A Crosswalk in the Park?

Converting from MARC 21 to Linked Art at Yale University

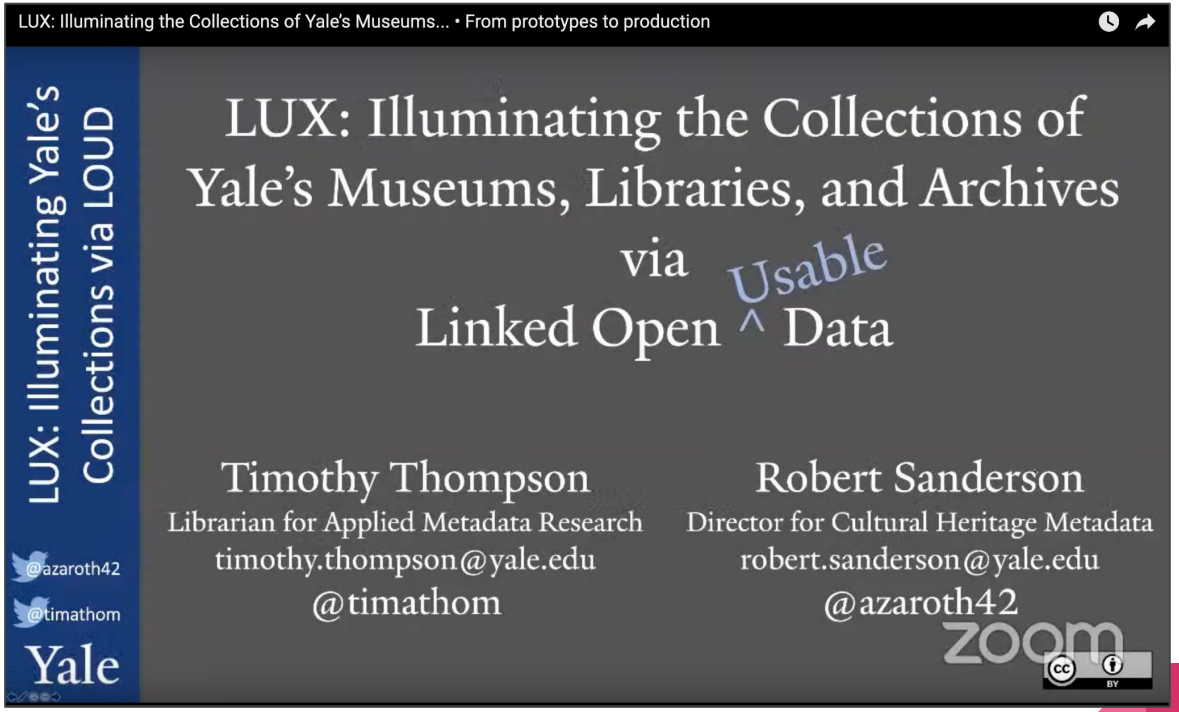
Semantic Web in Libraries | November 28, 2022

Martin Lovell (Yale University Library)
Timothy A. Thompson (@timathom@indieweb.social)

Outline

1. Yale, LUX, and Linked Art
2. MARC 21 to JSON-LD: Defining the Crosswalk
3. MARC 21 to JSON-LD: Implementing the Crosswalk

Background



https://www.youtube.com/watch?v=C4lAJHOs1gY

LUX Beta

LUX: Yale Collections Discovery

About LUX Open Access Help

Alison Saar, 1956-

Objects Made/Discovered | Works Created/Published | Works About

Snake Man

Agents: Alison Saar...

Supertypes: Engraving

Tobacco Demon

Agents: Alison Saar...

Supertypes: Engraving

Lost Boys

Agents: Alison Saar...

Supertypes: Engraving

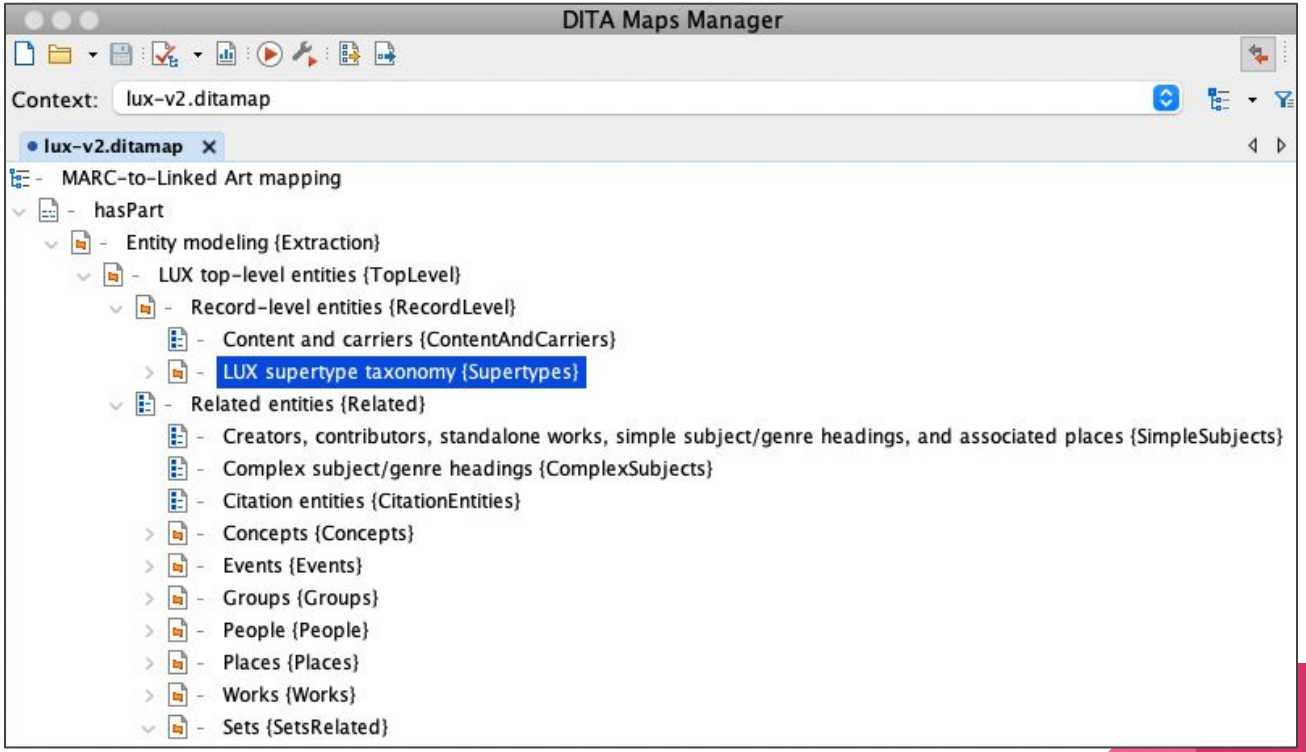
preferred terms	Saar, Alison Alison Saar Saar, Alison (American painter, sculptor, born 1956) アリソン・サール
Additional Names	Saar, Allison Saar, Alison, 1956-....
Type	Person
Birth	2/6/1956
Birth Place	Lotus
Nationality	American African Americans
Occupations	Photographers visual artist Women sculptors printmaker

Defining the Crosswalk: DITA

- Darwin Information Typing Architecture (DITA)
- Maintained by the OASIS standards organization.
- Modular (XML) framework for technical writing and documentation.
- Based on the idea of topics.
- Topics are organized using map documents.
- Topics and content can be reused and interlinked.

https://www.dita-ot.org

Defining the Crosswalk: DITA in oXygen



Defining the Crosswalk: Structure

- LUX Top-Level Entities
  - Record-Level Entities
  - Related Entities
    - Concepts
    - Groups
    - People
    - Places
- Descriptive Content
  - Identifiers
  - Names and Labels
  - Notes and Statements
  - Dates

“Supertype” Taxonomy Specifications

Source data

```
---
name: Collages
sampleBibs:
  - 7647390
fieldSpec:
  - ldr[6]
  - 006[0]
  - 007[0]
  - 007[1]
  - 300a
  - 655a
```

Processing steps and output

```
---
# Collages mapping
conditions:
  OR:
    # Two-dimensional nonprojectable graphic
    - - ldr[6]
      - k
    # Two-dimensional nonprojectable graphic
    - - 006[0]
      - k
  AND:
    - OR:
      - AND:
        # Nonprojected graphic
        - - 007[0]
          - k
        # Collage
        - - 007[1]
          - c
      - lower-case(300a) contains 'collage'
      - lower-case(655a) contains 'collage'
```


```
{
  "classified_as": [
    {
      "id": "http://vocab.getty.edu/aat/300033963",
      "type": "Type",
      "_label": "Collages",
      "classified_as": [
        {
          "id": "http://vocab.getty.edu/aat/300435443",
          "type": "Type",
          "_label": "Type of Object"
        }
      ]
    }
  ]
}
```

Narrative Specifications (Places from fixed field 008)

Introduction

Step

Substeps

Step result

Processing steps and output

`008[15–17]` contains a two- or three-letter code representing a country- or state-level place entity.

The code represents a place of publication, production, execution, or sometimes location (in the case of manuscript holdings).

1. Generate and store the top-level place resource, identified by an IRI.

   i. Normalize whitespace to test for null values and eliminate trailing whitespace after two-letter codes.

   ii. Match the two- or three-letter code against the Library of Congress's MARC List for Countries, which is available as a tab-delimited file, and add the corresponding URI as an `equivalent` reference, as shown below.

   iii. Use the place name from the Library of Congress file as a key to match against and merge with equivalent place entities.

   For example, references to "France" should point to the same entity IRI, regardless of the data source in MARC. `008 850723s1984 fr a b 00100 fre d` and `650 0 $a Opera $z France.` should both result in a link to the same place entity representing France.

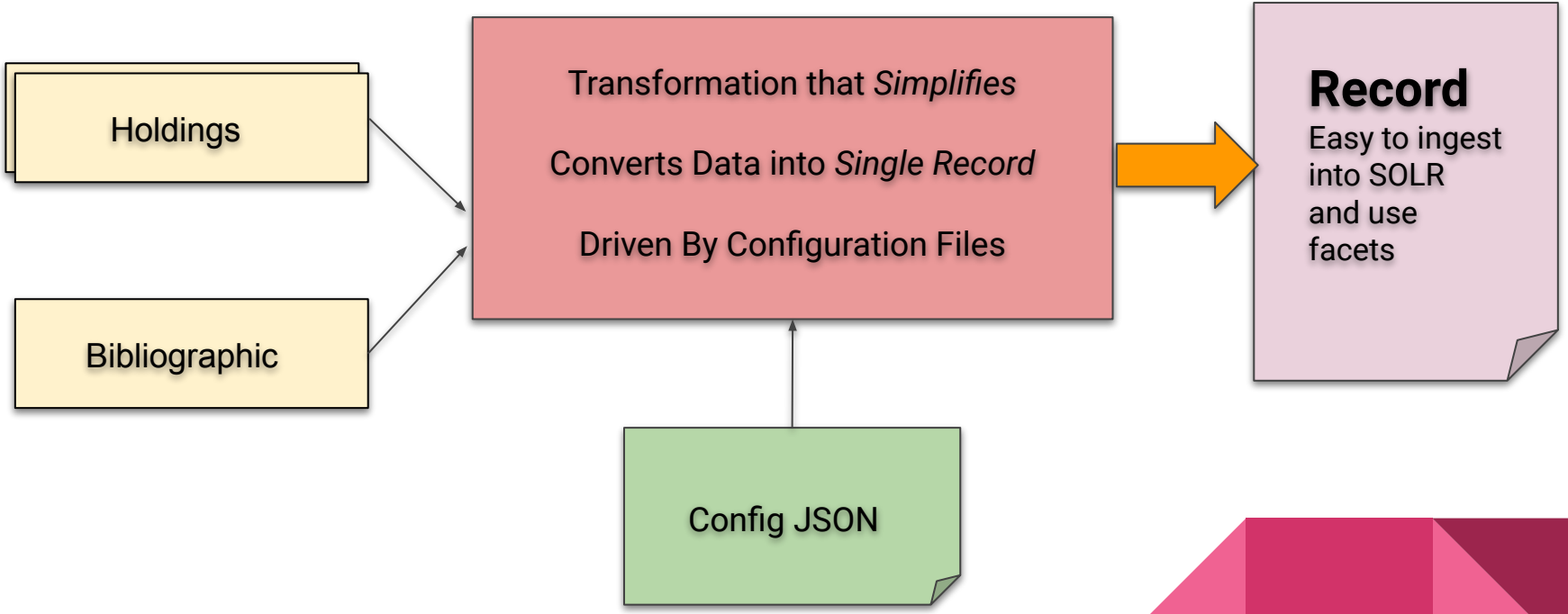
`3906934`


```
{
  "@context": "https://linked.art/ns/v1/linked-art.json",
  "id": "https://lux.collections.yale.edu/data/place/3dcbc9fa-ca9c-4fa1-bd0e-d25e93f461e5",
  "type": "Place",
  "_label": "France",
  "identified_by": [
    {
      "type": "Name",
      "content": "France",
      "classified_as": [
        {
```

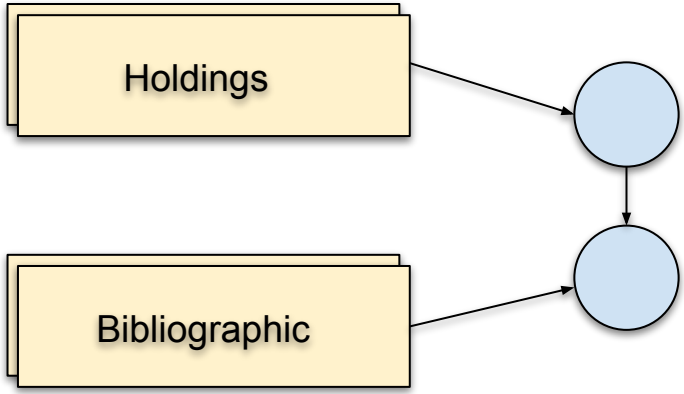
Technical Implementation

MARC Holdings and Bibliographic → Linked Art

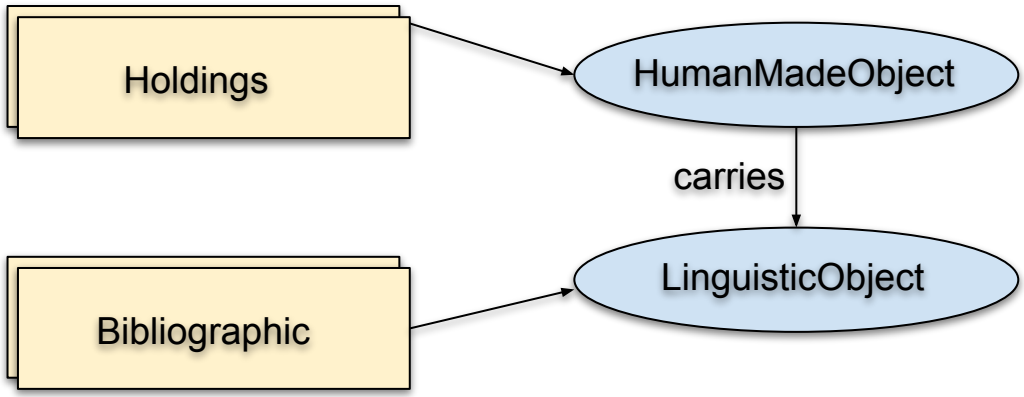
Typical Crosswalk



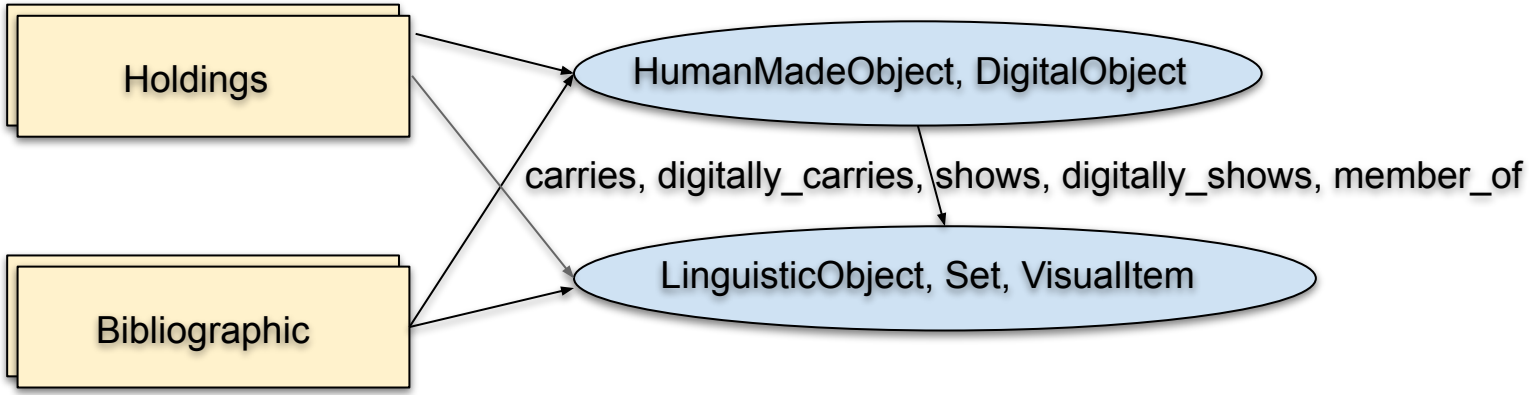
MARC Holdings and Bibliographic → Linked Art



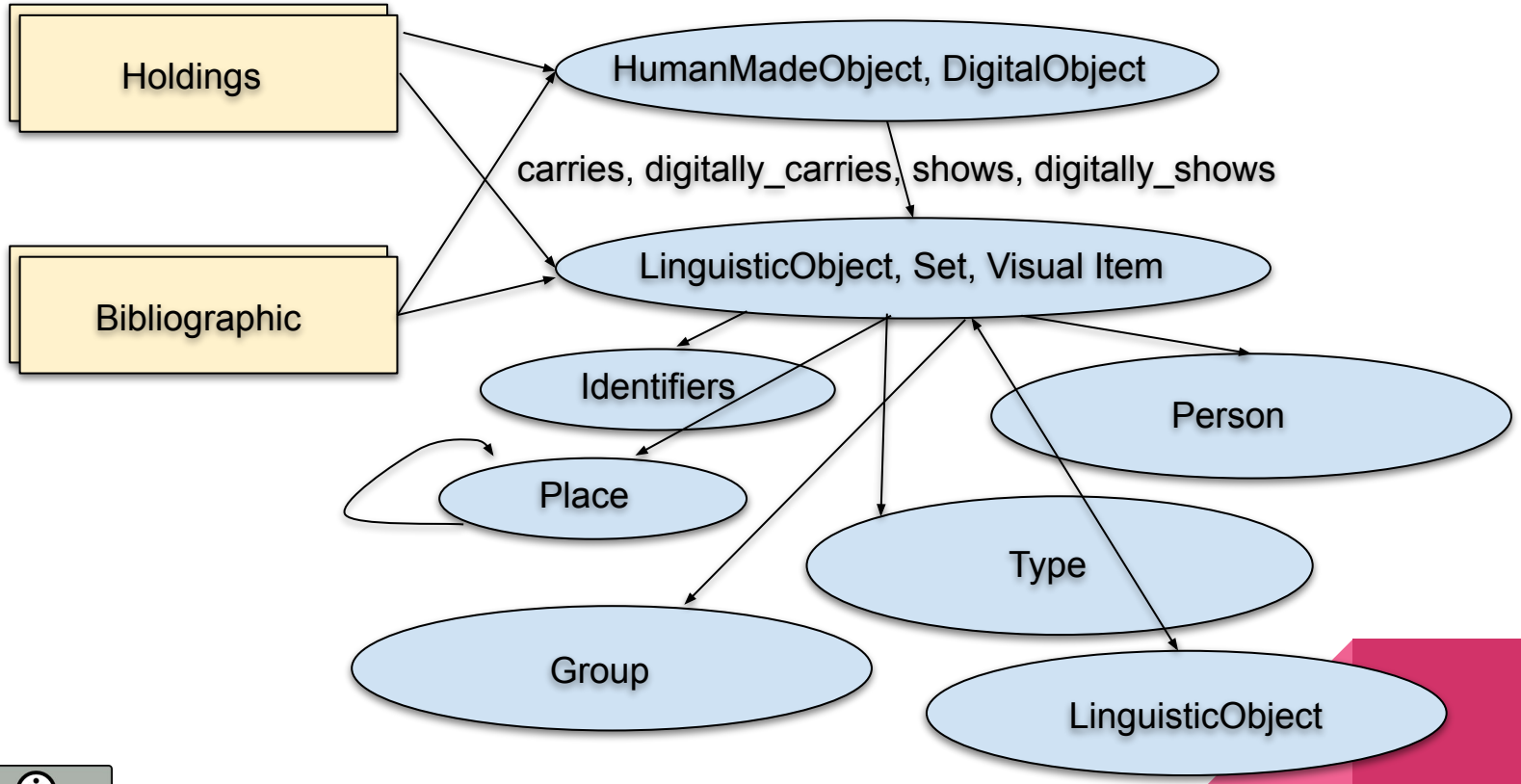
MARC Holdings and Bibliographic → Linked Art



MARC Holdings and Bibliographic → Linked Art

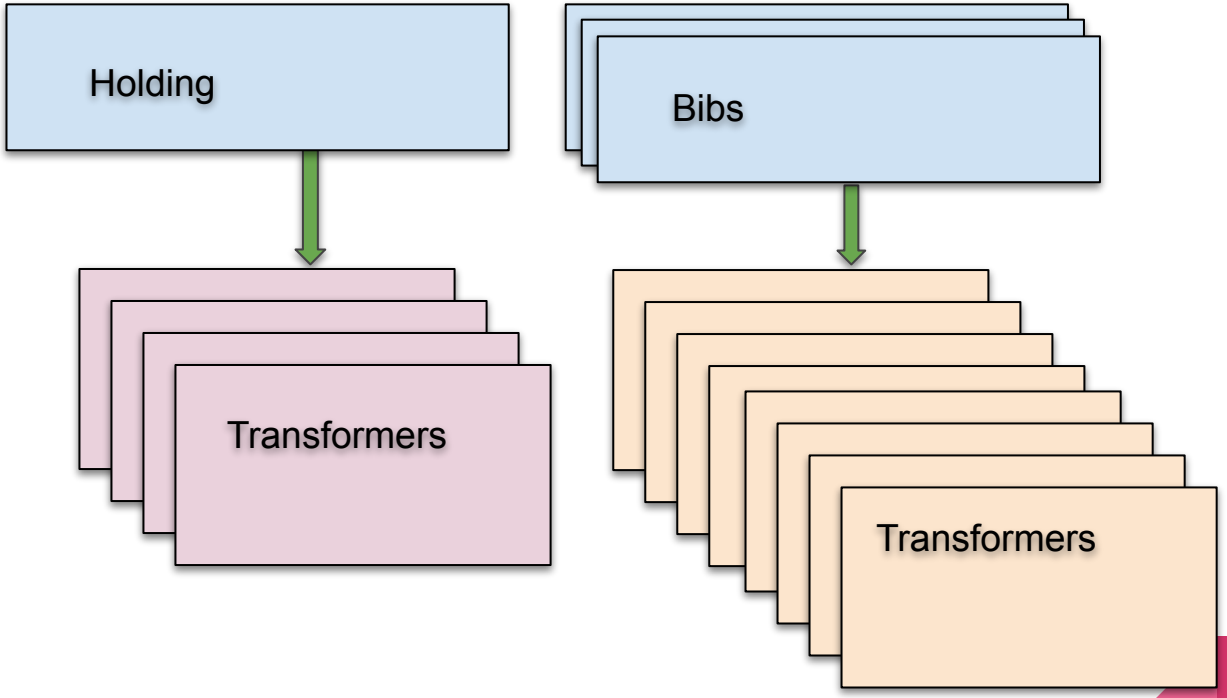


MARC Holdings and Bibliographic → Linked Art

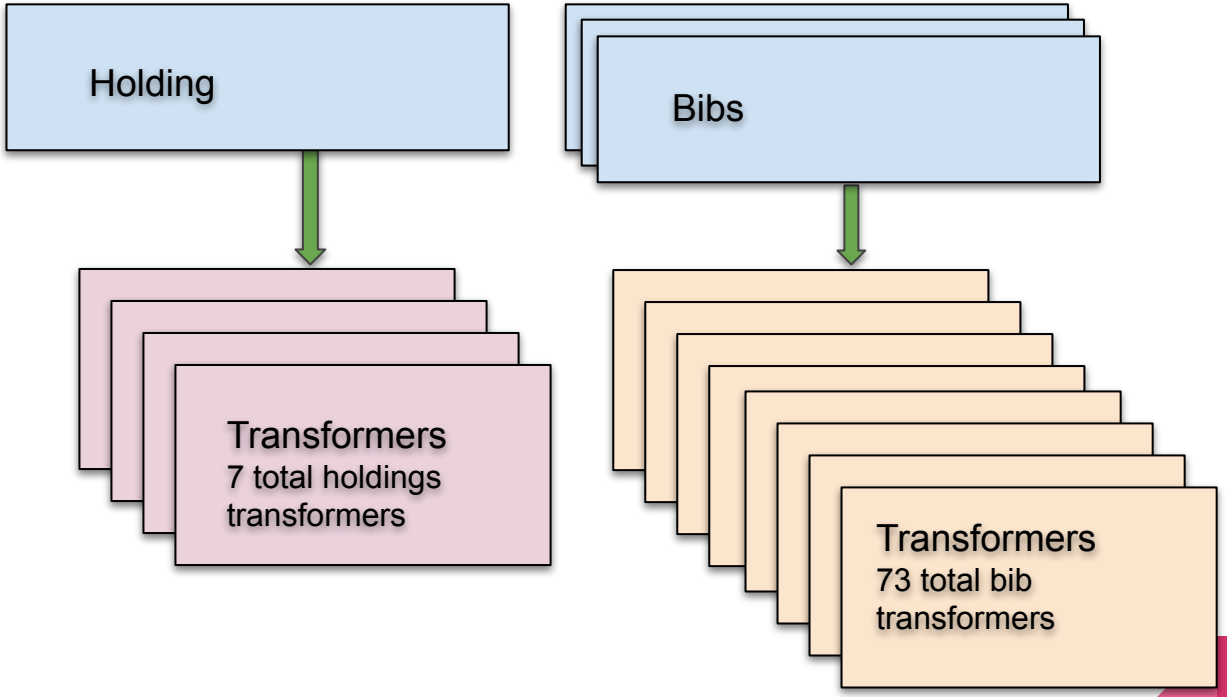


Processing the Data

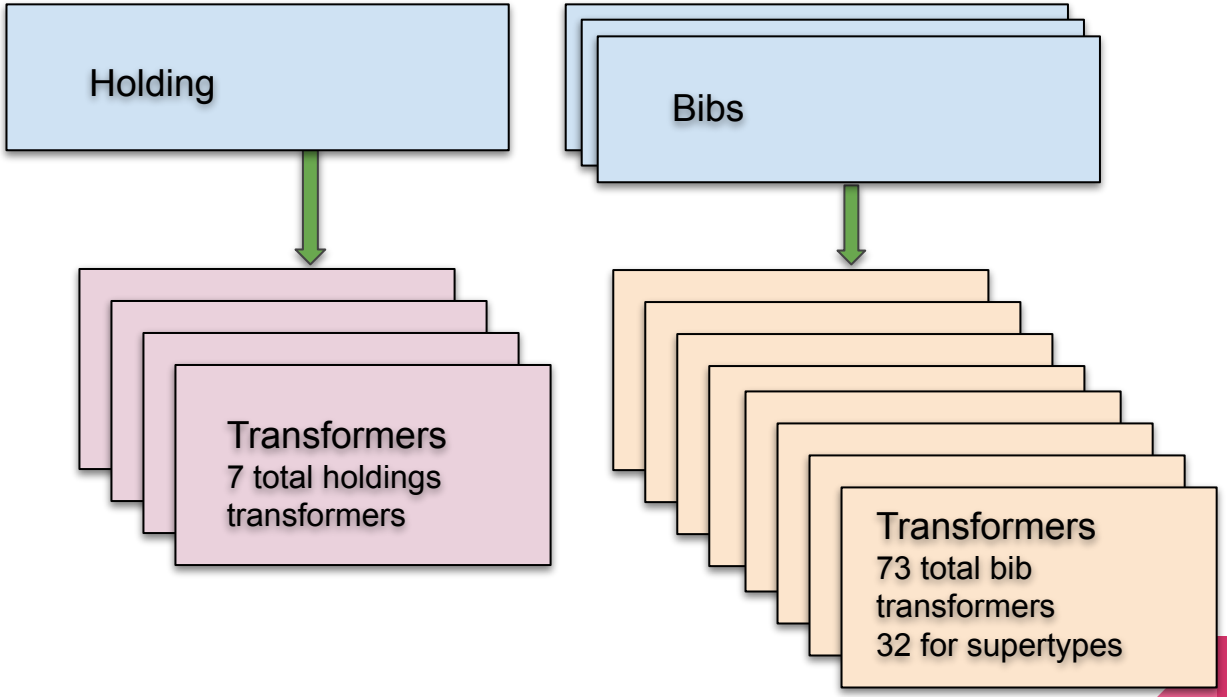
Processing the Data



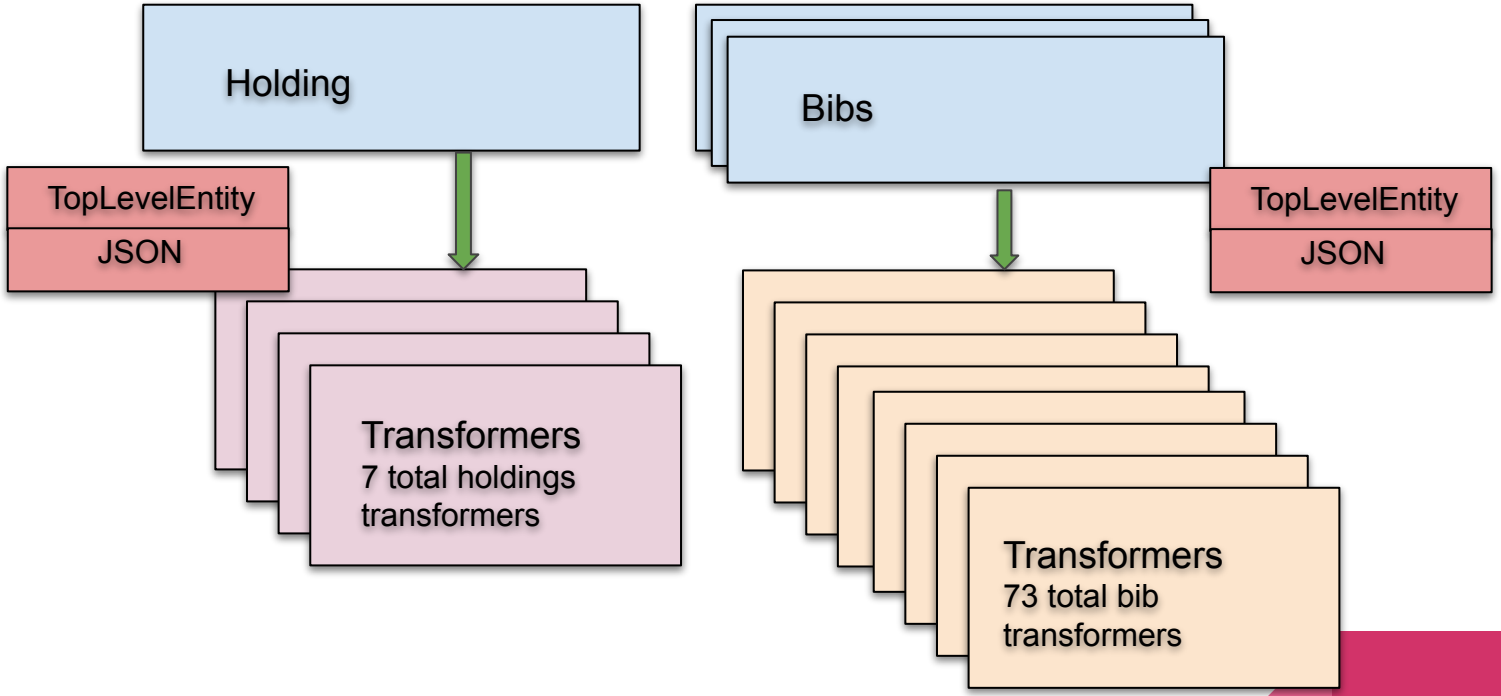
Processing the Data



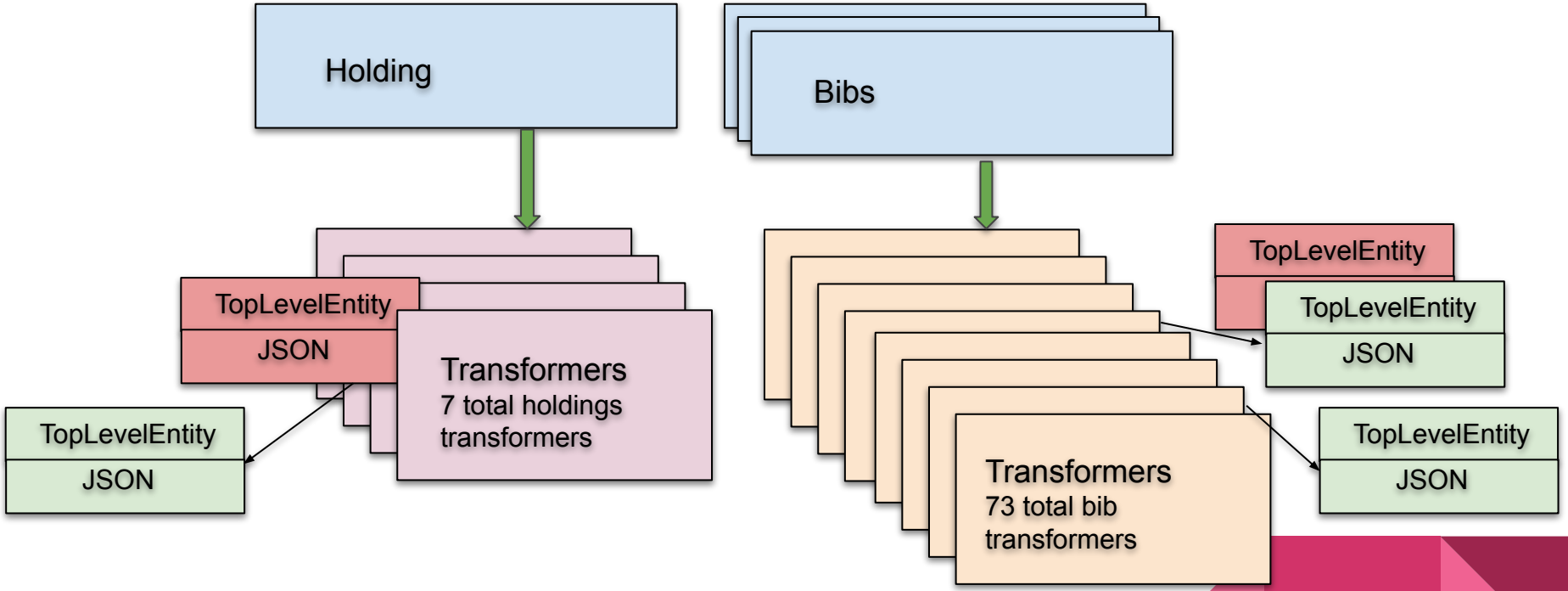
Processing the Data



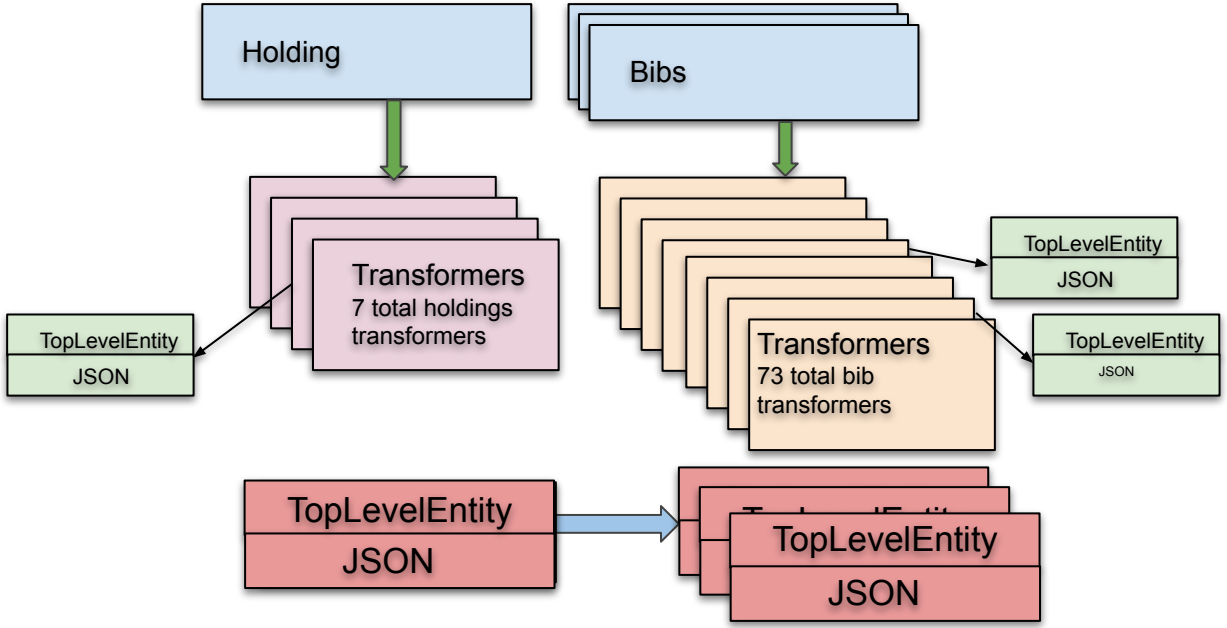
Processing the Data



Processing the Data

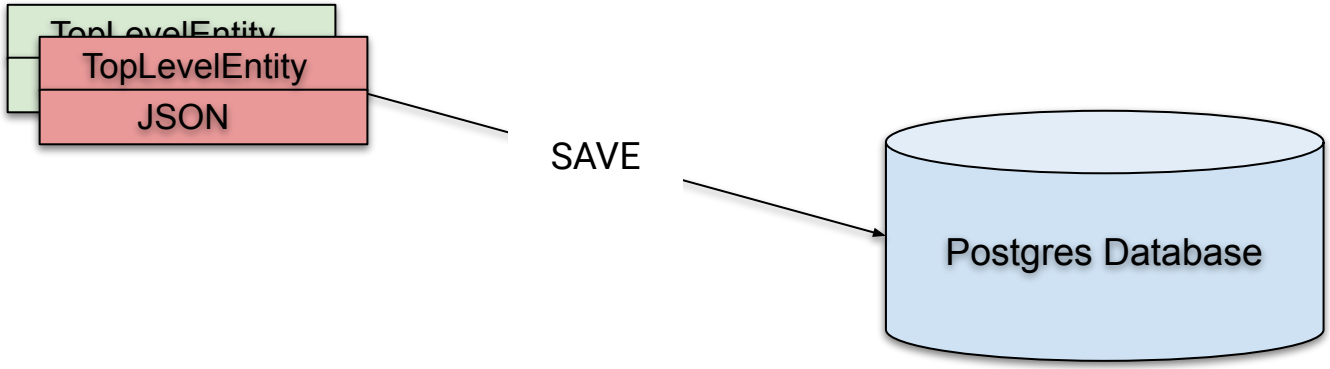


Processing the Data

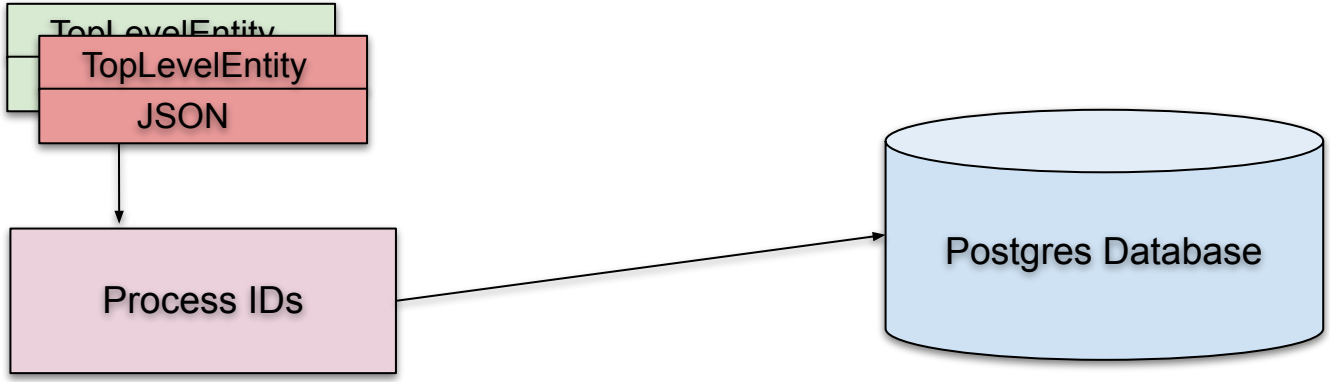


Storing the Entities

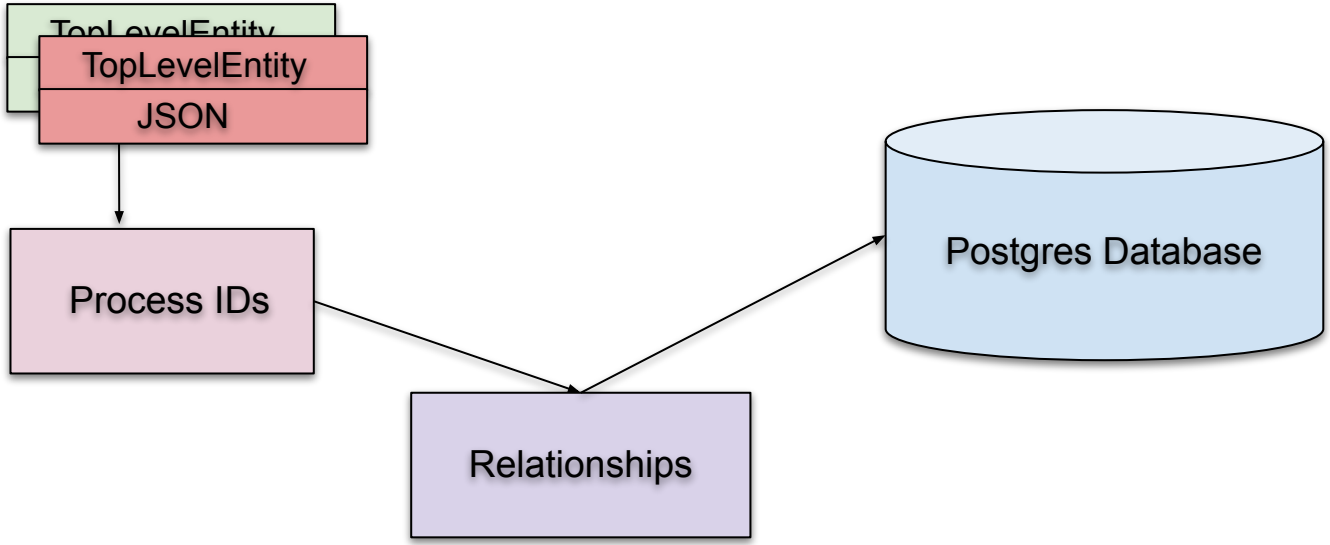
Storing the Entities



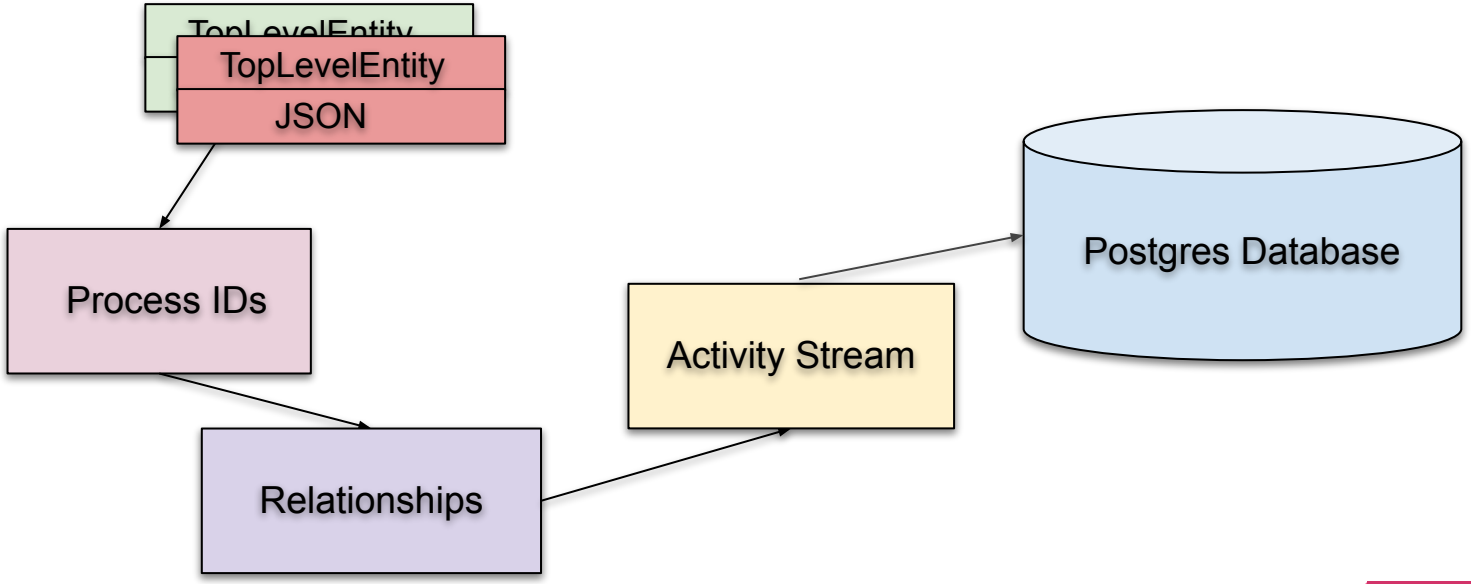
Storing the Entities



Storing the Entities

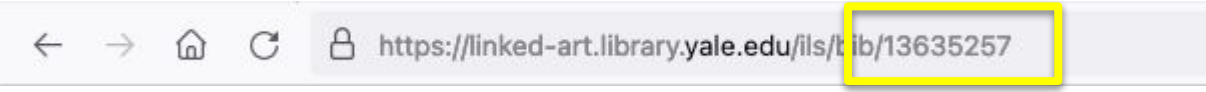


Storing the Entities



URL Schema

URL Schema



URL Schema

https://linked-art.library.yale.edu/ils/bib/13635257

https://linked-art.library.yale.edu/node/fc7e1846-aa0f-4d07-879a-e19e85cb9a8a

URL Schema

https://linked-art.library.yale.edu/ils/bib/13635257

https://linked-art.library.yale.edu/node/fc7e1846-aa0f-4d07-879a-e19e85cb9a8a

https://linked-art.library.yale.edu/node/fc7e1846-aa0f-4d07-879a-e19e85cb9a8a/relationships

Incoming Relationships

Incoming Relationships

https://linked-art.library.yale.edu/node/5925fc9b-7d4b-449a-9d70-63b06a738ab0/relationships

JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON


```
page:            0
total:           2
totalPages:      1
results:
  0:
    subject:         "https://linked-art.library.yale.edu/node/7cf270c2-8168-4e6f-a271-a46e63f77756"
    relationship:    ".created_by.part.carried_out_by"
    object:          "https://linked-art.library.yale.edu/node/5925fc9b-7d4b-449a-9d70-63b06a738ab0"
    _subject_label:  "Family of liars [electronic resource] : The prequel to we were liars"
  1:
    subject:         "https://linked-art.library.yale.edu/node/7da0360b-80c7-46ab-a20a-cb916aac723d"
    relationship:    ".created_by.part.carried_out_by"
    object:          "https://linked-art.library.yale.edu/node/5925fc9b-7d4b-449a-9d70-63b06a738ab0"
    _subject_label:  "French braid [electronic resource] : A novel"
```

Incoming Relationships

https://linked-art.library.yale.edu/node/5925fc9b-7d4b-449a-9d70-63b06a738ab0/relationships

JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON


```
page:                0
total:               2
totalPages:          1
results:
  0:
    subject:         "https://linked-art.library.yale.edu/node/7cf270c2-8168-4e6f-a271-a46e63f77756"
    relationship:    ".created_by.part.carried_out_by"
    object:          "https://linked-art.library.yale.edu/node/5925fc9b-7d4b-449a-9d70-63b06a738ab0"
    _subject_label:  "Family of liars [electronic resource] : The prequel to we were liars"
  1:
    subject:         "https://linked-art.library.yale.edu/node/7da0360b-80c7-46ab-a20a-cb916aac723d"
    relationship:    ".created_by.part.carried_out_by"
    object:          "https://linked-art.library.yale.edu/node/5925fc9b-7d4b-449a-9d70-63b06a738ab0"
    _subject_label:  "French braid [electronic resource] : A novel"
```

Incoming Relationships


```
▸ subject_of:                [...]
▾ created_by:
    type:                    "Creation"
  ▾ part:
    ▾ 0:
        type:                "Creation"
        _label:              "Creation"
      ▾ carried_out_by:
        ▾ 0:
          ▾ id:              "https://linked-art-test.library.yale.edu/node/4102fbdc-3fb9-44dd-ba5c-58fba3170e0d"
            type:            "Person"
            _label:          "Didion, Joan"
      ▾ classified_as:
        ▾ 0:
          ▾ id:              "https://linked-art-test.library.yale.edu/node/7a13c013-ee4b-41dd-a721-c4c2742707f6"
            type:            "Type"
            _label:          "Creator"
    ▾ 1:
        type:                "Creation"
        _label:              "Creation"
      ▾ carried_out_by:
        ▾ 0:
          ▾ id:              "https://linked-art-test.library.yale.edu/node/5925fc9b-7d4b-449a-9d70-63b06a738ab0"
            type:            "Person"
            _label:          "Farr, Kimberly"
      ▾ classified_as:
        ▾ 0:
```

Normalizing and Merging

Normalization and Merging

```
String normalizedValue = TransformerHelpers.stringNormalization(value, includePunctuation: "()-");
TopLevelEntity personEntity = linkedDataService.findBySourceAndTypeAndLabel(source: "its-bib", type: "Person", normalizedValue);
if (personEntity == null) {
```

Normalization and Merging

```
String normalizedValue = TransformerHelpers.stringNormalization(value, includePunctuation: "() ");
TopLevelEntity personEntity = linkedDataService.findBySourceAndTypeAndLabel( source: "ils-bib", type: "Person", normalizedValue);
if (personEntity == null) {
```

Normalization and Merging

```
String normalizedValue = TransformerHelpers.stringNormalization(value, includePunctuation: "()-");
TopLevelEntity personEntity = linkedDataService.findBySourceAndTypeAndLabel( source: "ils-bib", type: "Person", normalizedValue);
if (personEntity == null) {
```

Normalization and Merging

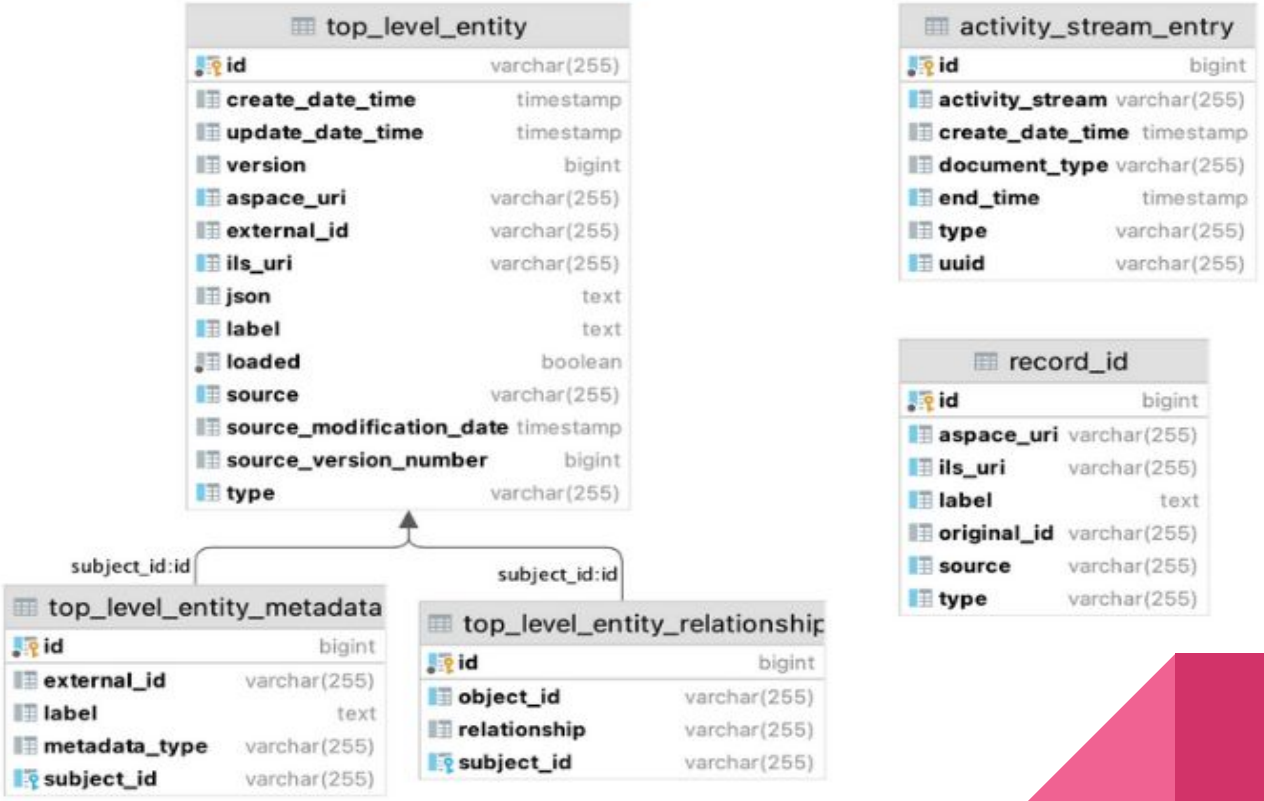
```
String normalizedValue = TransformerHelpers.stringNormalization(value, includePunctuation: "()-");
TopLevelEntity personEntity = linkedDataService.findBySourceAndTypeAndLabel( source: "ils-bib", type: "Person", normalizedValue);
if (personEntity == null) {
    personEntity = new TopLevelEntity( source: "ils-bib", type: "Person", normalizedValue);
    linkedDataService.save(personEntity);
    Node personAgentNode = linkedDataService.createNodeWithTypeAndLabel( type: "Person", value);
    personAgentNode.setId(personEntity.getId());
    if (equivalent != null) {...}
    Node identifyNode = linkedDataService.getNodeFactory().primaryName(value);
    linkedDataService.createRelationship(personAgentNode, relationship: "identified_by", identifyNode);
    linkedDataService.updateEntityJson(personEntity, personAgentNode);
}
Node creationNode = bibNode.findOrCreateChildNode( type: "Creation", relationship: "created_by");
Node partNode = linkedDataService.createNodeWithTypeAndLabel( type: "Creation", label: "Creation");
linkedDataService.createRelationship(creationNode, relationship: "part", partNode);
```

Normalization and Merging

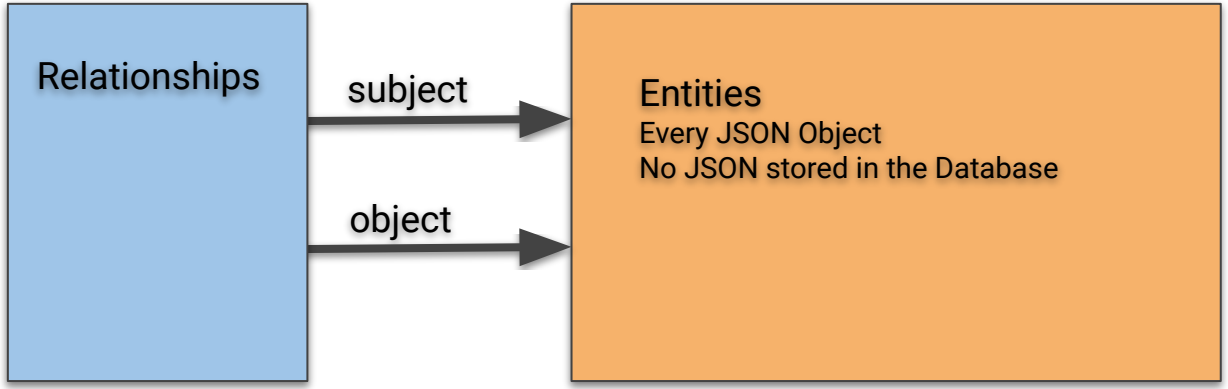
```
String normalizedValue = TransformerHelpers.stringNormalization(value, includePunctuation: "()-");
TopLevelEntity personEntity = linkedDataService.findBySourceAndTypeAndLabel( source: "ils-bib", type: "Person", normalizedValue);
if (personEntity == null) {
    personEntity = new TopLevelEntity( source: "ils-bib", type: "Person", normalizedValue);
    linkedDataService.save(personEntity);
    Node personAgentNode = linkedDataService.createNodeWithTypeAndLabel( type: "Person", value);
    personAgentNode.setId(personEntity.getId());
    if (equivalent != null) {...}
    Node identifyNode = linkedDataService.getNodeFactory().primaryName(value);
    linkedDataService.createRelationship(personAgentNode, relationship: "identified_by", identifyNode);
    linkedDataService.updateEntityJson(personEntity, personAgentNode);
}
Node creationNode = bibNode.findOrCreateChildNode( type: "Creation", relationship: "created_by");
Node partNode = linkedDataService.createNodeWithTypeAndLabel( type: "Creation", label: "Creation");
linkedDataService.createRelationship(creationNode, relationship: "part", partNode);
```

Database Schemas

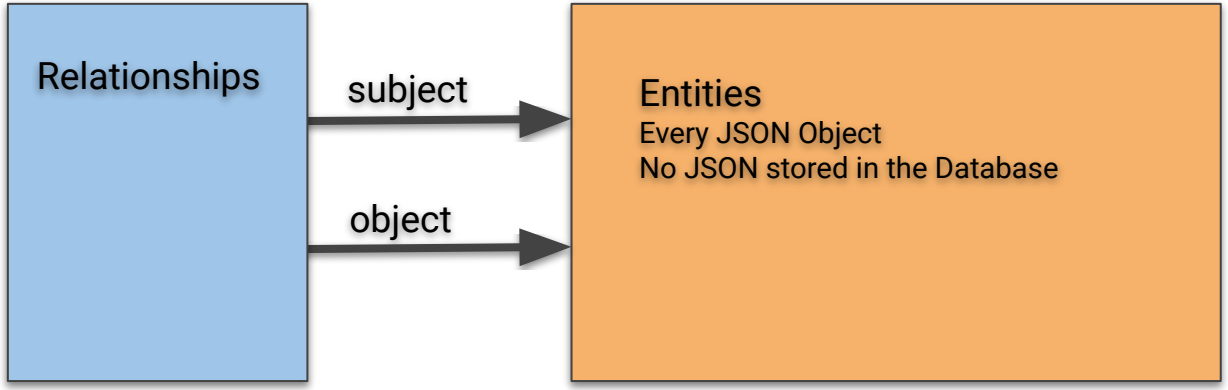
Database Diagram



Original Database Plan: Triple based



Original Database Plan: Triple based



Worked until we had about 10% of the data in the system.
We had to find another solution so we could have a faster turn around.

Simplified TopLevelEntity Design

TopLevelEntities
JSON stored in the database

Worked for our full set of data. Reingest is about 48 hours.

Simplified TopLevelEntity Design

Worked for our full set of data. Reingest is about 48 hours.

Some Statistics

Stats:

12,799,915
Holdings Records

11,962,919
Bibliographic Records

2M+
Archives Space Records

50,006,415
TopLevelEntities

193,784,739
TopLevelEntityRelationships

24,104,513
TopLevelEntityMetadata

50,133,873
ActivityStreamEntries

Thank you!