONNECT e-Infrastructures. Contract No. 654065

Outline

- What are PIDs?
- Use cases
- PID providers and systems
- PID usage in EUDAT

- The Handle system
  - The handle resolution system
  - The relation between Handle and ePIC
  - Hands-on tutorial

PIDs in EUDAT



PIDs in EUDAT – Why?

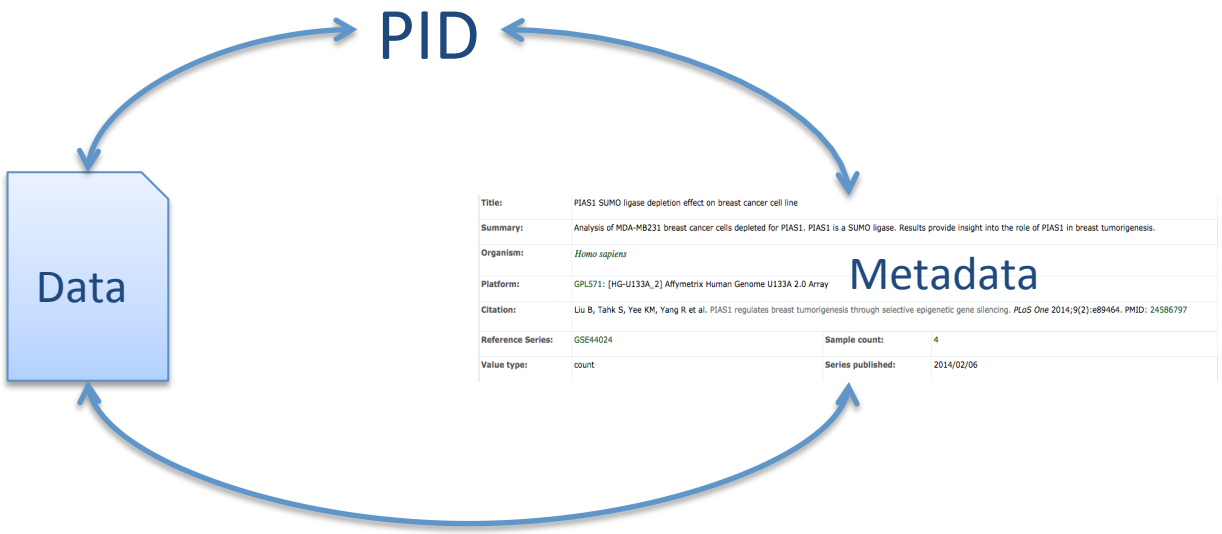
- Managing **increasing numbers of data objects**
- **Sharing data from different sources** amongst researchers
- Data needs to be **(globally) identifiable and addressable** → reuse of data
- Data citation
- Linking data from different sources

  → Pooling datasets
- Challenges
  - Object locations change over time
  - Object migration between repositories

What do we want from data?

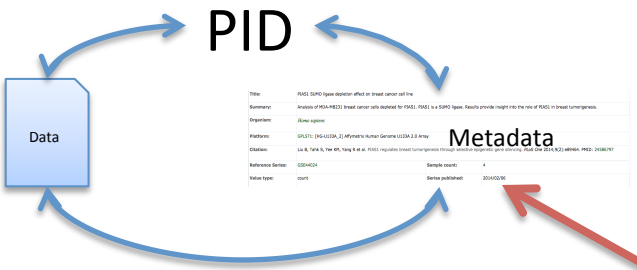
- **Findable** – Easy to find by both humans and computer systems → Metadata
- **Accessible** – Stored for long term, accessed and/or downloaded with well-defined license and access
- **Interoperable** – Ready to be combined with other datasets by humans as well as computer systems;
- **Reusable** – Ready to be used for future research and to be processed further using computational methods.
- The FAIR guiding Principles for scientific data management and stewardship, doi:10.1038/sdata.2016.18

What do we need?

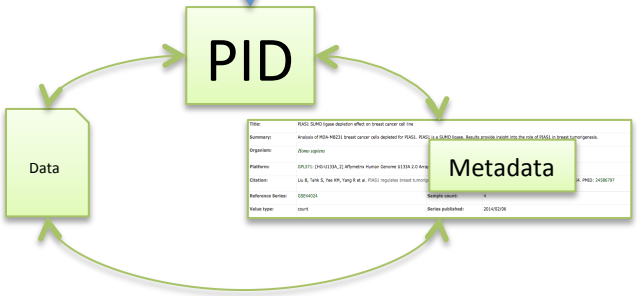


- Persistent Identifier: reference and identify object, either metadata or data object
- Synchronise PID, Data and Metadata during creation, maintenance, update and deletion of a digital object!

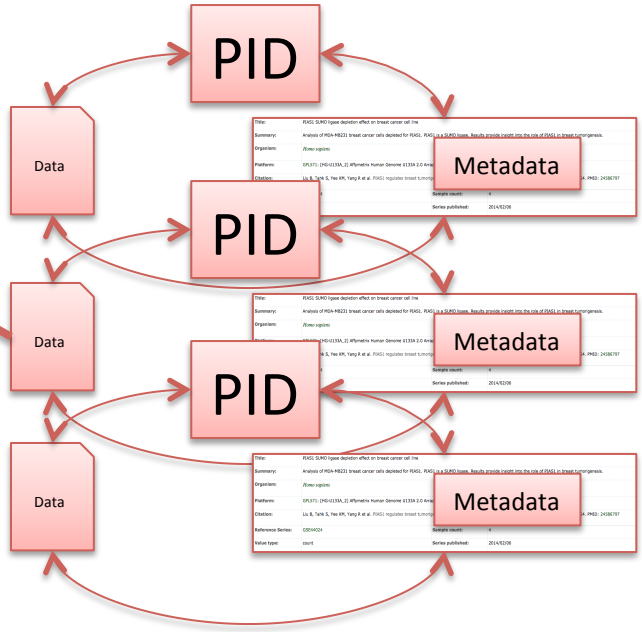
What do we need?



Analysed data



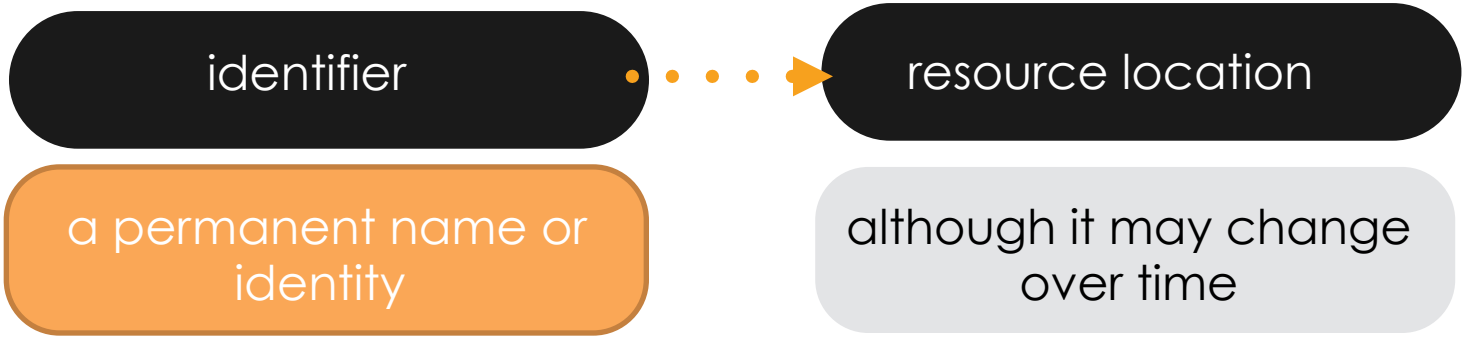
Published data



Raw data

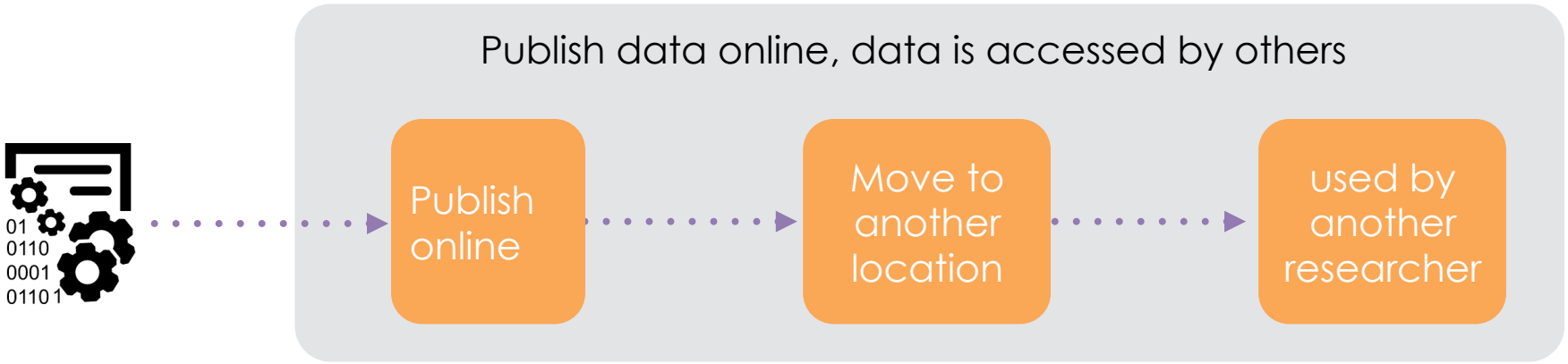
What do we know about Persistent Identifiers?

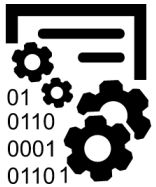
- A Persistent Identifier (PID) is an identifier that is effectively permanently assigned to a resource.



- Pointers to data resources
- Globally unique
- Exist infinitely long (the PID, not necessarily the data)

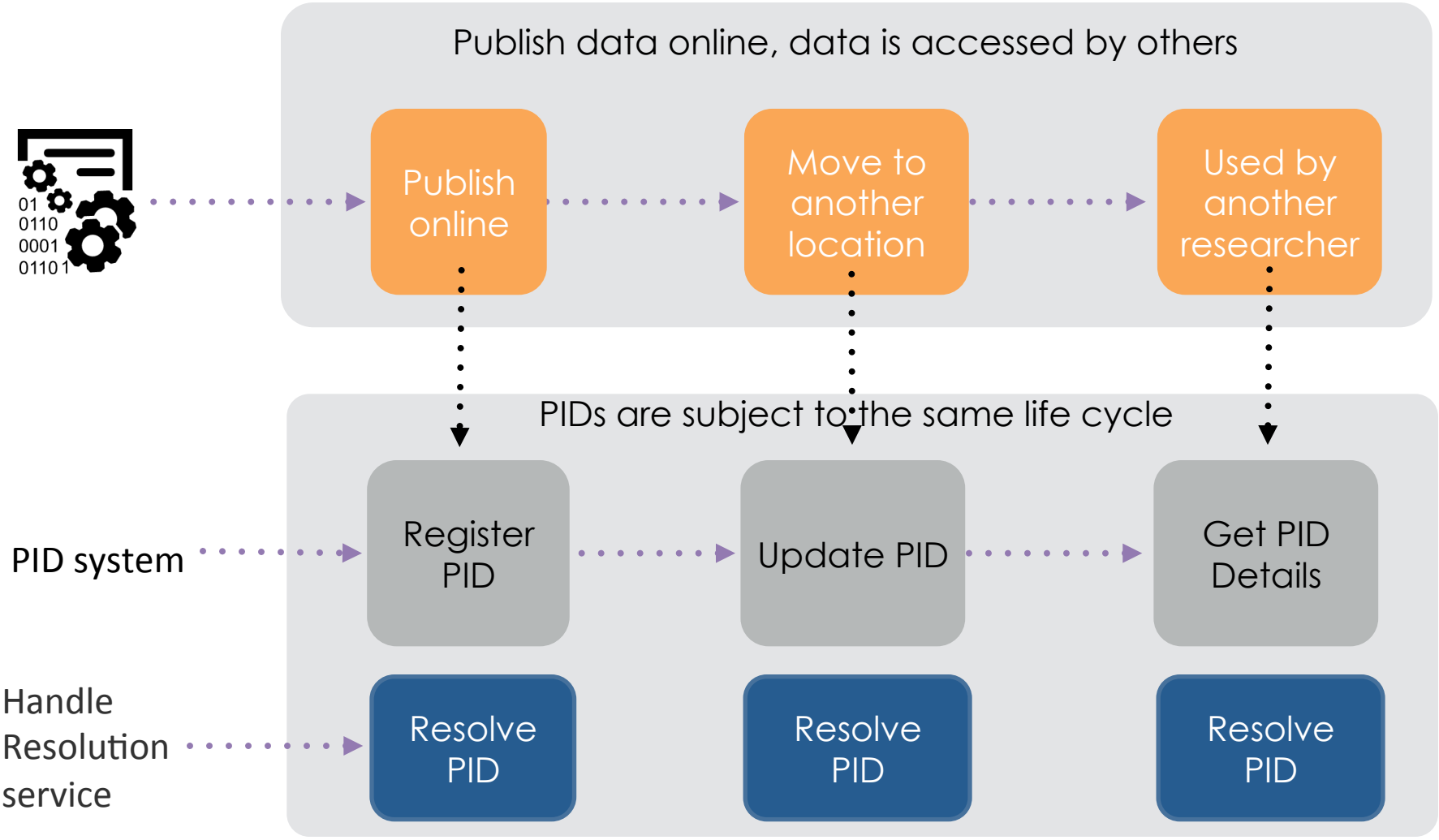
Simple data life cycle, linearised





- Published online: http://www.test.com/test.html
- Other users may cite, access, re-use this url
- Relocate the resource at http://www.example.com/
- Other users are not informed -> 404

Data Life Cycle with PID system



Advantages and Disadvantages

Pro:

- Static reference,
  even if data moves or changes
- Network of persistent links
  Data – metadata relations
  Provenance chains

Con:

- Extra effort
  - What to identify?
  - Coordination across organisations and people
- Organisational discipline to ensure persistence

Use cases

Use Case 1: Data publication

- PIDs point to landing page of the digital repository showing metadata
- "Real" data can be downloaded from this page with another link
- E.g. B2SHARE, FigShare, Zenodo, …
- PID

http://hdl.handle.net/11304/3265434c-4b34-11e4-81ac-dcbd1b51435e

resolves to landing page

https://b2share.eudat.eu/records/feafb12e810c489b9e878949c6c35345

EUDAT

B2SHARE
Store and Share Research Data

The persistent identifier for the **collection**

Climate station Waldhaeuser

by [Unknown]

Apr 12, 2017

Description: Climate data

DOI: 10.23728/b2share.7a70f943dcdd48a0822f0f135b3ac2bc Copy

PID: 11304/8220b208-b61b-4a05-bf36-df5d56b6247a Copy

LTER

Files

Name

Metadata_Meteodata WaldhÑuser LTER database.xlsx

Checksum: md5:3cdeaba5f3e9d99cb228161578699668

PID: 11304/f7...

doi®

HOME | HANDBOOK | FACTSHEETS | FAQs | RESOURCES | USERS | NEWS | MEMBERS AREA

Resolve a DOI Name

doi: 10.23728/b2share.7a70f943dcdd48a0822f0f135b3ac2bc

Go

Handle.Net®

Handle Values for: 11304/8220b208-b61b-4a05-bf36-df5d56b6247a

Index	Type	Timestamp	Data
1	URL	2017-04-12 11:59:52Z	https://b2share.eudat.eu/records/7a70f943dcdd48a0822f0f135b3ac2bc

EUDAT

B2SHARE
Store and Share Research Data

Climate station Waldhaeuser

by [Unknown]

Apr 12, 2017

Description: Climate data

DOI: 10.23728/b2share.7a70f943dcdd48a0822f0f135b3ac2bc Copy

PID: 11304/8220b208-b61b-4a05-bf36-df5d56b6247a Copy

The persistent identifier for **files**

LTER

Files

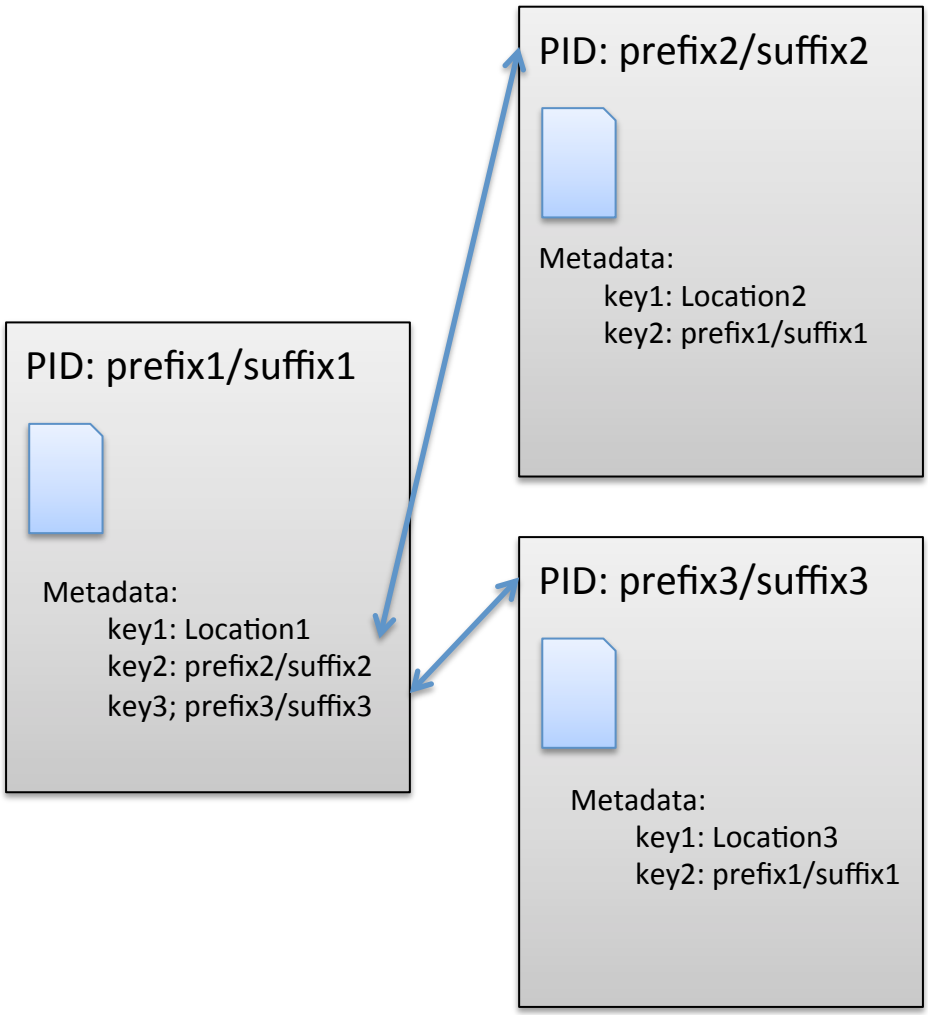
Name	Size
Metadata_Meteodata WaldhÑuser LTER database.xlsx Checksum: md5:3cdeaba5f3e9d99cb228161578699668 PID: 11304/f73726bb-fade-45aa-9700-bdb3a07ff692 Copy	9.24KB

Handle.Net®

Handle Values for: 11304/f73726bb-fade-45aa-9700-bdb3a07ff692

Index	Type	Timestamp	Data
1	URL	2017-04-12 11:59:54Z	https://b2share.eudat.eu/api/files/3d82b14b-8bbc-4bb3-b3b9-aaff408516cb/Metadata_Meteodata%20WaldhN%CC%83user%20LTER%20database.xlsx
2	CHECKSUM	2017-04-12 11:59:54Z	md5:3cdeaba5f3e9d99cb228161578699668

Use case 2: Modeling Relationships

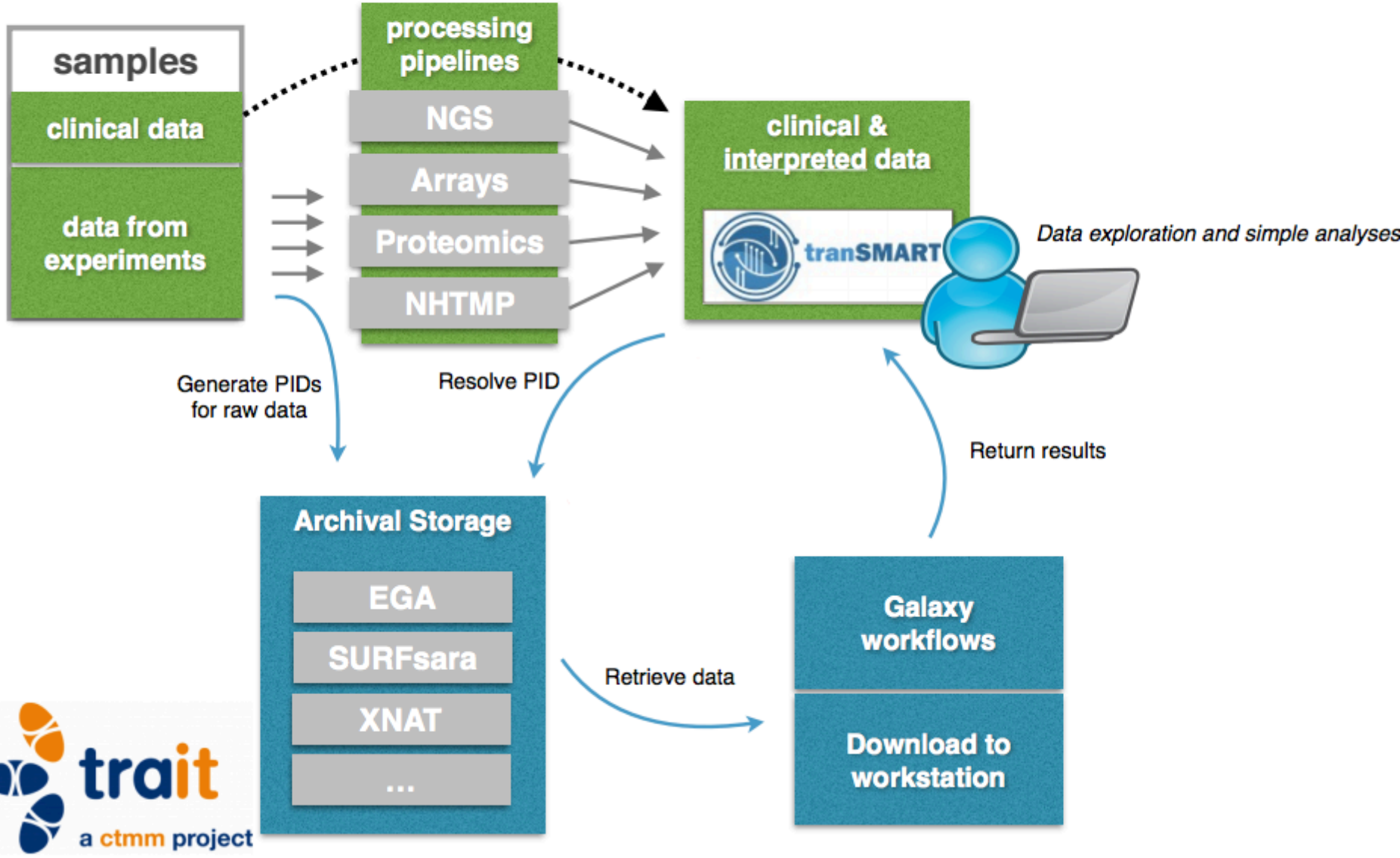


- Use tightly coupled metadata
- Part of/has part relationships
- Model cohort-patient relationship
- Model patient-samples relationship

Which metadata to store with the PID and which in an extra catalogue ?

Use case 3: Enabling data workflows

Molecular profiling dataflow in TraIT



TraIT data ontology

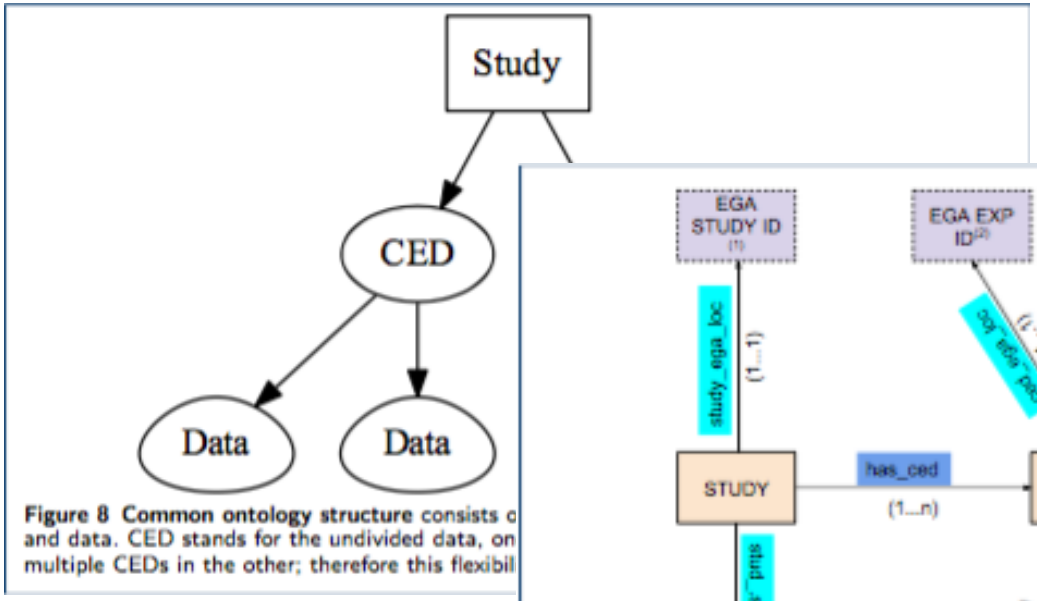


Figure 8 Common ontology structure consists o
and data. CED stands for the undivided data, on
multiple CEDs in the other; therefore this flexibil

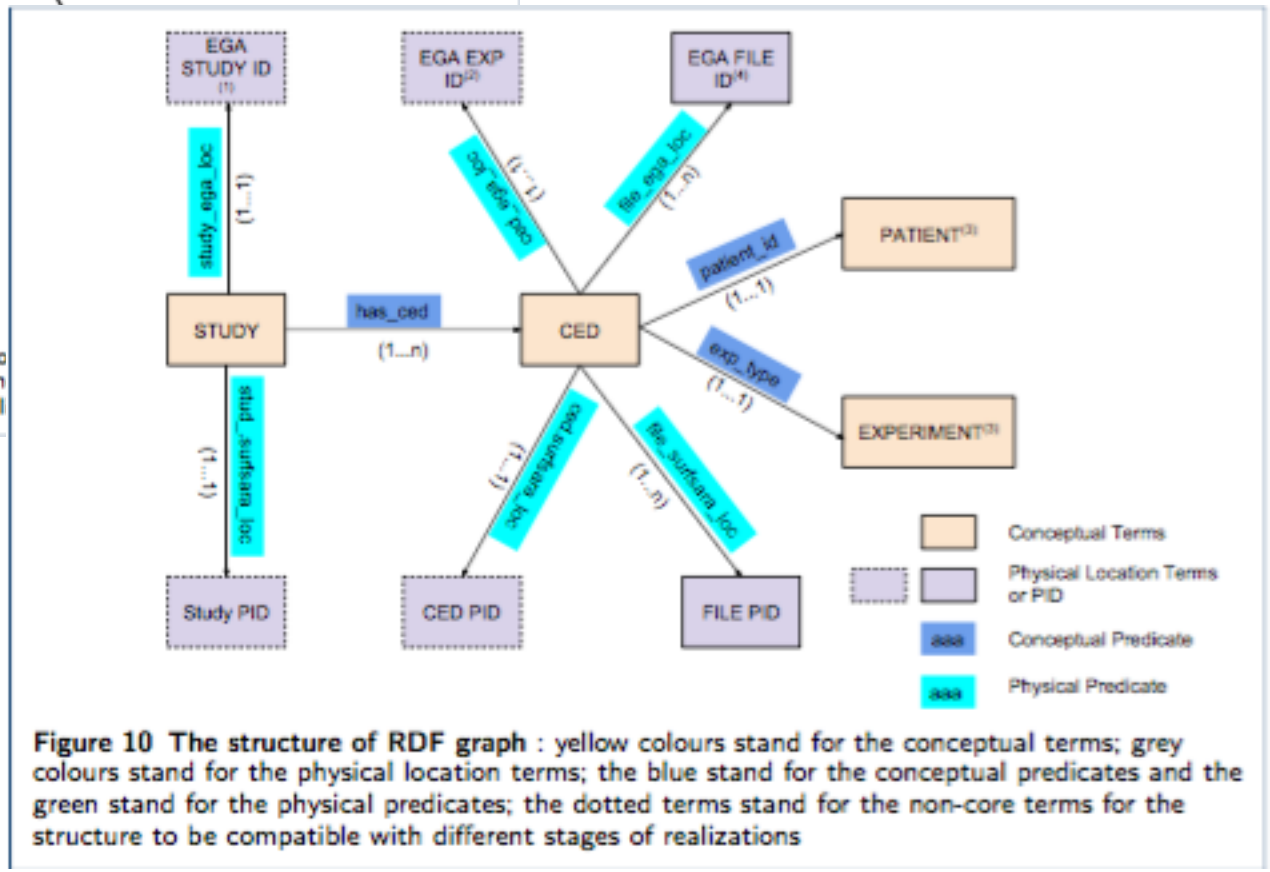
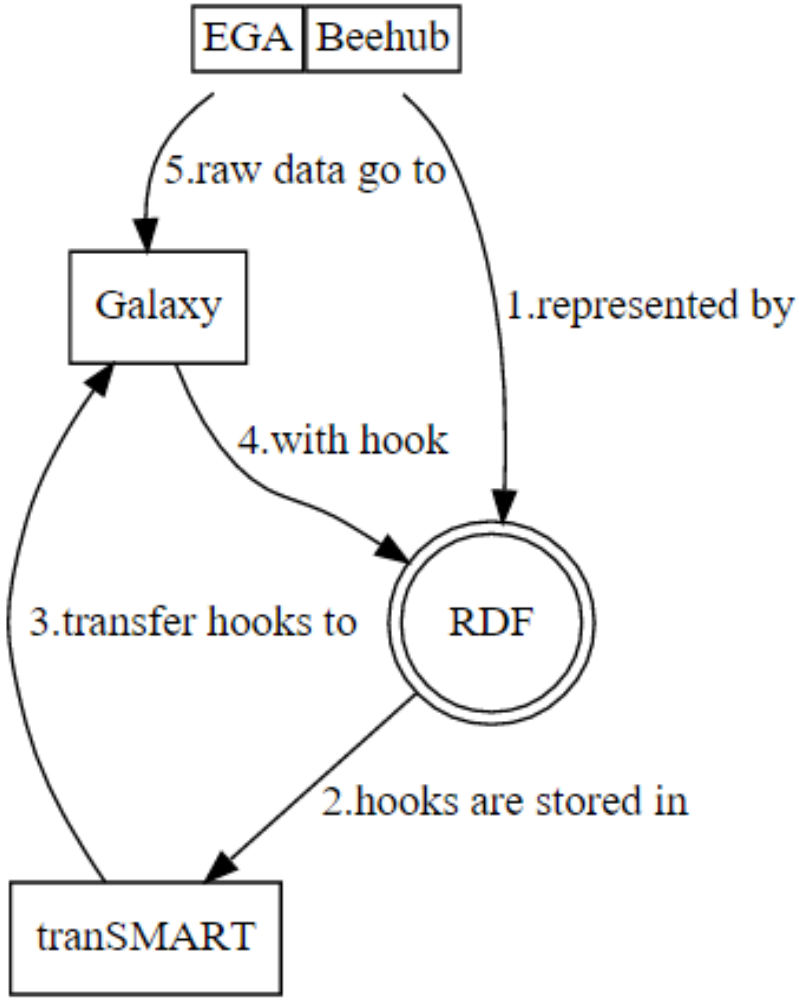


Figure 10 The structure of RDF graph : yellow colours stand for the conceptual terms; grey colours stand for the physical location terms; the blue stand for the conceptual predicates and the green stand for the physical predicates; the dotted terms stand for the non-core terms for the structure to be compatible with different stages of realizations

TraIT data infrastructure



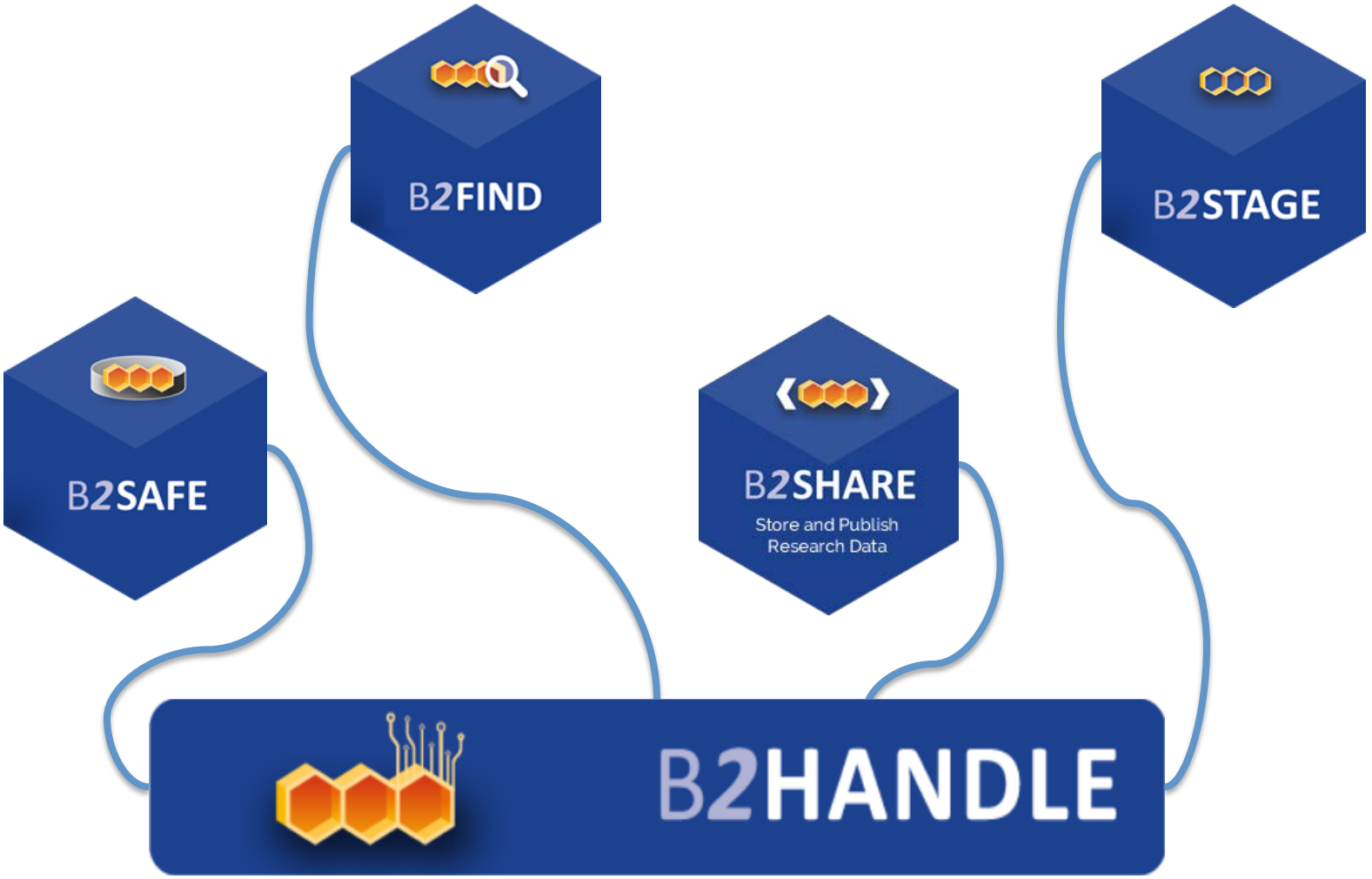
Chao (Cico) Zhang, VU
Sanne Ablen, VU
Jochem Bijlard, VU
Christine Staiger, SURFsara

Use Case 4: Enabling workflows

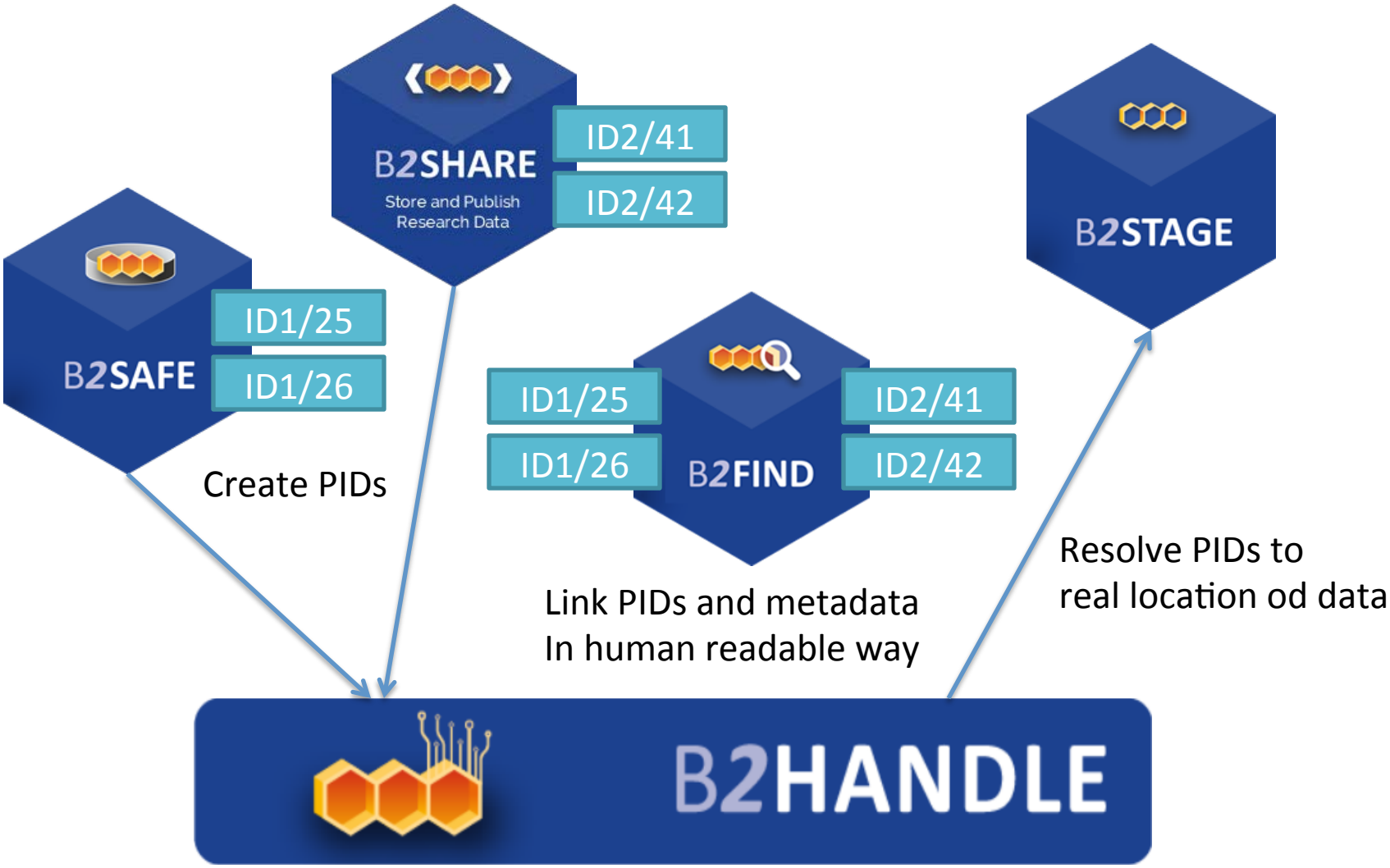
- Execute program hidden behind a PID
- Way to refer to workflows → reproducibility

```
In [16]: prefix = "841"

In [17]: suffix = "/5f6fb451-5841-11e4-9665-14109fe83170"

In [18]: ec.getValueFromHandle(prefix, "URL", suffix)
Out[18]: '/Users/christines/PIDs/helloWorld.py'

In [19]: pid = subprocess.Popen([sys.executable, ec.getValueFromHandle(prefix, "URL", suffix)])

In [20]: Hello World!
```

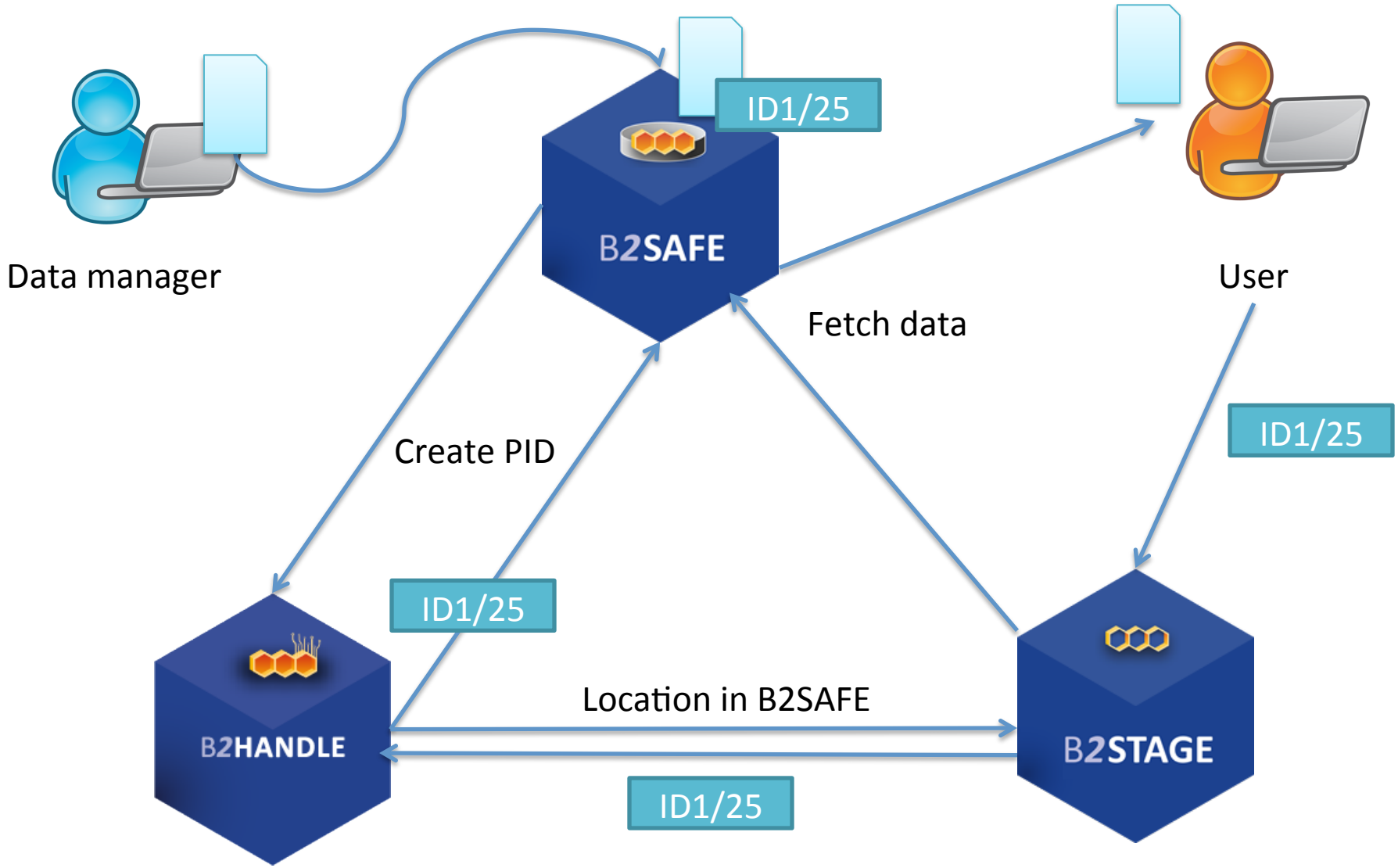
PIDs in EUDAT – Why?



PIDs in EUDAT – Why?



The data managers' workflow



PID systems

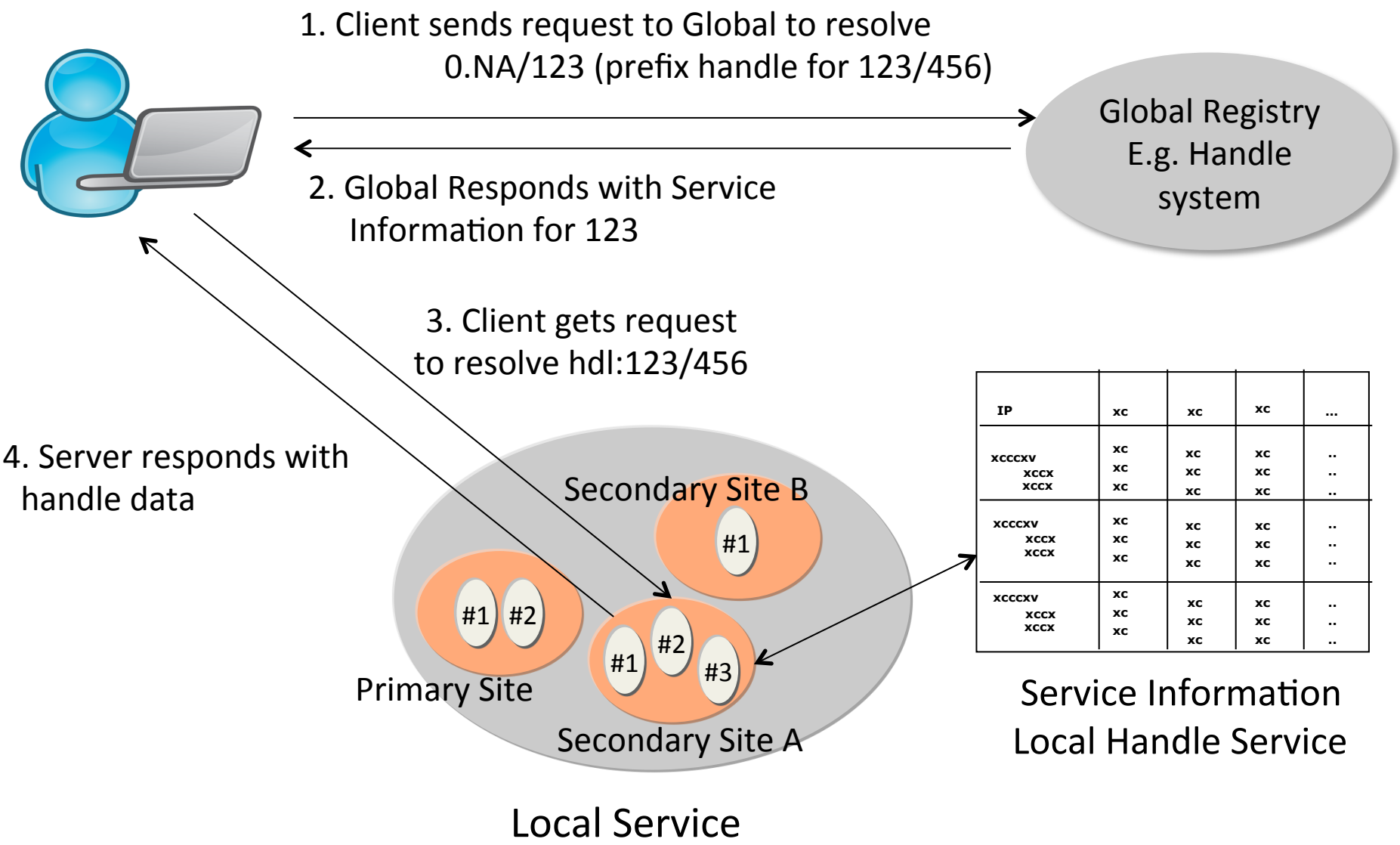
Resolution and the PID pattern

Exercise

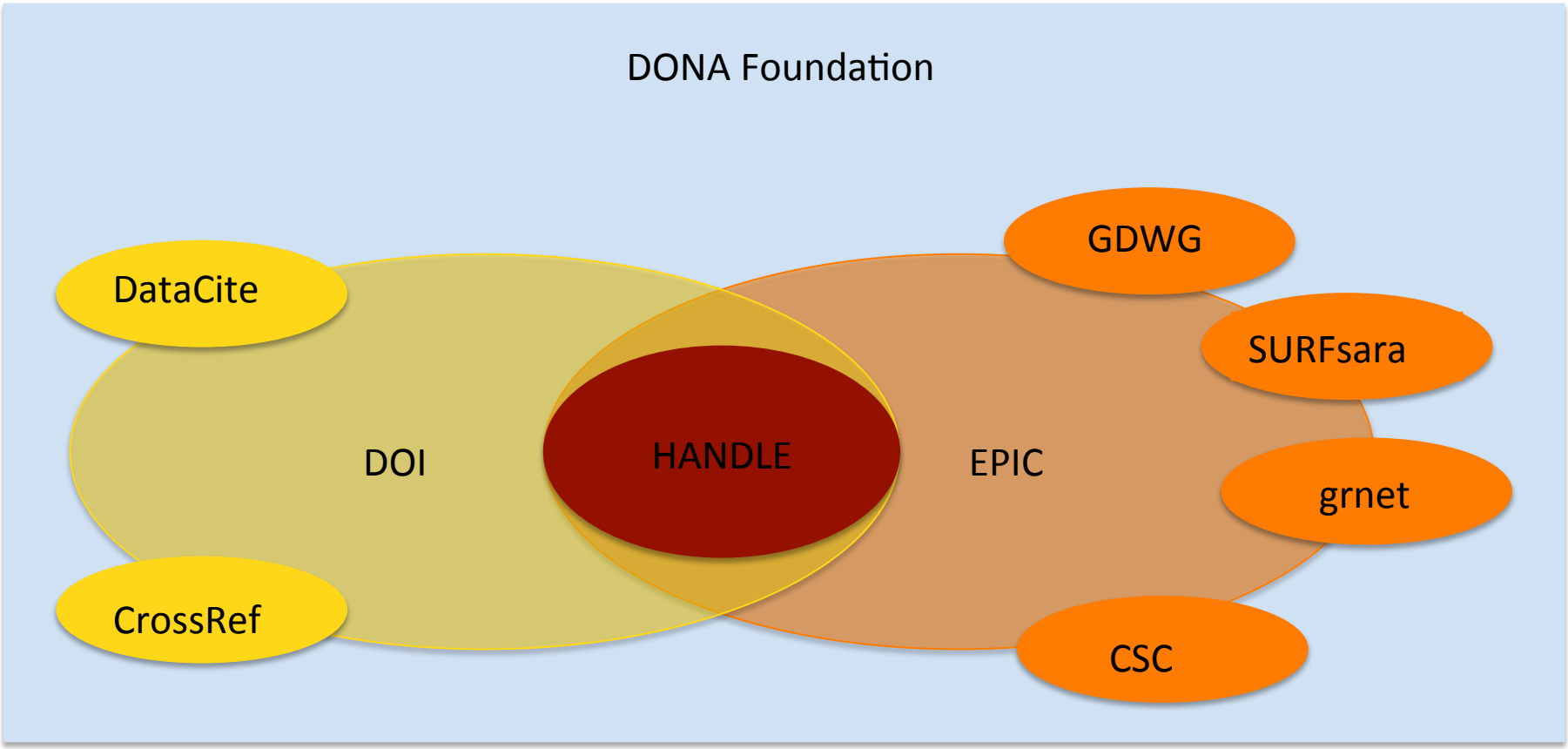
- Resolve the PIDs
- What happens if you resolve a PID with a foreign resolver?

http://hdl.handle.net/21.T12995/PID-training

Resolving PIDs



PID systems and issuing authorities



PID systems and issuing authorities

- **URN:NBN**
  - Policies: PID is persistent and the data it is dereferenced to
  - Wants to be independent from transfer protocols
    → Currently all identifiers start with *http*
    Might change in the future

- **DOI**
  - Policies: PID is persistent, data not
  - Based on the handle system
  - Datacite, Crossref are prefix issuing authorities
  - Requires extra metadata, stored in another database

- **Both:**
  - PIDs **point to a landing page**, not the file itself
    → **Taylored towards data citation**
  - User needs to provide **a minimum set of metadata (Dublin Core)**

PID systems and issuing authorities

- **ePIC (European PID consortium)**
  - Policies: PID is persistent, data is not
  - PIDs can point to anything
  - Based on the handle system
  - **Taylored towards data identification and resolving**

- **DONA foundation (www.dona.net)**
  - Maintains global handle registry
  - Partners:
    - CNRI (developer of the handle system)
    - GDWG (main partner in ePIC)
    - International DOI foundation (IDF)

The Handle system

- Metadata: You can create your **own keyword-value pairs** and store them with the PID

- EUDAT Policies:
  - Handles to be maintained beyond project life time
  - Enforce stability of PIDs to justify trust in them
  - Handles can point to anything
  - Handles can also be removed, they are not per se persistent

  …

  → Great flexibility for adjusting the system towards your own needs

  → EUDAT provides implementations for replica tracking

  → You have to think even more carefully about how you want to facilitate data management

For whom?

- PIDs allow to make a **distinction between data users and data managers**
  - Data users get a PID and can directly access the data, or the metadata stored with the PID
  - Pipelines can programmatically access the metadata and start specific applications

- Requires some serious thoughts about **data organisation** and developing the **code to put data policies into practice**, including code **maintenance**

  → For **bigger research groups or consortia** working in a **distributed data environment**

  → For **repositories** who are in need of a host for their PIDs

Step by Step:
Using the B2HANDLE python library

- Register data with a Handle
- GET the details of a Handle
- Modify a Handle record
- Link two files on PID level
- Reverse look-up (not possible via normal Handle API)

www.eudat.eu

Thank you

Authors

Themis Zamani, GRNET
Willem Elbers, CLARIN
Christine Staiger, SURFsara

Contributors

Ellen Leenarts, DANS
Kostas Kavoussanakis, EPCC

EUDAT receives funding from the European Union's Horizon 2020 programme - DG CONNECT e-Infrastructures.
Contract No. 654065