Developing a Zenodo Jupyter Lab Extension
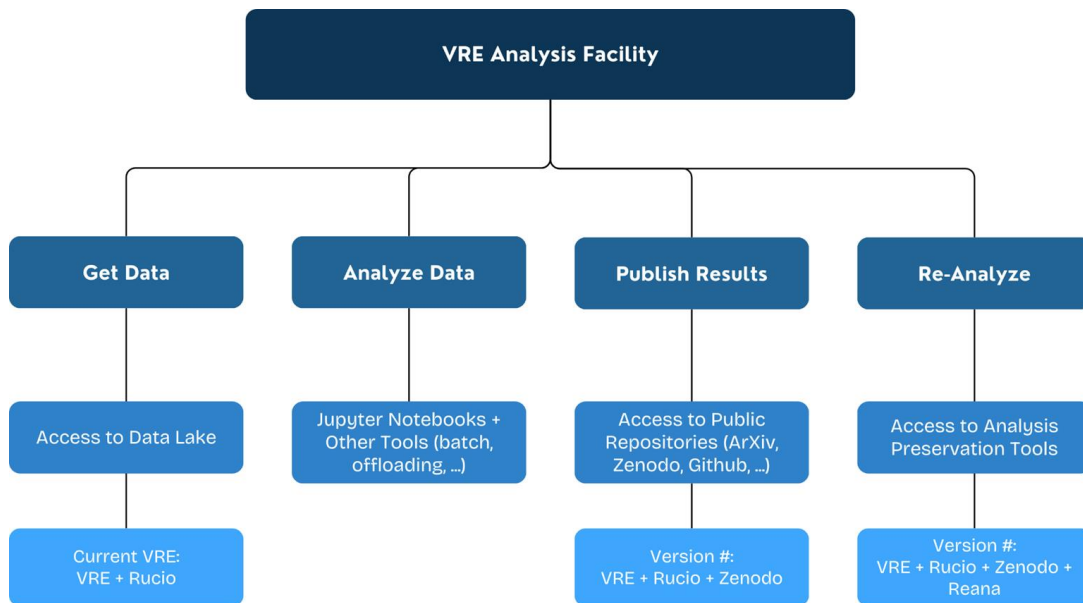
Michael Zengel

08/08/2024

The Virtual Research Environment (VRE)

Grand Purpose of VRE

- An analysis facility based on the Jupyter framework
- Simplify life for physicists by aggregating software and infrastructure
- Developed by the ESCAPE collaboration
- Goal: End-to-end Scientific Analysis Workflow in a Cloud-Based Environment

VRE Analysis Facility

Get Data	Analyze Data	Publish Results	Re-Analyze
Access to Data Lake	Jupyter Notebooks + Other Tools (batch, offloading, ...)	Access to Public Repositories (ArXiv, Zenodo, Github, ...)	Access to Analysis Preservation Tools
Current VRE: VRE + Rucio		Version #: VRE + Rucio + Zenodo	Version #: VRE + Rucio + Zenodo + Reana

This project: Context

**Zenodo:** Open-source database software for sharing results/code

- Goal: Incorporate into VRE via Jupyter Lab, widely used interactive framework

  - Increases speed and ease of downloading and uploading data

  - Removes the need for local storage interaction; **fully cloud-based**

  - Exploits command line interface (CLI) in a visual way

  - Incorporates the "Publishing Results" step of the VRE end goal
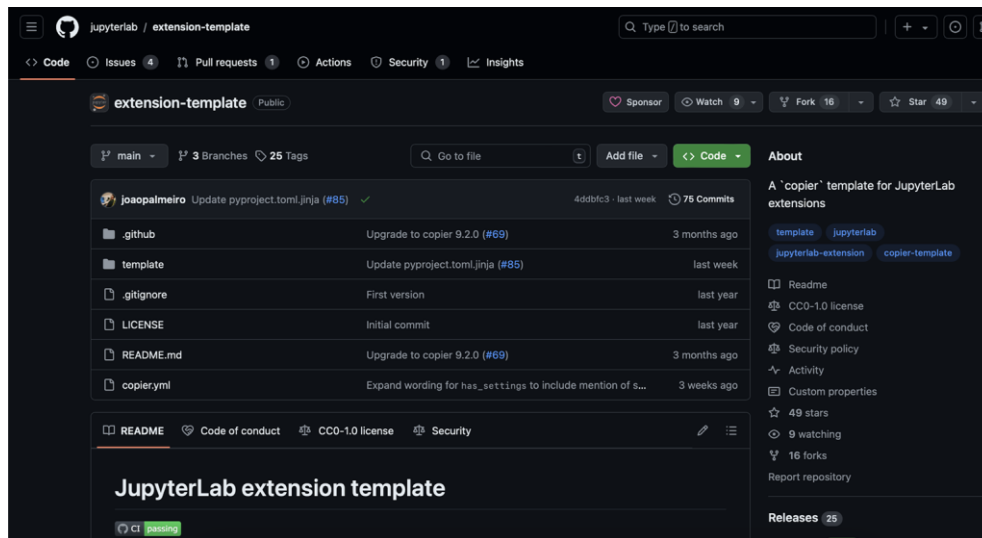
    - https://github.com/vre-hub/zenodo-jupyterlab-extension
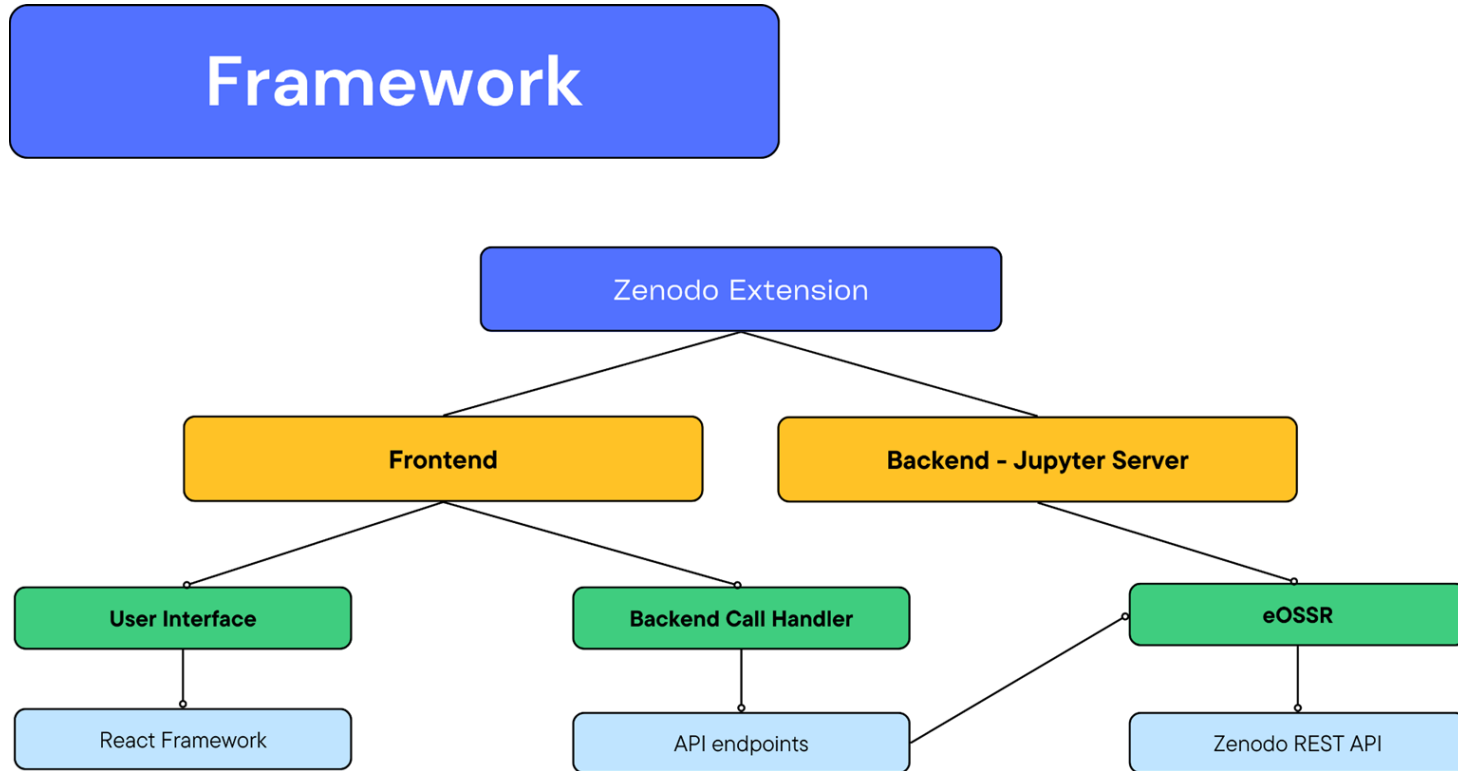
General Framework

# General Framework

Frontend Design

- Built off of copier extension template
- Developed via Nodejs 20 and React
- Rendered as a Sidebar Widget
- Extends JupyterFrontEnd app

General Framework

Backend Design

- Jupyter Server Extension (separate from Frontend Extension)
- Hosts API calls
- Runs eOSSR scripts for searching, logging in, and uploading data
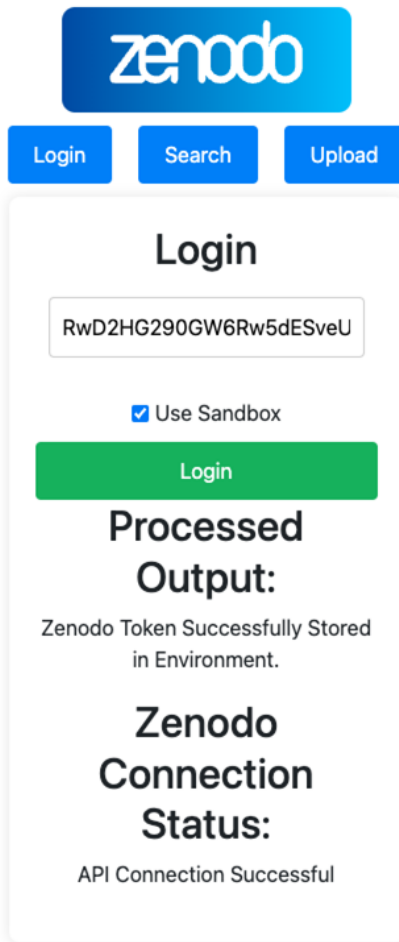 - eOSSR is a python library developed as a part of the ESCAPE Project
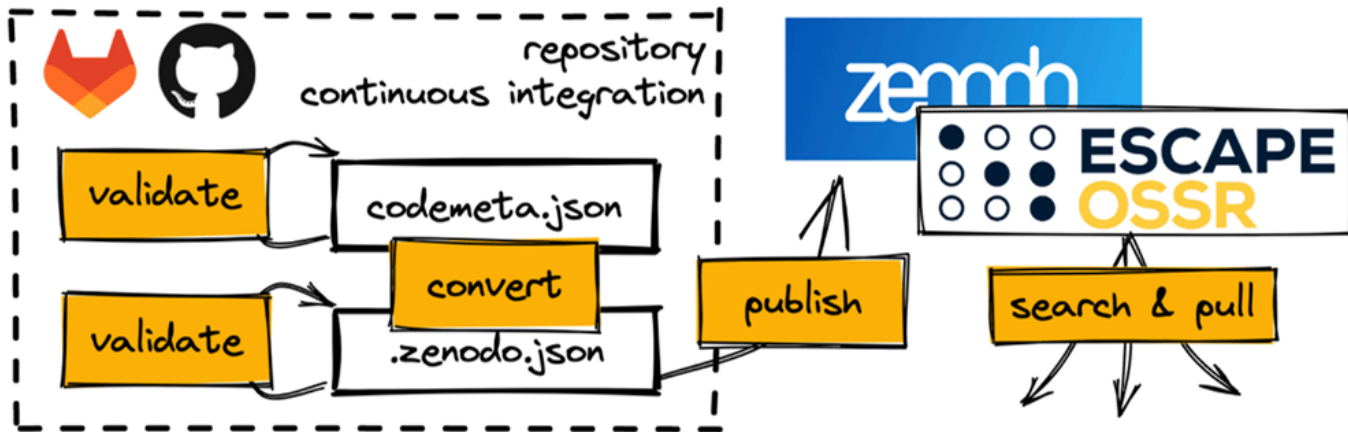
Major Components

Logging in to Extension

- Takes in API Access Token

- Validates via deposit query status code

- Stores in env var for use throughout JupyterLab instance

  - Securely only accessible to user within session

- Sandbox functionality (stored for use in uploading)

  - Searching is exclusively non-sandbox

Note: This is simply a draft of the application; cosmetic details, such as spacing, will be addressed in the future.
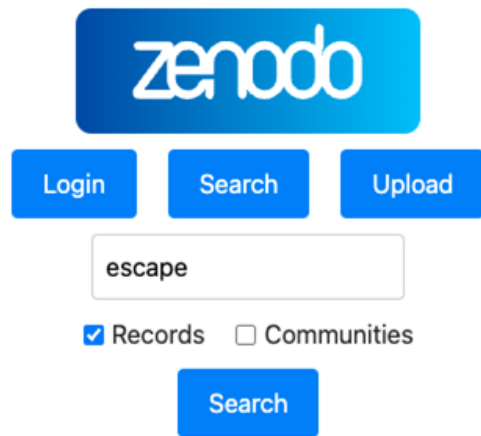
Major Components

Searching

- Uses built in Elasticsearch query string syntax from REST API

  - Searches for DOI, Title, Description, Creators, Communities

- Returns Title, Resource Type, Date Published

- Sorted by Most Recently changed (same as REST API)

Title	Resource Type	Date Published
Science Clusters: Position statement on operational commitment to EOSC and Open Research	Publication	2024-03-01
AfterSSHOC: synergies along the journey to EOSC and a view into the future	Presentation	2022-04-11

Major Components

Searching

- Uses built in Elasticsearch query string syntax from REST API
- Returns Title, Resource Type, Date Published
- Sorted by Most Recently changed (same as REST API)
- Clicking a Record gives more information
    - Title with link to Zenodo record
    - Authors (with affiliations upon hover)
    - Download links on listed files (*WIP*)
        - Now on the PC
        - Future: $HOME directory in Jupyter

Major Components

Searching

- Searchable Communities (same Elasticsearch query)
- Returns title and date published sorted by most recently changed
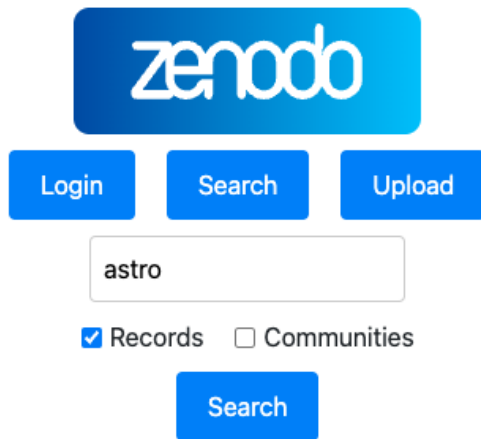
Title	Date Published
ESCAPE	2022-12-05
ESCAPE OSSR	2019-06-27
ESCAPE-NET H2020	2019-05-24
Community for ESCAPE demos	2021-04-01
ESCAPE 2018 - Workshop on Energy Scale Calibration in Anti-neutrino Precision Experiments	2018-07-18

Major Components

Searching

- Searchable Communities (same Elasticsearch query)

- Returns title and date published sorted by most recently changed

- When clicked:
  - Allows for searching of records within that community

- Possible future goal: More advanced search settings

Major Components

Uploading

- Takes in basic required info
    - Files to upload (from $HOME directory), Resource Type, Title, Creator
    - Optional: DOI (otherwise automatic), Description, Creator affiliation, multiple Creators

Major Components

Uploading

- Takes in basic required info
  - Files to upload (from $HOME directory), Resource Type, Title, Creator
  - Optional: DOI (otherwise automatic), Description, Creator affiliation, multiple Creators
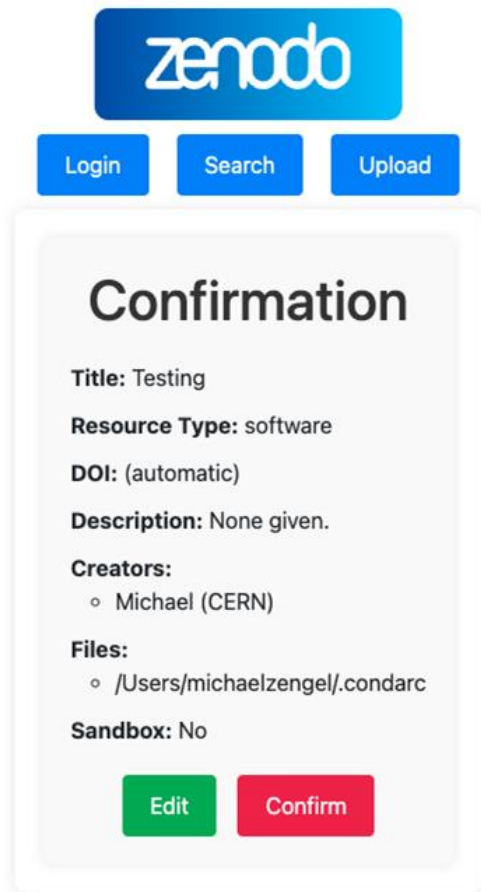- Confirmation Page of Info to Upload
- "Confirm" does the following:
  - Creates deposit
  - Sets metadata (WIP)
  - Adds files to deposit (WIP)

Install

You will need NodeJS >= 20 for these steps.

Now, install `yarn` :

```
npm install -g corepack   corepack enable
```

Install the Python dependencies from within the main project directory:

```
python -m pip install -r requirements.txt
```

Install Yarn Dependencies:

```
jlpm
```

Install and Build the Extension:

```
python -m pip install .
```

Enable the Extension:

```
jupyter server extension enable zenodo_jupyterlab.server
```

Now open a local instance of Jupyter Lab, and it should be present on the sidebar.

Docker

Rather than manually cloning the repository, it is possible to run the extension in a Docker container. To do this, use the following command:

```
docker run -d -p 8888:8888 ghcr.io/vre-hub/zenodo-jupyterlab-extension:<version>
```

All available versions can be found here

Now the instance of Jupyter Lab with the extension installed and enabled should be avilable on localhost:8888

Usage

- Downloadable and Installable via git repository

- For Developers:
  - Easy installation in development mode (-ve after install command)
  - Simple building of front-end via jlpm

- Up to date docker image available for download
  - Automatically downloads dependencies, software, installs, and runs Jupyter Lab with extension active
  - Can easily be added to Jupyter Hub distributions

Future Steps

- Continued Development of software
  - Implementation of downloads to the Jupyter $HOME directory
  - Ability to upload files to Zenodo Records
  - Advanced search settings
  - Improved cosmetic design
- Presentation of results at ADASS
  - Astronomical Data Analysis Software & Systems

Conclusions

The Software:

- Jupyter Lab Extension
- Provides Visual interface between users and the Zenodo Service
- Easily integrated into existing VRE

Why it's useful:

- Capability for **fully cloud-based** interaction (downloading and uploading) with Zenodo
- Adds another step into the VRE based analysis workflow
- Allows for more seamless downloading and uploading of results and software
- Applicable to any Jupyter-based environment

Thank you!
Any Questions?

ESCAPE project & ESCAPE Data Lake

ESFRI RIs

- Addressed RI's needs in Data Management, Access and Analysis for **Astro-particle**, **Radio-astronomy**, **Gravitational Waves**, **Cosmology** and **Particle Physics**.

- Provided a fully working **common data infrastructure** "The ESCAPE Data Lake" to test novel data management tools and models, giving the RI's the opportunity to influence and steer its development.

- Expanded **collaborations** and fostered involvement with other Scientific Communities. Maintained and strengthened collaborations with related EC initiatives and projects.

- ESCAPE finished Jan '23 and become an open collaboration [link]   -> link

Data centres