

<a href="

">

Permanent Identification

Permanent Identifiers

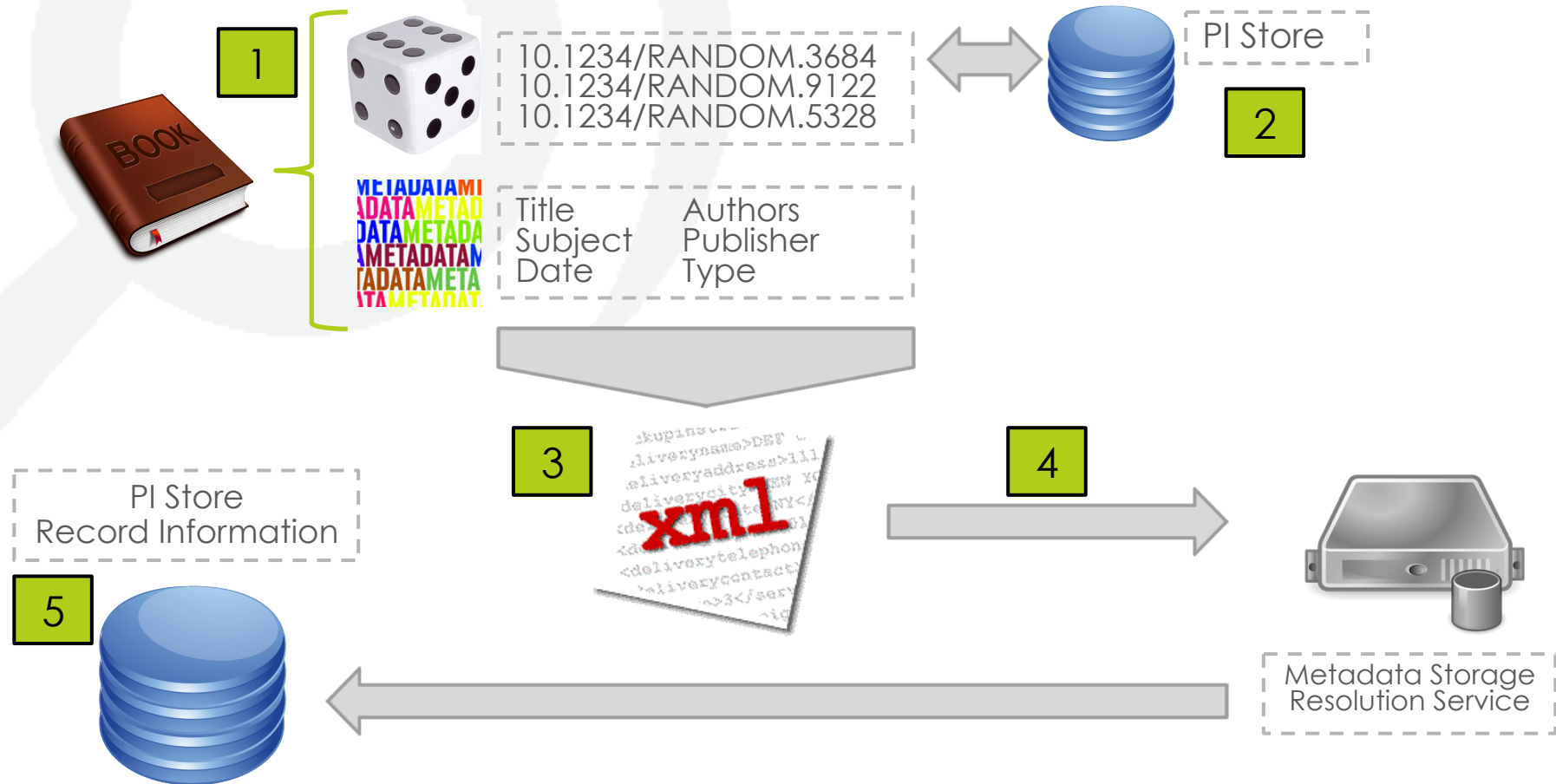
- How to point objects that can move around?
 - By adding an extra level of indirection
 - Permanent ID → Resolver → Final location
 - Resolvers save the ID, the location and, most of them, metadata describing the object
- By creating Permanent IDs you commit yourself to offer the permanent storage and metadata **maintenance!**



How will Invenio manage it?

- ▣ By now, persistent IDs are saved in MARC:
 - ▣ 0247_ \$\$2DOI\$\$a10.1103/PhysRevC.85.035210
- ▣ What about minting them by ourselves?
 - ▣ Manual minting
 - ▣ 'Batch' minting
- ▣ Our use-case, by now:
 - ▣ System: DOI
 - ▣ Service Provider: DataCite
 - ▣ Objects: HEPData

Permanent Identifiers



Technically...

- 1 Random names have many advantages (but you may prefer meaningful ones...)
- 2 PI Store structure: (+ *format?*)

Type	Record ID	Object ID	ID System	Provider	Name	Date	Status
Record	12345	-	DOI	DataCite	10.1234/ RAND.438	14/Nov/ 2012	0
Object	-	45678	ARK	EZID	12025/65 4xz31/s3/	15/Nov/ 2012	4
...							

Technically...

3

Metadata Output format:

- Bibformat-ed per:

- type
- provider
- set of metadata
- ...

```
20 from textutils import transliterate_ala_lc
21 from dateutils import guess_datetime
22
23 def format_element(bfo):
24     """
25     Generates the Datacite Metadata XML with the kernel information:
26     http://schema.datacite.org/
27
28     Mandatory properties:
29     - ID (+type)
30     - Creators (+name)
31     - Title
32     - Publisher
33     - Year
34     """
```

4

Provider's API

- Our case: DataCite offers a nice RESTful API
- Living in miscutil, *should we move it into a module?*

```
227
228 def post_metadata(metadata, endpoint=ENDPOINT, username=USER, password=PASSWORD, testing=TESTING):
229     """
230     Sends a new metadata set to complete an existing DOI.
231     Metadata should follow the DataCite Metadata Schema 2.2
232     http://schema.datacite.org/
233
234     @param metadata: XML format of the metadata
235     @type metadata: str
```

Technically...

5

Now that is minted:

- ▣ Update our pitstop
- ▣ Update the record information (and again: bibupload, bibindex...)
- ▣ What about minting while bibuploading the first time?
 - ▣ → You need the final URL to mint the DOI →
 - ▣ ← You need the DOI to fill the record ←

What else?

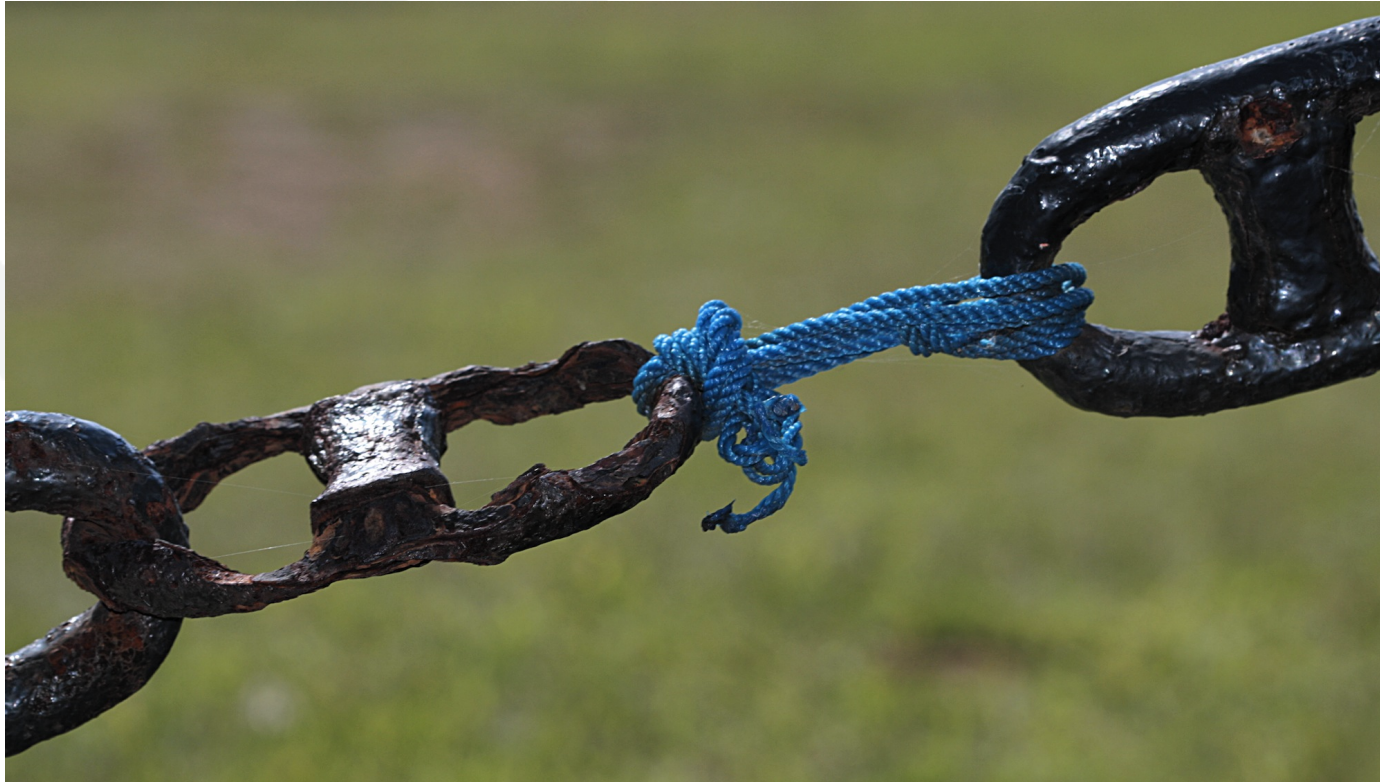
- ▣ Manual DOI minting
 - ▣ Let catalogers mint DOIs manually (from the Record Editor)
 - ▣ But don't let them modify them (sorry, this is **permanent!**)
- ▣ If the metadata changes:
 - ▣ Should trigger an update to our PI provider
 - ▣ On the fly? Save and schedule updates?
 - ▣ Which changes? Performance!
- ▣ External services can be non-reliable
 - ▣ Store the request and try? Retry? Again? Agaiiin?
- ▣ Versioning... !

It will come...

- Permanent person IDs
 - Open Researcher and Contributor ID

ORCID

Questions?



Thanks! 😊
