MLExchange: bringing machine learning to scientific discovery

Presenter: Zhuowen Zhao

Advanced Light Source, Lawrence Berkeley National Laboratory

SMP Materials Seminar, 1/17/2023

Acknowledgement

MLExchange architecture team

Adam Green (LBNL, now at Apple)

Alexander Hexemer (LBNL)

Dylan McReynolds (LBNL)

Elizabeth Holman (LBNL, now at Stanford)

Hari Krishnan (LBNL)

Howard Yanxon (ANL)

Tanny Chavez (LBNL)

Tibbers Hao (LBNL)

Zhuowen Zhao (LBNL)

Collaborators

AIMM team

Apurva Mehta (SLAC)

Dani Ushizima (LBNL, CAMERA)

Ed Bernard (LBNL, Molecular Foundry)

Eric J. Roberts (LBNL, CAMERA)

Nicholas Schwarz (ANL)

Petrus H. Zwart (LBNL, CAMERA)

Ronald J. Pandolfi (LBNL, CAMERA)

Sergei V. Kalinin (ORNL)

Stuart I. Campbell (NSLS-II)

Subramanian Sankaranarayanan (ANL)

Funding

financial support by the U.S. Department of Energy through Collaborative Machine Learning Platform for Scientific Discovery (award No. 107514).

Table of contents

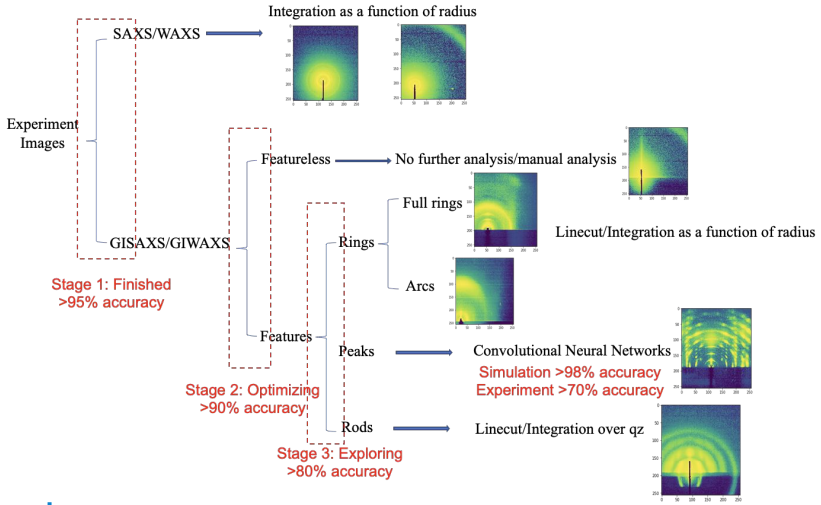
- Introduction (challenges)
- MLExchange architecture
- Live demonstration
- Use cases

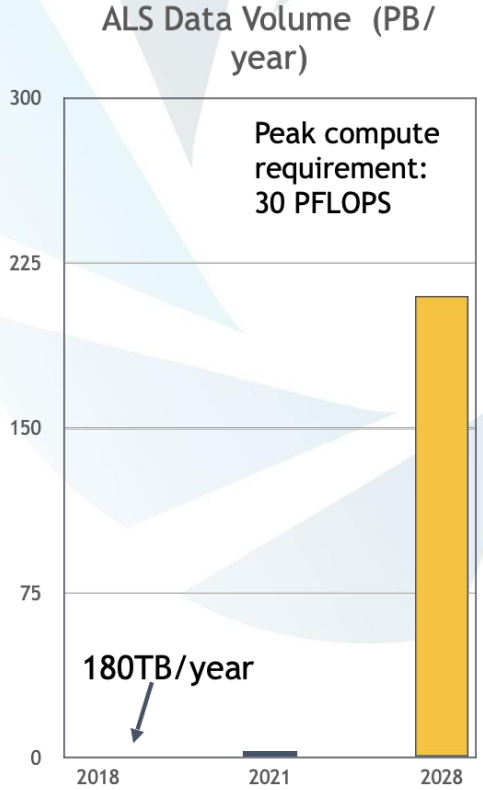
Challenges

Data challenges for beamline users

- Big amount of data, growing annually
- On-the-fly analysis dealing with various data types (spectra, image data, etc.)
- Interactive visualization tools

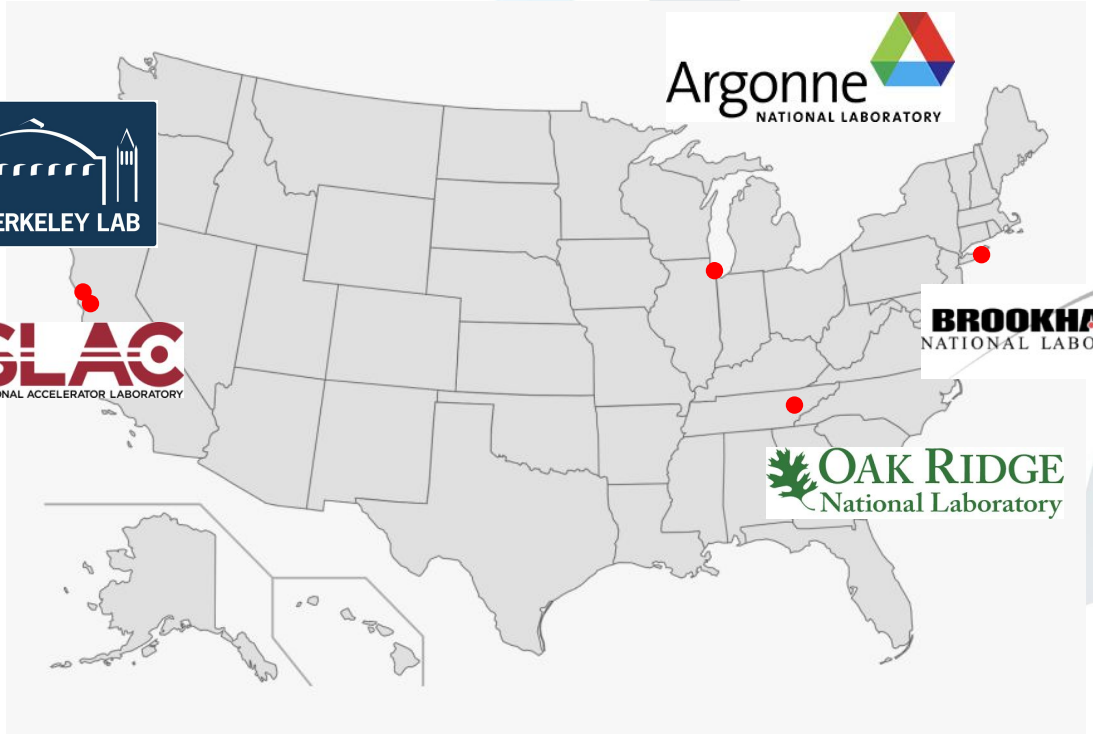
Machine learning tools can help!





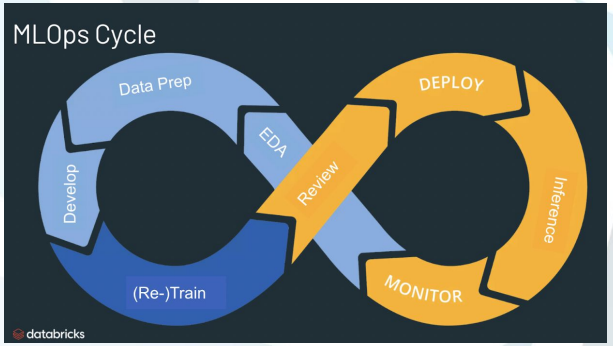
Challenges to use machine learning (ML)

- How to deploy and share ML workflows across DOE beamline facilities?



Challenges to use machine learning (ML)

- Many available ML tools require programmatic skills.
- Existing MLOps software:
  - not open-source (Weights & Biases)
  - do not have user system (MLflow)
  - do not have interactive interfaces customized for beamline users



General ML tool

Deep learning tools

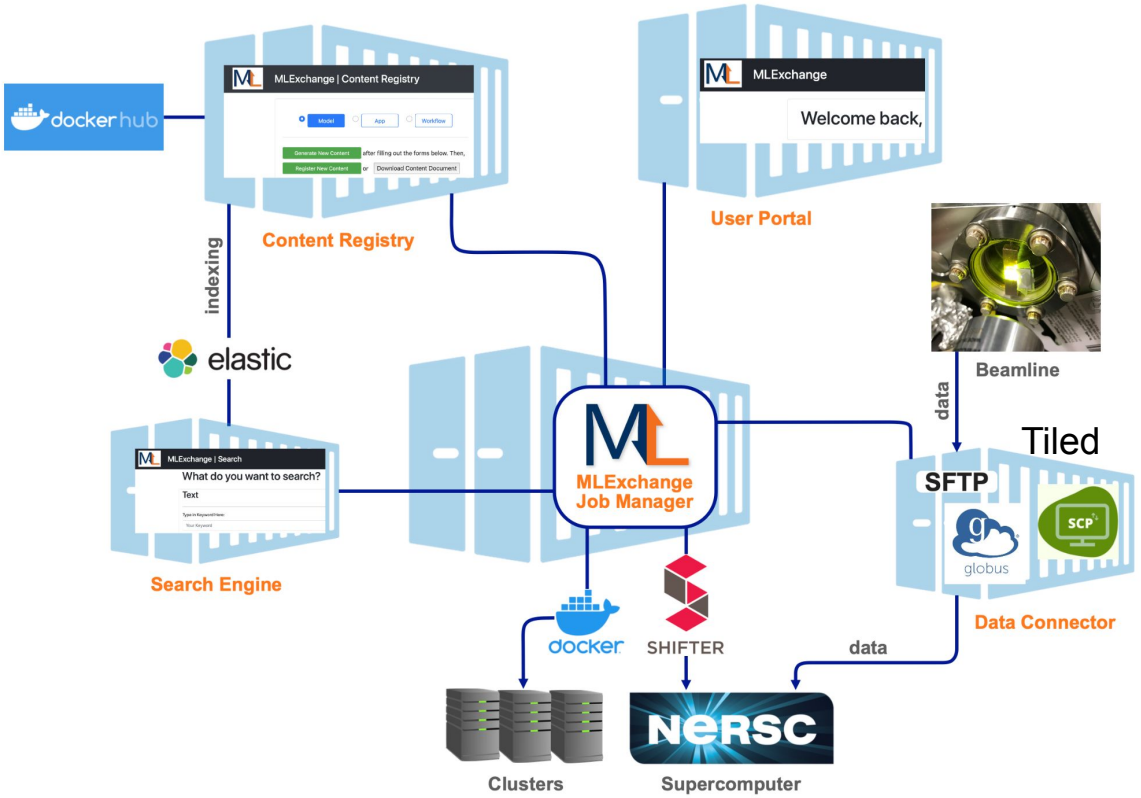
Data processing tools

Visualization tools

MLExchange platform

MLExchange architecture

- 5 major “containerized” components
- Communication through application programming interfaces (APIs)

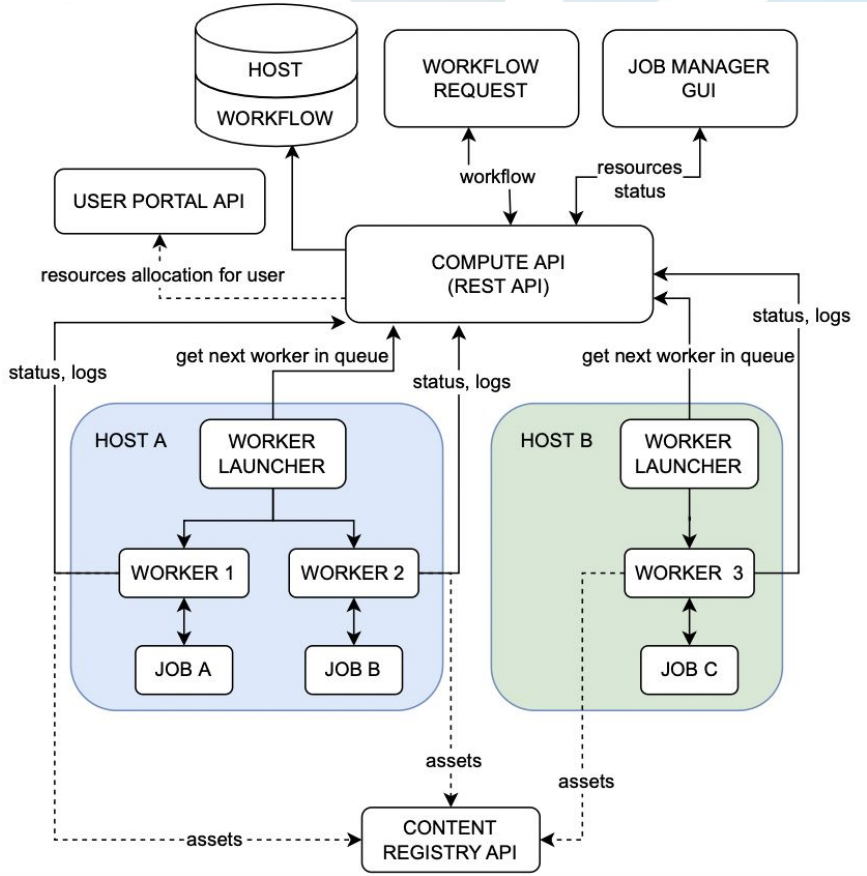


Major components

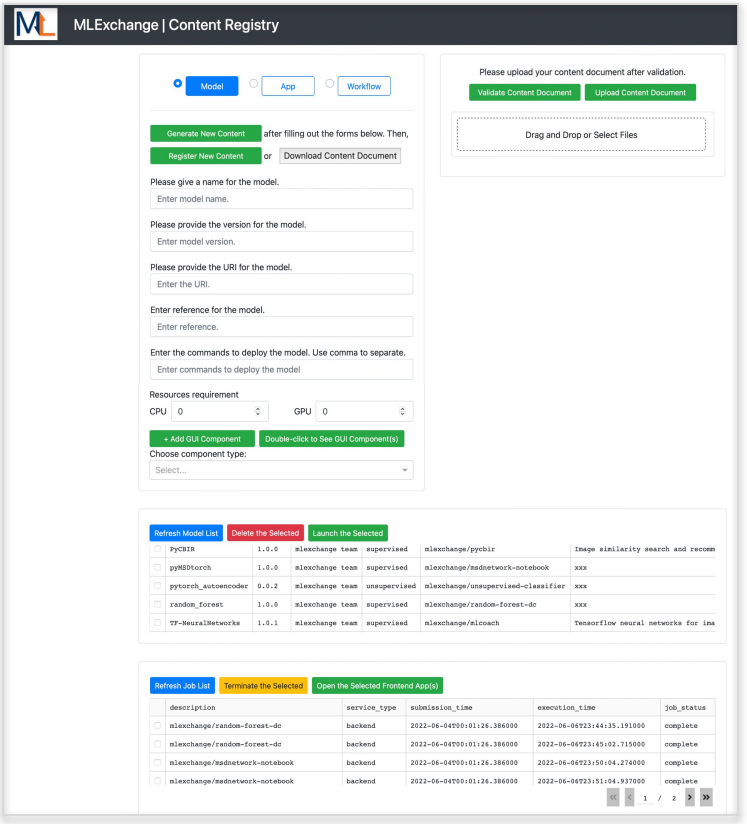
Job manager

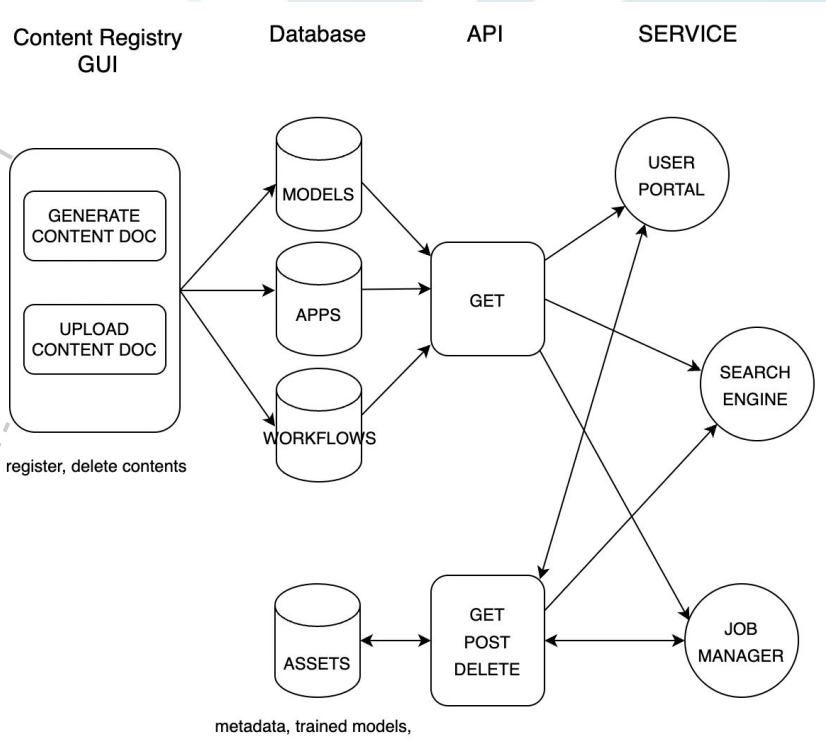
- A central job coordinator: a queue based on the timestamp and available computing resources.

Image source: Katerina Limpitsouni



Content registry





Content registry: **adaptive GUI components**

9 Dash component types

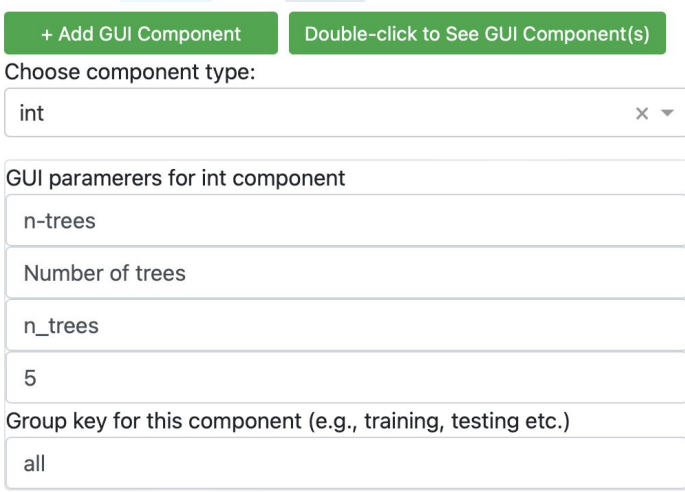
- 3 input forms (int float, str)
- Slider
- Dropdown window
- Radio items
- Boolean toggle switch
- Image
- Graph uploader

component id

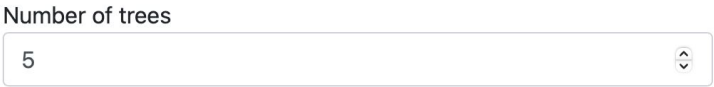
title (optional)

default value (optional)

group key (optional)

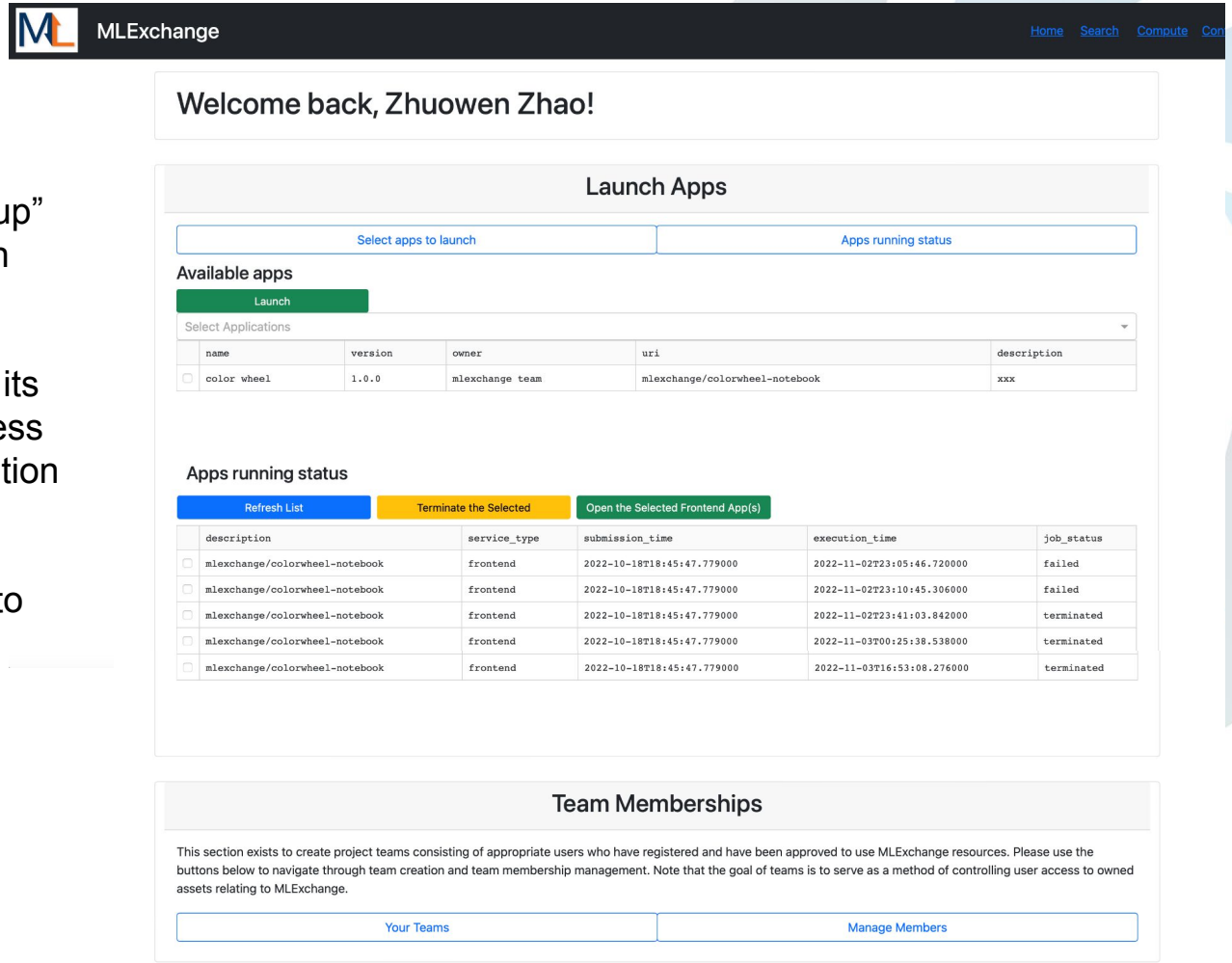


Automatically updates GUI components in frontend apps



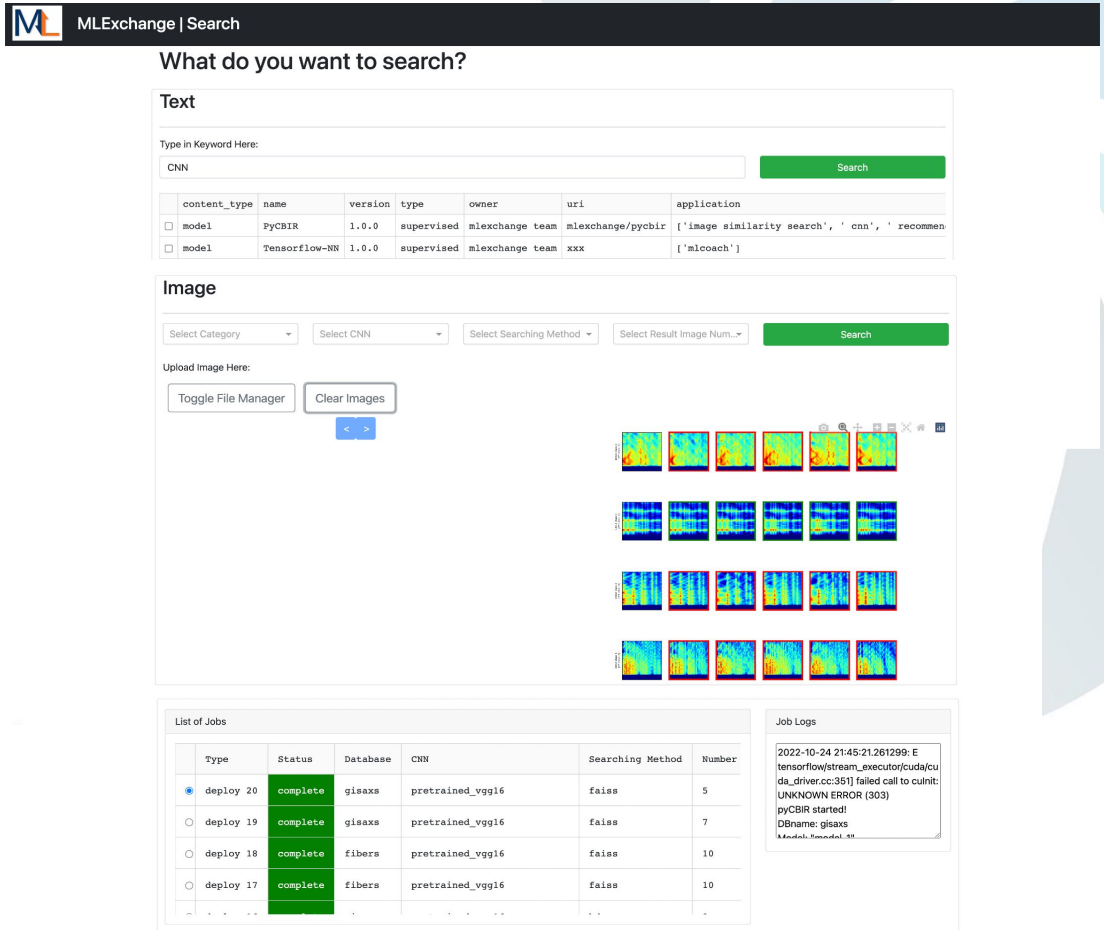
User portal

- Create “user-role-group” relationship in a graph database (neo4j)
- Insert the user id and its authority into the access token after authentication (Google)
- Launch/terminate/go to apps on behalf of individual user



Search

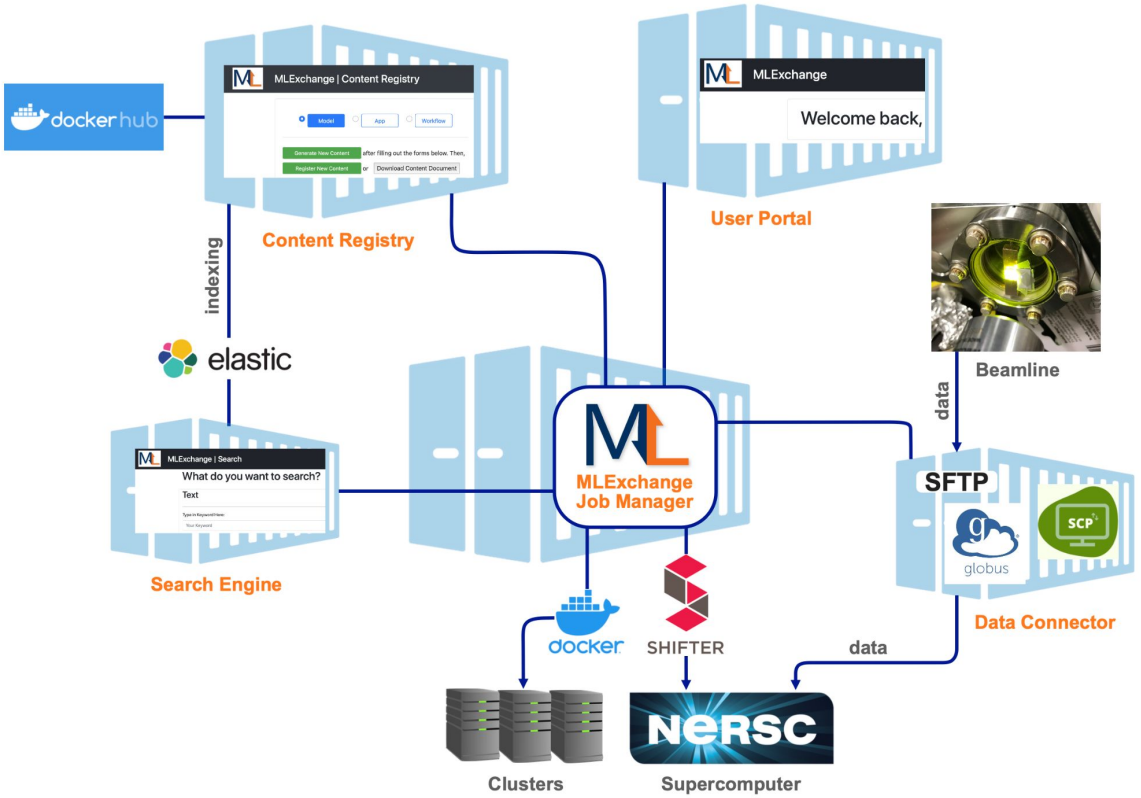
- Fast search for the contents stored in the content registry (based on user authority); it will evolve into a recommendation system.
- Search for similar images using pyCBIR[1] software



[1]pyCBIR is Python-based tool for content-based image retrieval (Araujo et.al. 2018)

Revisit MLExchange architecture

- Versatile deployment
- High modularity
- Scalability
- Easy accessibility



How-to live demo

How-to

Try MLExchange

GitHub: https://github.com/mlexchange/mlex

Docker Hub repository: https://hub.docker.com/u/mlexchange1

The official MLExchange portal: https://mlexchange.als.lbl.gov (Google authentication)

The development (beta) portal: https://mlexchangebeta.als.lbl.gov (Google authentication)

* Please note that Gmail account must be registered with both firstname and last name; o.w., you will see it keeps redirecting you back to the login page. The mlexchangebeta runs on our VM. We only have limited resources that support running max 6 (app) containers at the same time (automatically terminate in 5 mins).

How-to

Useful resources

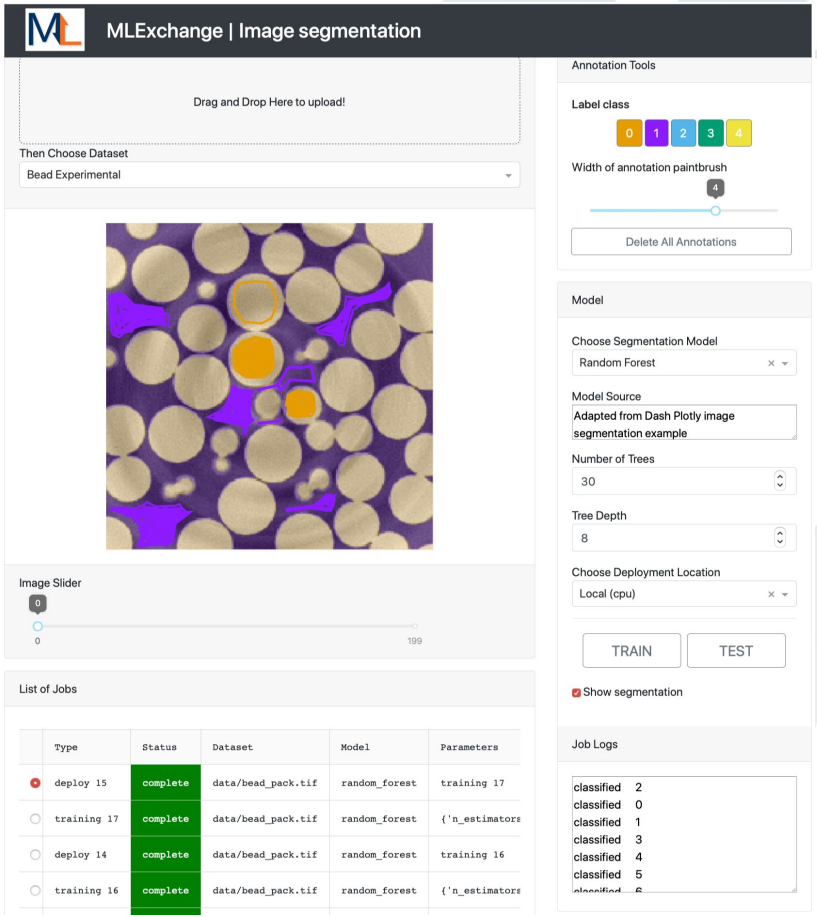
MLExchange documentation: https://docs.mlexchange.als.lbl.gov

MLExchange website: https://sites.google.com/lbl.gov/software-solutions-for-als/mlexchange

Use cases

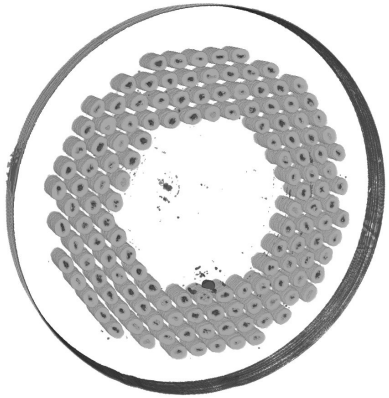
Use cases

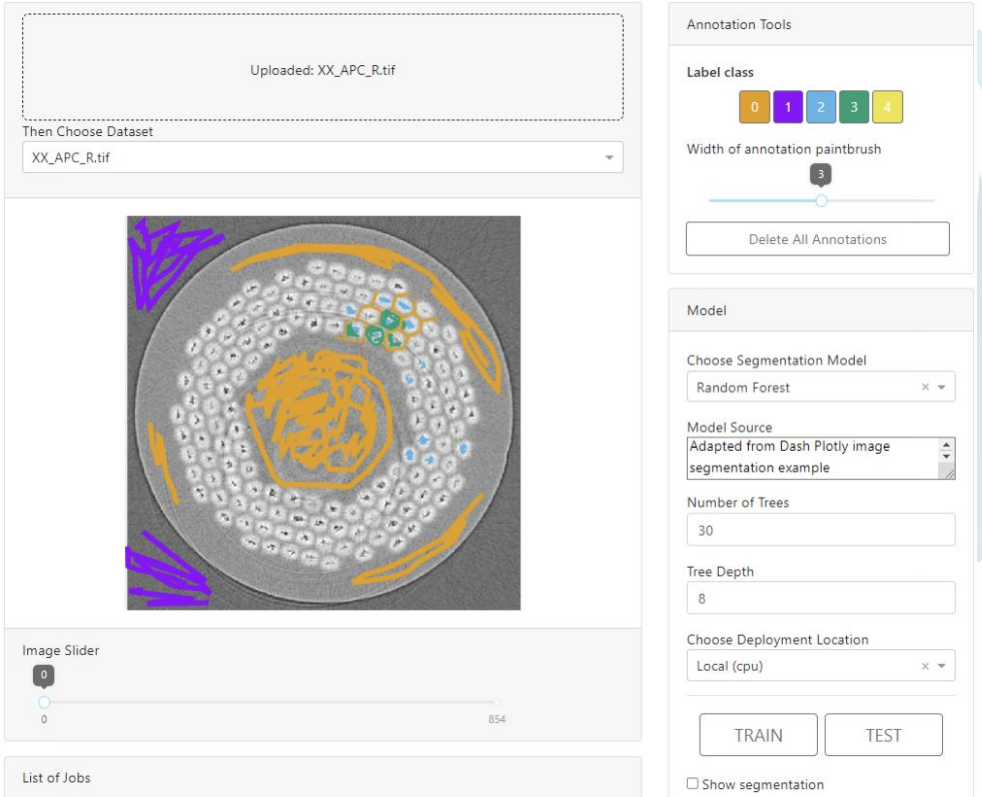
- Image segmentation: a “standalone” frontend app



APS tomography data

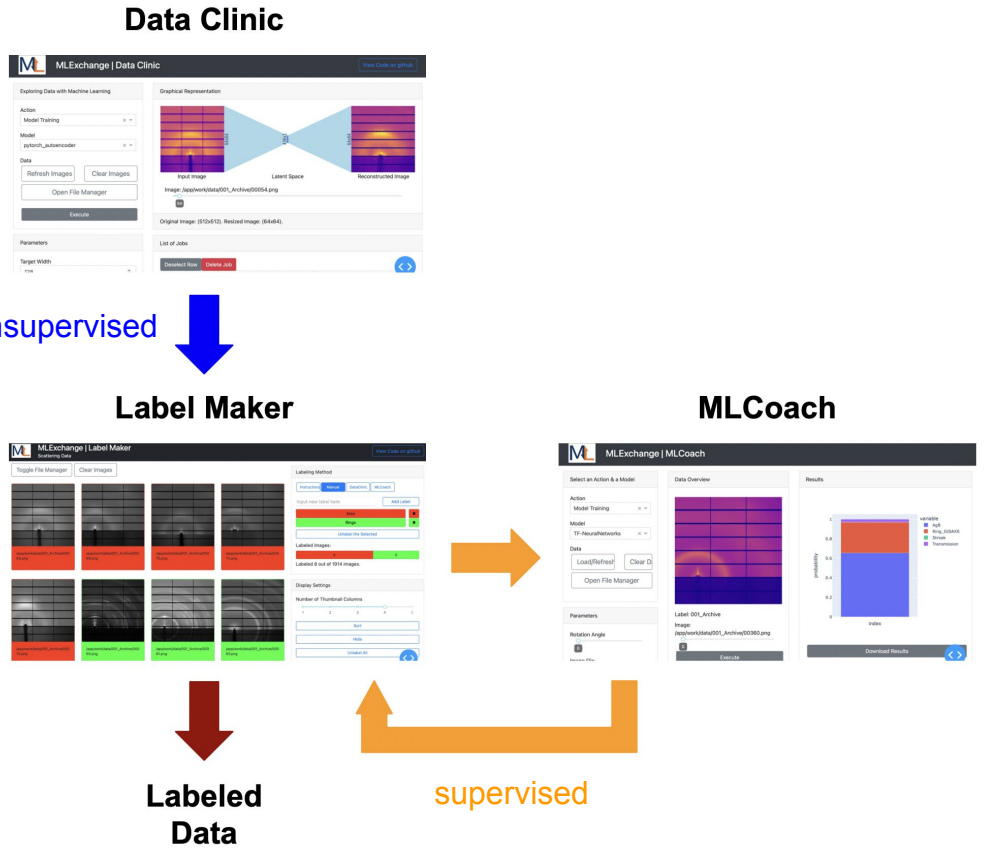
- Segmented area visualization (ImageJ 3D viewer) by Jean-Francois Croteau





Use cases

- Image labeling pipeline: a combination of multiple frontend apps



Thank you !

Q & A

Contact: zzhao2@lbl.gov