Persistent Identifiers

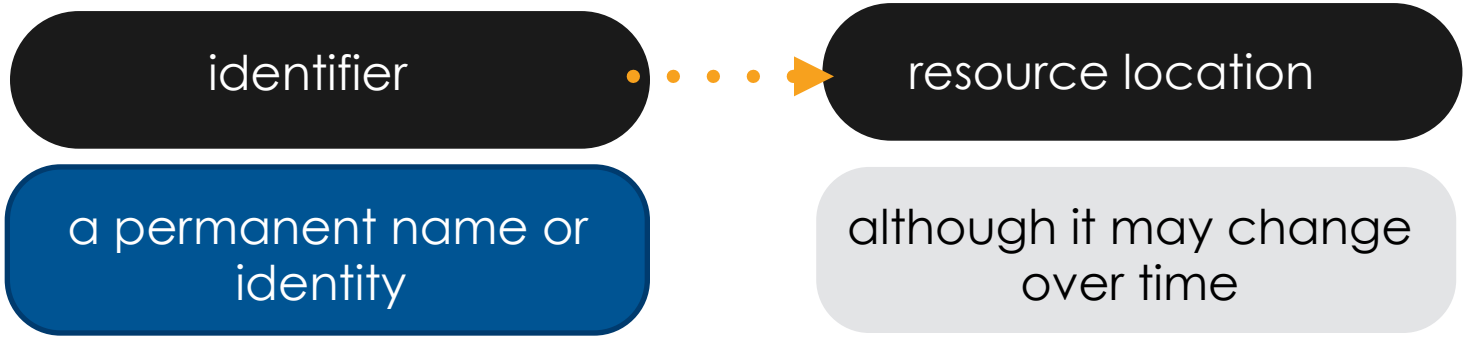
Data registry and transfers with Handles

Christine Staiger
SURFsara

European Life Sciences Infrastructure for Biological Information
www.elixir-europe.org

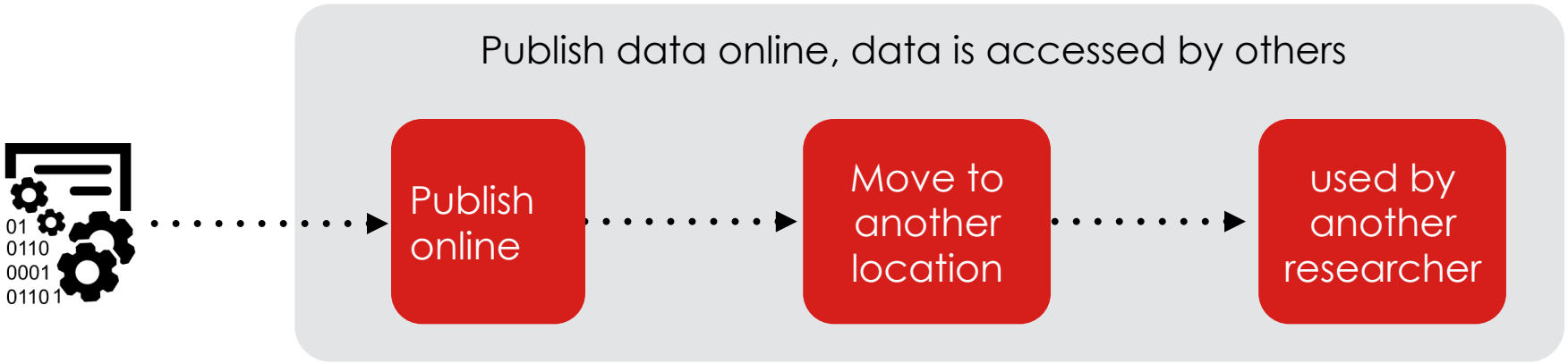
What do we know about Persistent Identifiers?

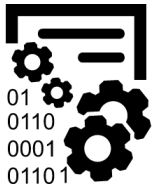
- A Persistent Identifier (PID) is an identifier that is effectively permanently assigned to a resource.



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

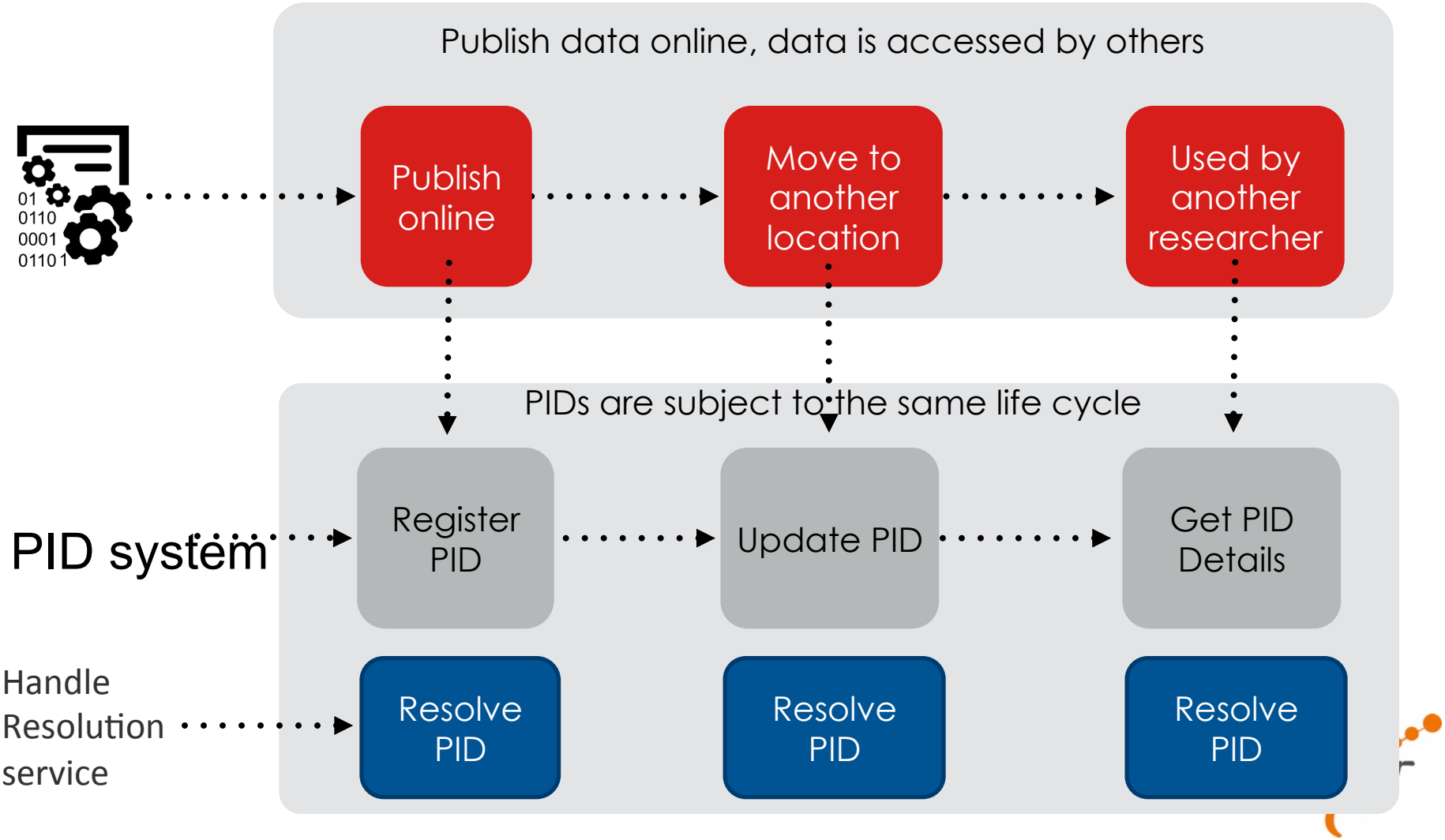
Simple data life cycle, linearised





- 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



Handles, DONA, DOI, EPIC

The Handle system

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

- Policies: **Do it yourself**!

  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

  → You have to implement all by yourself

PID systems and prefix issuing authorities

- **DOI - Policies**
  - PID is persistent, data not
  - PIDs **point to a landing page**, not the file itself
  - Extra metadata required and stored externally
- Well accepted amongst researchers
- **Datacite, Crossref** are prefix issuing authorities
- Taylored towards the needs of **repositories and journals**

  → **Publishing of data and articles!**

- **ePIC (European PID consortium) - Policies**
  - PID is persistent, data is not
  - PIDs can point to anything
- Prefix issuing authority

ePIC
Persistent Identifiers for eResearch

elixir

PID systems and issuing authorities

DONA foundation

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

Dataset registry - Exemplar

Technology and scope

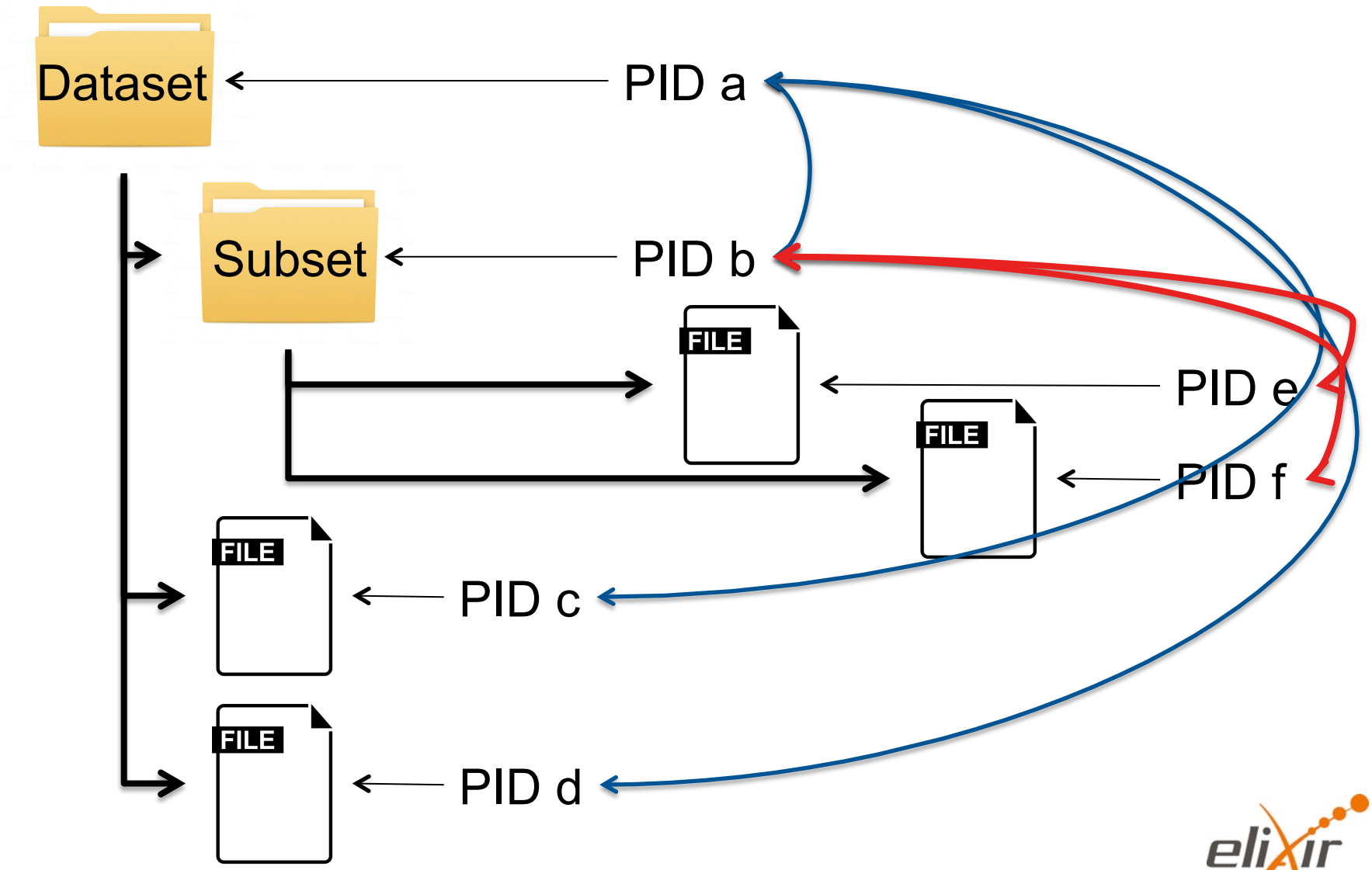
Scope:

- PIDs for management of data on system level
- Enable creation of Handles and resolving of Handles on the client side of gridFTP
- Globus-url-copy
- Calls to Handle system via B2HANDLE python library

→A python script

- User will get access to data by other identifiers or via portals

Data structure



Handle.Net®

Handle Values for: 21.T12995/A890EC3E-E947-11E6-A26B-040091643BEA

Index	Type	Timestamp	Data
1	URL	2017-02-02 14:01:25Z	/home/admincentos/Test2/SubCollection/
2	TYPE	2017-02-02 14:01:25Z	Folder
3	PROTOCOL	2017-02-02 14:01:25Z	gsiftp
4	SITE	2017-02-02 14:01:25Z	nlnode.elixirgridftp-sara.surf-hosted.nl/
5	PARENT	2017-02-02 14:01:25Z	21.T12995/A866A7A8-E947-11E6-A26B-040091643BEA
6	CHILDREN	2017-02-02 14:39:06Z	21.T12995/A8A3075C-E947-11E6-A26B-040091643BEA,
100	HS_ADMIN	2017-02-02 14:01:25Z	handle=0.NA/21.T12995; index=200; [create hdl,delete hdl,

Trace all data belonging to a dataset given:

- A PID of dataset
- A PID of a data object in the chain
- Reverse lookups:
  The local Handle server and
  a checksum of a file in the chain

Exemplar functions

Upload and synchronisation

- Supports currently **gridFTP**
- **Upload:**
  1. Globus-url-copy of dataset
  2. Assign PID to dataset
  3. Recurse in the uploaded file tree and assign a PID to every file and folder
  4. Introduce parent-child relations
- **Synchronisation:**
  - Local copy synced with copy on gridFTP server
  - Globus-url-copy sync-level 0
  - Label new data with PID
  - Fix parent-child relations

Download and update of location

- **Download given a PID:**
  1. Resolve PID
  2. Globus-url-copy to destination (recursively)
- **Update URL:**
  - Data is moved on the gridFTP server
  - Update URL field recursively for whole dataset

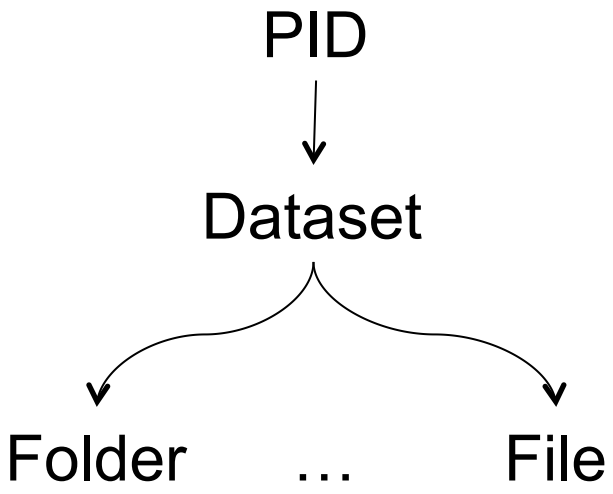
Exemplar limitations

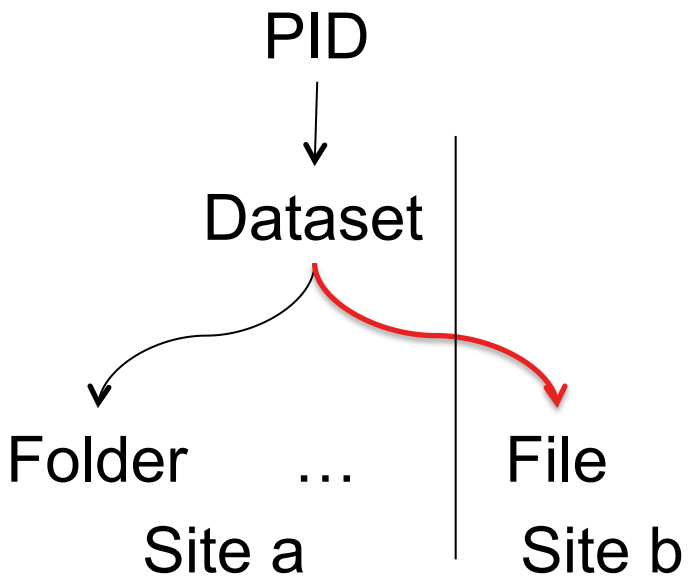
PID advantages and disadvatages

- Dataset is traceable
  - Technology/service highly sustainable
  - PID stays the same even if data location changes → stability in the ELIXIR data network
- Integrity checks
  - supported via an external system → no need to have access to every copy of the data in the ELIXIR network
- Data of one dataset can potentially reside at different locations and is still traceable
- Handle best practice: **use uids**, no human-readable suffixes

Limitations

- All data under <path dataset>
- Globus-url-copy PID → works





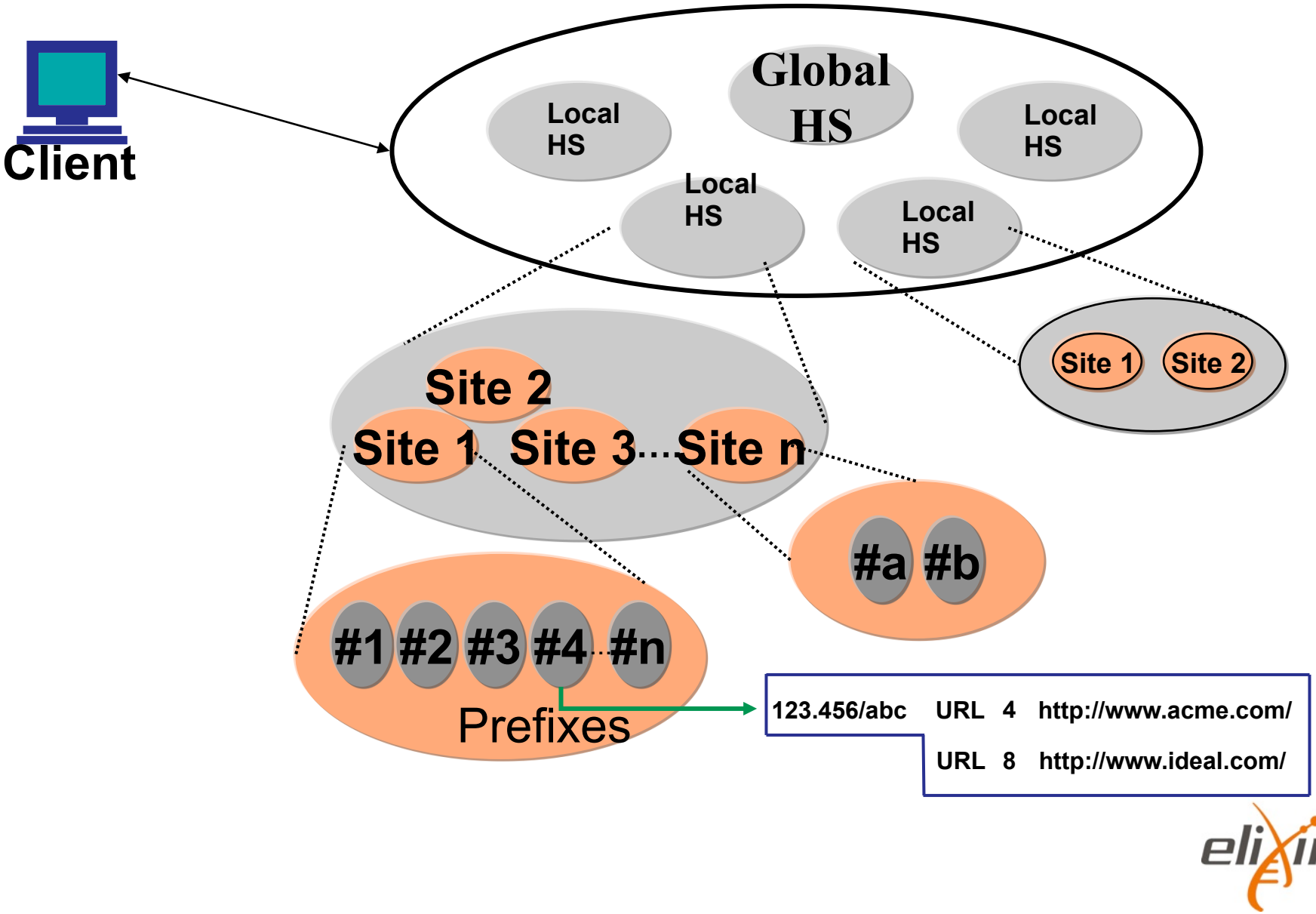
- Data still traceable
- BUT
  - Recursive resolving
  - URLs of all PIDs need to be checked

Thank you!

Code here:
https://github.com/chStaiger/ELIXIR-gridftp-PID

The Handle system: technical details

Resolution system



Resolving PIDs