

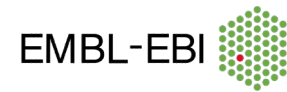


OMERO: source for image analysis data on the cloud

NEUBIAS 2023

OME Team

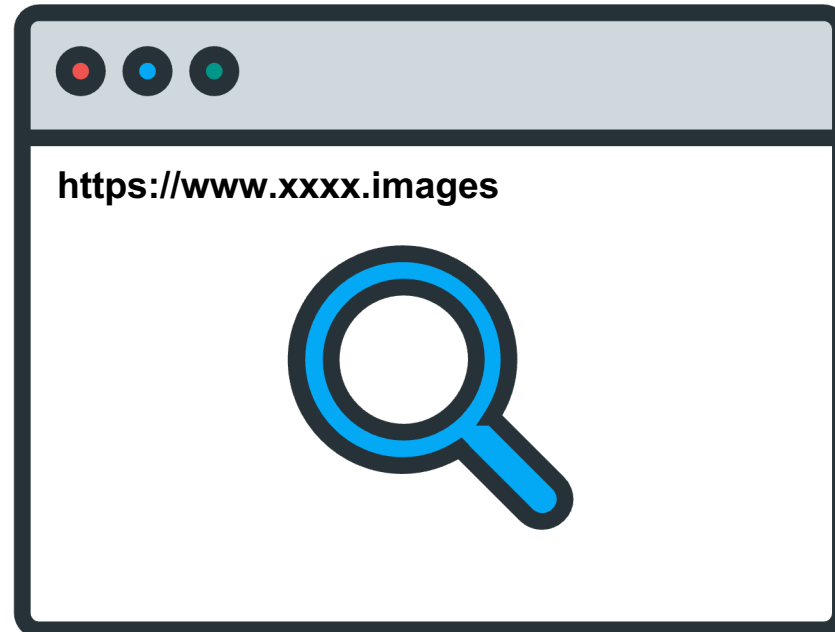
HORIZON 2020



Motivation

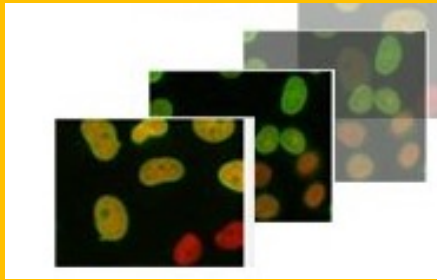
Have you ever analyzed large, heterogeneous image data ?

Have you ever wanted to find existing data to reuse ?



Analysis landscape - local

Environment  ANACONDA.



Data

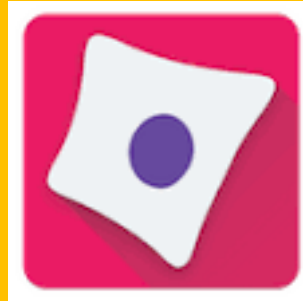
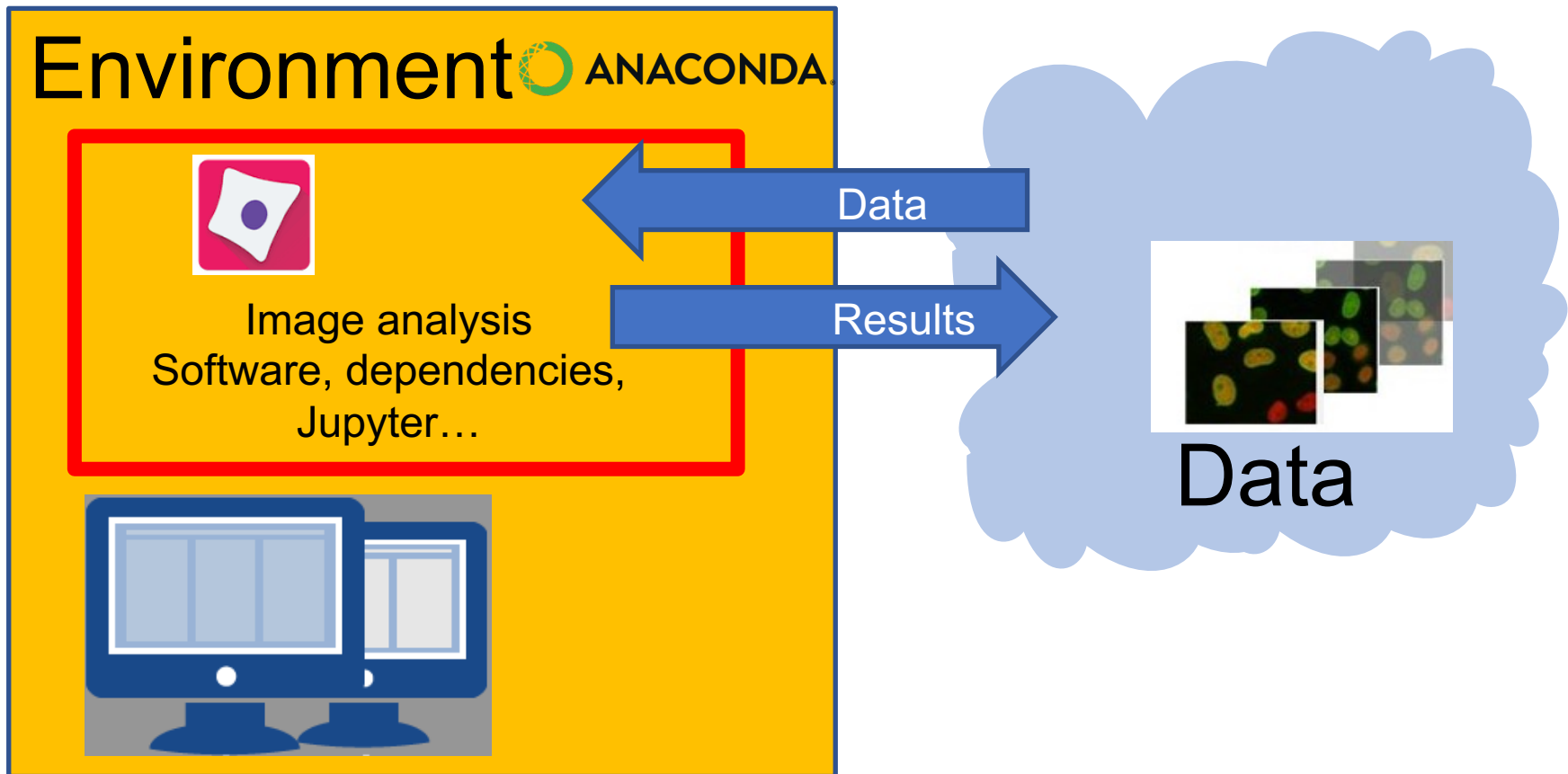


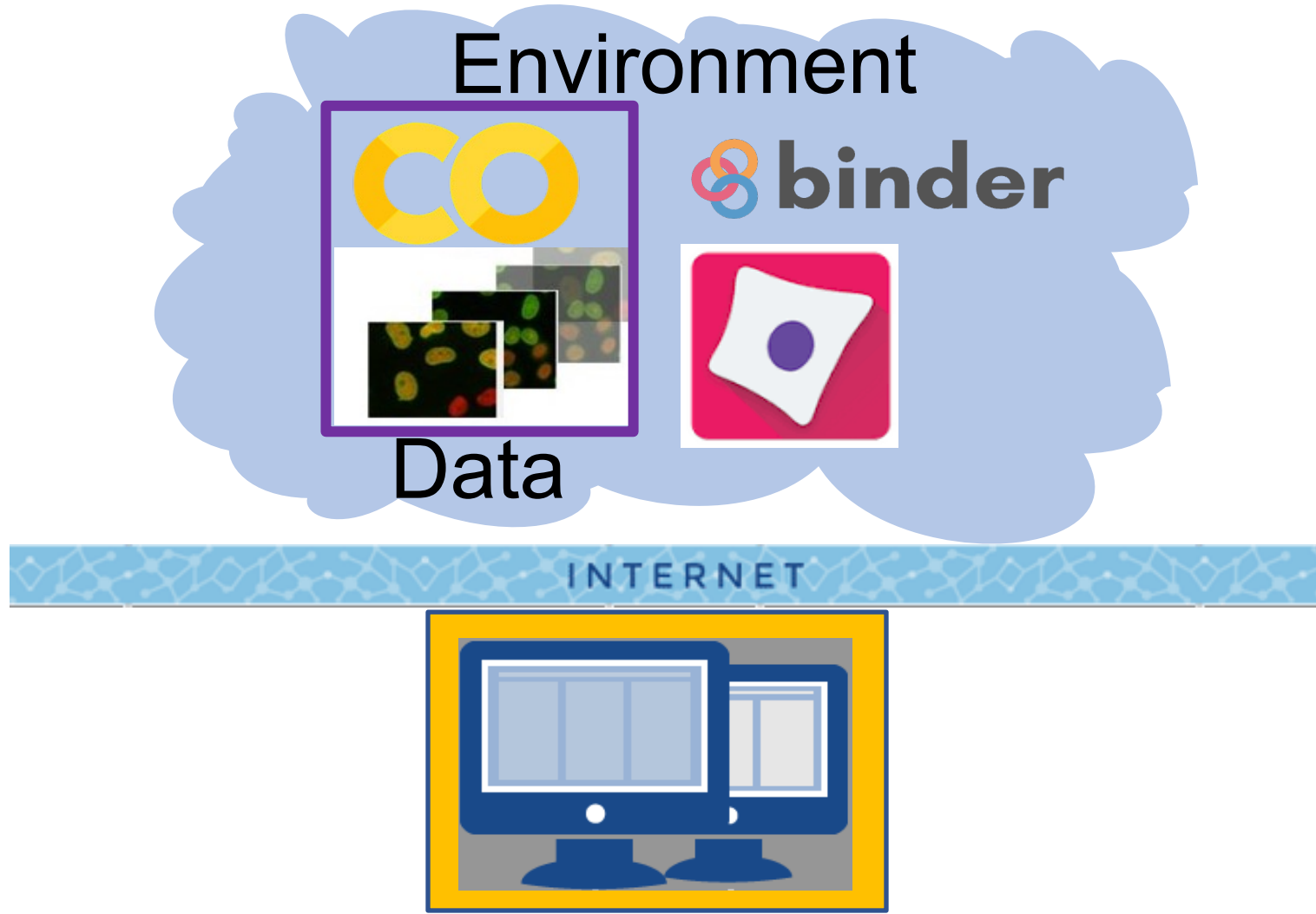
Image analysis
Software,
dependencies,
Jupyter...



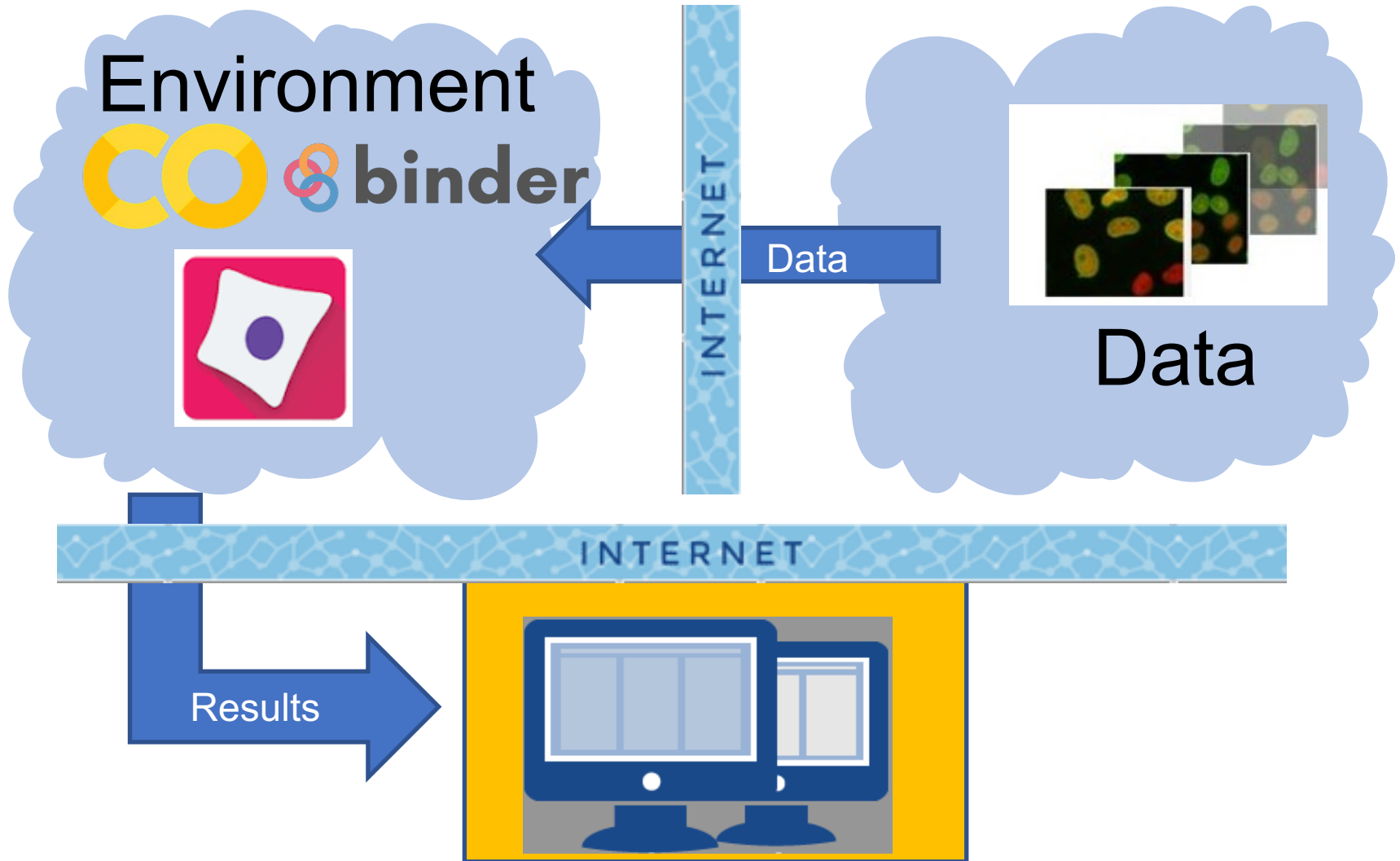
Analysis landscape – remote data



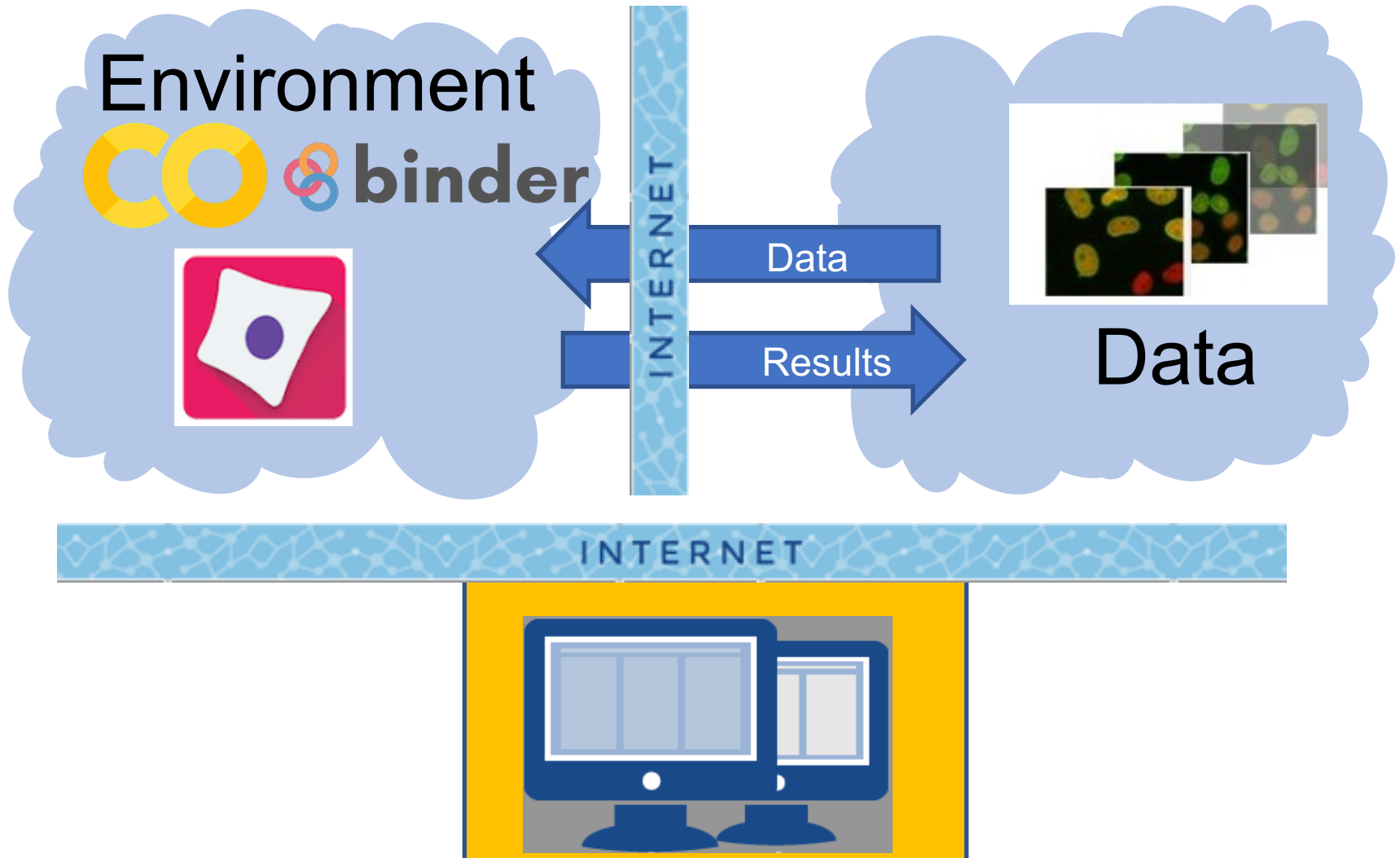
Analysis landscape – remote



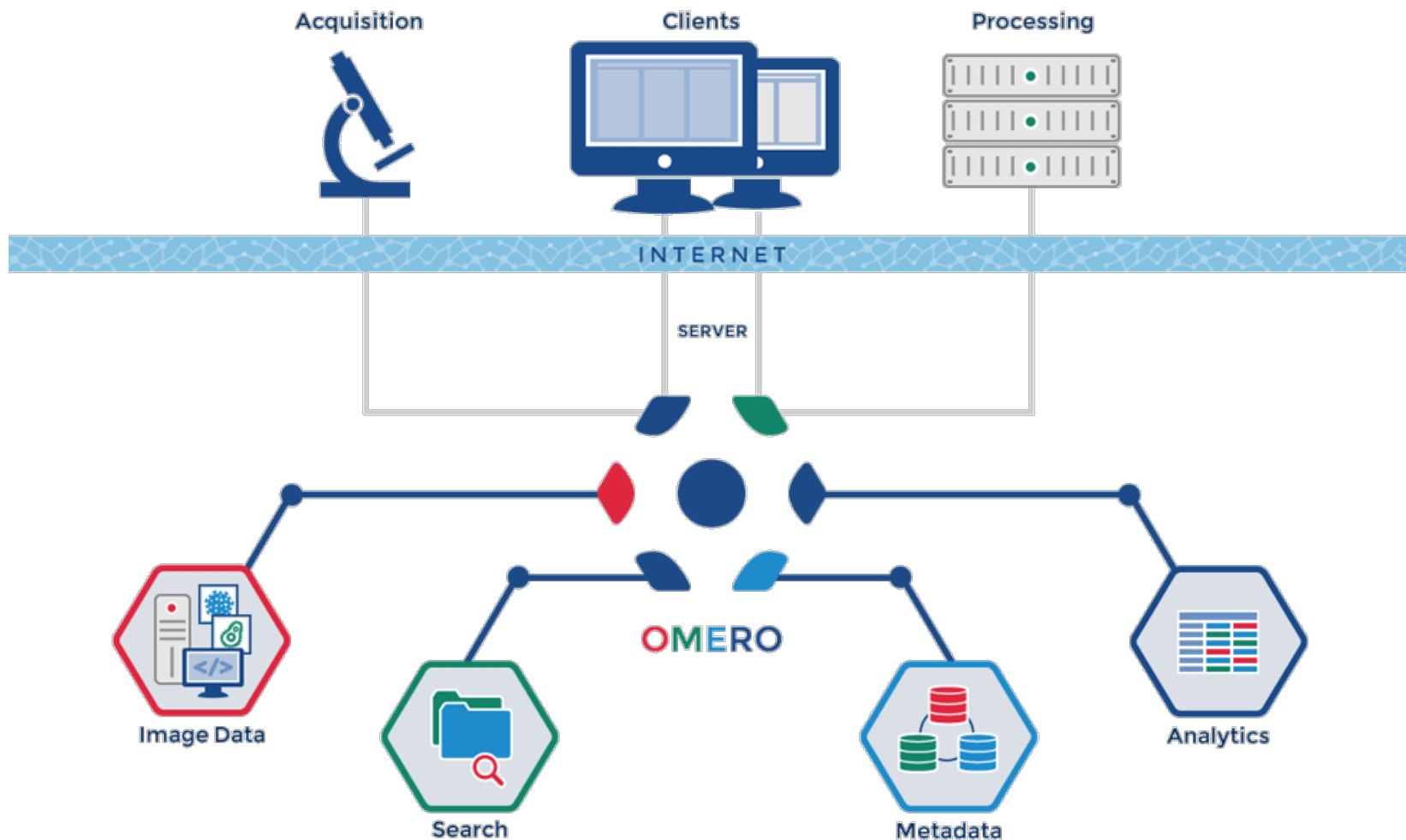
Analysis landscape – remote



Analysis landscape – remote

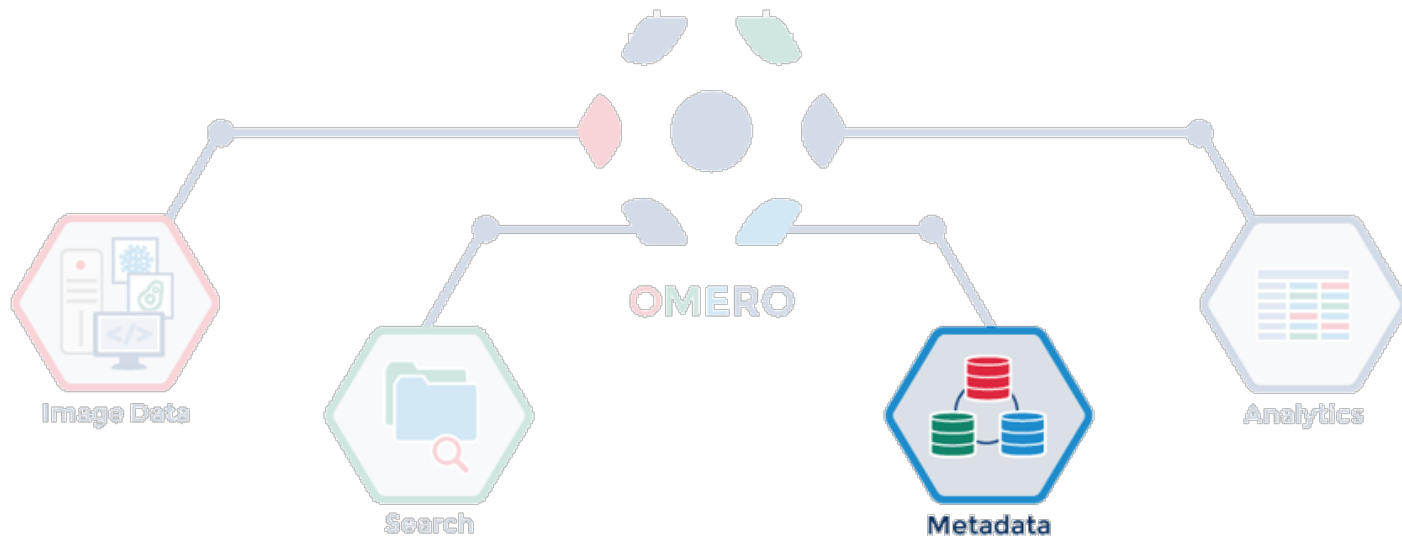


OMERO is a server with clients



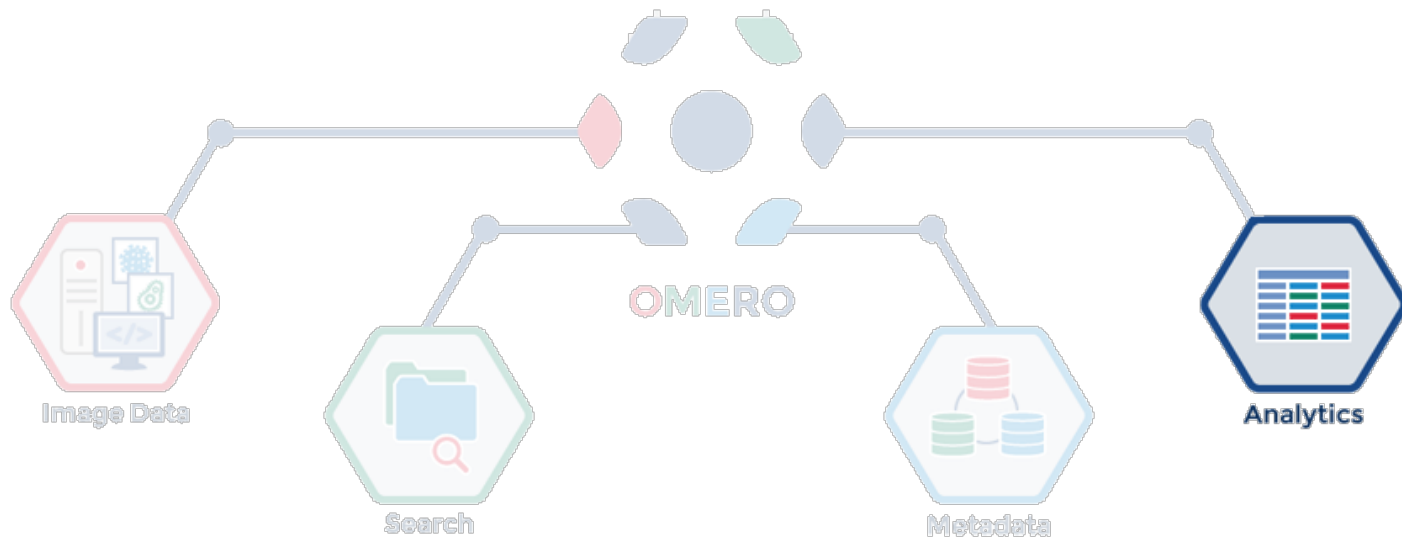
Metadata

- Stored in relational database
- Acquisition metadata
- User-added annotations
- ROIs / Segmentations



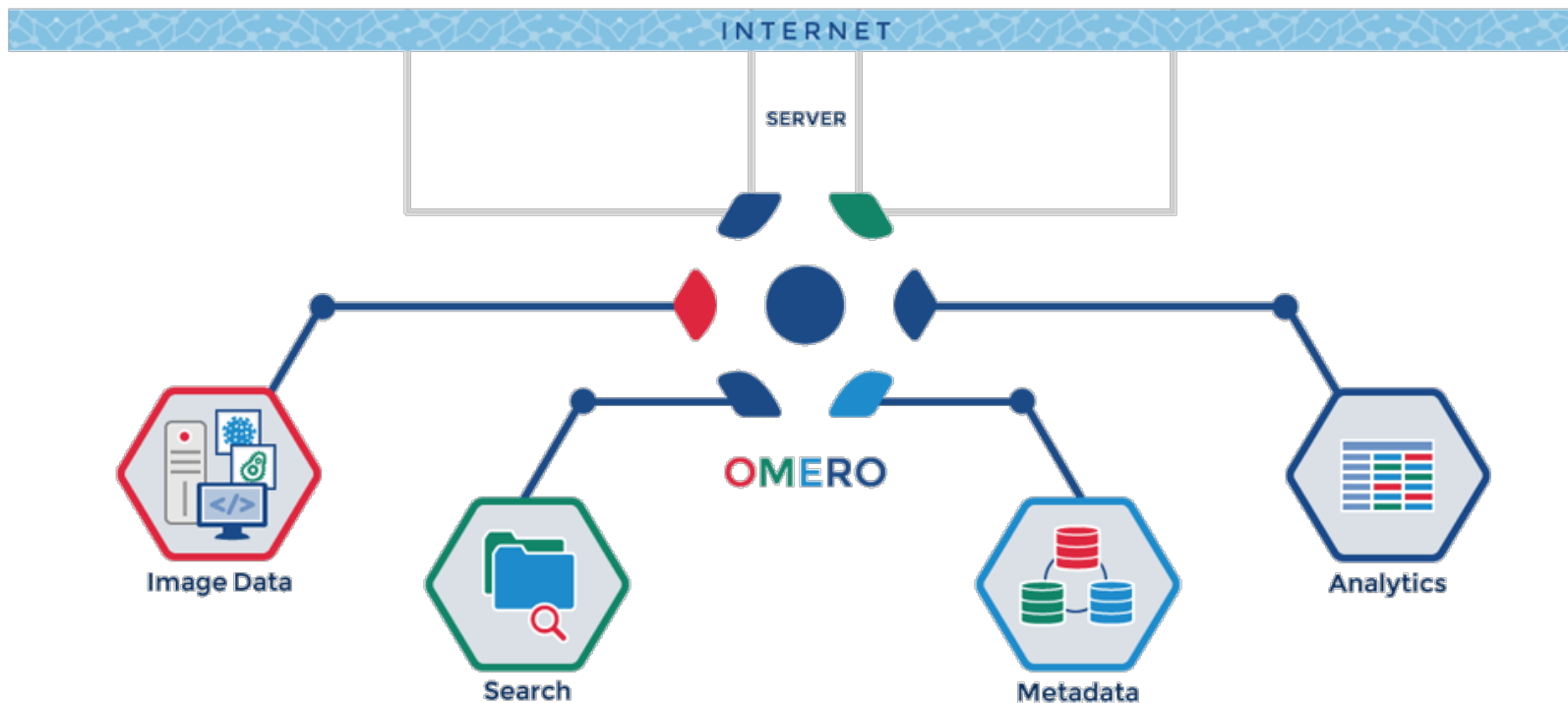
OMERO.tables

- HDF5 tables for analysis results
- Flexible columns



OMERO API

Python | Java | Matlab | R

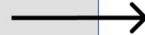
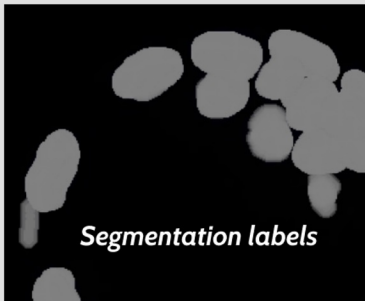
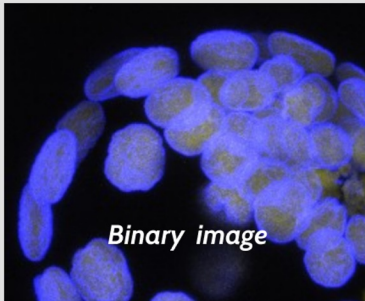


Integration



Analysis using OMERO API

Data stored in IDR
as .tiff with OMERO ROIs



Python-based analysis
environment

Notebook with StarDist

Load Image with labels from IDR, analyze using StarDist and compare results

The notebook shows how to load an IDR image with labels.

The image is referenced in the paper "NesSys: a novel method for accurate nuclear segmentation in 3D" published August 2019 in PLOS Biology: <https://doi.org/10.1371/journal.pbio.3000388> and can be viewed online in the [Image Data Resource](#).

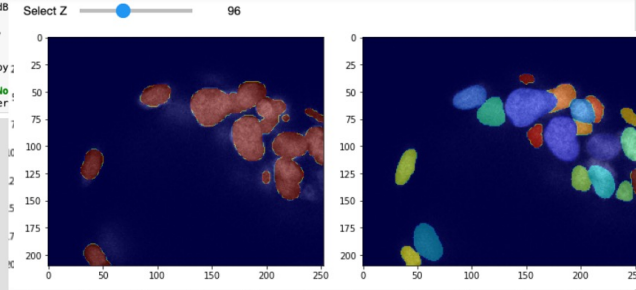
In this notebook, the image is loaded together with the labels and analyzed using StarDist. The StarDist analysis produces a segmentation, which is then viewed side-by-side with the original segmentations produced by the authors of the paper obtained via the loaded labels.

Labels have been saved as mask.

```
In [9]: from omero_zarr import masks
```

```
In [10]: roi_service = conn.getRoiService()  
result = roi_service.findB
```

```
dims = (image.getSizeT(),  
shapes = []  
for roi in result.rois:  
    shapes.append(roi.copy)  
saver = masks.MaskSaver(  
labels, fillColors, proper
```



Segmentation labels from original

Segmented by StarDist in the Notebook

Notebook segments images using StarDist and produces new segmentation labels

Notebook compares the labels



The Image Data Resource (IDR) is a public repository of image datasets from published scientific studies, where the community can submit, search and access high-quality bio-image data.

Cell - IDR

Tissue - IDR

Choose search field (optional) ▼

pax6

2342 Images matched **Gene Symbol:** pax6

991 Images matched **Antibody:** pax6

9 Images matched **HGNC Gene Symbol:** pax6

6 Images matched **Nearest Human Homologue:** pax6

4 Images matched **Original Gene Target:** pax6

3 Images matched **Gene Symbol Synonyms:** eyeless tumor-head-63 l(4)102cdh ey eyel pax-6 ok107 ey/pax6 eye l(4)33 pax6 ey dpax-6 l(4)102cdr pax6 cg1464

118 Studies

13,666,719 Images

353 TB

The Image Data Resource (IDR)

- **Public access**
- **Reference datasets** - complete datasets containing molecular and functional annotations, associated with an existing or upcoming publication.
- **Study integration** - integrating studies or datasets with other datasets via **genes**, **compounds** or **phenotypes**.
- **Curated metadata**
- **Cloud re-analysis**

Which diabetes related genes are expressed in the pancreas?

HumanMine
v12 February 2022
An integrated database of *Homo sapiens* genomic data

<https://www.humanmine.org/humanmine>

TISSUE = "Pancreas"
DISEASE = "diabetes"

```
query.add_constraint('proteinAtlasExpression.tissue.name', "=", TISSUE)
query.add_constraint('proteinAtlasExpression.level', "ONE OF", ["Medium", "High"])
query.add_constraint('organism.name', "=", "Homo sapiens")
query.add_constraint('diseases.name', "CONTAINS", DISEASE)

<BinaryConstraint: Gene.diseases.name CONTAINS diabetes>

Collect the genes

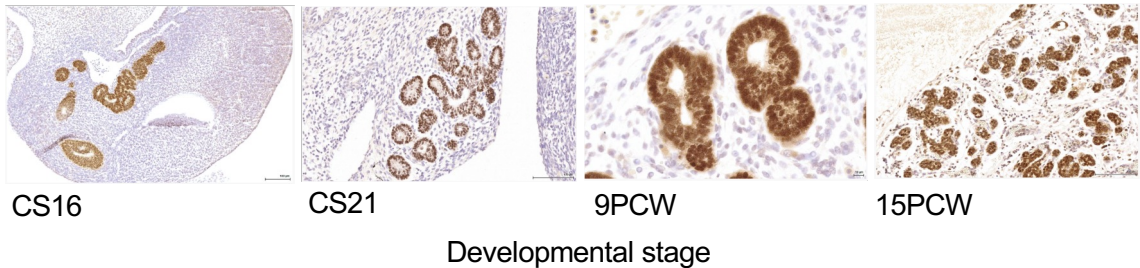
upin_tissue = list()
for row in query.rows():
    upin_tissue.append(row['symbol'])
unique = set(upin_tissue)
genes = sorted(genes, reverse=True)
```

Genes found

WFS1	VEGFA	TCF7L2	TBC1D4	SOD2	SLC30A8	PTPN22	PDX1
MIA3	KCNJ11	IRS2	IRS1	INSR	INS	IGF2BP2	IER3IP1
HNF4A	HNF1B	HMGAL	HPE	GPD2	GCK	ENPP1	EIP2AK3
DNAJC3	CEL	CAPN10	APPL1	AKT2	ABCC8		

IDR MULTIOMICS API

Images linked to gene PDX1



Search for images in IDR associated to the genes found in Humanmine

From the list of genes found using the intermine API, we are now looking in [Image Data Resource](#) for studies linked to those genes and with tissue as Sample Type.

```
TYPE = "gene"
SAMPLE_TYPE = "tissue"
EXPRESSION_KEY = "Expression Pattern Description"
EXPRESSION = "islets" # "Brain"
KEYS = {'phenotype':
    {'Phenotype Term Name',
     'Phenotype Term Accession',
     'Phenotype Term Accession URL',
    }}

projects = list()
for gene in genes:
    qs1 = {'key': TYPE, 'value': gene}
    url1 = URL.format(**qs1)
    json = session.get(url1).json()
    for m in json['maps']:
        qs2 = {'key': TYPE, 'value': gene}
        url2 = SCREENS_PROJECTS_URL.format(**qs2)
        json = session.get(url2).json()
        for p in json['projects']:
            value = find_type("project", p['id'])
            if value > -1:
                projects.append(value)
```

idr0070, Kerwin et al, *Journal of Anatomy* DOI: 10.1111/j.1469-7580.2010.01290.x

<https://workflowhub.eu/workflows/242>

Access possibilities of images

image from idr0044, McDole et al.	Download	IDR API Access	OME-Zarr Access via S3
Load image subregion, e.g., single chunk or tile	No , only per file	Yes	Yes
Lazy loading	No	No	Yes . Use Dask collections: da.from_zarr(endpoi nt_url)
Easily analyze in parallel	No , depends on file format which may require a translation library.	Difficult due to the transfer protocol used (zeroc-ice)	Yes . Use Dask schedulers: dask.delayed(analyz e)(t, c, z)

Adapted from Table 3, Josh Moore et. al.: OME-Zarr: a cloud-optimized bioimaging file format with international community support, bioRxiv 2023.02.17.528834

What is OME-NGFF ?

Brief Communication | [Open Access](#) | [Published: 29 November 2021](#)

OME-NGFF: a next-generation file format for expanding bioimaging data-access strategies

[Josh Moore](#), [Chris Allan](#), [Sébastien Besson](#), [Jean-Marie Burel](#), [Erin Diel](#), [David Gault](#), [Kevin Kozlowski](#), [Dominik Lindner](#), [Melissa Linkert](#), [Trevor Manz](#), [Will Moore](#), [Constantin Pape](#), [Christian Tischer](#) & [Jason R. Swedlow](#) 

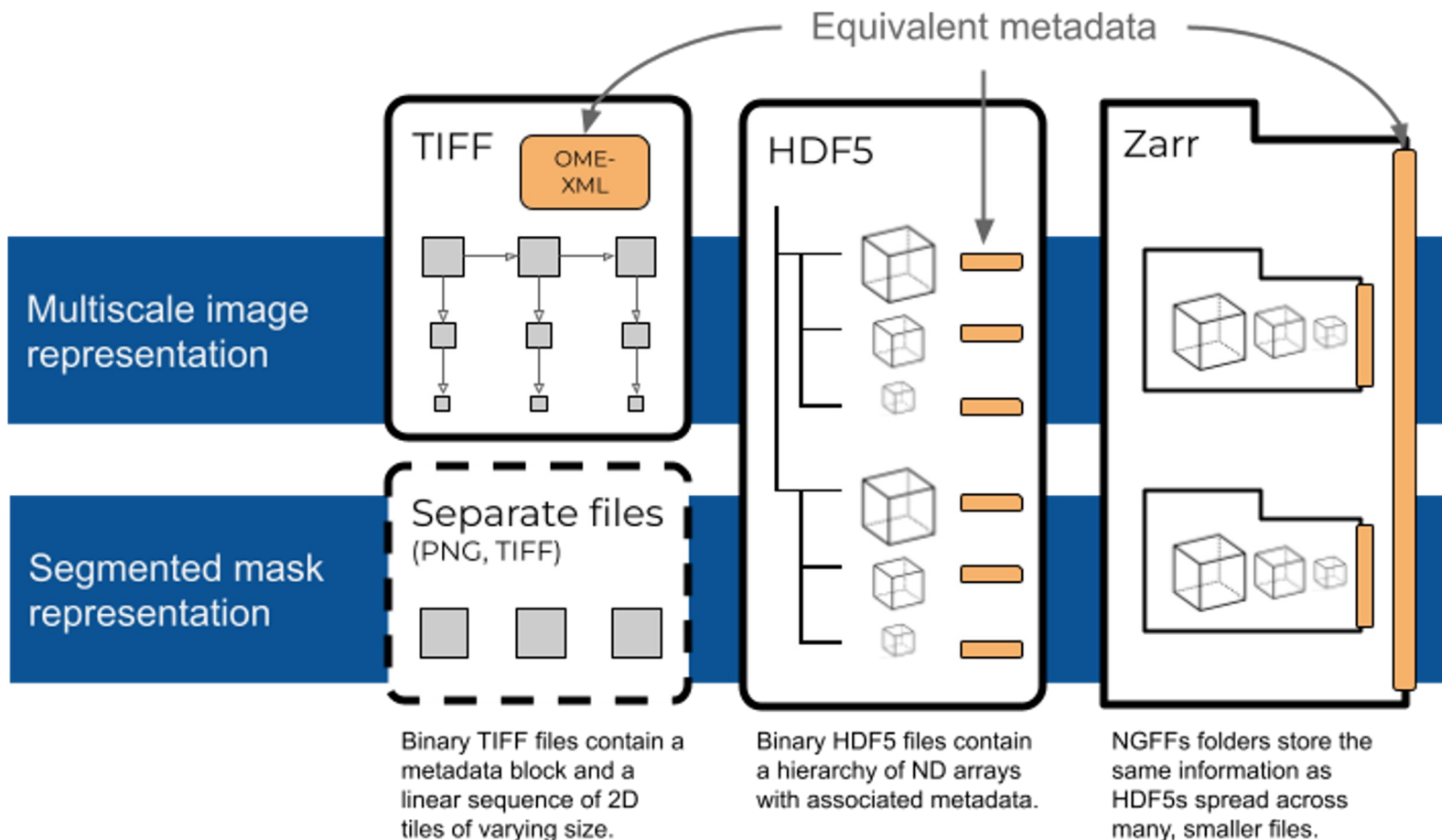
[Nature Methods](#) **18**, 1496–1498 (2021) | [Cite this article](#)

6045 Accesses | **4** Citations | **80** Altmetric | [Metrics](#)

<https://ngff.openmicroscopy.org>

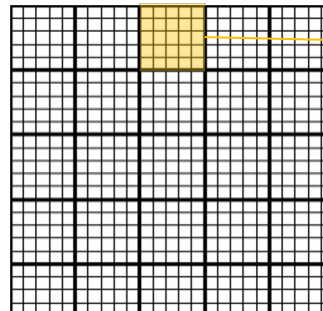
<https://ngff.openmicroscopy.org/data>

OME-NGFF uses Zarr



Zarr

chunk = file =
object



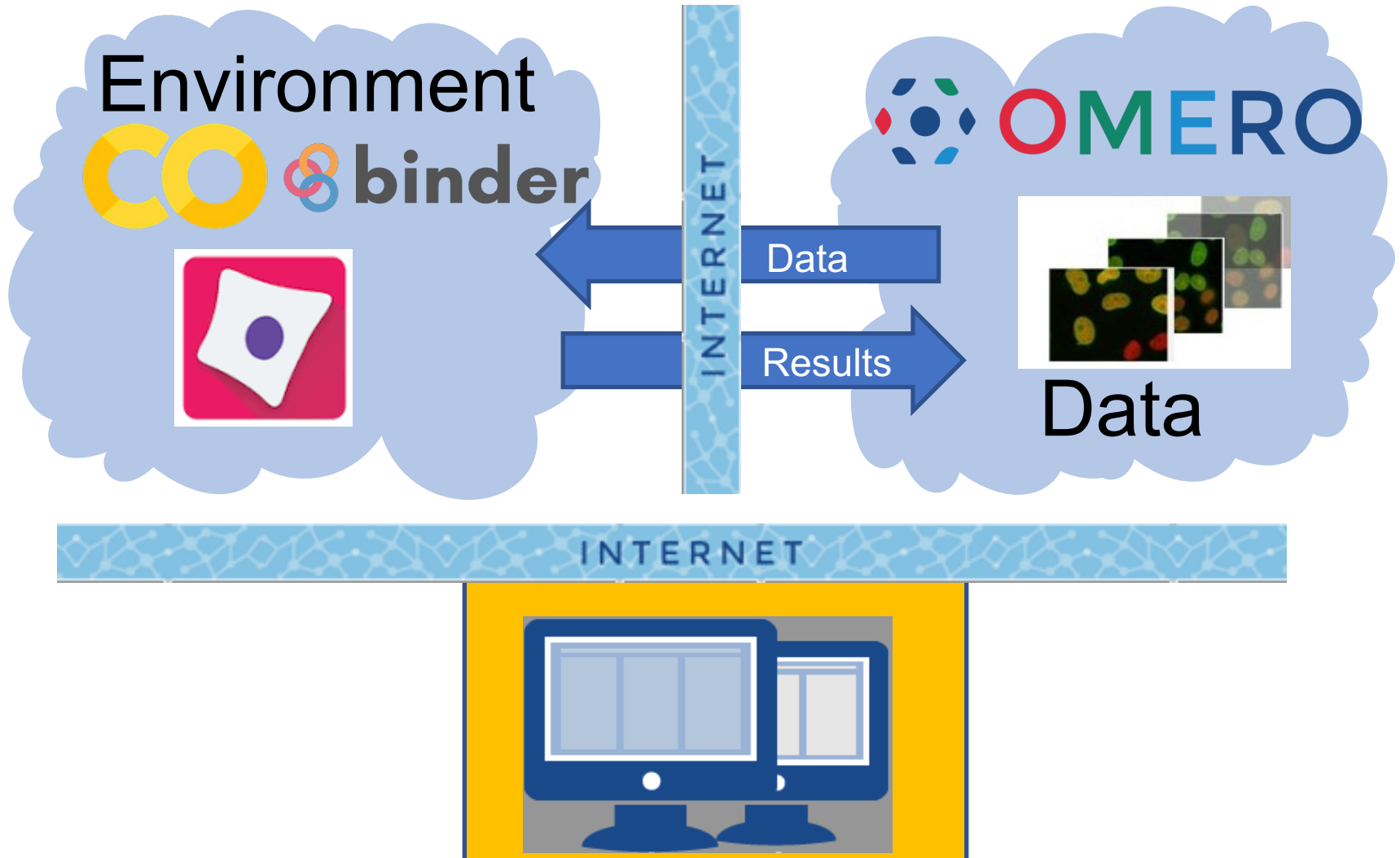
▼	folder	s0	--
▼	folder	0	--
▼	folder	0	--
▼	folder	0	--
▼	folder	0	--
	file	0	175 KB
	file	1	173 KB
	file	2	173 KB
	file	3	177 KB
	file	4	174 KB
	file	5	173 KB
	file	6	173 KB
	file	7	174 KB
▶	folder	1	--
▶	folder	2	--
▶	folder	3	--
▶	folder	4	--
▶	folder	1	--
▶	folder	2	--
▶	folder	3	--

Remote cloud storage

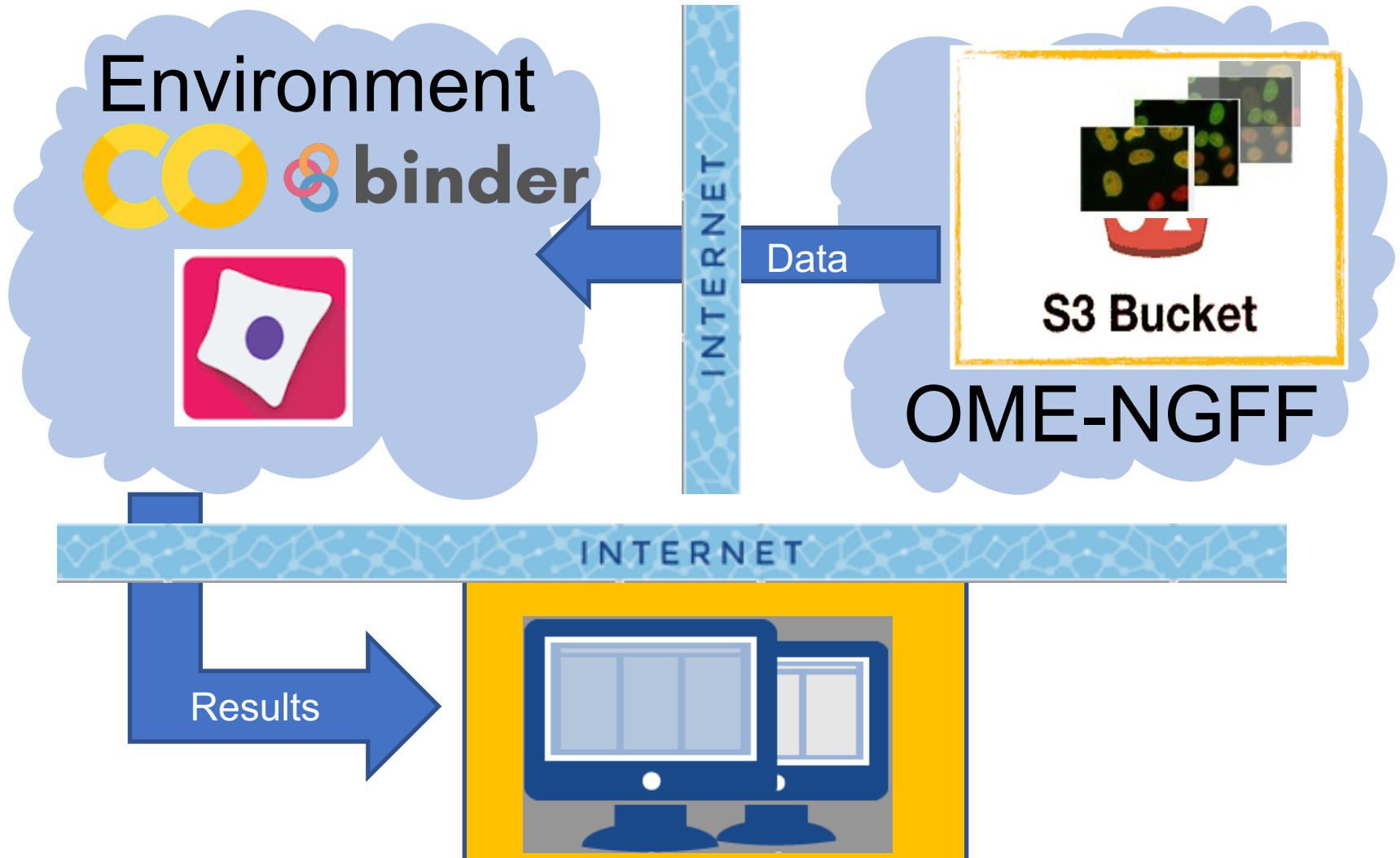


S3 Bucket

This workshop - OMERO



This workshop – OME-NGFF



Suggested workflow

- **Find data of interest in IDR**
- **Set-up analysis environment**
- **Run examples to access IDR data**
- **Edit the code to add your own analysis**

YouTube **openmicroscopy** channel:

1. **Exploring IDR**
2. **IDR Analytical workflows** playlists

Materials

- Link to the walkthrough pdf

<https://downloads.openmicroscopy.org/presentations/2023/Neubias>

- Videos

- [openmicroscopy YouTube channel](#)

- [IDR homepage](#)

- [I2K video \(minute 28\) for NGFF](#)

Thank you

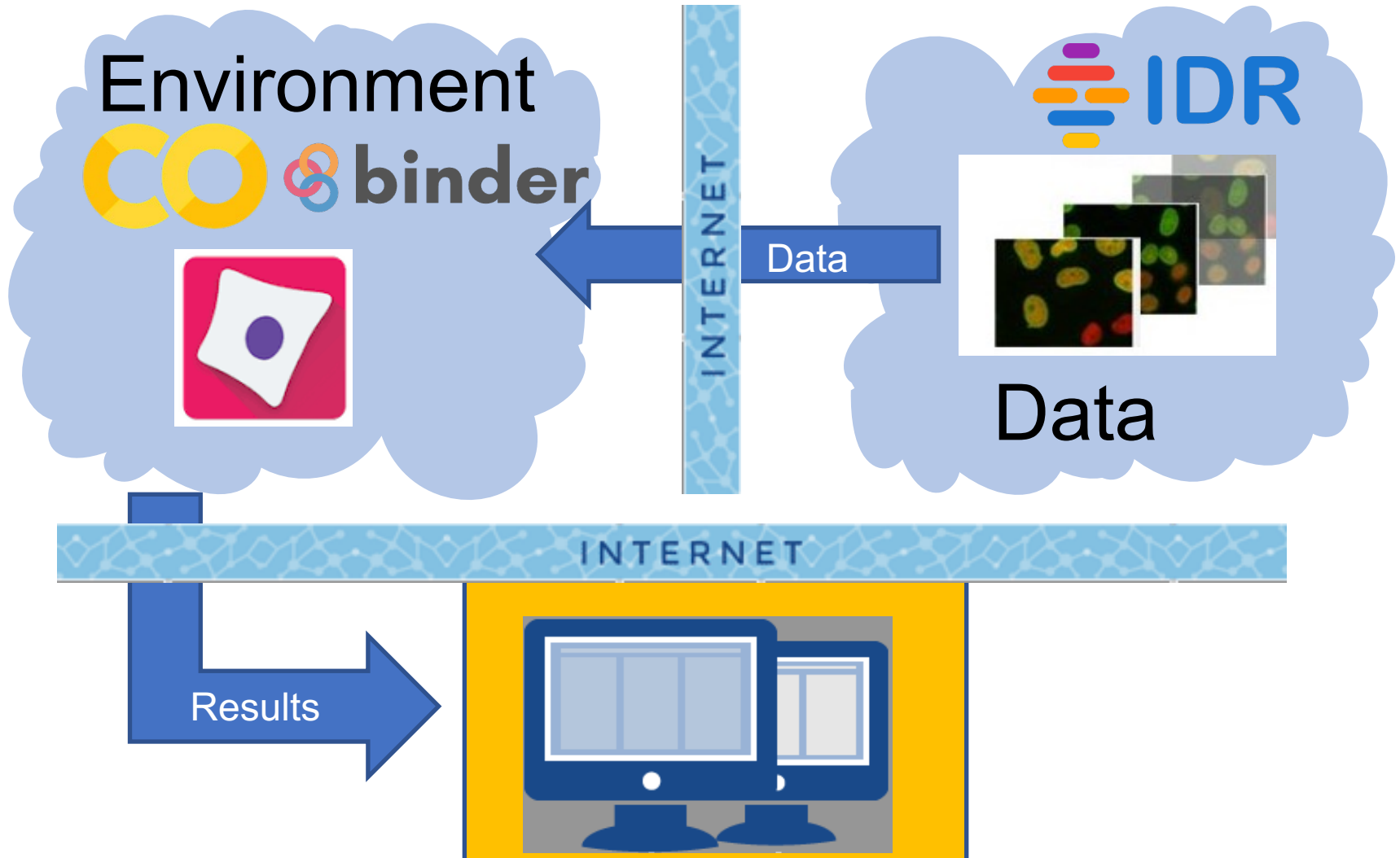


Parallel analysis

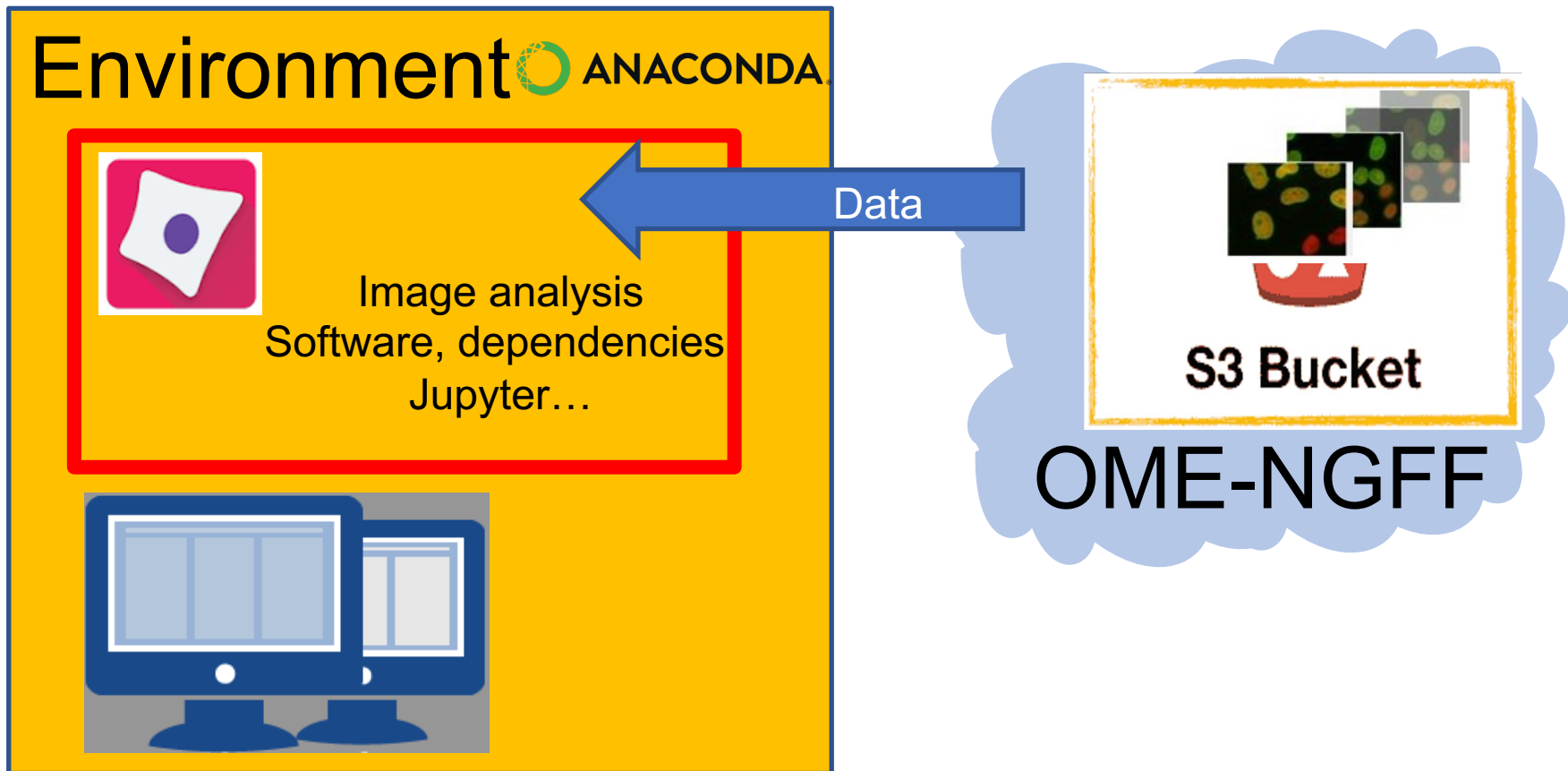


- Natively scales Python
- Advanced parallelism for analytics
- See <https://dask.org/index.html>

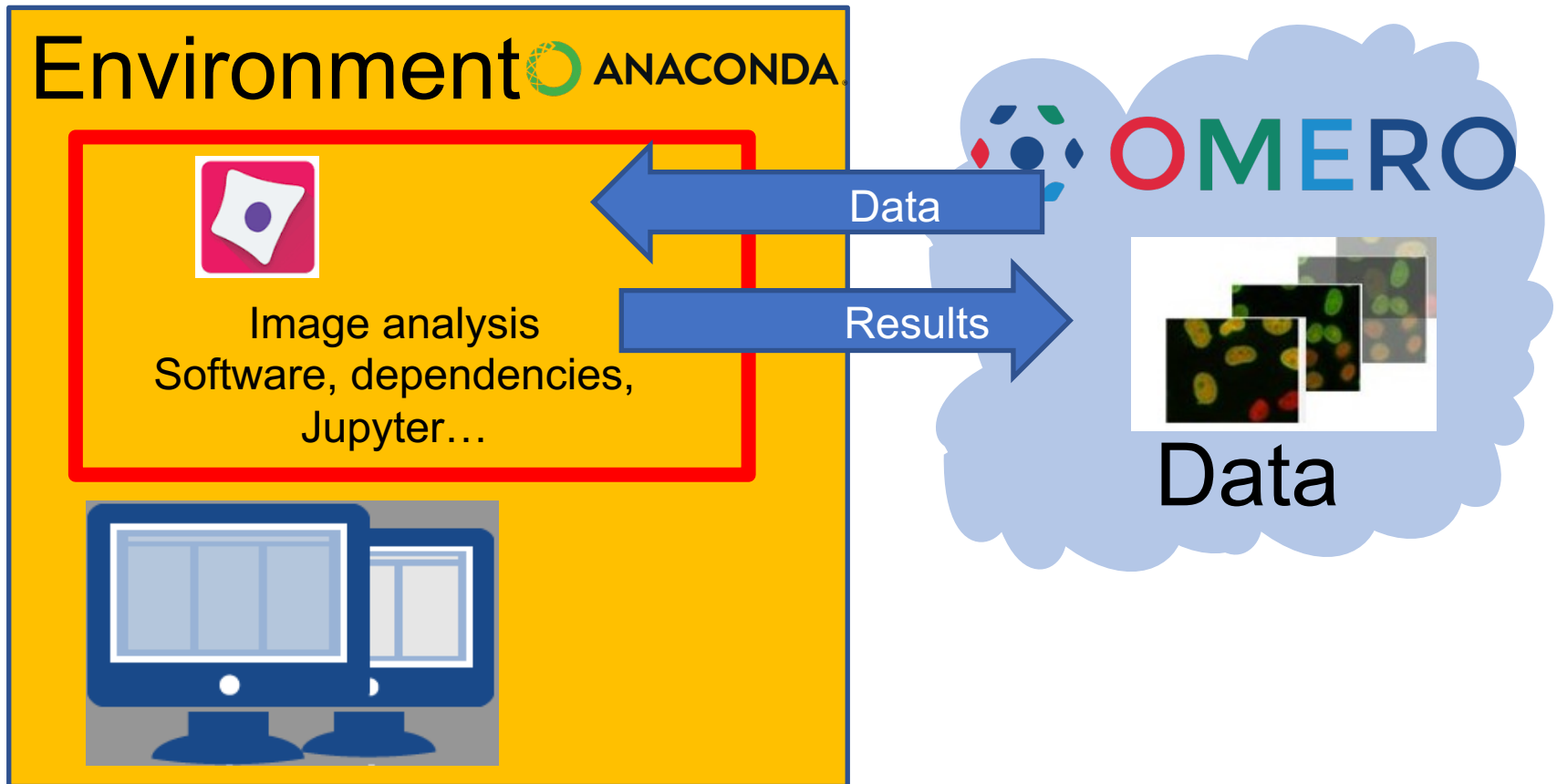
This workshop – IDR



This workshop – local env



This workshop – local env



This workshop – local env

