

[illegible]

Enhancing Library Integration: Streamlining Data Imports

Elisa Dell'Ambrogio, Head of the Renouvaud Coordination Team, BCU Lausanne

Noémie Payot, Authorities and Subject Indexing Coordinator, Renouvaud Coordination Team, BCU Lausanne

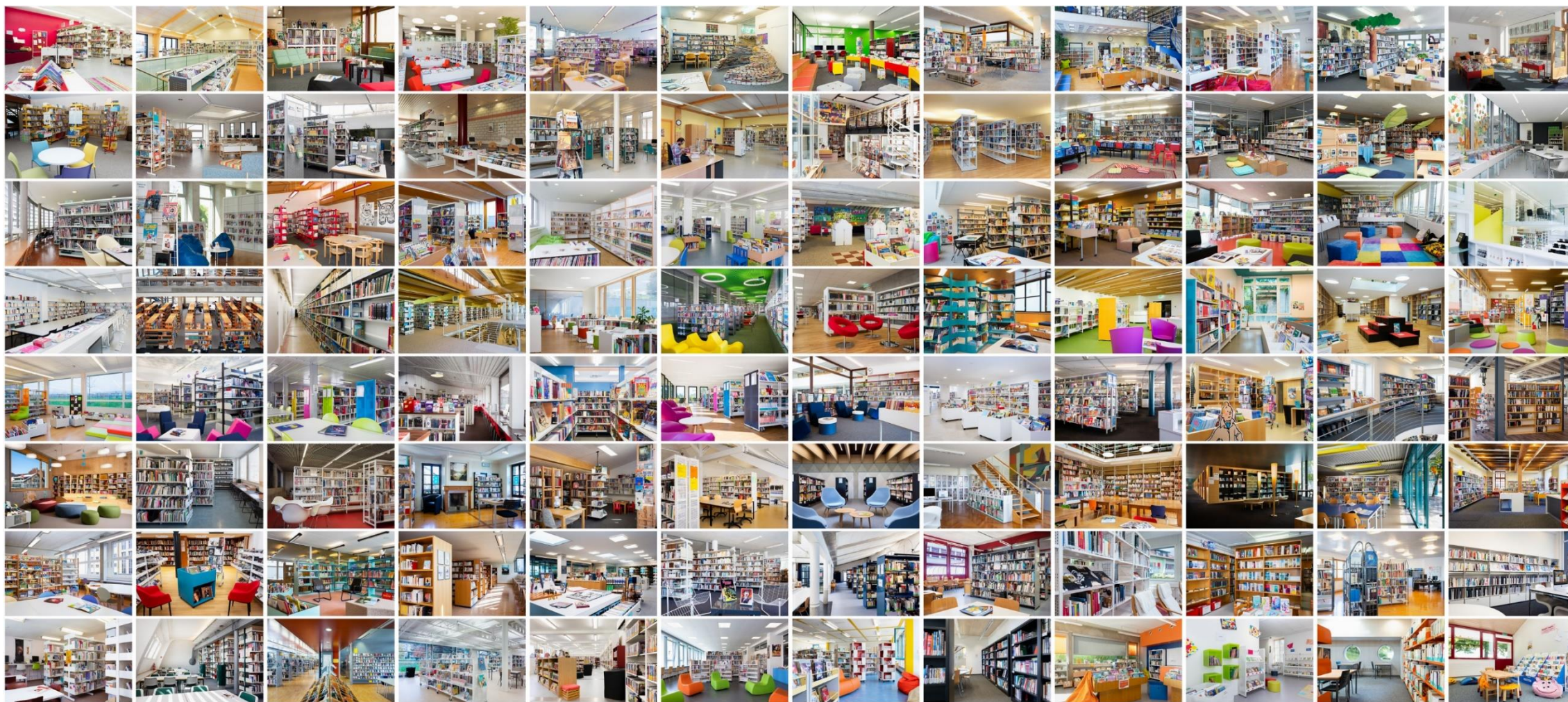
Agenda

- The Renouvaud Library Network
- Expanding Renouvaud: Library Integration
- Derivaud
- Sharing our Toolbox
- Conclusion



© BCU Lausanne. Photo by Laurent Dubois.

The Renouvaud Library Network



© BCU Lausanne. Photos by Laurent Dubois.

A regional network

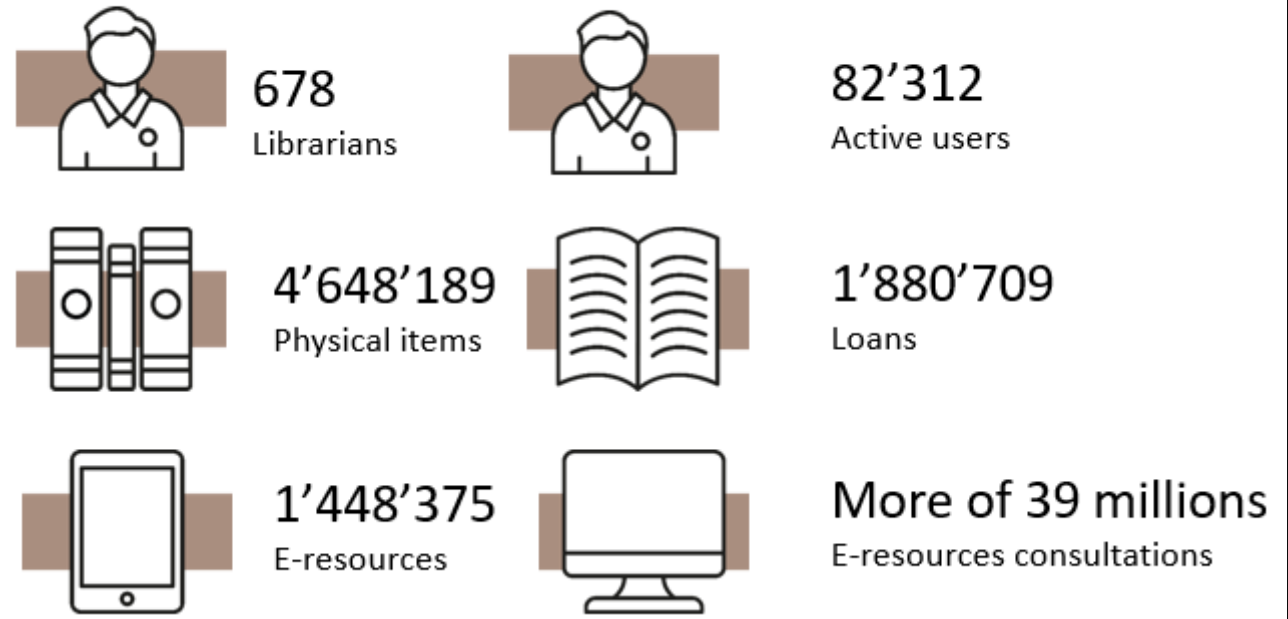
- 157 member libraries
- State of Vaud (Switzerland)
 - Capital : Lausanne
 - Language : French
 - Population: 855'700*



A regional network

- 157 member libraries
- State of Vaud (Switzerland)
 - Capital : Lausanne
 - Language : French
 - Population: 855'700*

Key figures* about Renouvaud



An heterogeneous network

18 Academic and research libraries

20 Heritage and special libraries

23 Higher education libraries

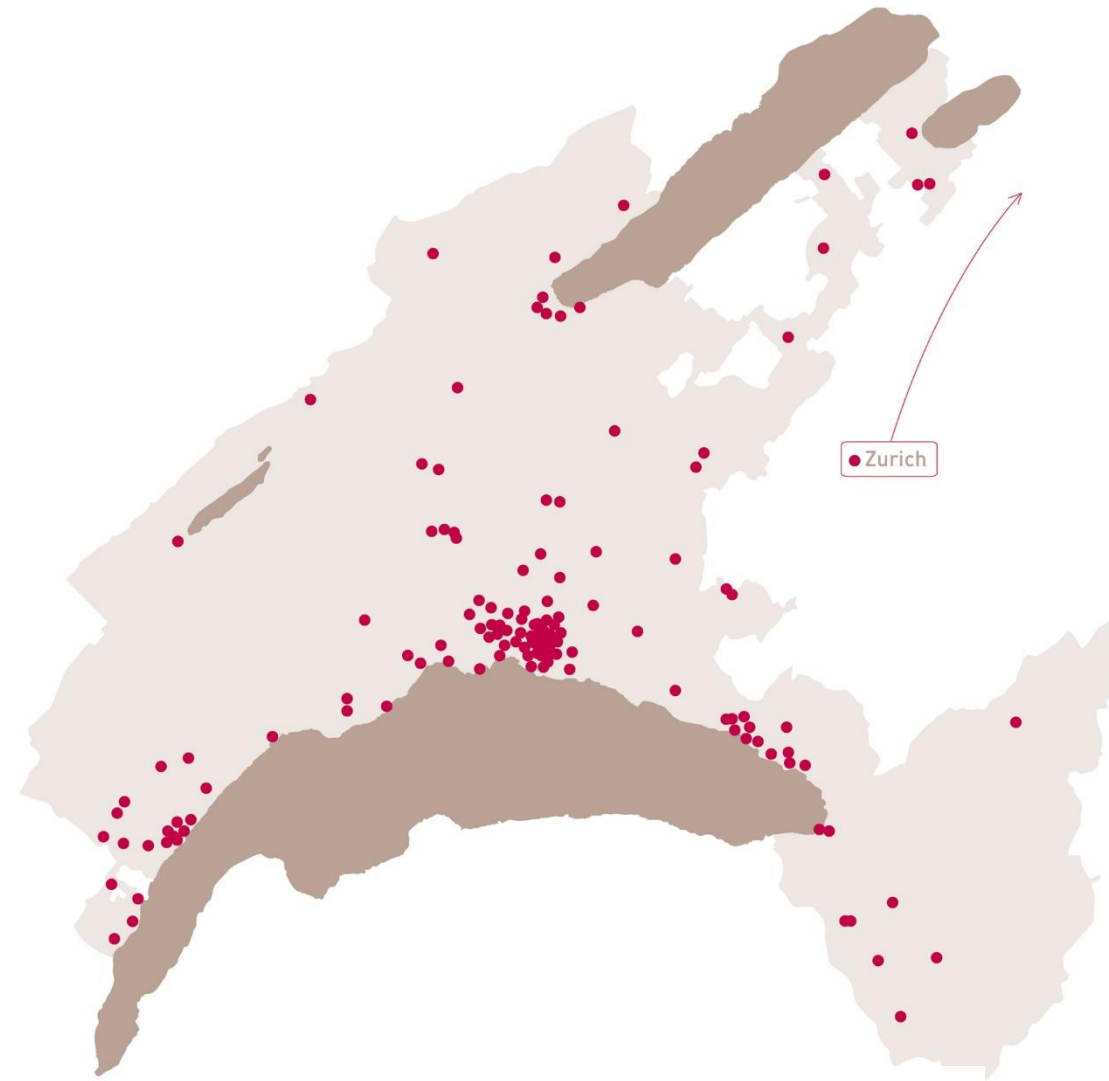
58 School libraries

32 Public and school libraries

6 Public libraries

→ 157 libraries using Alma (NZ and 2 IZ)

→ More than 60 distinct Primo VE views for end-users



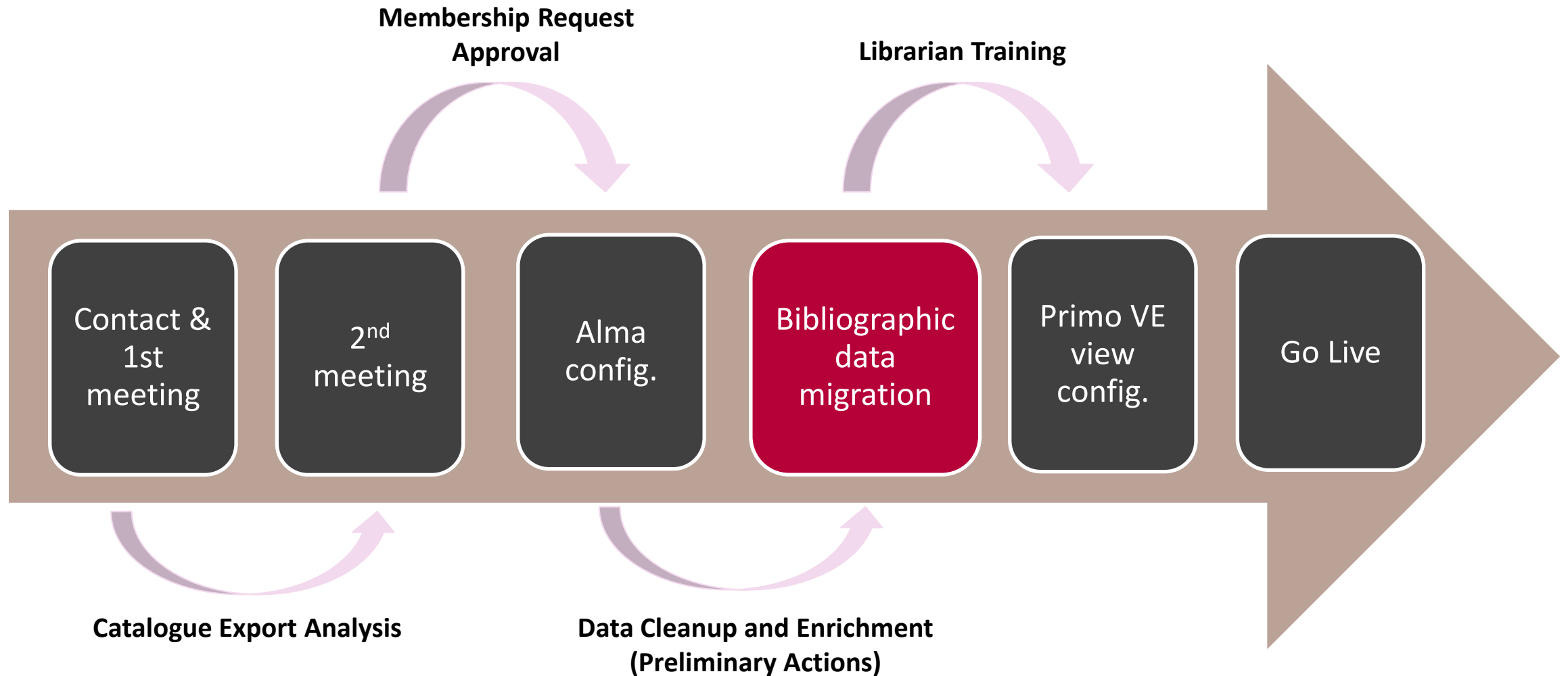
Renouvaud member libraries

A steady growth

- More than 60 new members since 2016
... and 13 libraries waiting to join the network
- Political driven consolidation :
 - Academic Libraries
 - Cantonal Museum
 - School Libraries
 - Higher education libraries
- Supported by voluntary participation



The library integration process



Challenges and goals

- Diversity of collections
- Import high-quality metadata from non-MARC data
- Custom mapping: not feasible (8 FTE for the network management)
- Avoid duplicates: rely on records already in the Renouvaud catalog
- Automate record creation: less manual cataloging, more efficiency

Entering... Derivaud (“DERivation” + “Canton of VAUD”)

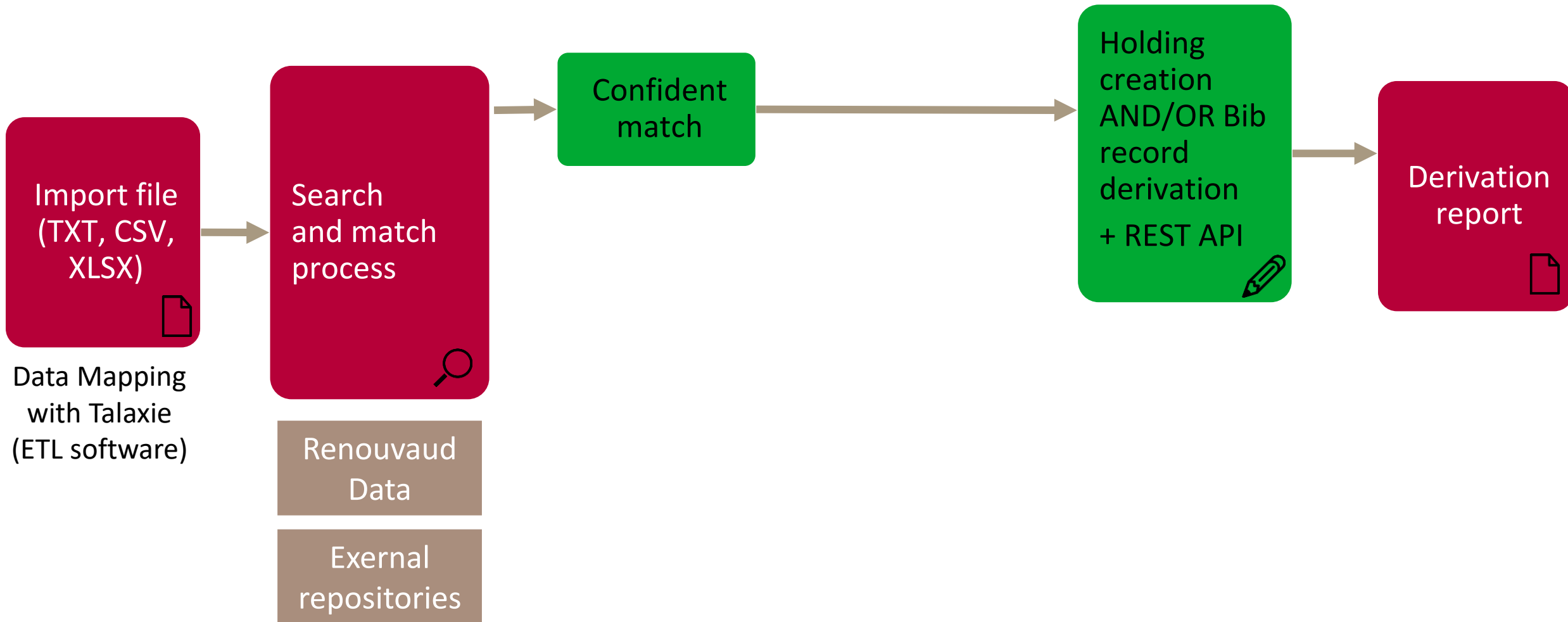
- First try: the Inventory Hooktool
- Web Application
- Collaboration with Datuman SàRL
- In use since 2019
- Programming language: Scala
- Integration with Alma REST APIs



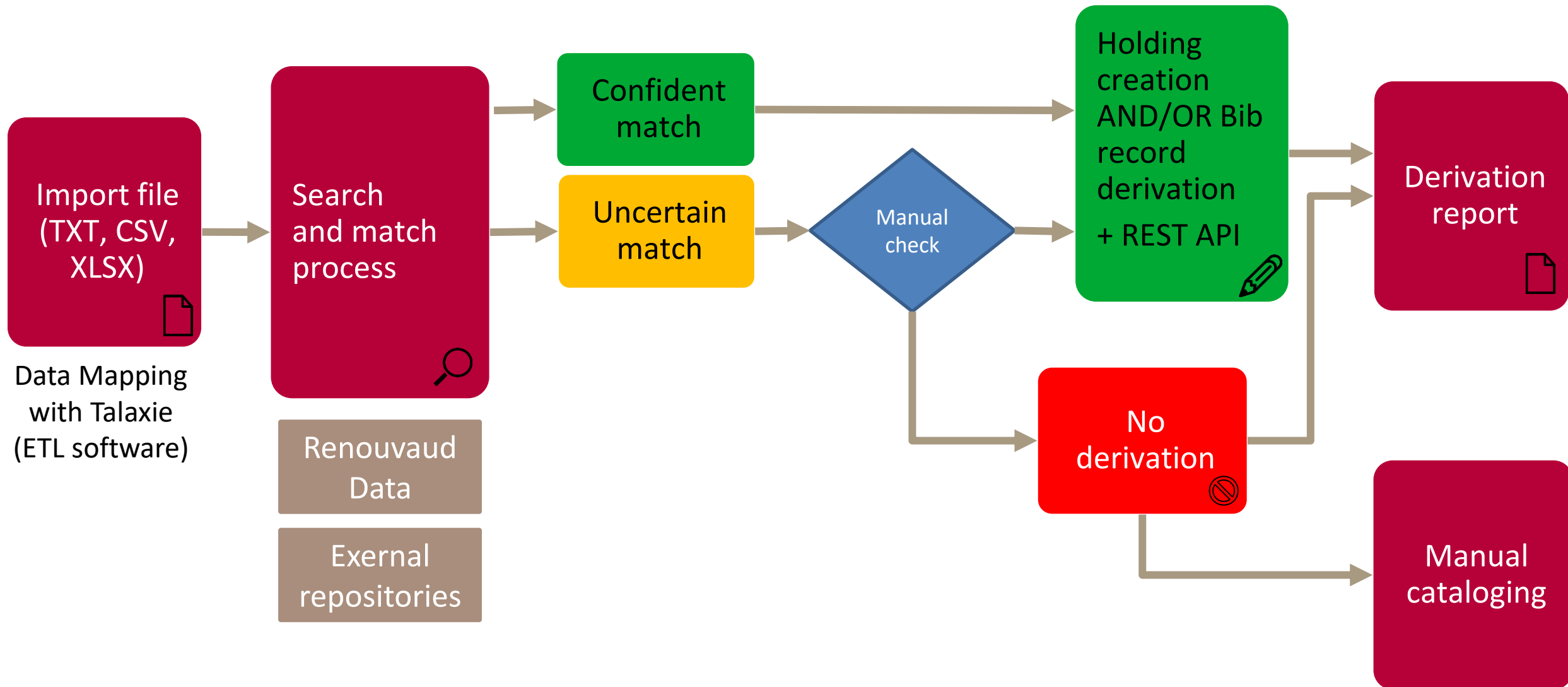
How does it work?

- Research of possible matches in the Renouvaud catalog
 - Automatically link local items to existing records in the catalog
- Record derivation
 - Retrieve records from external sources and import them into the catalog
- Required data to identify a bibliographic record match:
 - ISBN, Title, Author(s), Year of Publication, Publisher, Document type.
- Manual check via the Derivaud user interface to validate or reject the match

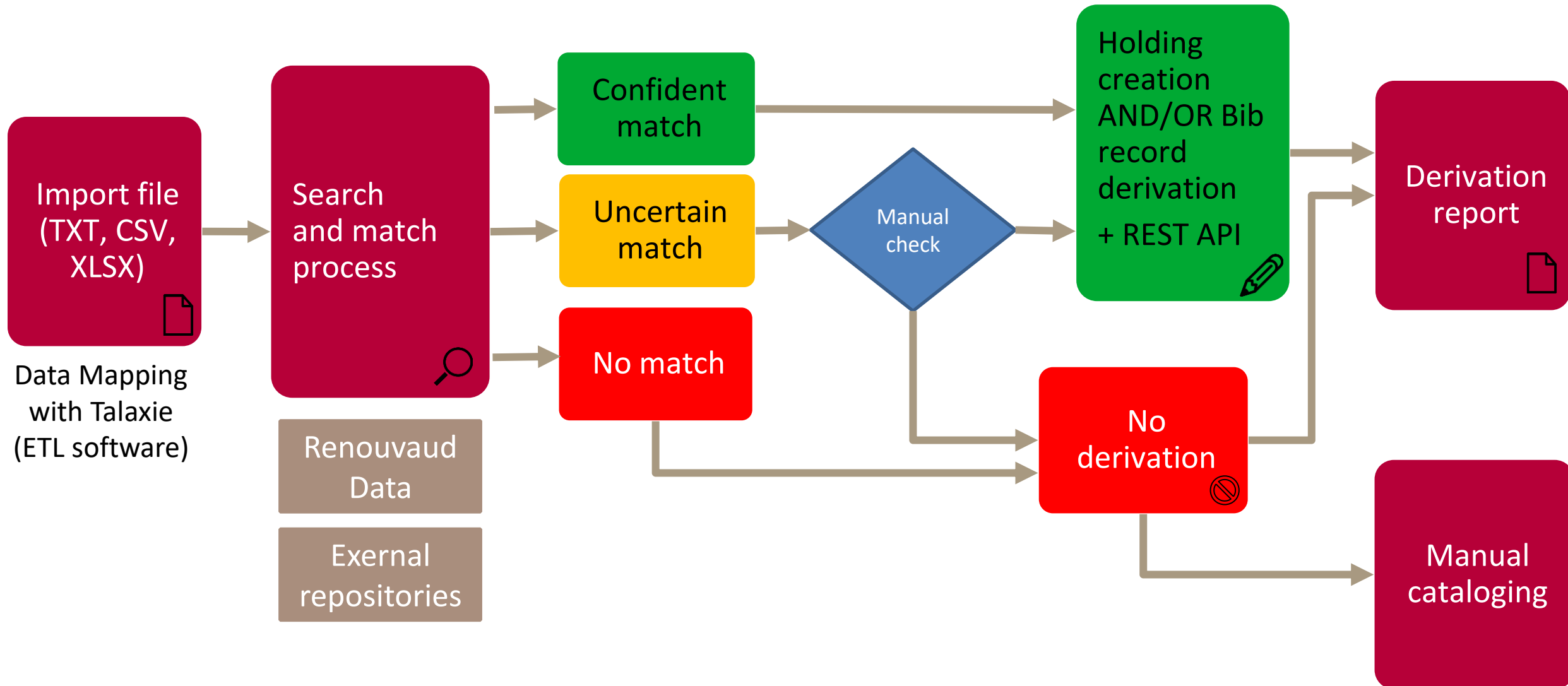
The derivation process



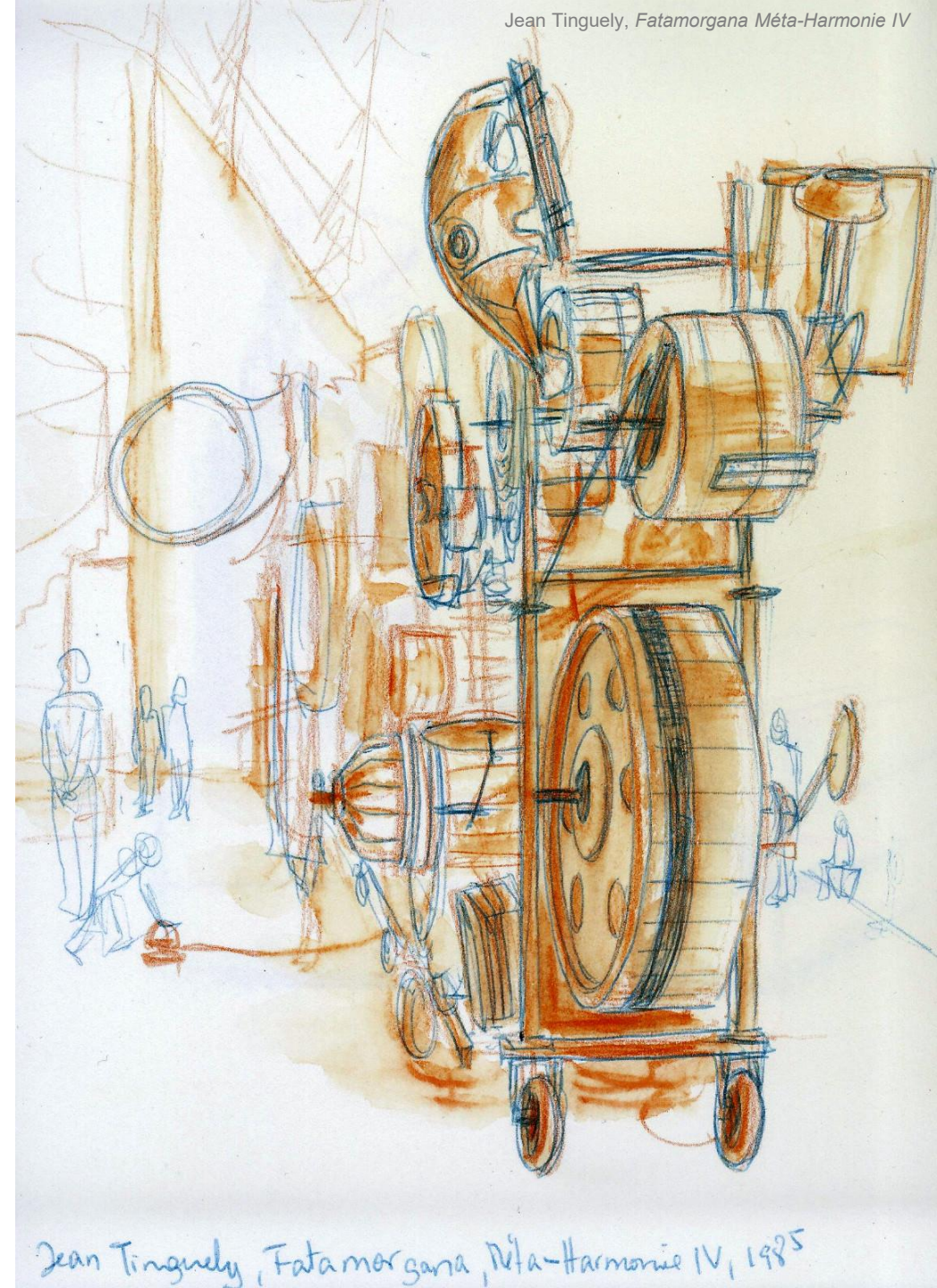
The derivation process



The derivation process



Sharing our Toolbox



Creative context

Optimize mapping for XML files

Import specialized data

Avoid manual cataloging :

Import of Derivaud data without matching

Import of bibliographic data that did not require deduplication

- Unpublished documents such as Grey literature
- Special types of documents that are rarely found in Renouvaud network (e.g. vinyl records)

Why develop additional tools?

Speed of development and cost

The Alma import profiles have been tested but deemed unsatisfactory:

- Work doubled due to importing into NZ and then IZ for local fields and inventory data.
- Problems encountered with multiple holdings and items.
- No log report available with details of mms id, holding id, item pid, and barcodes created.

Our toolbox

Marc2Derivaud : Available on demand

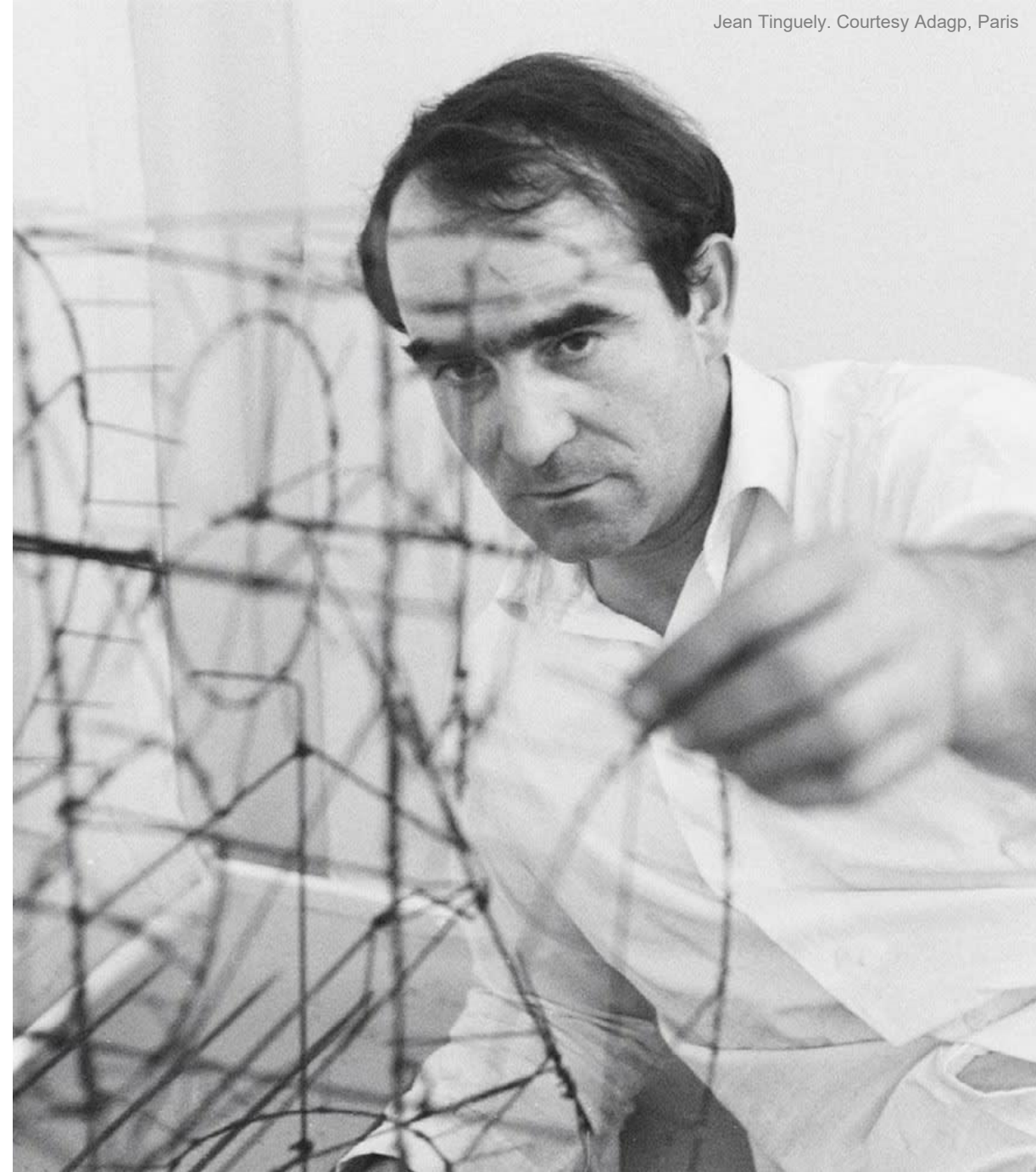
ExcelMarc-Bridge : Available on GitHub

Marc2Alma : Available on GitHub

How-to use them?

- Clone the GitHub repository to your machine
- Define your configuration in json files (from template)
- Run the script

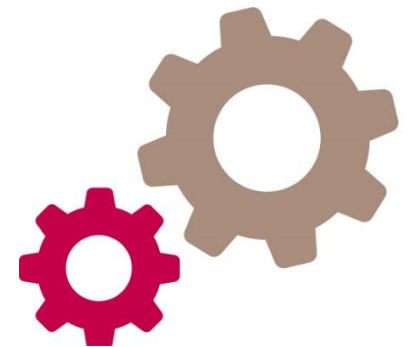
Step by step



ExcelMarc-Bridge

Download link : <https://github.com/Renouvaud/ExcelMarc-Bridge>

This project is a Python script that converts an .xlsx file into a MARC21/XML format ready to be uploaded into the Alma ILS (including holdings and item data).



Copy repository

<https://github.com/Renouvaud/MARC2Alma>

The screenshot shows the GitHub repository page for **MARC2Alma** by user **Renouvaud**. The repository is public and has 1 branch and 0 tags. The main branch is selected. The repository contains files: `.gitignore`, `LICENSE`, and `README.md`, all from the initial commit. The license is GPL-3.0. The 'Code' button is highlighted with a red box. The dropdown menu is open, showing options to clone the repository using HTTPS, SSH, or GitHub CLI, or to open it with GitHub Desktop. The 'Download ZIP' option is also highlighted with a red box.

MARC2Alma Public

main 1 Branch 0 Tags

Go to file t Add file <> Code ?

Local Codespaces

Clone ?

HTTPS SSH GitHub CLI

`https://github.com/Renouvaud/MARC2Alma.git`

Clone using the web URL.

Open with GitHub Desktop

Download ZIP

Configuration of gen_params

```
"xlsx_file_input": "input_file_example",
"xml_file_output": "output_file",
"global_dict":{
    "codes_bib": {
        "code_bib_alma" : "test"
        "code035": "TEST"
    },
"mapping":[
    ["035",
        ["a", "(#codes_bib['code035'])@IDJeux"
        ],
    ["040",
        ["a", "RNV #codes_bib['code_bib_alma']"
        ],
    ["245",
        ["a", "@Nom_jeu"
        ],
    ["264_2",
        ["a", "[S.1.] : "],
        ["b", "if $ ' '!=@Editeur $ @Editeur, $ [s.n.], "],
        ["c", "if $ re.match('[0-9]{2}[0-9\\?]{2}', @Annee_sortie) $ @Annee_sortie $ [s.d.]"
        ],
    ],
```

...

ExcelMarc-Bridge : Use case

GameID	Name	Year	Publisher
1911	SmartMax : My First Sounds & Senses	2021	SmartMax
704	Kikou le coucou (As d'Or enfants, Cannes 2017)	20??	HABA
1445	La Maison des Souris	2020	Gigamic
1699	Malette vétérinaire 2-en-1	?	Smoby
1697	Zig & Go Wroom (45 pcs.)	2021	Djeco
1758	Concept (As d'Or 2014)	2012	Dance Production

```
[ "264_2",
  [ "a", "[S.1.] : ",
    [ "b", "if $ '!=@Publisher $ @Publisher, $ [s.n.], ",
      [ "c", "if $ re.match('[12][0-9][0-9\\?]{2}', @Year) $ @Year $ [s.d.]" ]
    ],
  ],
```

```
<datafield tag="035" ind1="" ind2="">
  <subfield code="a">(TEST)1445</subfield>
</datafield>
<datafield tag="245" ind1="0" ind2="3">
  <subfield code="a">La Maison des Souris</subfield>
</datafield>
<datafield tag="264" ind1="" ind2="2">
  <subfield code="a">[S.1.] : </subfield>
  <subfield code="b">Gigamic,</subfield>
  <subfield code="c">2020</subfield>
</datafield>
```

Example of output file

```
<?xml version="1.0" encoding="UTF-8"?>
<collection>
  <record>
    <leader>00625nam a2200145 c 4500</leader>
    <controlfield tag="008">110624s1994    xxu||||| |||| 00| 0 zxx d</controlfield>
    <datafield tag="245" ind1="1" ind2="0">
      <subfield code="a">Bernard Reichel : </subfield>
      <subfield code="b">la découverte d'un musicien et compositeur romand /</subfield>
      <subfield code="c">Micheline Tschanz</subfield>
    </datafield>
    <datafield tag="985" ind1="" ind2="">
      <subfield code="2">ctest</subfield>
      <subfield code="a">M- 130</subfield>
      <subfield code="9">local</subfield>
    </datafield>
    <holding_data>
      <leader>      nx  a22      1n 4500</leader>
      <datafield tag="852" ind1="" ind2="">
        <subfield code="b">vdtest</subfield>
        <subfield code="c">814080003</subfield>
        <subfield code="h">M- 130</subfield>
      </datafield>
    </holding_data>
    <item_data>
      <item_location>814080003</item_location>
      <barcode>TEST-115226</barcode>
      <policy>3000</policy>
      <physical_material_type>BOOK</physical_material_type>
    </item_data>
  </record>
</collection>
```

ExcelMarc-Bridge: Possibilities

What we saw today :

- Add fixed value
- Add value from the input Excel file using columns header (@column_header)
- Add value using simple condition (if § condition § true value § optional false value)

ExcelMarc-Bridge: Possibilities

Other possibilities :

- Add value using multiple conditions (if, elif, elif, .., else)
- Add value using Python dictionary (`#dict_name['input value']`)
- Convert value (using Python dictionary) → E.g. input value is EN, FR, DE and output value is eng, fre, ger
- Merge values using loops (with external Excel file or Python dictionary) → E.g. items and records are in different files
- Convert date format → E.g. input value is 16.09.2025 10:09:30 and output value is 2025-09-16

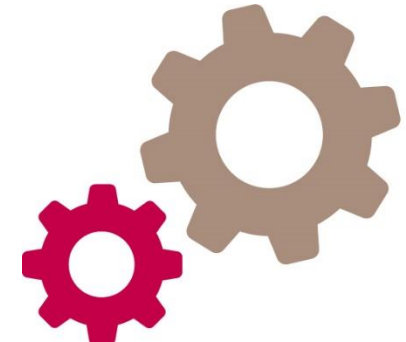
Marc2Alma

Download link : <https://github.com/Renouvaud/MARC2Alma>

This project is a Python script designed to upload a MARC21/XML file containing bibliographic records, holdings, and items into the Alma ILS.

Requirements :

- Network Zone (NZ) administrator role
- Institution Zone (IZ) administrator role
- Similar setup as MarcExcel-Bridge
- 2 configuration files in json format



Format used for Alma APIs :

[Bibliographic records](#)

[Holdings and items](#)

```
<collection>
```

```
  <record>
```

```
    <mms_id></mms_id>
```

```
    <leader></leader>
```

```
    <controlfield tag=""></controlfield>
```

```
      <datafield tag="" ind1="" ind2="">
```

```
        <subfield code=""></subfield>
```

```
      </datafield>
```

```
    <holding_data></holding_data>
```

```
      <datafield tag="" ind1="" ind2="">
```

```
        <subfield code=""></subfield>
```

```
      </datafield>
```

```
    <item_data></item_data>
```

```
  </record>
```

```
</collection>
```

Configuration of secure_params

```
"api_keys": {  
  "api_key_nz_prod" : "your api key nz prod",  
  "api_key_nz_sb" : "your api key nz sb",  
  "api_key_iz1_prod" : "your api key iz1 prod",  
  "api_key_iz1_sb" : "your api key iz1 sb",  
},
```



prod → production environment
sb → sandbox environment



nz → Network Zone (NZ)
iz1 → to rename with your
Institution Zone (IZ) code

```
"sru_link": {  
  "url_sb" : "https://  
  "url_prod" : "https://  
}
```

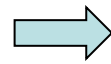
Alma documentation :

[API keys](#)

[SRU](#)

Configuration of general_params

```
"env_param" : {  
    "env" : "sb"  
},  
"inst_param" : {  
    "inst" : "iz1"  
},  
"file_param" : {  
    "XMLfile_name" : "test_books_2rec",  
    "XMLfile_path" : "./test_files/"  
},  
...
```



IZ code



use the same code as for API
keys, e.g. :
"api_key_iz1_prod"

Marc2Alma : Possibilities

What we saw today :

- Create/update bibliographic records in NZ/IZ
- Create/update holdings
- Create items

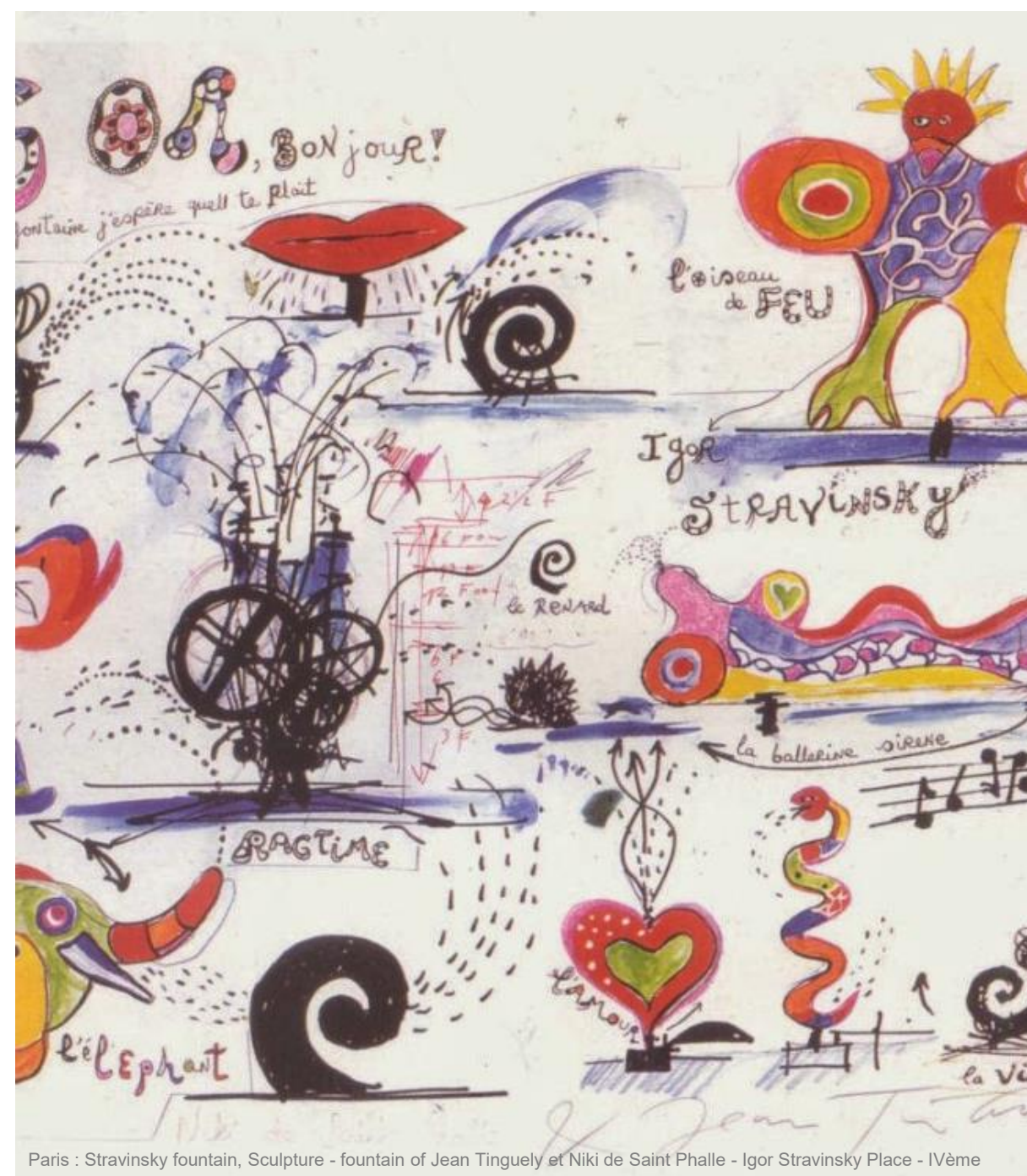
Marc2Alma : Possibilities

Other available options :

- Apply normalization rules for created/updated records
- Record deduplication based on NZ ID, IZ ID, 035 tag (other system number) or SRU search (020, 022 and 024 tags used for search, other tags can be used to filter results)
- Merge options : choose which datafield and subfield to add for updated records
- Holding hanging options : holding is choosen according to the item call number (regular expressions can be used)

What you will find on GitHub

- Full documentation
- Input file to test
- Example of log report / output file



Conclusion

Goal of zero (or almost zero) manual cataloguing → achieved

Goal of a high-quality catalogue → need additional tools to:

- Deduplicate records within the Renouvaud catalogue
- Improve record quality (e.g. remove duplicate information)

Still a work in progress!

Thank you!

elisa.dellambrogio@bcu.unil.ch

noemie.payot@bcu.unil.ch

<https://github.com/renouvaud>



© BCU Lausanne. Photo by Laurent Dubois.