

Expanding the OME ecosystem for imaging data management

Erick Ratamero

SciPy 2024

July 11th, 2024



Erick Ratamero, PhD

Manager, Imaging Applications, Research IT
<https://orcid.org/0000-0002-7545-3675>

PhD – University of Warwick, UK

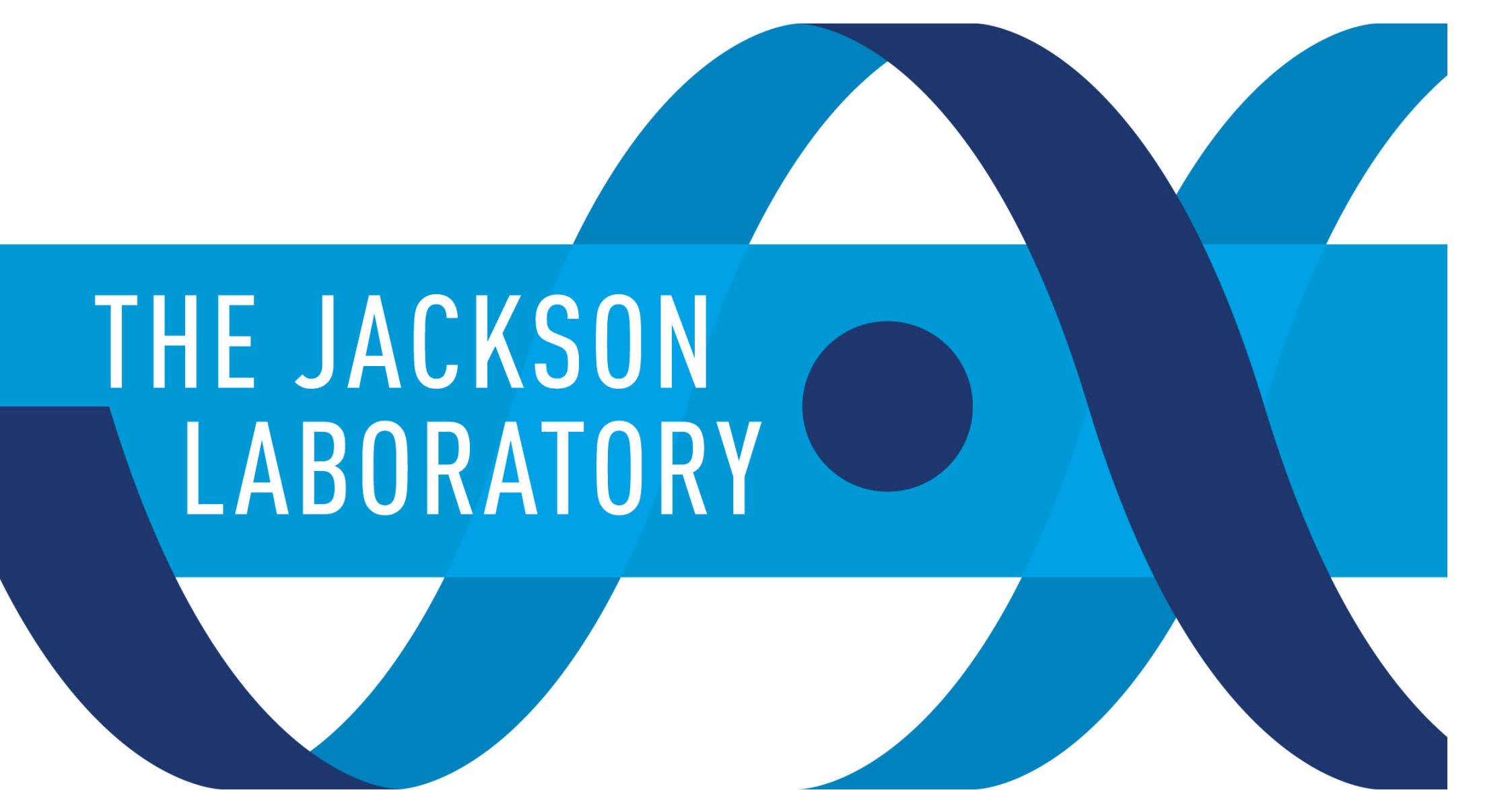
- Bacterial Division
- Mathematical Modelling

“PostDoc” – Warwick Medical School

- Responsible for data analysis and management for a microscopy facility
- ~100 users, ~20 microscopes, ~80TB of data over 2 years
- Work in multiple programming languages using multiple tools



Moved to JAX almost 5 years ago, currently managing the Imaging Applications team



THE JACKSON
LABORATORY

We distribute mice and provide services to organizations around the globe

JAX® MICE
& CLINICAL
RESEARCH
SERVICES





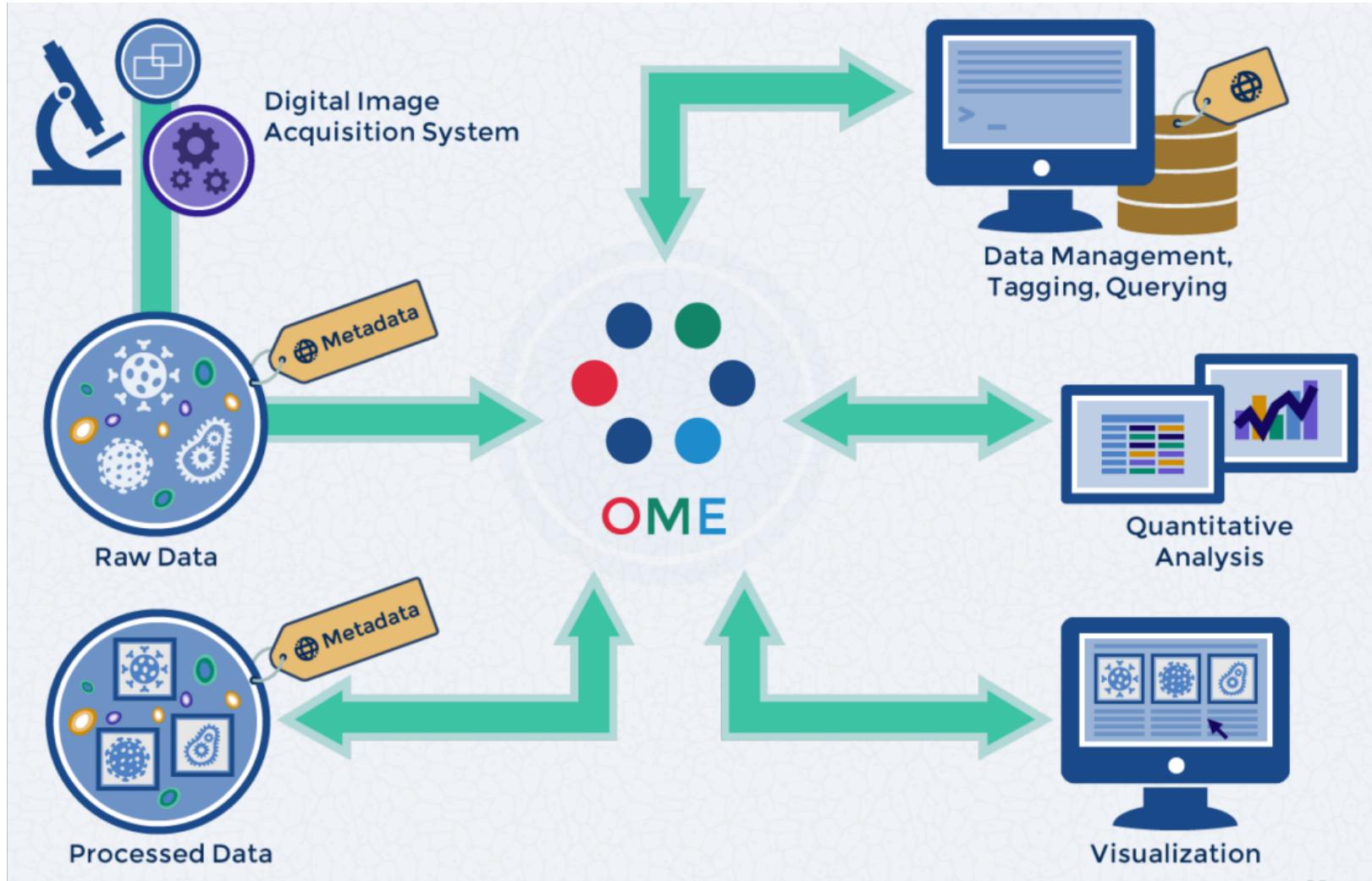
THREE LOCATIONS

BAR HARBOR, MAINE

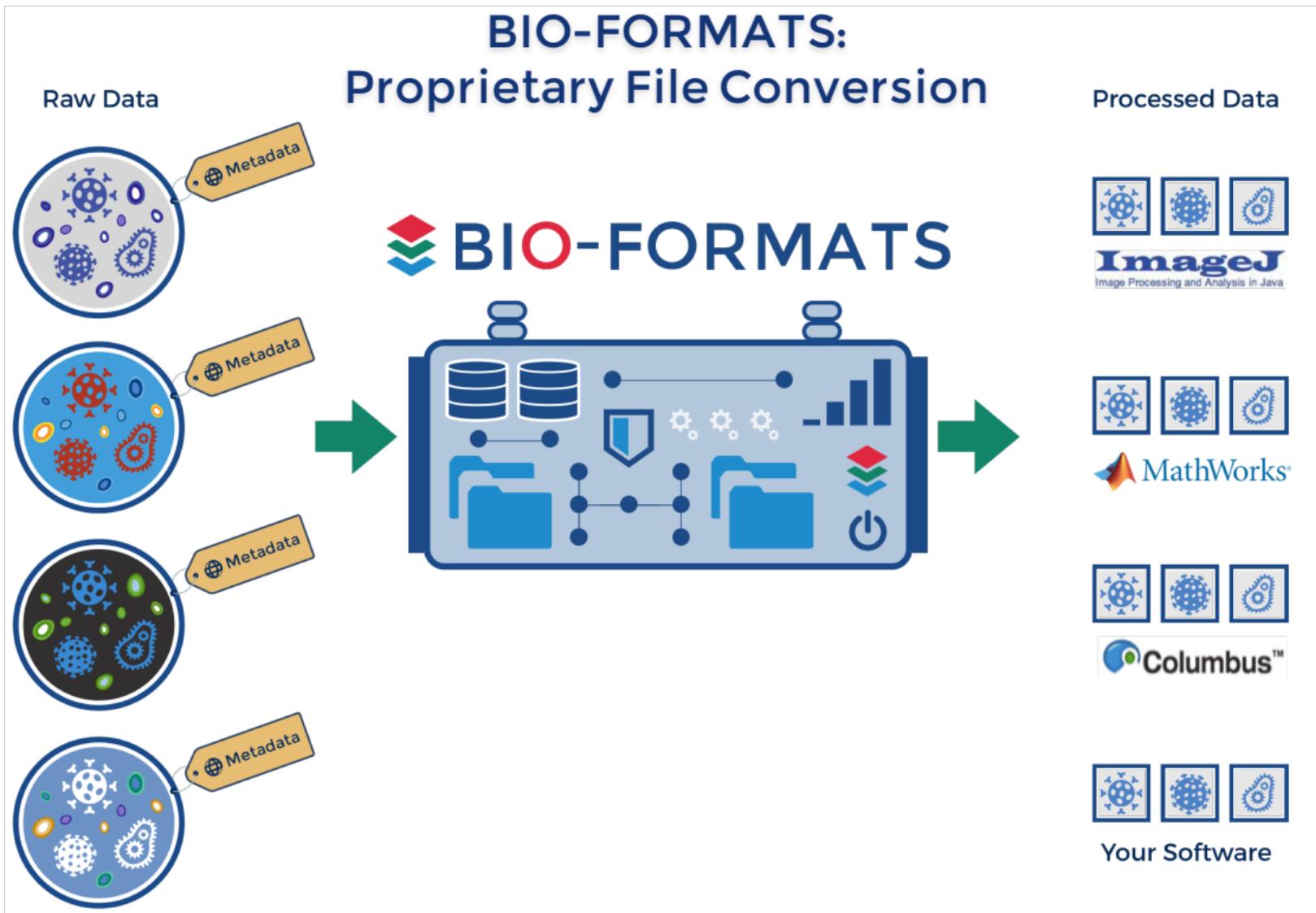
FARMINGTON, CONN.

SACRAMENTO, CALIF.

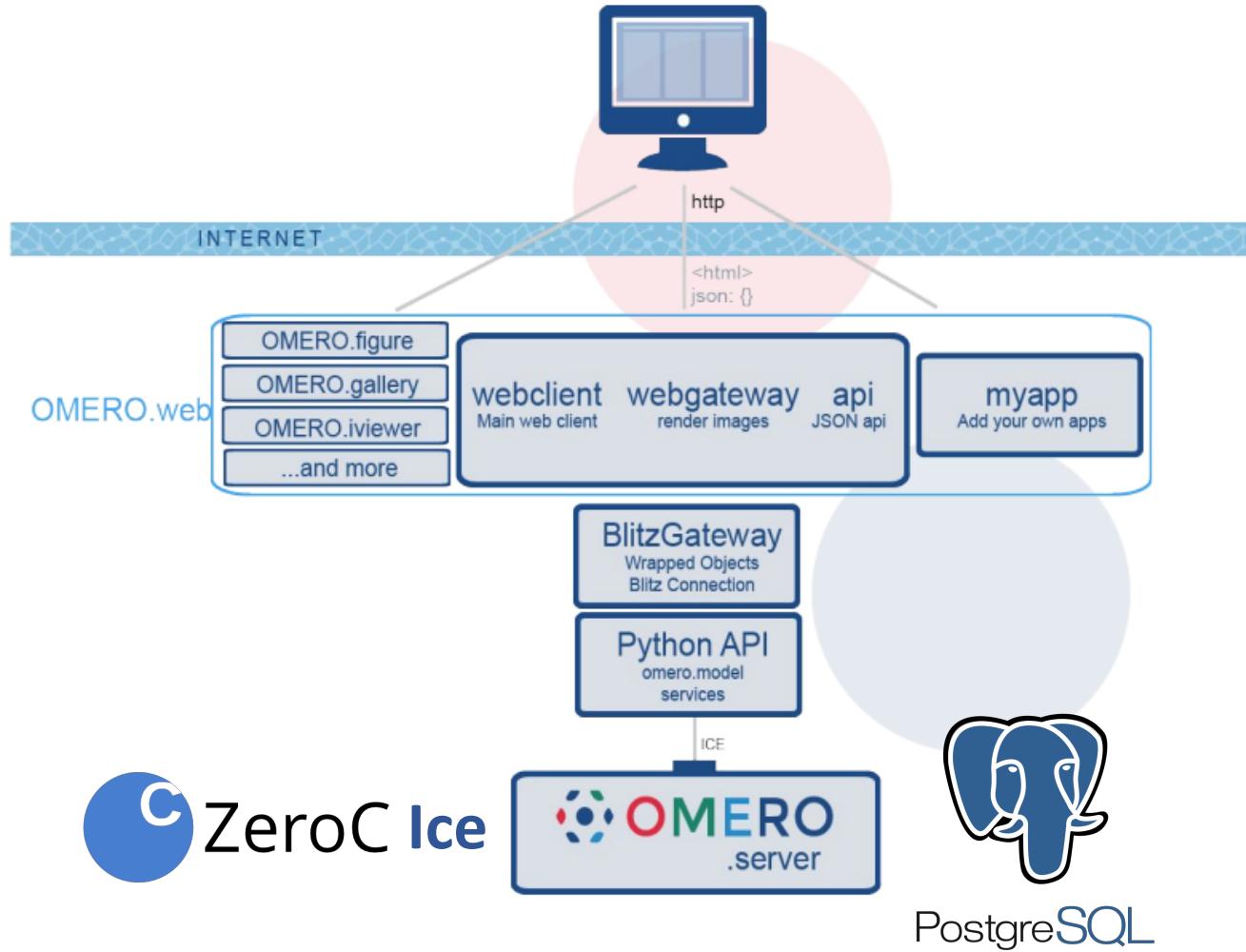
OMERO: a primer



BIO-FORMATS: Proprietary File Conversion



The OMERO stack



django
gunicorn
NGINX

OMERO.web

outreach.openmicroscopy.org/webclient/userdata/?experimenter=2

110% ome 7 dimensional

Search: trainer-1 trainer-1

General Acquisition Preview

Full viewer

Cep120_gtub_001_SIR_PRJ.dv

Image ID: 9594
Owner: trainer-1 trainer-1

Image Details

Add Description

Import Date: 2018-05-17 15:59:15
Dimensions (XY): 256 x 256
Pixels Type: float
Pixels Size (XYZ) (μm): 0.04 x 0.04 x 0.13
Z-sections/Timepoints: 1 x 1
Channels: CEP120, TUBG1-N
ROI Count: 5

Tags 0

Key-Value Pairs 7

Add Key Add Value

omero.batch_roi.export.map_ann

Added by: trainer-1 trainer-1

filter_shapes_by_channel	1
shape_count	6
min_intensity	21.5446281433
max_intensity	96540.1933125
mean_intensity	18782.1641302
min_points	12
max_points	110
mean_points	50

Tables

Attachments 1

Comments 0

Ratings 0

Others 0

Explore Tags Shares

Add filter

Lab1 trainer-1 trainer-1

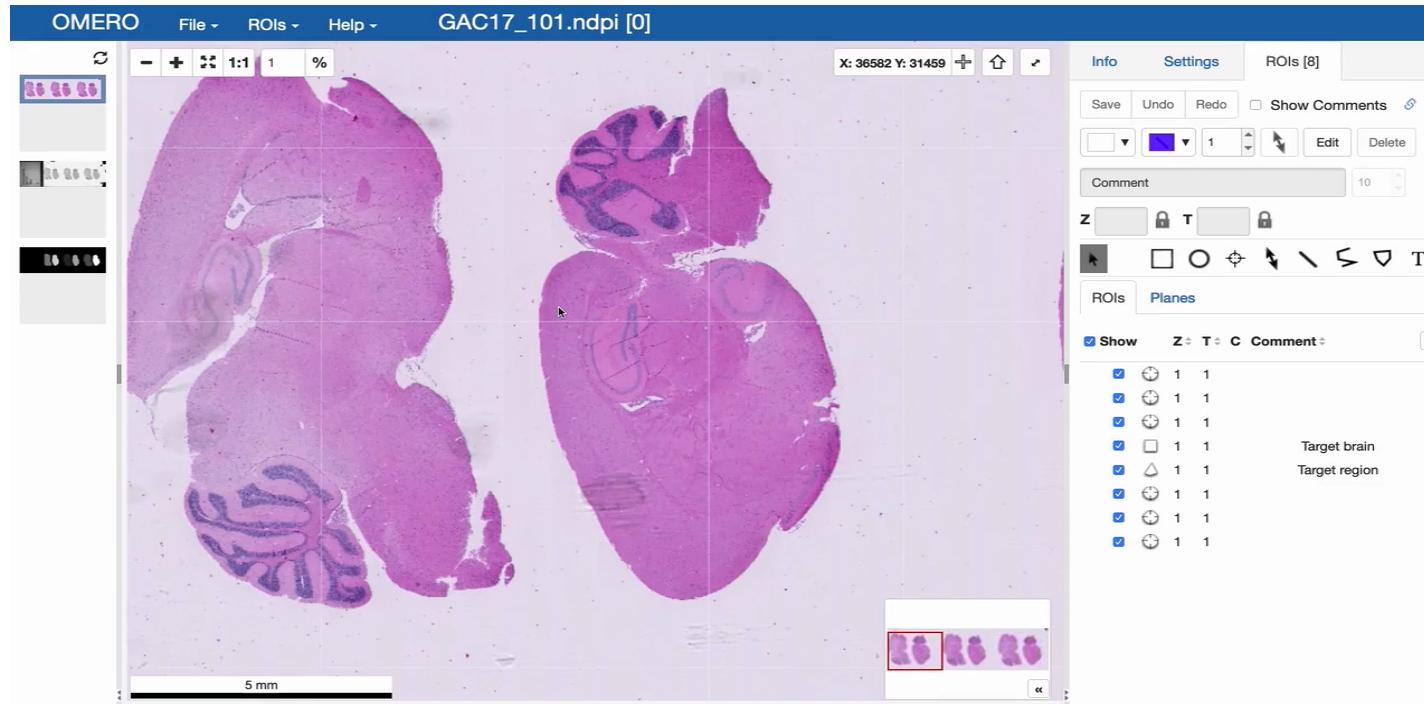
Explore Tags Shares

id0021 10

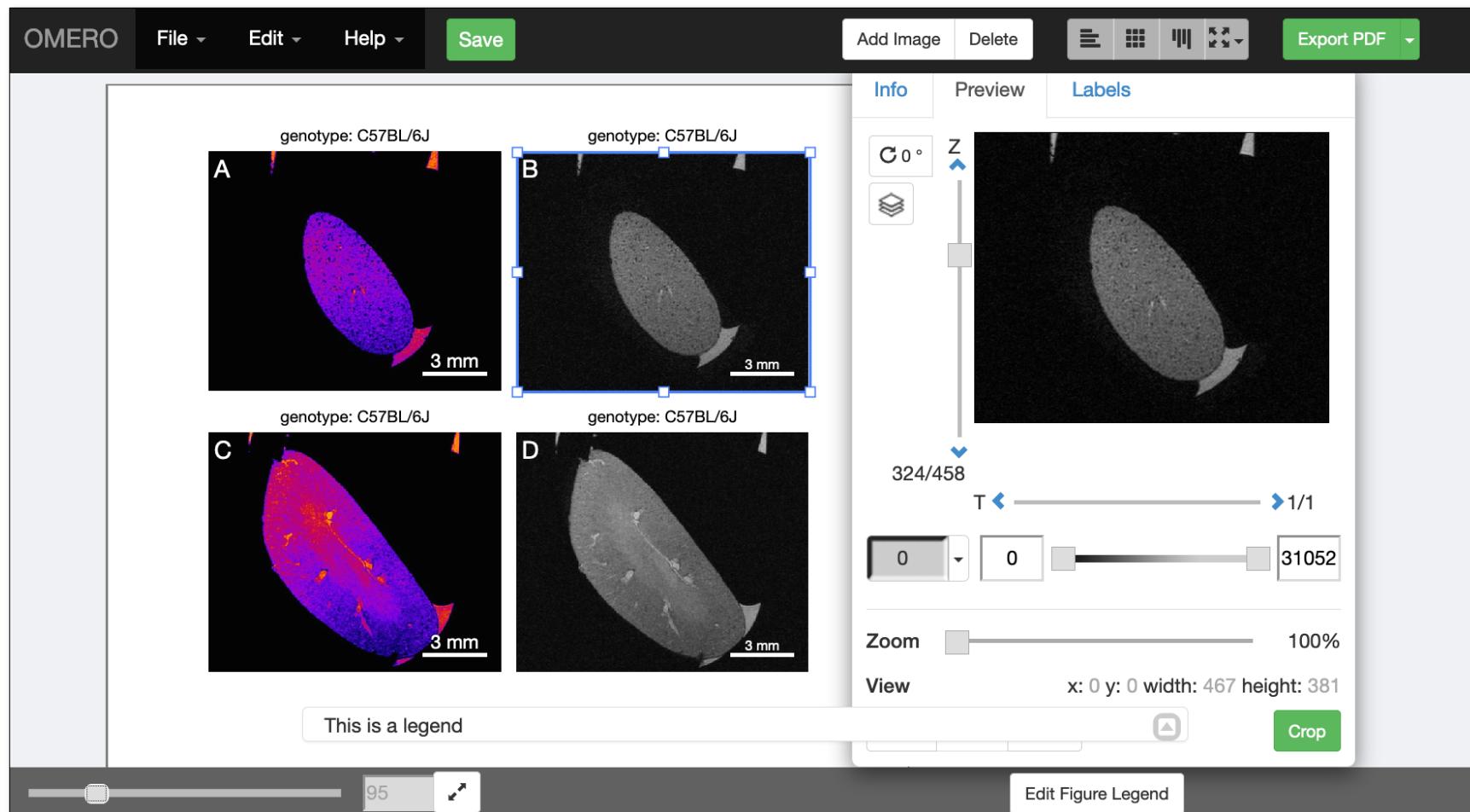
- CKS5RAP2-C 33
- CENT2 22
- CEP120/20111106 38
 - Cep120_gtub_001_SIR_PRJ.dv
 - Cep120_gtub_002_SIR_PRJ.dv
 - Cep120_gtub_007_SIR_PRJ.dv
 - Cep120_gtub_008_SIR_PRJ.dv
 - Cep120_gtub_009_SIR_PRJ.dv
 - Cep120_gtub_010_SIR_PRJ.dv
 - Cep120_gtub_015_SIR_PRJ.dv
 - Cep120_gtub_016_SIR_PRJ.dv
 - Cep120_gtub_020_SIR_PRJ.dv
 - Cep120_gtub_023_SIR_PRJ.dv
 - Cep120_gtub_024_SIR_PRJ.dv
 - Cep120_gtub_025_SIR_PRJ.dv
 - Cep120_gtub_026_SIR_PRJ.dv
 - Cep120_gtub_027_SIR_PRJ.dv
 - Cep120_gtub_028_SIR_PRJ.dv
 - Cep120_gtub_030_SIR_PRJ.dv
 - JG_Cep120_gtub_001_SIR_PRJ.dv
 - JG_Cep120_gtub_002_SIR_PRJ.dv
 - JG_Cep120_gtub_003_SIR_PRJ.dv
 - JG_Cep120_gtub_005_SIR_PRJ.dv
 - JG_Cep120_gtub_006_SIR_PRJ.dv
 - JG_Cep120_gtub_007_SIR_PRJ.dv
 - JG_Cep120_gtub_008_SIR_PRJ.dv
 - JG_Cep120_gtub_009_SIR_PRJ.dv
 - JG_Cep120_gtub_010_SIR_PRJ.dv
 - JG_Cep120_gtub_011_SIR_PRJ.dv
 - JG_Cep120_gtub_012_SIR_PRJ.dv
 - JG_Cep120_gtub_014_SIR_PRJ.dv
 - JG_Cep120_gtub_015_SIR_PRJ.dv
 - JG_Cep120_gtub_018_SIR_PRJ.dv
 - JG_Cep120_gtub_020_SIR_PRJ.dv
 - JG_Cep120_gtub_023_SIR_PRJ.dv
 - JG_Cep120_gtub_024_SIR_PRJ.dv
 - JG_Cep120_gtub_026_SIR_PRJ.dv
 - JG_Cep120_gtub_027_SIR_PRJ.dv

Zoom: [progress bar]

OMERO annotations and multiscale image exploration



OMERO.figure



How we adapt and extend OMERO

Develop in Python, test with pytest



Ephemeral test environment; extend and
integrate official releases using Docker
Compose

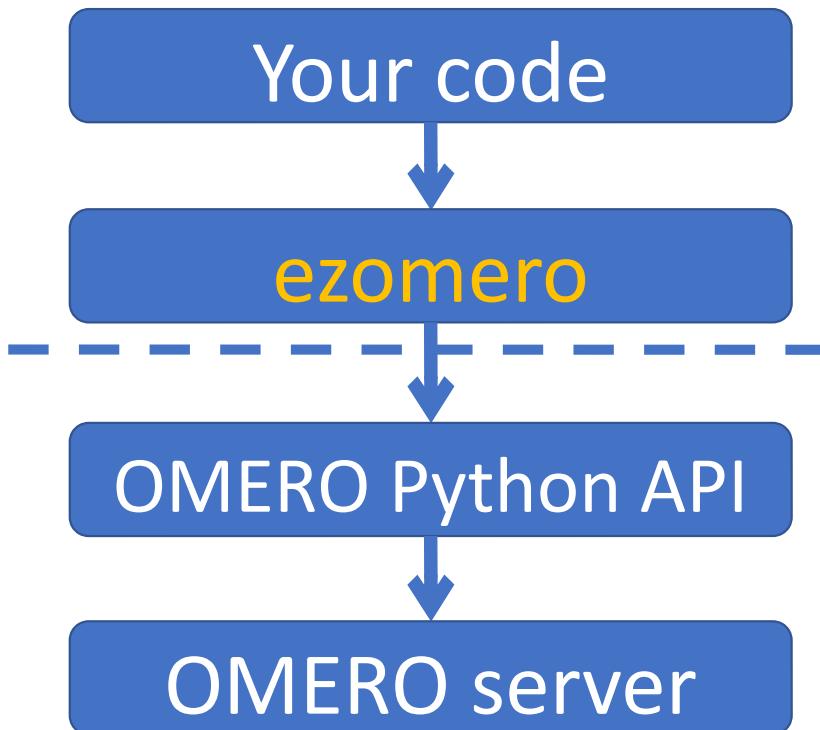


Continuous integration with
GitHub + Github Actions



Two open-source libraries
extending OMERO capabilities

Contributing to the community: ezomero



```
import omero
from omero.rtypes import rstring
conn = BlitzGateway(USERNAME, PASSWORD, host=HOST, port=PORT)
conn.connect()
dataset_obj = omero.model.DatasetI()
dataset_obj.setName(rstring("New Dataset"))
dataset_obj = conn.getUpdateService().saveAndReturnObject(dataset_obj, conn.SERVICE_OPTS)
dataset_id = dataset_obj.getId().getValue()
print("New dataset, Id:", dataset_id)
link = omero.model.ProjectDatasetLinkI()
link.setChild(omero.model.DatasetI(dataset_obj.id.val, False))
link.setParent(omero.model.ProjectI(projectId, False))
conn.getUpdateService().saveObject(link)
```

vs.

```
import ezomero
conn = ezomero.connect()
dataset_id = ezomero.post_dataset(conn, "New Dataset", project_id=projectId)
```

<https://github.com/TheJacksonLaboratory/ezomero>

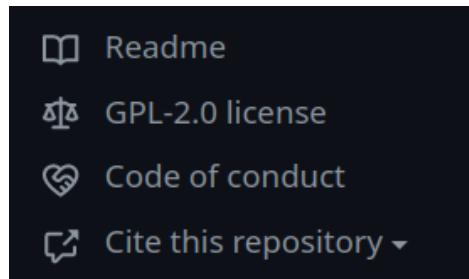
ezomero: design principles

- Input/output types should be built-in or Scientific Python
- Do not duplicate OMERO API
- REST-like interface (POST, GET, PUT)
- Avoid new dependencies, abstract nuances
- and keep it simple!

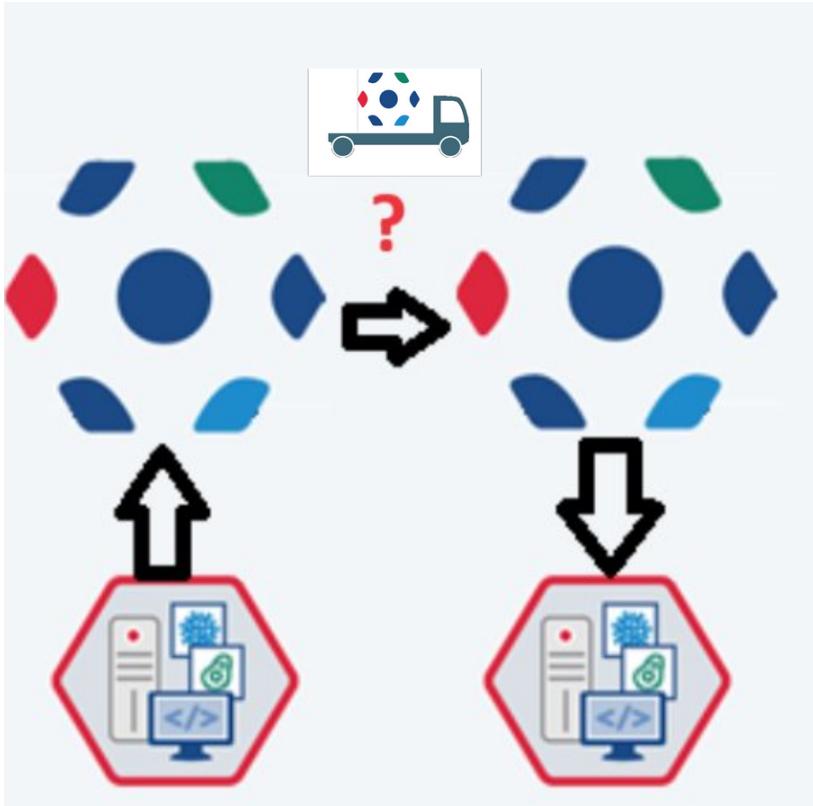
ezomero: community development

Bug report
Create a report to help us improve [Get started](#)

Feature request
Suggest an idea for this project [Get started](#)



Contributing to the community: omero-cli-transfer



```
omero transfer pack Image:123 transfer_pack.zip  
omero transfer pack Dataset:1111 /home/user/new_folder/new_pack.zip  
omero transfer pack 999 zipfile.zip # equivalent to Project:999
```

```
omero transfer unpack transfer_pack.zip  
omero transfer unpack --output /home/user/optional_folder --ln_s
```

omero-cli-transfer: the problem

Key-Value Pairs 2	
jax.org/omeroutils/user_submitted/v0	
Added by: Data Importer	
species	Mus Musculus
genotype	P/J
antibodies/stains	PAS
tissue description	Kidney
Sex	female
Diet	lithium

- Only exists inside OMERO
- Would not move over with images

Serializing OMERO data

ome-types

A pure-python implementation of the OME data model

Maintainers

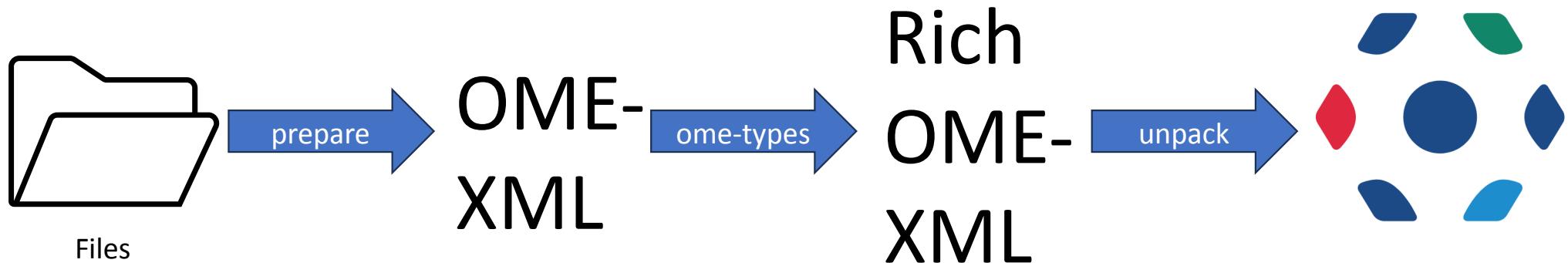
`ome_types` provides a set of python dataclasses and utility functions for parsing the [OME-XML format](#) into fully-typed python objects for interactive or programmatic access in python. It can also take these python objects and output them into valid OME-XML. `ome_types` is a **pure python** library and does not require a Java virtual machine.



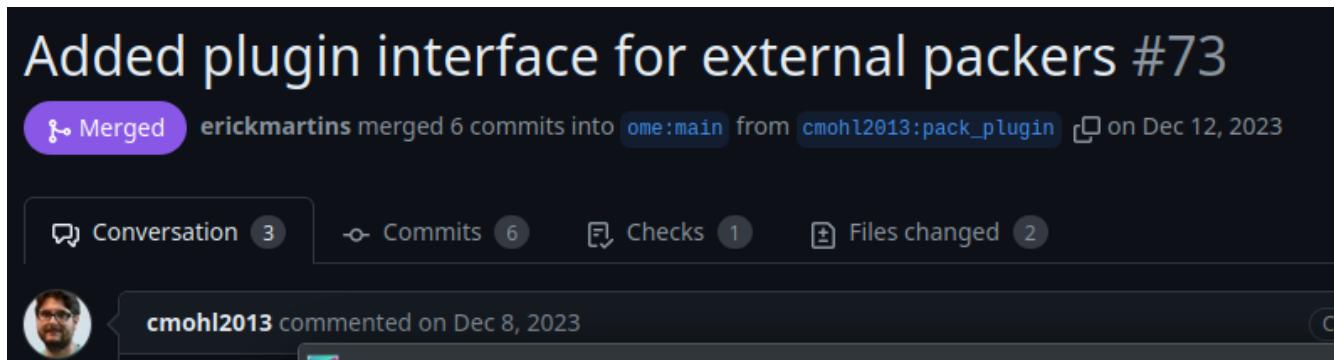
[talley](#)

```
In [19]: from ome_types import to_xml  
  
In [20]: print(to_xml(ome))  
<OME ...>  
        <Plate ColumnNamingConvention="letter" Columns="12" ID="Plate:1" ...>
```

Other use cases: rich import



Other use cases: plugins and export formats



OMERO-ARC

A plugin for [omero-cli-transfer](#) to export imaging projects from the OMERO database to ARC repositories.

Bioimage Archive submission contents

- Folder structure in the generated zip/tar follows project/dataset structure rather than original ManagedRepository folder structure, and instead of a `transfer.xml` file, a `submission.tsv` file is generated.

Summary and conclusions

- OMERO facilitates imaging data management at scale
- Python and command-line ways of interfacing with an OMERO server already exist; they are also easily extensible
- We have developed two open-source libraries in Python that simplify and extend functionality in OMERO

Acknowledgments

JAX Imaging Applications

- Fernando Cervantes
- Peter Sobolewski
- Kiya Govek

JAX Research IT

- Dave McKenzie
- Jason Macklin
- Vishal Thummala
- Dave Mellert

The Open-source community

- OpenCV
- Scikit-image
- ImageJ/Fiji
- OMERO & BioFormats
- Zarr-python
- N5-lib

OME & Friends

- Josh Moore
- Will Moore
- Seb Besson

Come talk to me if you are interested in...

- Anything imaging, really!
- But ESPECIALLY...
 - imaging data management
 - how we get researchers to add rich metadata!
 - larger-than-memory imaging analysis/visualization
 - (in particular if you are in the Zarr/OME-Zarr world!)
 - Weird careers that are research-but-not-quite