

Persistent Identifiers

Adam Carter (EUDAT/EPCC)

0000-0001-9544-9211

including content originally created by

Tobias Weigel (DKNZ)
and Larry Lannom (CNRI)



What I'll cover

- Why PIDs?
- What are PIDs?
- How to use PIDs

Q. Why PIDs?

A. Data Intensive Science

- Not only the **volume** is increasing, but also the **number** of digital objects
- Data generation is getting easier/cheaper
- A greater emphasis is being placed on *data as infrastructure* and the use of data as a new way to do science: datascopes
- In science, data is increasingly shared across communities

So, Why PIDs?

- These objects need to be (globally) addressable so that they can be reusable
- Once data is addressable, it
 - can be found more easily by you, other people, and computers
 - can be cited
 - can be linked together



WHAT ARE PIDS?

Persistent Identifiers come in various formats

- 10876/abc123
- 10.1594/WDCC/CMIP5.NCCNMpc
- ark:/13030/tf5p30086k
- <http://purl.org/dc/elements/1.1>
- urn:lsid:ubio.org:namebank:11815

There are several different PID systems / infrastructures

- Handle System
- Digital Object Identifier (DOI)
- Archival Resource Key (ARK)
- Persistent URL (PURL)
- Life Science Identifier (LSID)
- Uniform Resource Name (URN)

There are several different PID systems / infrastructures

- **Handle System** 📱
- Digital Object Identifier (DOI)
- Archival Resource Key (ARK)
- Persistent URL (PURL)
- Life Science Identifier (LSID)
- Uniform Resource Name (URN)

There are several different PID systems / infrastructures

- **Handle System** 📱
 - <http://hdl.handle.net/1234/56>
- **Digital Object Identifier (DOI)**
 - <http://doi.org/10.1002/prot.9999>
- **Archival Resource Key (ARK)**
 - <http://www.nmah.org/ark:/13030/tf5p30086k>
- **Persistent URL (PURL)**
 - <http://purl.oclc.org/keith/home>
- **Life Science Identifier (LSID)**
 - `urn:lsid:<Authority>:<Namespace>:<ObjectID>[:<Version>]`
- **Uniform Resource Name (URN)**
 - `urn:isbn:0451450523`

PIDs identify resources

- The resource, generally, is a black box which can contain
 - Data
 - Metadata
 - Document
 - Software code
 - ...
- PIDs can also identify things
 - e.g. a species, in the case of an LSID
- PIDs normally *point* to the thing they identify

PIDs are globally unique

- Very useful from the point of view of someone following using the PID to find a resource
- ...but can make creating/assigning PIDs a little more complicated

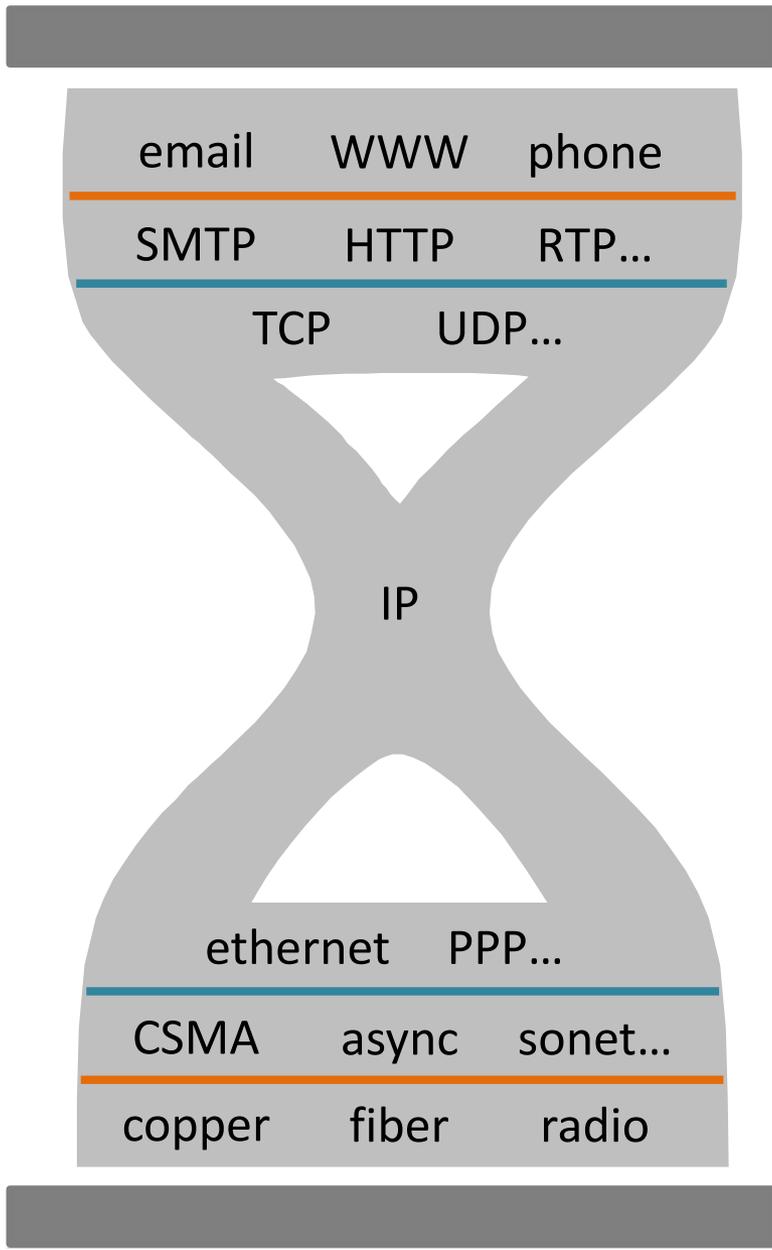
PIDs are persistent over time

- ...by design
- Ensuring they continue to point to the right thing requires (some) work on behalf of the entity responsible for the data
 - but the design of the handle system, for example, is such that this is made easier by decoupling the identifier from both *where* the data is stored and *who* is currently storing it

PIDs establish a redirection layer

"All problems in computer science can be solved by another level of indirection" – David Wheeler

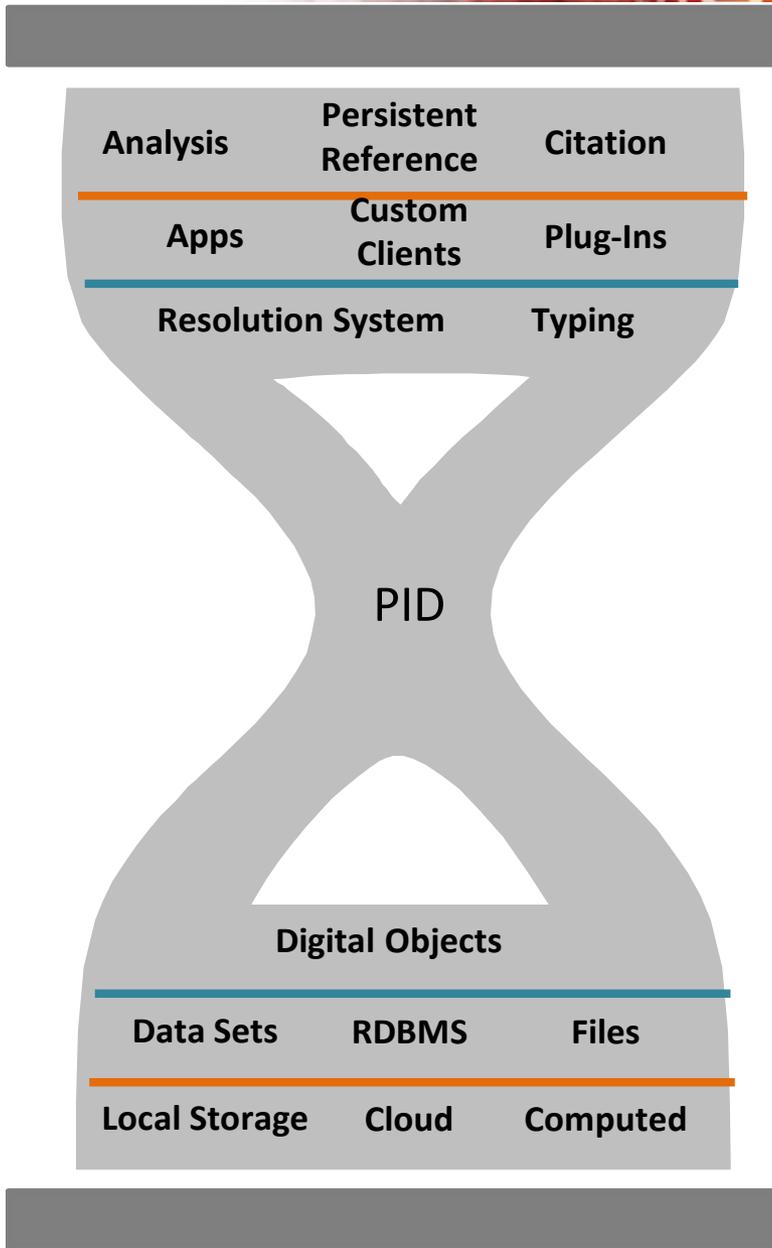
- PID can point to a URL which points to the digital object
- If the digital object is moved, its ownership is changed, or the organisation of the objects is changed, the URL is often changed...
 - ...but the PID can be made to point to the new URL



Value Added Services

Internet Protocol Suite

Network Technology



Value Added Services

Persistent Identifiers

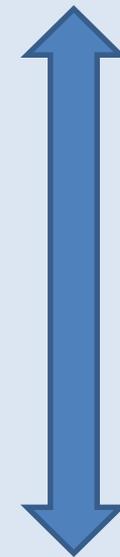
Data Sources

Handles go beyond PIDs

Minimal metadata (key-metadata)

- Checksum
- PID creation time stamp
- Graph structure (links)
- Collection membership

static



dynamic

PIDs: Advantages

- Persistent Identity via Indirection
 - Static Identity via Indirection
 - Data on networks moves
 - Ownership/responsibility change
 - Formats change
 - Embedded Ids
 - For data object in hand – current state data
 - Updates
 - New related entities
 - Networks of Persistent Links
 - Data / metadata links
 - Provenance chains
 - Inheritance across a broad set of entities

PIDs: Disadvantages

- Extra level of effort / cost on creation
 - Analysis – what to identify / granularity
 - Coordination across organisations
 - Need to maintain resolution system
- Persistence requires sustained effort
 - Organisational discipline
 - Technology necessary but not sufficient
- Analyse cost/benefit ratio
 - Don't start unless its worthwhile
 - Is your data worth it?

What are requirements for a “good” identifier?

- Not based on any changeable attributes of the entity
 - Location
 - Ownership
 - Any other attribute that may change without changing identity
- Opaque, preferably a “dumb number”
 - A well known pattern invites assumptions that may be misleading
 - Meaningful semantics invite intellectual property disputes, language problems
- Unique
 - Avoid collisions
- Nice to have
 - Human-readable
 - Cut’n’paste-able
 - Fits with common systems, e.g., URI specification
- All of the above contribute to persistence

How to use a PID

- If you've got one for your data, *write it down / type it in*
 - Online, in publications, in other linked data
- If you've got one from someone else's data
 - Use it to get the data, or to refer to it
 - Handles resolve:
 - <http://hdl.handle.net/1234/56>

How to get a PID

- The easy way: Upload your data to EUDAT
 - Saving to a DSpace repository might get you one too, depending on how it's been configured
- How EUDAT gets the PID:
 - An EPIC service, provided by SARA, conforming to the RESTful EPIC API
 - <http://www.pidconsortium.eu/>
- EUDAT members also have access to a Python script which wraps the web service

EPIC API (Version 2)

- «root»/
- └─discovery/
- └─NAs
- └─«NAsegment»/ ← PREFIX
- └─└─handles/
- └─└─└─«LNsegment»/ ← SUFFIX
- └─└─└─└─...
- └─└─profiles/
- └─└─└─«profile»
- └─└─status/
- └─└─└─«id»
- └─templator

The Handle Resource in the Handle API

«root»/NAs/«NAsegment»/handles/«LNsegment»/

- GET Returns a Handle
- PUT Submits a Handle.
- DELETE Deletes the handle.
- POST Accepts a Value Set which **MUST NOT** contain a handle member.
 - In this case, the «LNsegment» is interpreted as a suffix template. A suffix template is a suffix containing exactly one unescaped '*' character. This character will be replaced with a unique string by the server, resulting in a new and unique handle.
 - Creates a new handle with the provided metadata. An HTTP/1.1 201 Created status is returned upon success, with the location of the new resource in the Location: response header. The new handle is returned in an X-Handle: response header, encoded as per [RFC5987](#) if necessary.



```
POST /NAs/10/handles/  
Host: example.com  
Content-Type: application/json  
...
```

```
[ { "handle" : "handleOne",  
    "values/": { "1": { "type": "URL",  
                        "data": "http://www.example.com" } } },  
  { "handle" : "handleTwo",  
    "values/": { "1": { "type": "URL",  
                        "data": "http://mail.example.com" } } } ]
```

IDs for People

- One kind of entity that you often want to link to or refer to is a person
 - Possibly as the subject of the data (with obvious data protection caveats),
 - but more commonly as the creator / contributor
 - Can encourage credit, credibility
- Examples:
 - ORCID, International Standard Name Identifier

Summary

- Persistent Identifiers *uniquely identify* data, and make it addressable
 - Facilitates: citation, linking, unambiguous reference
- Several ID systems exist, but their goals are similar, and they seem to be converging
- Persistence requires some effort but this is often not on the part of the end-user
- Upload your data to EUDAT to get a PID
 - at the moment, this is through a member community, such as VPH