



NIE-INE

National Infrastructure for Editions

Roberta Padlina

Conference "Open Access in Action" by swissuniversities and ZHAW
19th October 2020

Supporting Institutions

- swissuniversities
- Schweizerische Akademie der Geistes- und Sozialwissenschaften (SAGW)
- Universitätsbibliothek Basel
- Zentralbibliothek Zürich
- Universität Basel
- Universität Bern
- Universität Zürich

Edition Projects I

- An Online Repertory of Commentaries on Peter Lombard's Sentences
- Anton Webern Gesamtausgabe (AWG)
- Karl Barth-Gesamtausgabe
- Bernoulli-Euler-Online (BEOL)
 - Leonhardi Euleri opera omnia (LEOO)
 - Basler Edition der Bernoulli-Briefwechsel (BEBB)
- C.F. Meyers Briefwechsel. Historisch-Kritische Ausgabe
- Digitale Historisch-kritische Gesamtausgabe der Werke und Briefe von Jeremias Gotthelf (DHKG)
- Heinrich Wölfflins Gesammelte Werke

Edition Projects II

- Johann Caspar Lavater: Historisch-kritische Edition ausgewählter Briefwechsel
- Das Kloster-Tagebuch des Einsiedler Paters Joseph Dietrich, 1670-1704. Kommentierte Online-Edition
- Kritische Robert Walser-Ausgabe (KWA)
- Kuno Raeber: Online-Edition des lyrischen Nachlasses
- Online-Edition der Paippalada-Rezension des Atharvaveda
- Parzival-Projekt
- Reconstruire Delille
- Der späte Nietzsche. Manuskriptedition des Nachlasses 1885-1889 (KGW IX)

Scientific Edition

critical open access

scholarly edition = access to text + critical information



FAIR data & FAIR knowledge

Findable - **A**ccessible - **I**nteroperable - **R**Reusable

Semantic Web

To solve the issue of interpretation or meaning of information coming from different sources

- standard formal languages (W3C)
- for **knowledge representation**
- based on model and set theory, and first order logic

Semantic Web - Benefits

- machine interpretable semantics
- semi-automated semantic interoperability
- FAIR formal linked data
- natural language independence → global information exchange
- explicit, self-descriptive semantics → data transparency and information disambiguation
- machine reasoning
 - automated data quality assurance (consistency checking)
 - enhanced expressivity by inferring data from data
 - calculations

Semantic Web - Services and Products

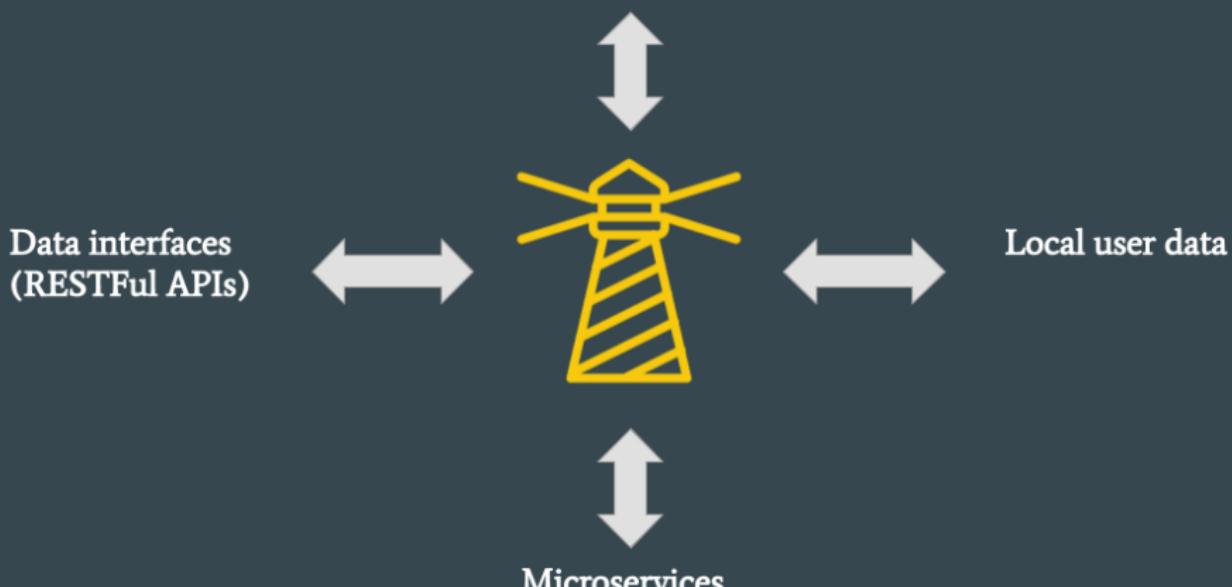
→ machine interpretable and interoperable semantics for the
Humanities

- ontologies
 - domain knowledge
 - based on W3C, CIDOC and FRBRoo standards
 - basic modelling patterns
- best practices and methodologies
- two-step formalization
- machine reasoning based on Notation3 rules

inseri Web Framework

online collaborative working and publishing environment

Online Components / inseri apps

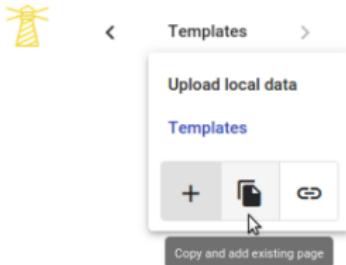


inseri - Local User Data (Cloud Storage)

The image shows a screenshot of the inseri - Local User Data (Cloud Storage) application interface. It consists of two main windows:

- Left Window:** Titled "Upload local data". It features a search bar "Search all appgroups" and a navigation bar with "Add, Edit & Manage Data" and "Visualis" tabs. Below this are several appgroup icons:
 - Data management
 - Comment App
 - My Files
 - Text Editor
 - Spreadsheet
- Right Window:** Titled "Upload local data". It shows a list of appgroups with icons:
 - HTML Viewer
 - Machine Reasoning
 - Comment App
 - Json Environment
 - Audio Player
 - My Files
 - Text Editor
 - Plaintext Viewer
 - PaintCanvas
 - OpenBis Login
 - Image Viewer
 - Simple Image App
 - Data list viewer
 - Tree navigation
 - Youtube Video Viewer
 - Synopsis Viewer

inseri - RESTful APIs and Templates



	Gallica (Bibliothèque nationale de France)	50149 manifests Digitized manuscripts from Gallica, the digital library of the National Library of France (BnF)
	BVMM (IRHT-CNRS)	8099 manifests Digital library of medieval manuscripts (IRHT-CNRS)
	The British Library, Polonsky Pre-1200 Project	405 manifests The Polonsky Foundation England and France Project (700-1200): Manuscripts from the British Library
	Cambridge Manuscripts	template test
	Oxford IIIF	Bodleian Libraries' digital collections (Oxford) 5883 manifests
	Nietzsche Pages	Import Pages of the Nietzsche Source http://www.nietzschesource.org/
	e-codices manuscript import	Read, comment on and analyse manuscripts from e-codices

inseri - e-codices Template & Comments

New comment

Add comment



<https://www.e-codices.unifr.ch/metadata/W/Mu-MurF0084/manifest.json>

e-codices Manuscript Import | ref: /W/Mu-MurF0084
Please paste the link of the manuscript above and click "save" to import it to inseri



V2



This composite manuscript consists of an intiale from Freiburg I. Br. from 1494 that has parts in my manuscript, which was written in 1495 and 1496 by Bernhard Jenichen, Canon of Agaibach Abbey in the Forest. The three texts on solvivalent logic are by Peter of Spain and by Petrus Tartareus, a contemporaneous Peripatetic philosopher whose mnemonic device, a logical figure called pars astrovare has also been copied.



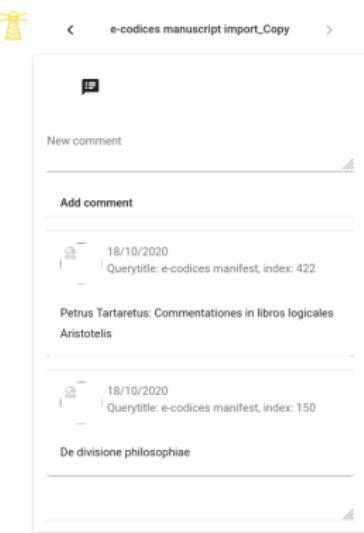
DOI: 10.5071/e-codices-Abbe-MurF0084
Metadata | ref: /W/Mu-MurF0084
This app enables you to browse

New comment

Add comment

18/10/2020 | Querytitle: e-codices manifest, index: 422
Petrus Tartareus: Commentationes in libros logicales Aristotelis

18/10/2020 | Querytitle: e-codices manifest, index: 150
De divisione philosophiae



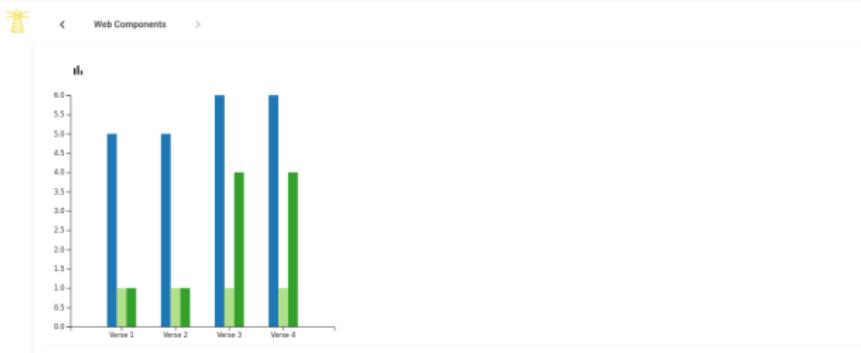
inseri - Web Components (Apps)

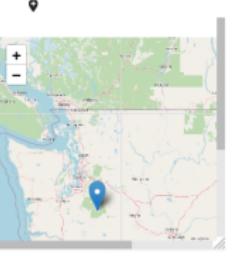
The screenshot shows the inseri application interface with a sidebar icon and a header bar labeled "Apps".

The main area is divided into three horizontal sections:

- Add, Edit & Manage Data** (selected):
 - Data management
 - Comment App
 - My Files
 - Text Editor
 - Spreadsheet
- Visualise**:
 - HTML Viewer
 - Plaintext Viewer
 - PaintCanvas
 - Image Viewer
 - Simple Image App
 - Data list viewer
 - Tree navigation
- Web Applications**:
 - Microservice
 - Audio Payer
 - Youtube Video Viewer
 - Synopsis Viewer
 - Sparql - Visualiser
 - Crispr
 - Salsah 2
 - Anton Webern Gesamtausgabe

inseri - Web Components (Apps)





Editor Code

Normal **Sans Serif** **B** **I** **U** **A** **M** **E** **E** **Q** **C** **T**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Save Text and update Page

inseri - Microservices

The screenshot shows a web-based interface for managing microservices. At the top, there's a navigation bar with a lightning bolt icon, the text "Microservices", and back/forward arrows. Below the header, there are three main sections: "INPUT", "PYTHON CODE", and "OUTPUT".

INPUT: This section displays a portion of a JSON object under the key "results". The visible part includes "bindings" and two nested objects: "stropheOrdinalNumeral" and "verseOrdinalNumeral", each with "type" and "value" fields.

```
"results": {
  "bindings": [
    {
      "stropheOrdinalNumeral": { "type": "literal" , "value": "1" } ,
      "verseOrdinalNumeral": { "type": "literal" , "value": "001" } ,
```

PYTHON CODE: This section contains a Python script for processing the input. The script imports json, defines a pieChart function, and iterates through the input data to calculate citation counts and generate chart input.

```
1 import json
2
3 def pieChart(input, labels):
4
5     with open(input) as json_file:
6         # Parse JSON into a Python dictionary
7         data = json.load(json_file)
8
9         # Getting iterable of actual results
10        bindings = data["results"]["bindings"]
11
12        # Dict for the total number of citations per element of interest
13        sums = {}
14
15        # Populating sums() with the given elements and count = 0
16        for k, v in labels.items():
17            sums[k] = 0
18
19        # Iterating through query results
20        for l in bindings:
21            # For each verse, check if element is present and if yes,
22            # add the citation count to the total counts in sums()
23            for k, v in labels.items():
24                if k in l:
25                    sums[k] += int(l[k]["value"])
26
27        # Total of all sums in sums()
28        sum_total = sum(sums.values())
29
30        # Container to create chart input
31        results = []
32
33        # Iterating through elements of interest and
34        # creating according chart input
35        for k, v in labels.items():
36            results.append({
```

OUTPUT: This section is currently empty, indicated by a large white box.

At the bottom left, there are buttons for "Save Python Code and Update dependent apps" and "Run".

Questions?

<https://www.nie-ine.ch>

<https://github.com/nie-ine>

<https://nie-ine.github.io/inseri/>

<http://e-editiones.ch/>

roberta.padlina@unibas.ch