Hands-On Lab: Build Your Own Index

> Heute baue ich meinen eigenen Artikelindex - Hands-On Workshop zur leichtgewichtigen Metadatenverarbeitung.

106. Deutscher Bibliothekartag, 2017-06-02, Frankfurt am Main

Martin Czygan, Tracy Hoffmann, Leipzig University Library

https://ub.uni-leipzig.de,
https://finc.info,
https://amsl.technology,
itprojekte@ub.uni-leipzig.de

Intro

- Hands-On: Wer möchte, kann mitprogrammieren und einen eigenen Metadatenindex bauen.
- Leichtgewichtig?

Warum leichtgewichtig?

- Nachnutzung existierender Tools
- nicht zu viel Code
- relativ leicht reproduzierbar

Warum leichtgewichtig?

- circa 5000 Zeilen Code für Orchestrierung

```
|-------------------------------------+----------|
|  siskin/sources/crossref.py         | 600      |
|  siskin/sources/amsl.py             | 597      |
|  siskin/sources/genios.py           | 334      |
|  siskin/sources/jstor.py            | 185      |
|  siskin/sources/doaj.py             | 177      |
|  siskin/sources/mag.py              | 175      |
|  siskin/sources/elsevierjournals.py | 159      |
|  siskin/sources/degruyter.py        | 144      |
|  siskin/sources/ieee.py             | 124      |
|  siskin/sources/khm.py              | 111      |
|  siskin/sources/arxiv.py            | 102      |
|  siskin/sources/thieme.py           | 92       |
|  siskin/sources/dawson.py           | 81       |
|  siskin/sources/springer.py         | 78       |
...
```

Warum leichtgewichtig?

- `lucene/index/IndexWriter.java` - 5097
- Nachnutzung von existierenden Frameworks wie Metafacture, Kommandozeilentools, ...

Konzepte

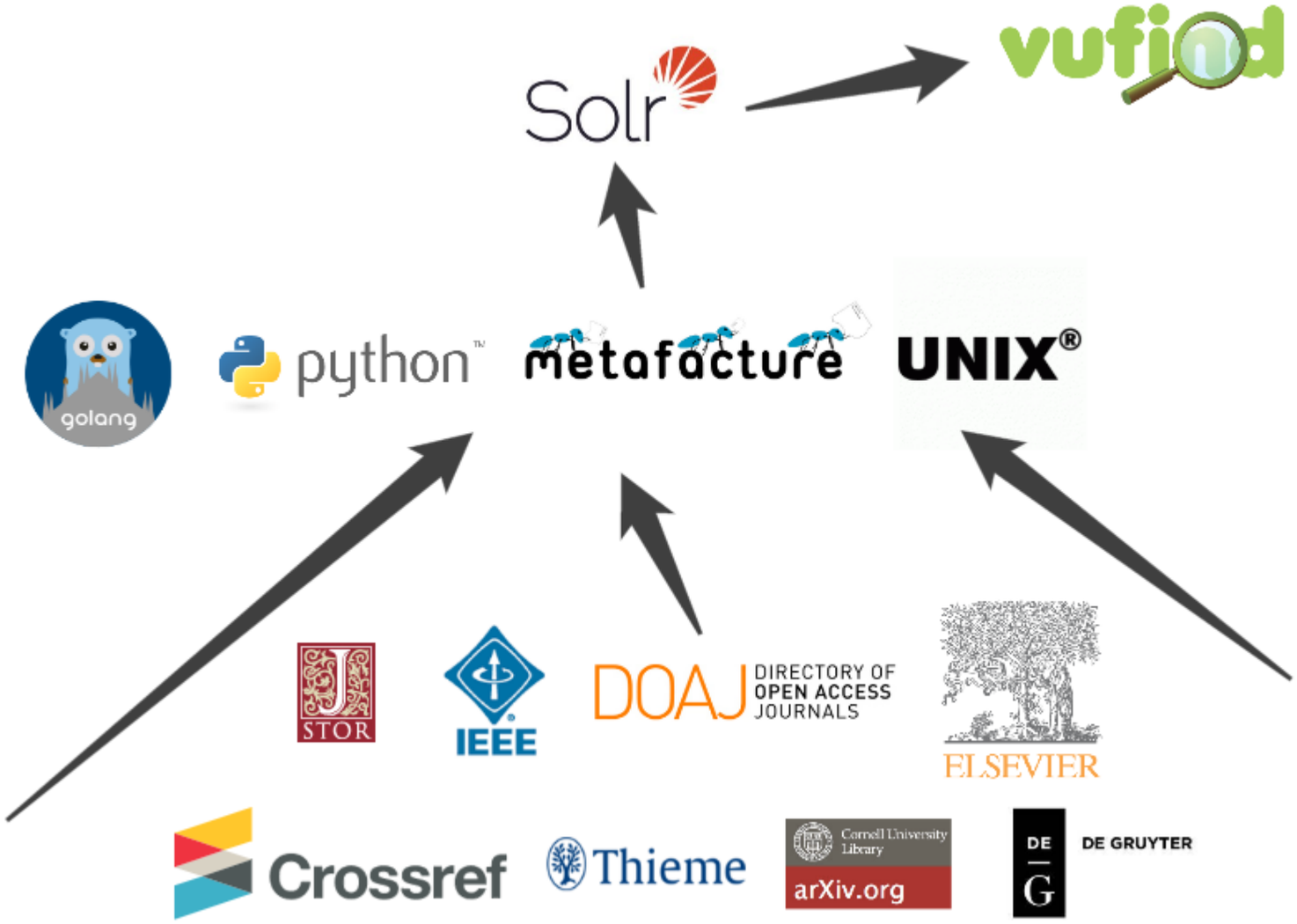
- iterative Entwicklung
- Unveränderlichkeit (immutability) von erstellten Daten
- Reproduzierbarkeit

Andere ETL-Tools

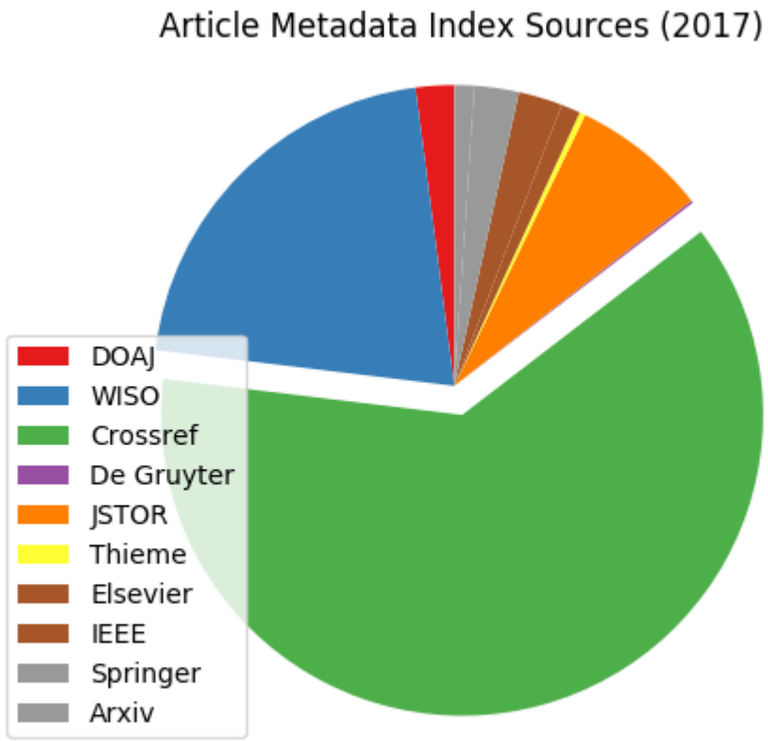
- Toolsets:
  - Metafacture
  - Catmandu
- Data management platforms:
  - OpenRefine
  - KNIME

Warum ein eigener Artikelindex?

- *Leaving block boxes behind* - commercial index (Primo Central)
- Seit 04/2015 in Produktion
- 14% Klicks/Ansichten im Katalog


Solr
vufind
golang
python
metafacture
UNIX®
STOR
IEEE
DOAJ DIRECTORY OF OPEN ACCESS JOURNALS
ELSEVIER
Crossref
Thieme
Cornell University Library
arXiv.org
DE G
DE GRUYTER

Quellenverteilung



Circa 118,221,121 Artikel insgesamt. Für Bibliotheken nur Teile sichtbar.

Weiter

Abfolge

- Datensynchronisation
- Normalisierung von Formaten
- Anwenden von Lizenzinformationen
- Postprocessing (e.g. DOI-Deduplikation)
- Export (SOLR)

Orchestrierung

- viele (meist kleine) Aufgaben: FTP-Update, Filtern von Dateien, Konvertierung, Lizensierung, Deduplizierung, Export

Orchestrierung

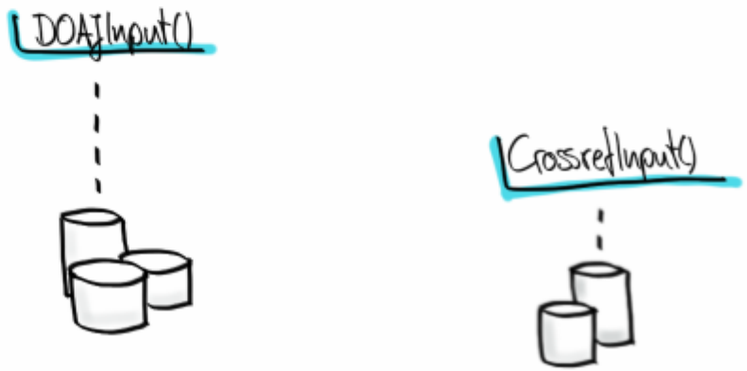
- Wie dokumentiert man so etwas?
- Luigi
- Beschreibung von Abhängigkeiten
- eine *Aufgabe* kapselt einen Prozessierungsschritt
- eine *Aufgabe* besteht aus Vorbedingungen (requirements), business logic und Ergebnissen

Orchestrierung

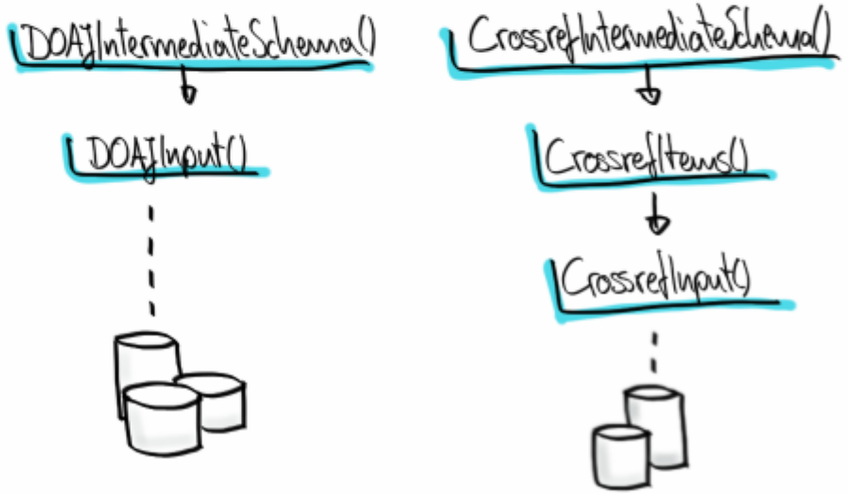
Abhängigkeiten (Analogie):

- Pizza

Harvesting / Synchronisierung



Normalisierung



Normalisierung

Konvertierung in ein *Zwischenformat*.


```
{
  "finc.format": "ElectronicArticle",
  "finc.mega_collection": "DOAJ Directory of ...",
  "finc.record_id": "ai-28-000011857dbc42afb0f1a8c7e35ab46f",
  "finc.source_id": "28",
  "ris.type": "EJOUR",
  "rft.atitle": "Study progresses on continuous ...",
  "rft.genre": "article",
  "rft.issn": [
    "1672-5123"
  ],
  "rft.jtitle": "Guoji Yanke Zazhi",
  "rft.pages": "1737-1740",
  "rft.pub": [
    "Press of International Journal of Ophthalmology ..."
  ],
  ...
```

Metafacture

> Metafacture: Werkzeuge zur Metadatenprozessierung (DNB)

- Beispiel: Arxiv
- Flux + Morph

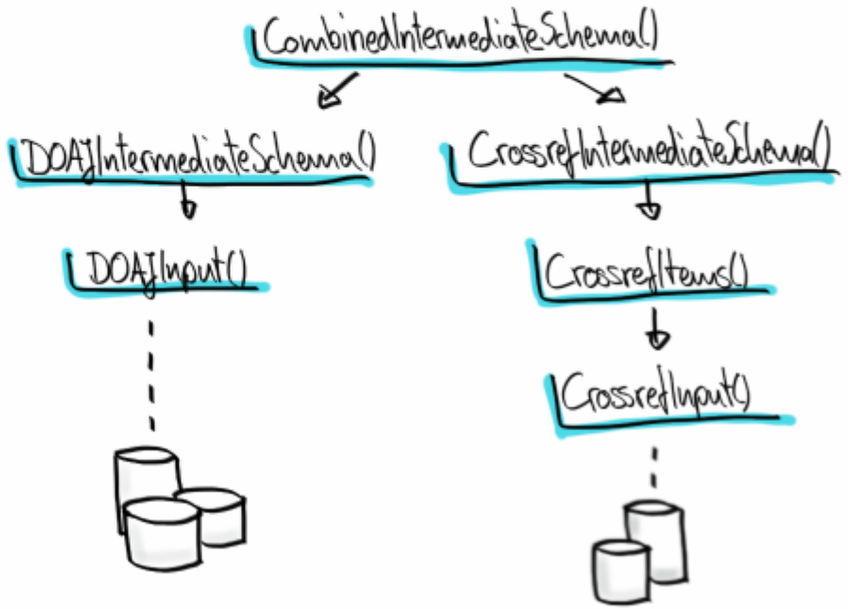
```
// Example flux script.

fileName|
open-file|
decode-xml|
handle-generic-xml("Record")|
morph(FLUX_DIR + "morph.xml", *)|
encode-json|
write("stdout");
```

Normalisierung mit Metafacture

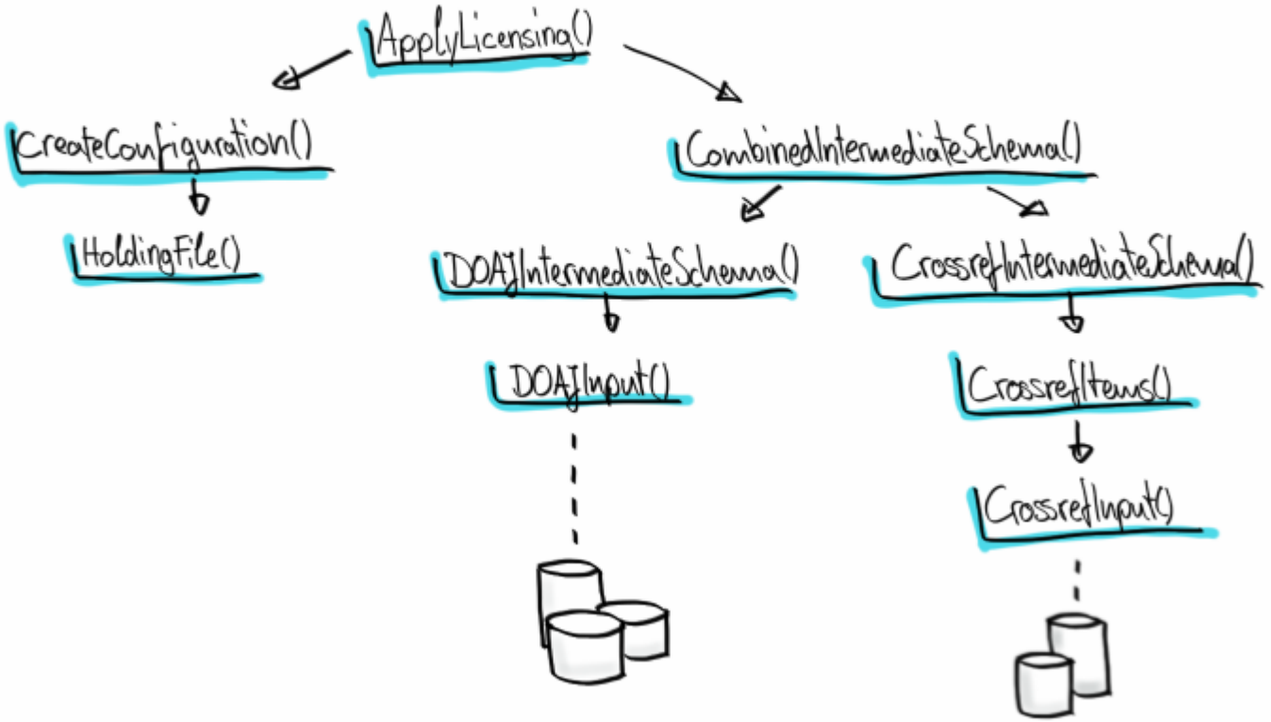
Beispiel Morph:

```
<entity name="url[]" flushWith="record">
    <data source="metadata.dc.identifier.value">
        <regexp match="^(http.*)" format="${1}"/>
    </data>
</entity>
...
<data source="metadata.dc.type.value" name="rft.genre">
    <lookup in="genre_liste"/>
</data>
...
```

Zusammenführen in *ein*e normalisierte Datei



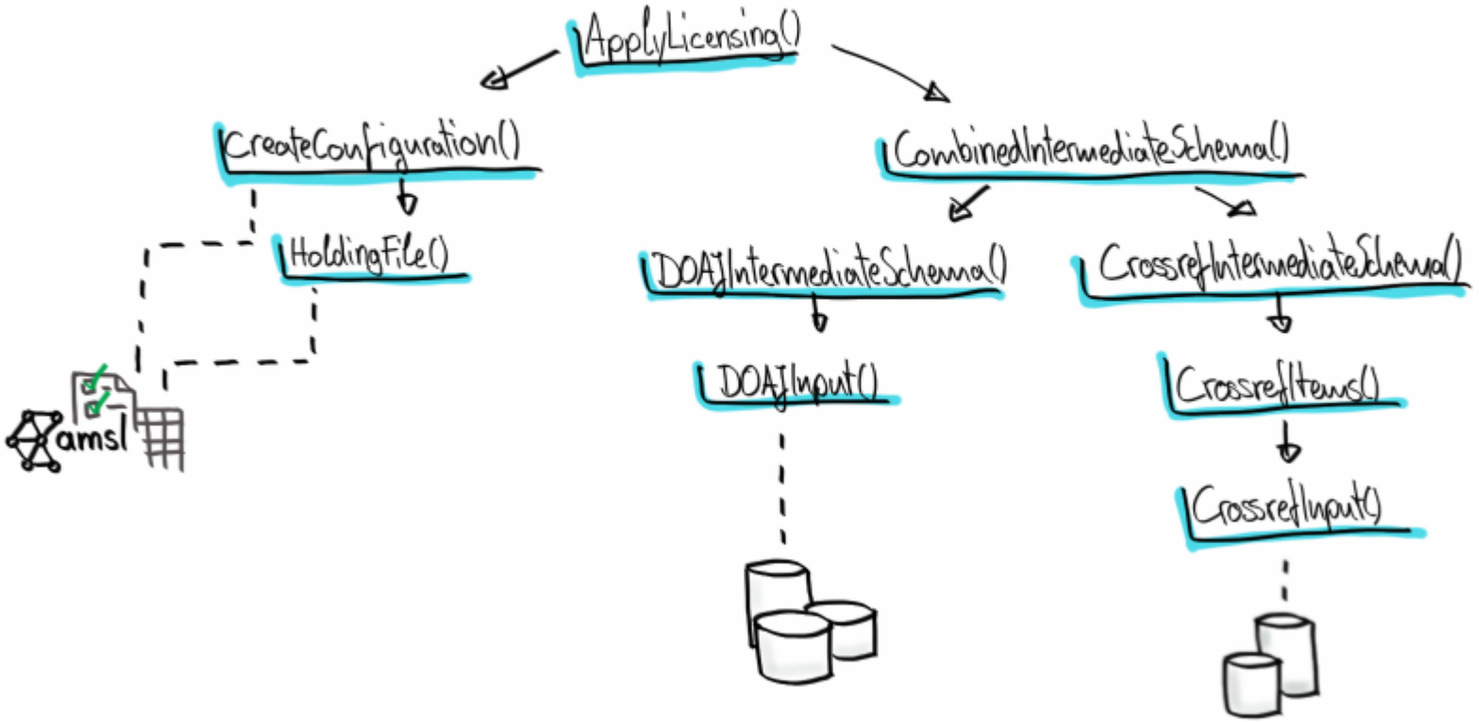
Lizenzinformationen

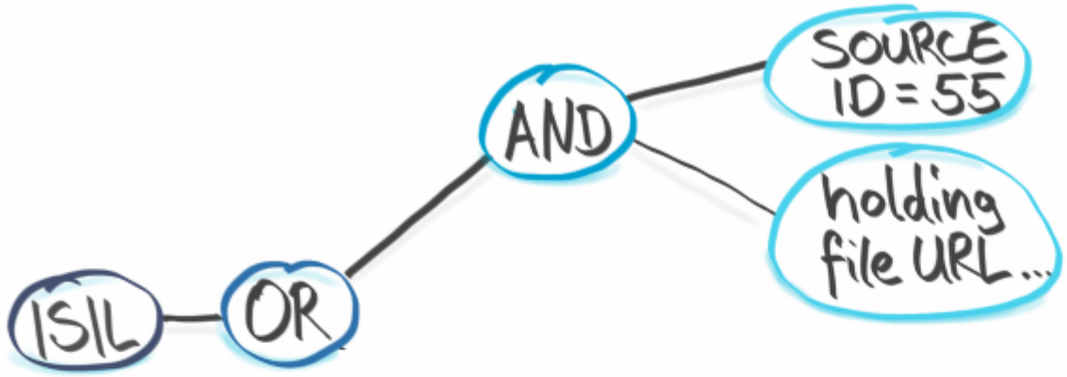


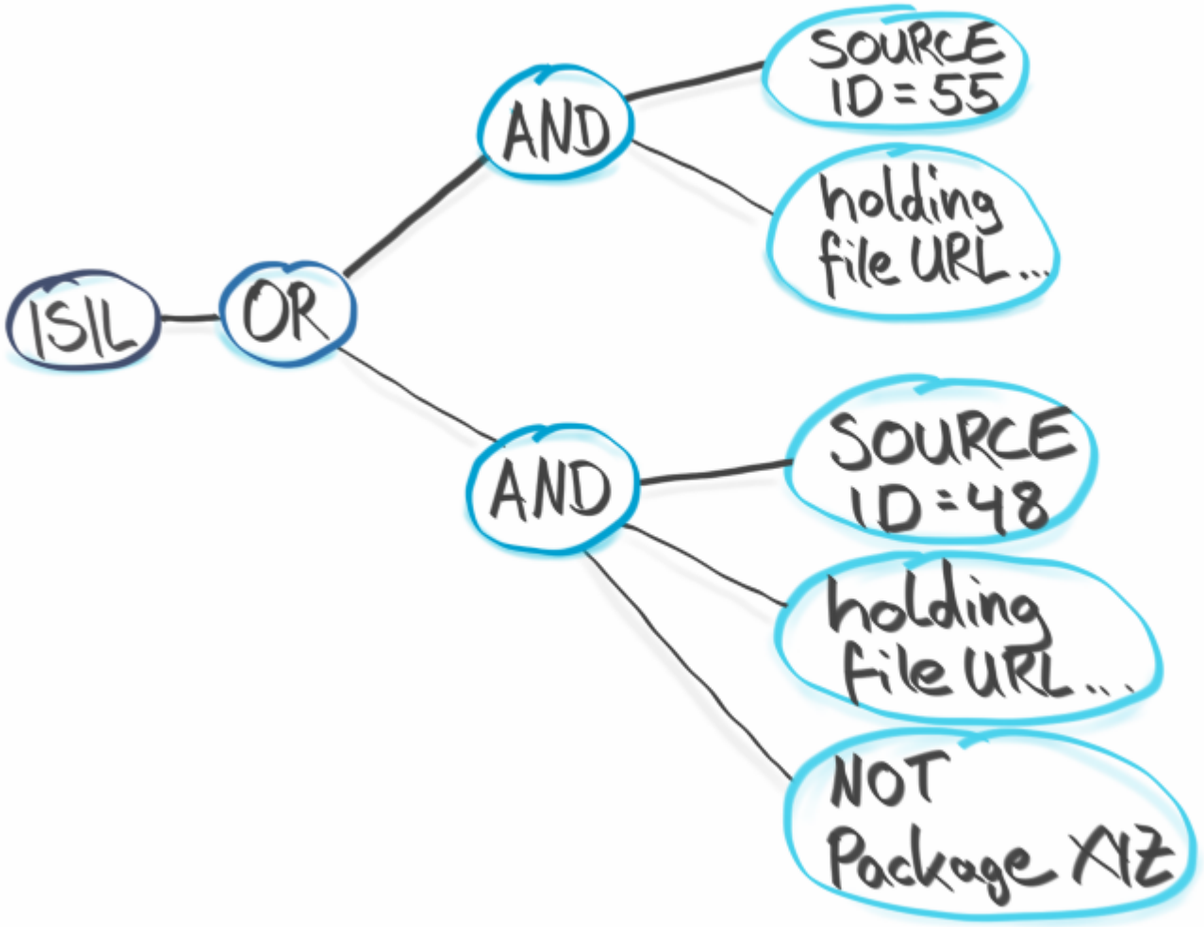
Lizensierung

- Beispiel-Konfiguration
- AMSL Website + Screenshots

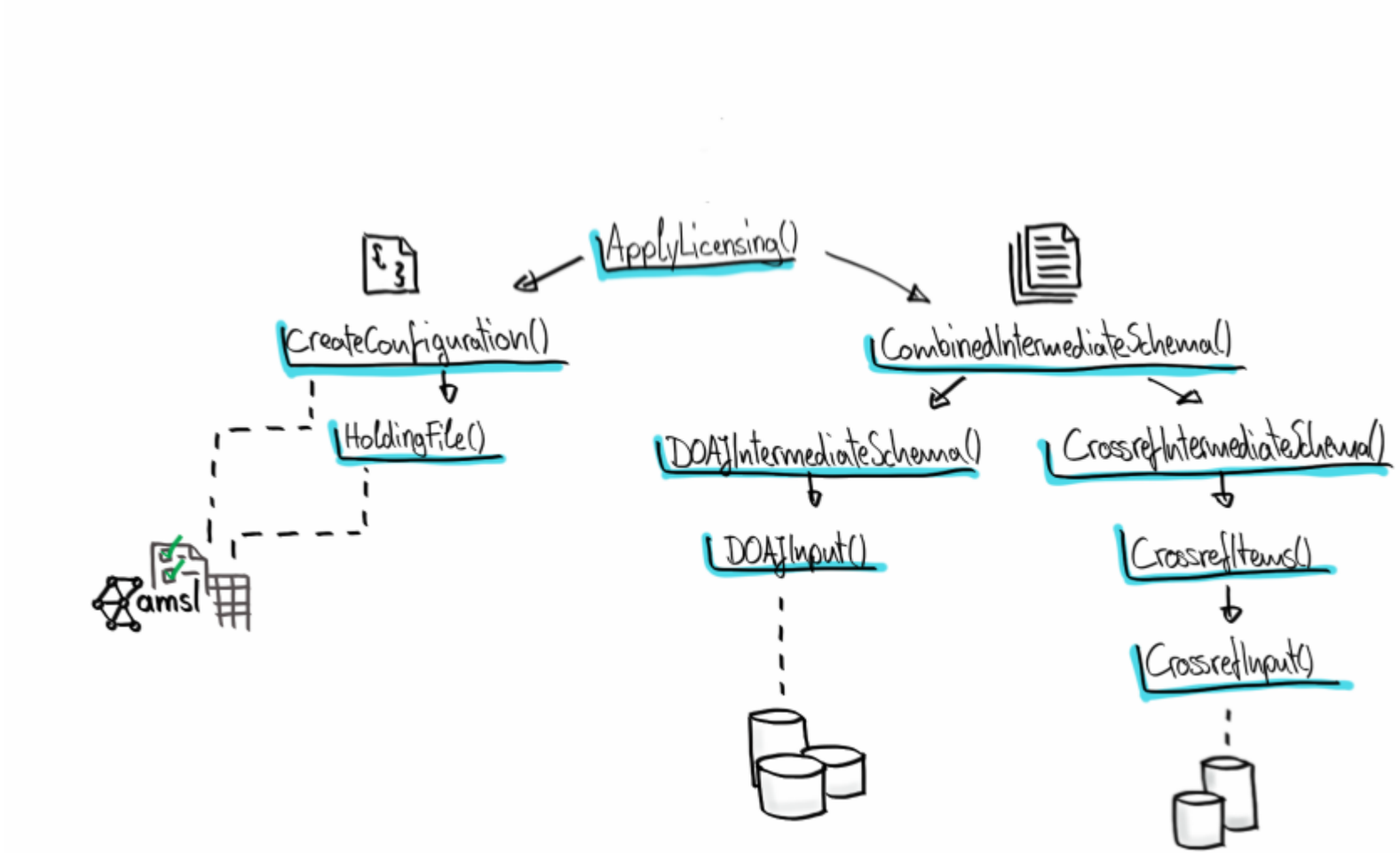
Lizensierung




SOURCE
ID = 55
AND
holding
file URL ...
ISIL
OR


ISIL
OR
AND
SOURCE ID = 55
holding file URL...
AND
SOURCE ID = 48
holding file URL...
NOT Package XYZ

Postprozessierung (z.B. Deduplizierung)



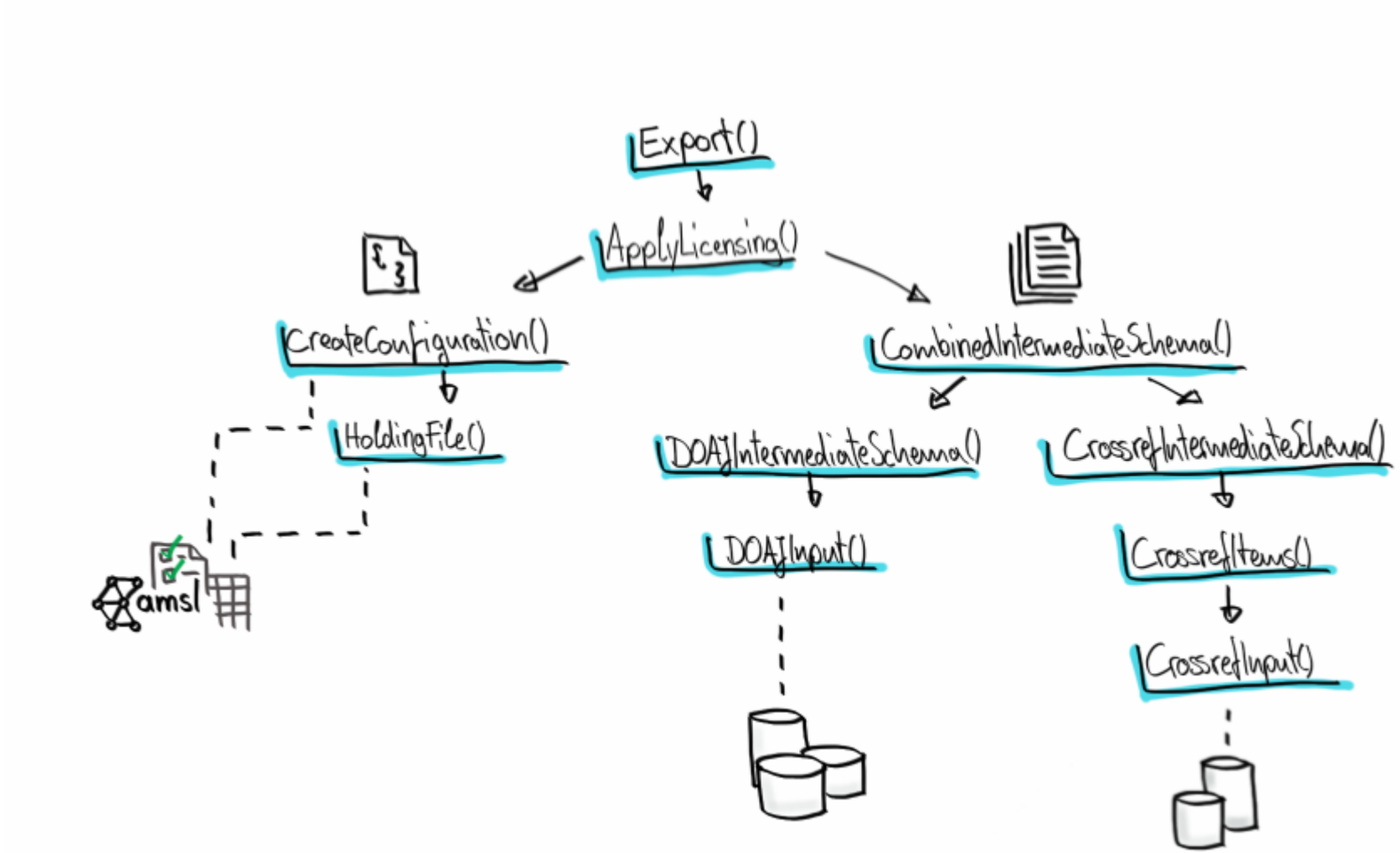
Deduplizierung

- groupcover

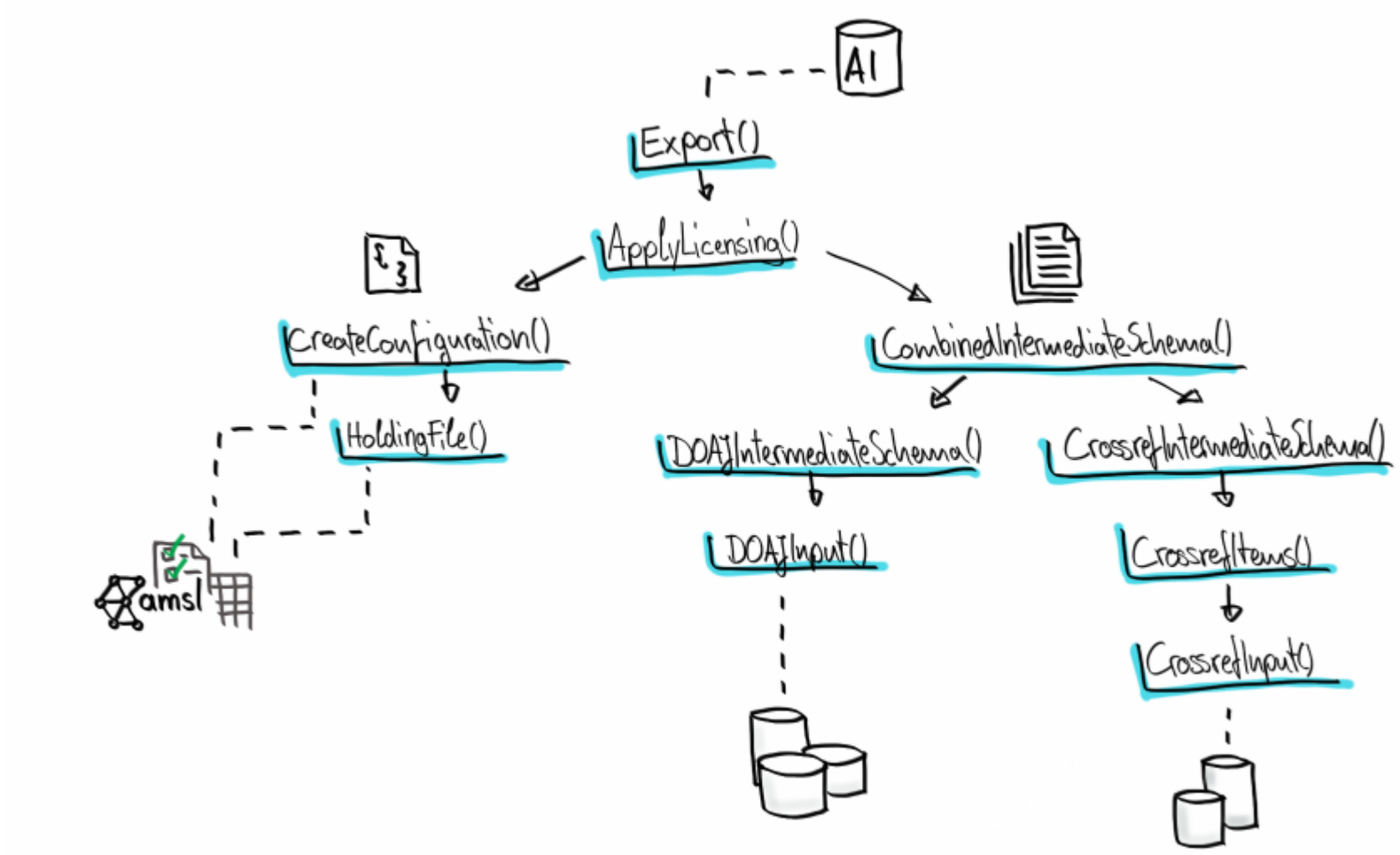
Postprocessing: Anreicherung von Daten

- Beispiel: Microsoft Academic Graph

Konvertierung in SOLR-Schema



Indexierung



Indexierung und letzte Schritte