

# Invenio v3 data model and indexing

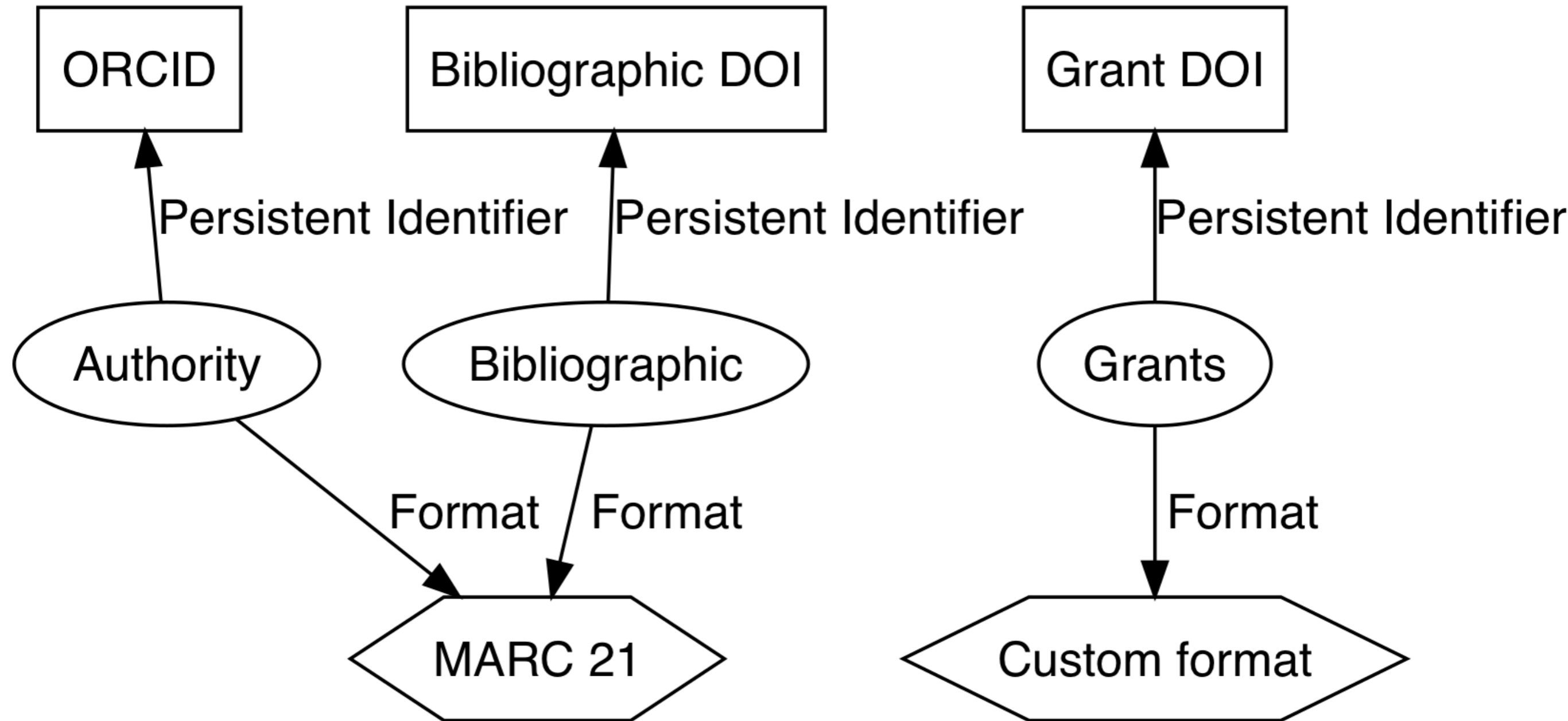
IUGW 2017

*Lars Holm Nielsen  
Nicolas Harraudeau*

# Outline

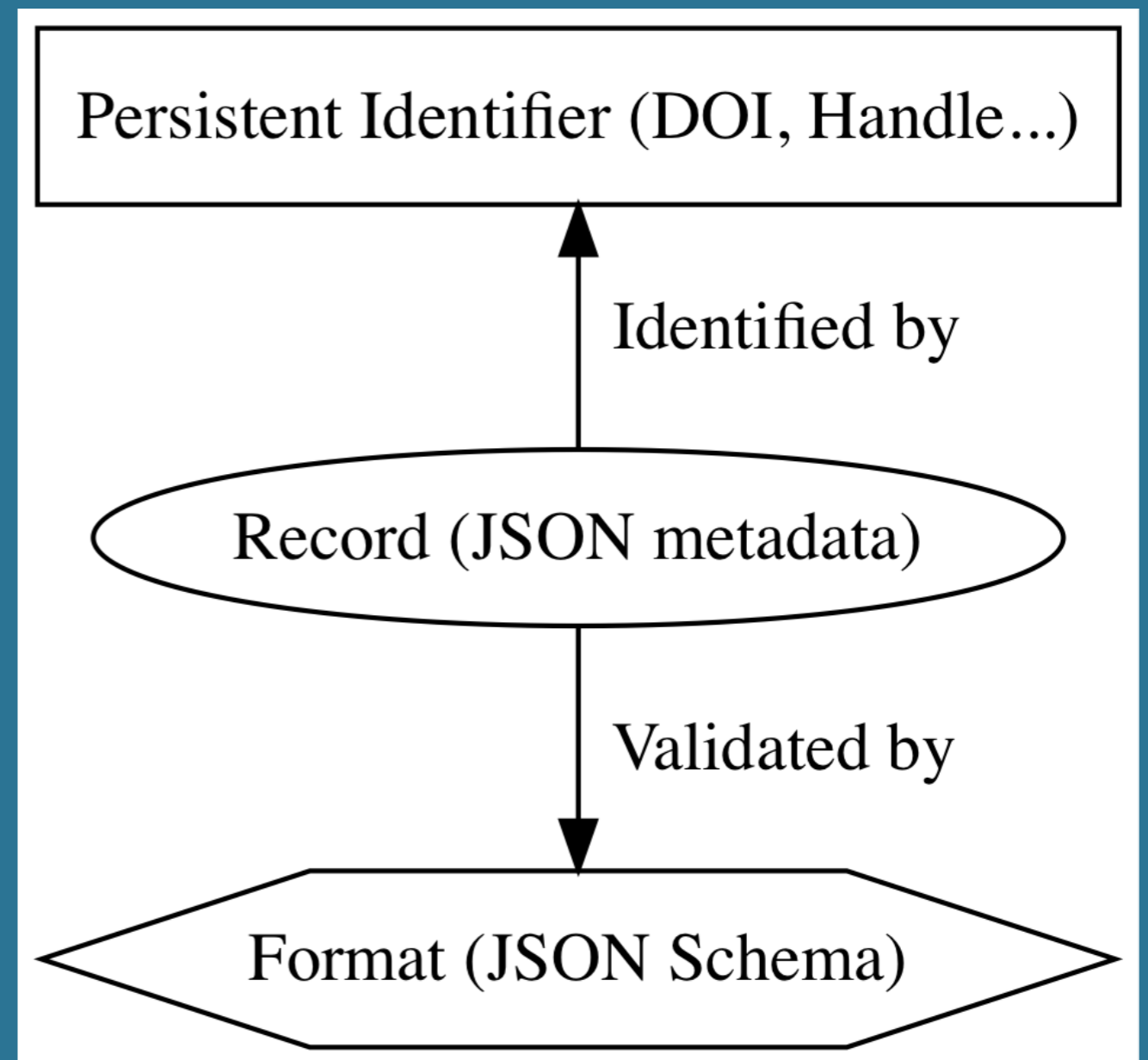
- Goal: see how Invenio data model can be customized.
- Plan:
  1. high level view
  2. a few commands + questions
  3. back to step 1
- Rule: take it easy and ask questions.

# Use cases



# Generic Invenio Datamodel

- **Optional MARC21** data model.
- Enables **custom** data models.



# Tutorial

- **Instructions:**

<https://github.com/inveniosoftware/iugw2017/tree/master/3-datamodels/>

- **Code**

USB KEY> iugw2017/3-datamodels/custom-data-module

# Installation

1. Go to your virtual environment
2. Install the custom-data-module
3. change config and reindex
4. create a few test records

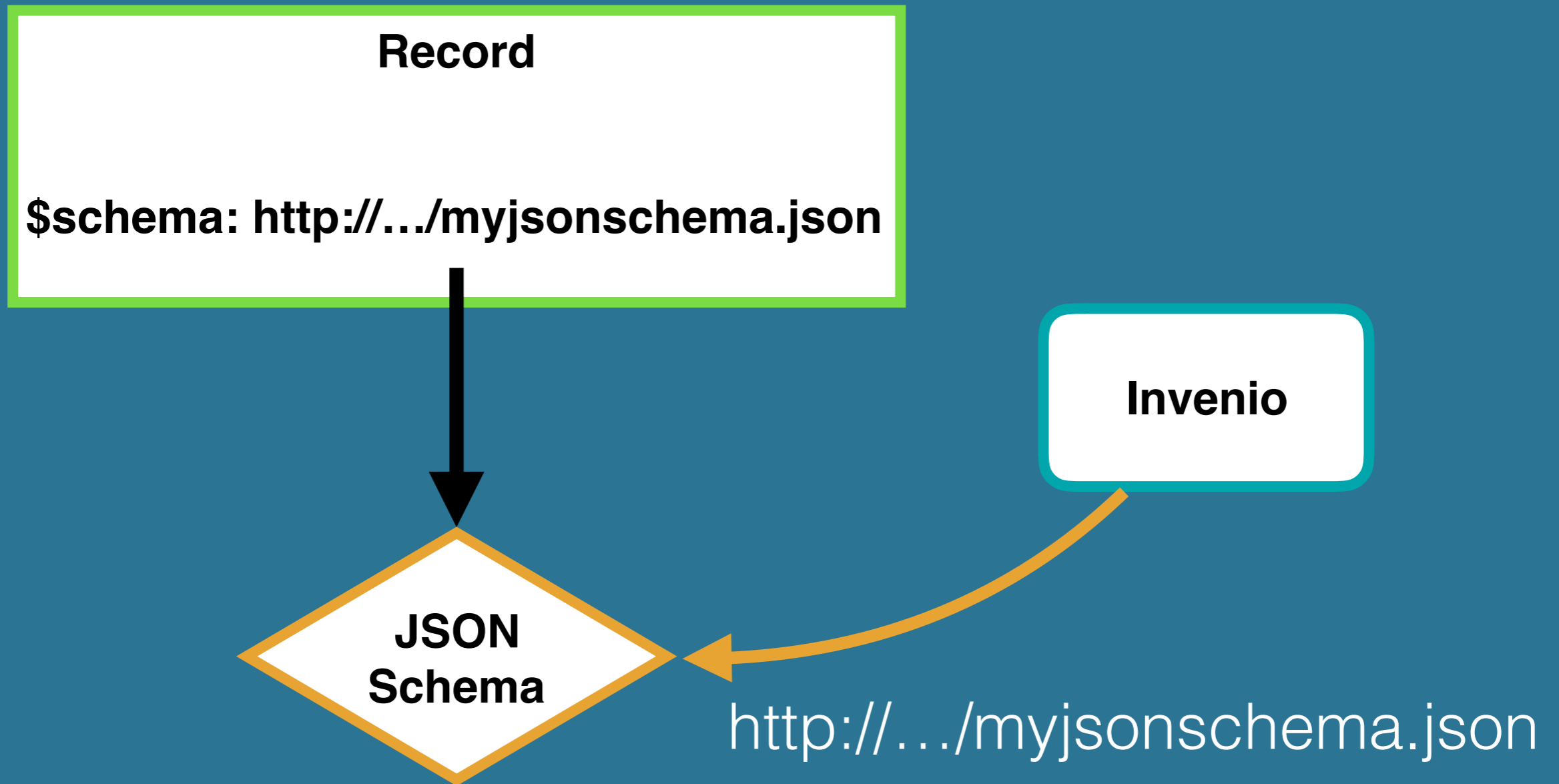
# QUESTIONS (first round)

# Record Storage

- Record:
  - Metadata = JSON
  - Date of creation and last update
- Validation = JSON Schema



# Records - JSON schemas

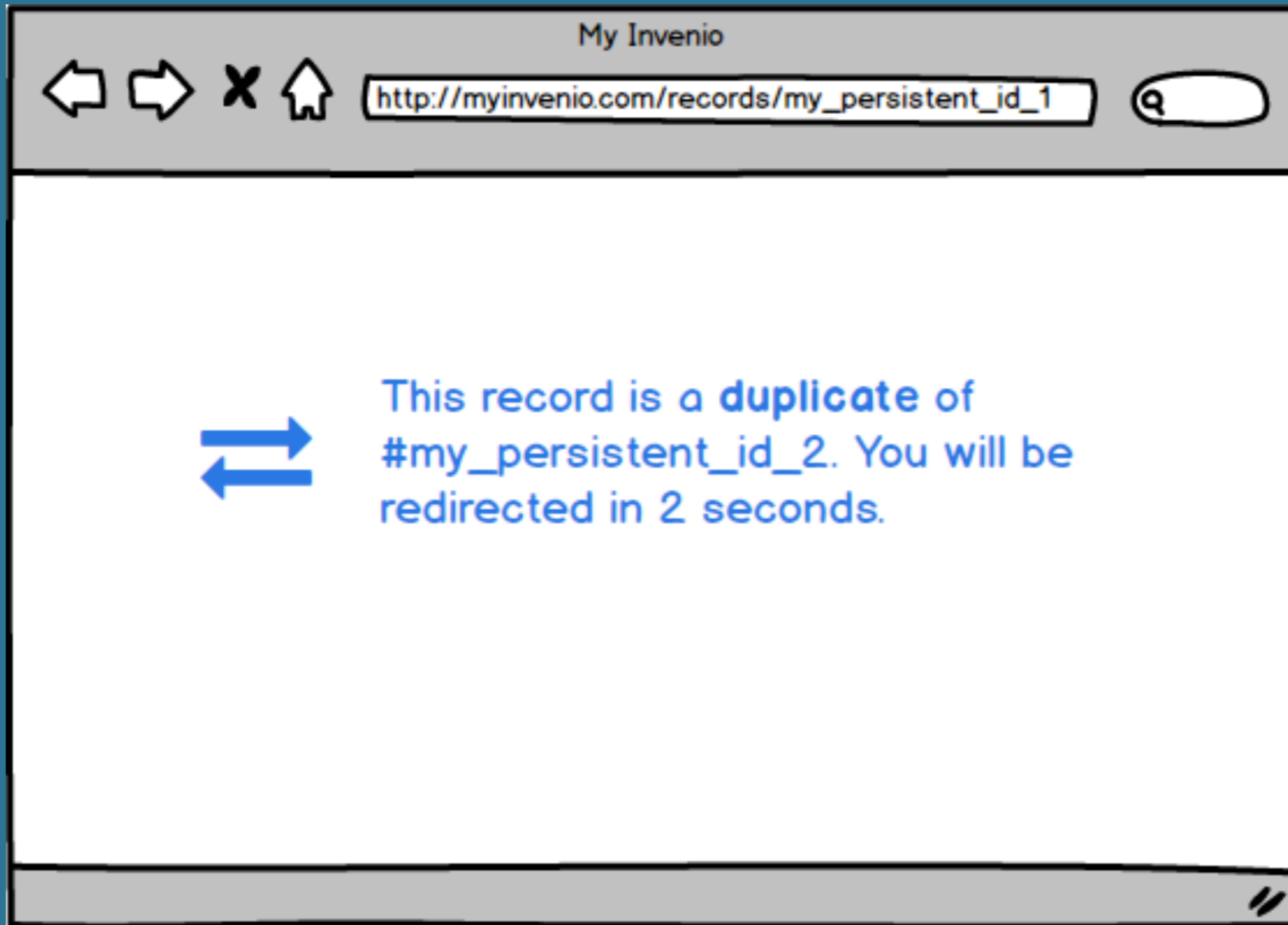


# QUESTIONS (round 2)

# Persistent Identifiers



# Persistent Identifiers



# PID exposed as URL

<http://records/<PID>>

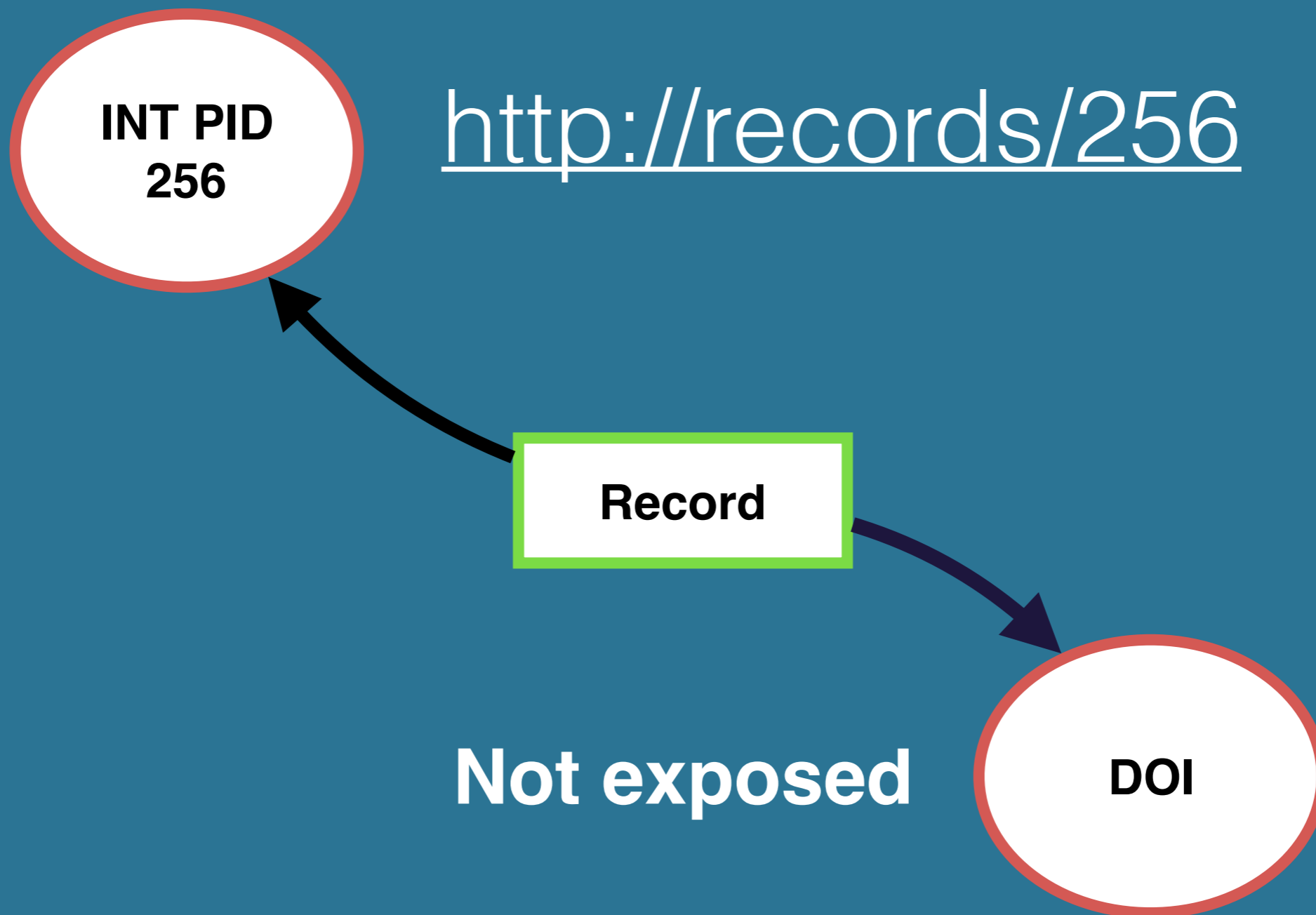
<http://authors/<PID>>

<http://grants/<PID>>

# Persistent Identifiers

- PIDs are giving access to records.
  - They are part of records URLs.
- Records disappear. PIDs remain.
- PID type separate types of records: Authors, Papers, Datasets, Grants, ...

# Multiple PIDs use case



# PID Minting

## Record

**control\_number: 54**

**doi: 10.6284/myinvenio.377042**

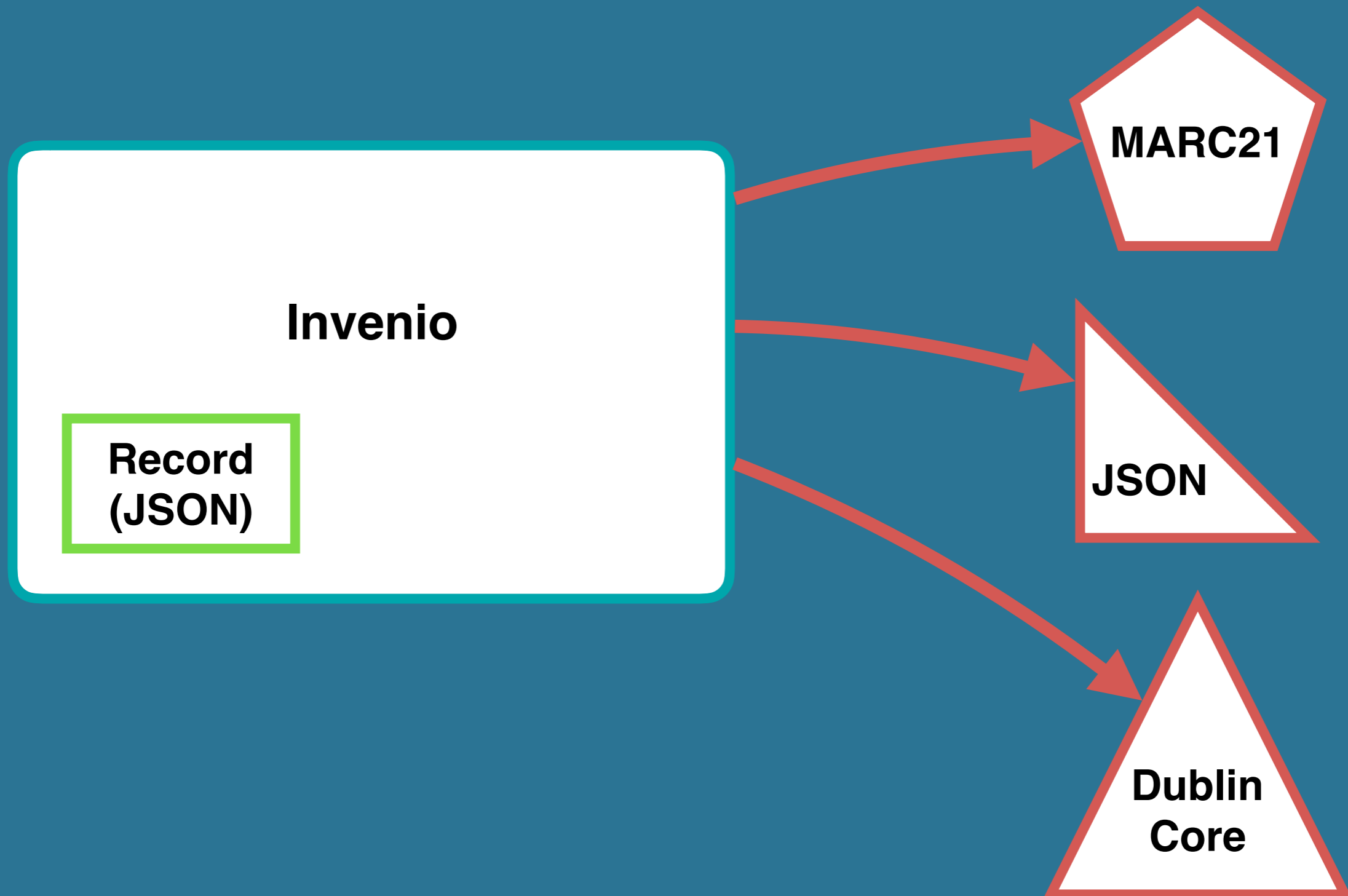


# QUESTIONS (round 3)

# REST API

- READ, DELETE, UPDATE Records with other programs

# Serialization



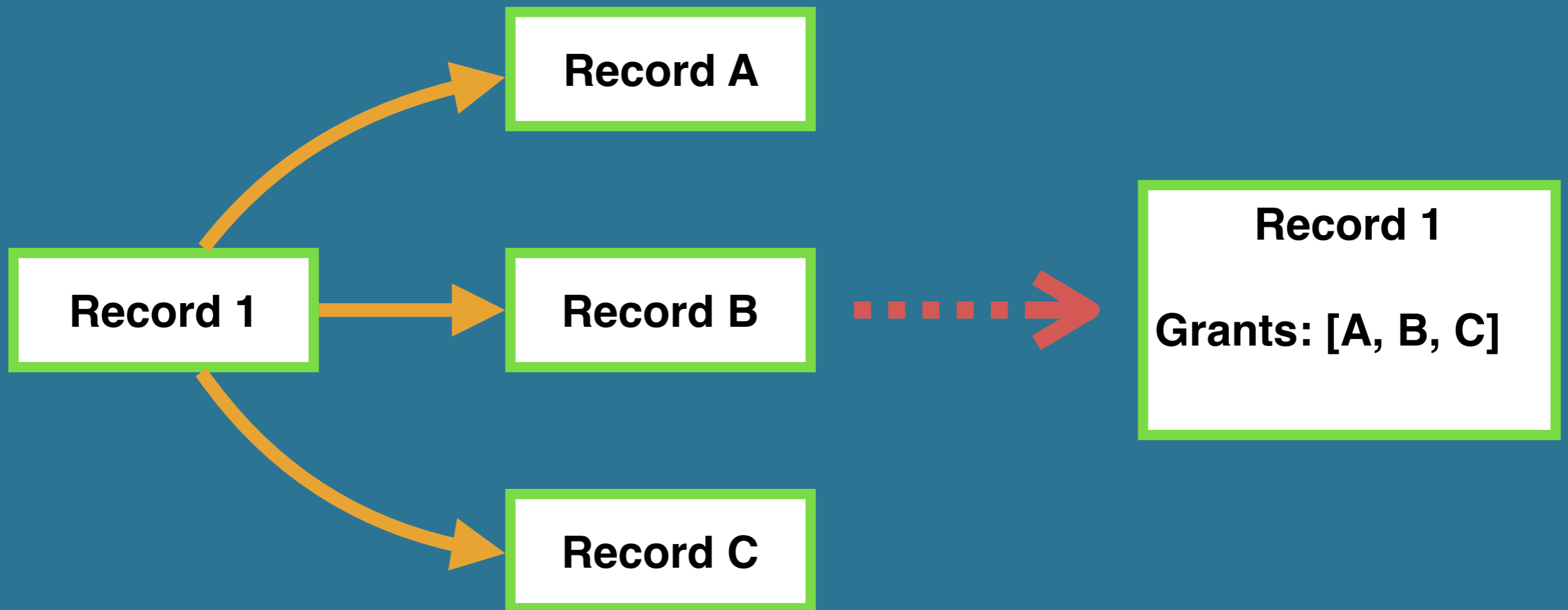
# QUESTIONS (round 4)

# Search

- Elasticsearch = power
  - example: searching for a specific language
- But it requires configuration

# QUESTIONS (round 5)

# Linking records



# LAST QUESTIONS