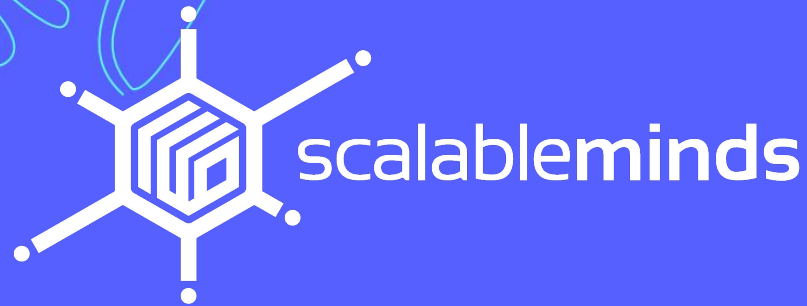


X @normanrz

m normanrz@mastodon.social



# Interoperability for large images with OME-Zarr and WEBKNOSSOS

29 May 2024

Norman Rzepka ~ OME Meeting

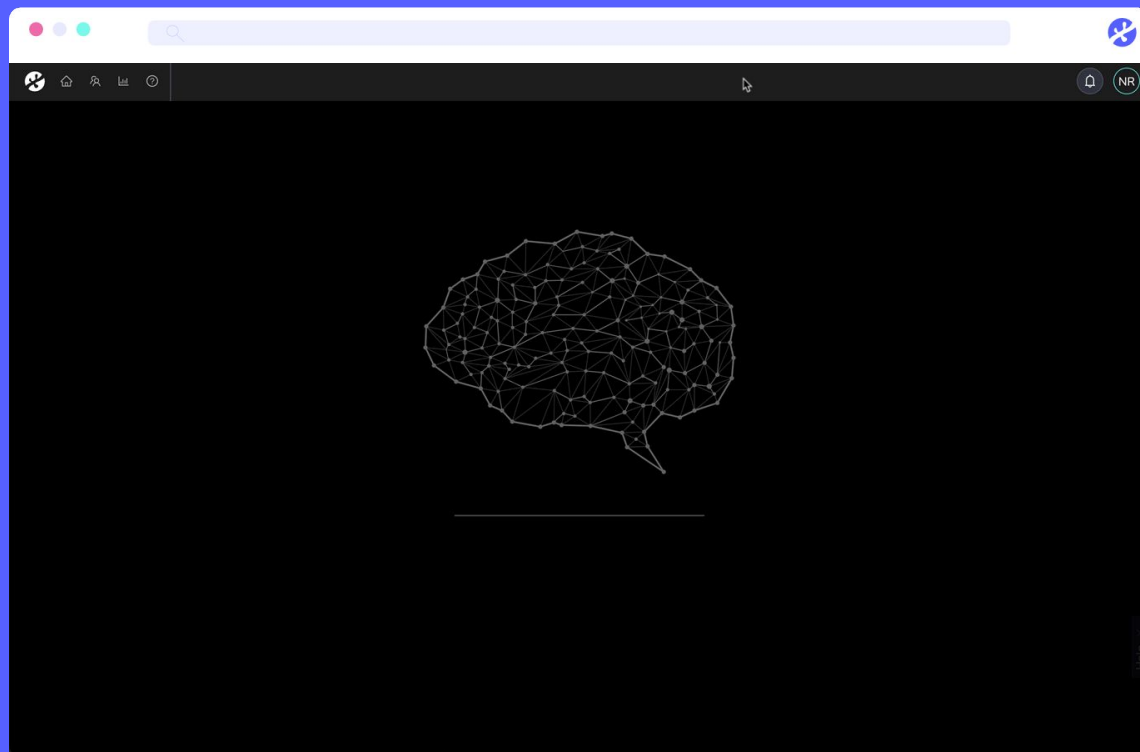


WEBKNOSSOS

A large-image data platform

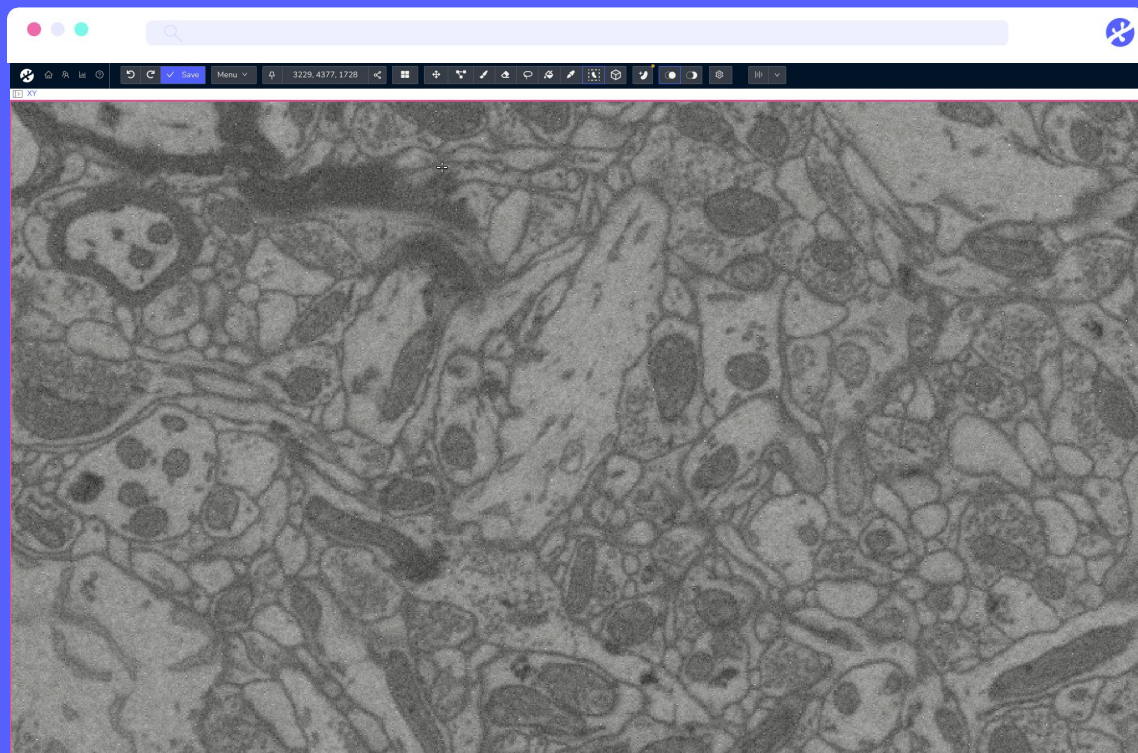
Easily browse  
through  
terabytes of  
data

over the  
web



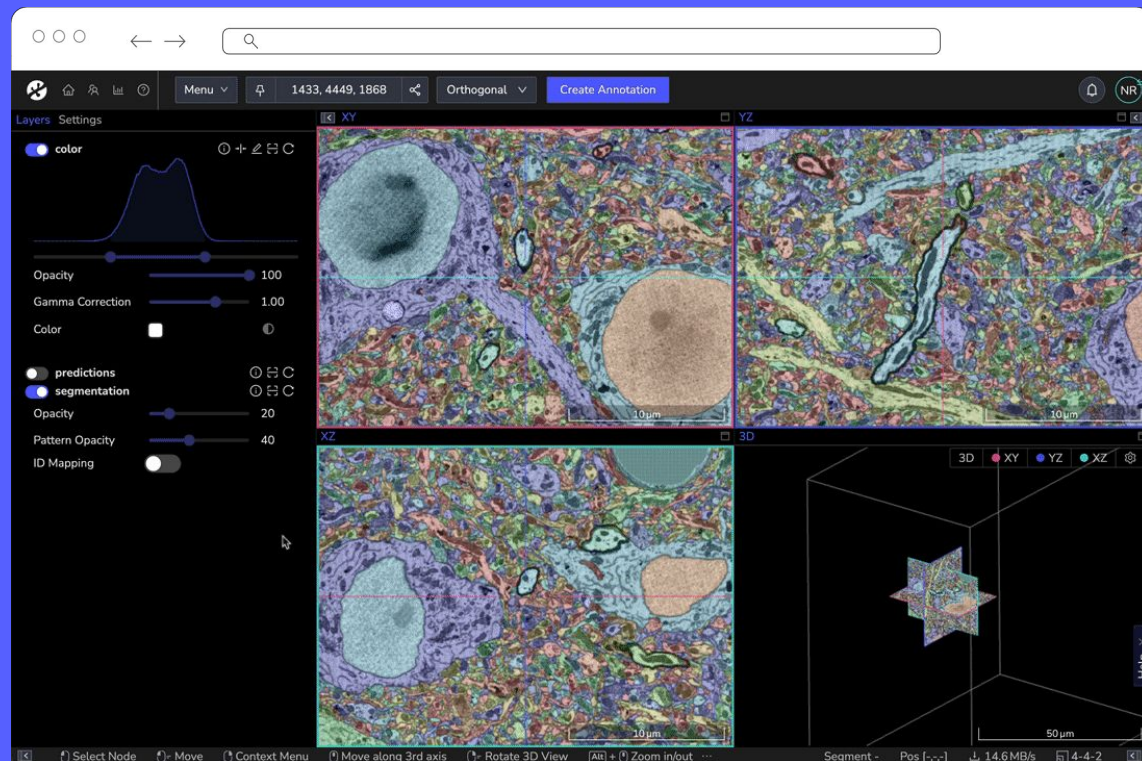
Annotate  
data

Quickly and  
collaboratively





Train and run  
AI models  
(paid feature)



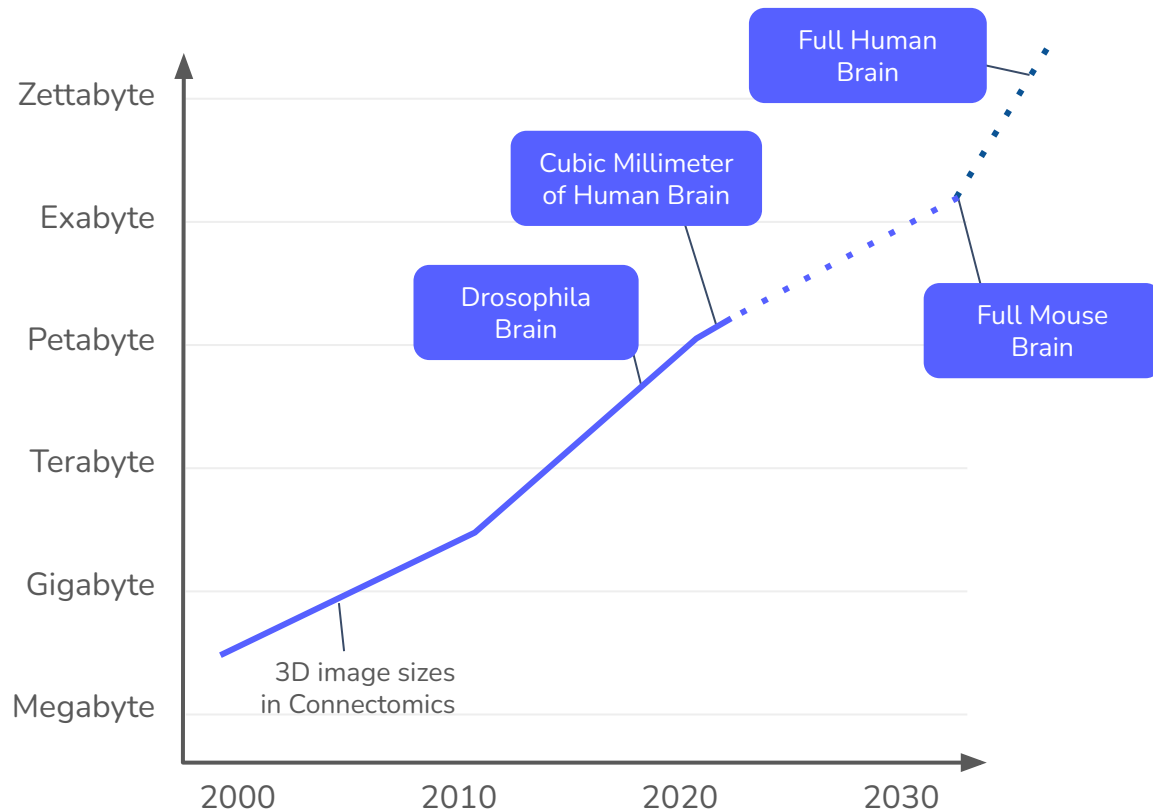
③ Analyses data

① Acquires data

② Annotates data

④ Explores data after publication

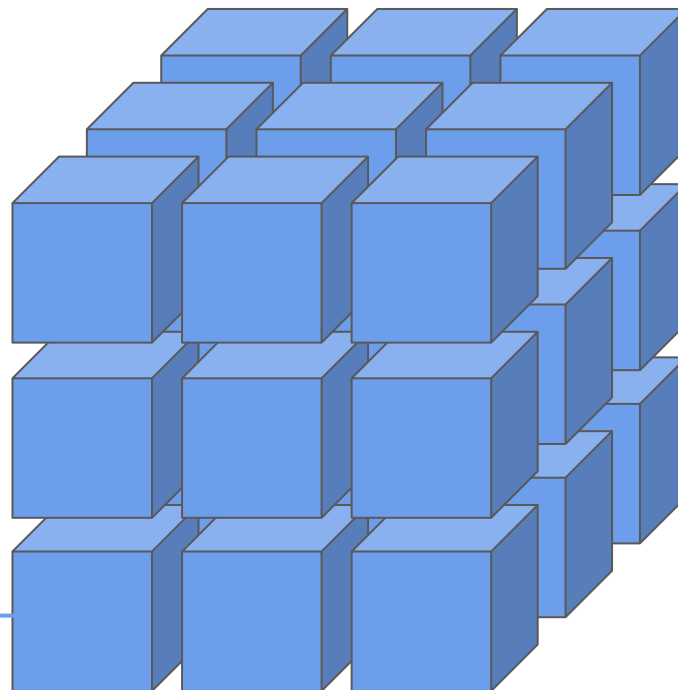
How can we store images that are 100s of TBs?



Key enabler:  
Chunked 3D data  
streaming and  
storage

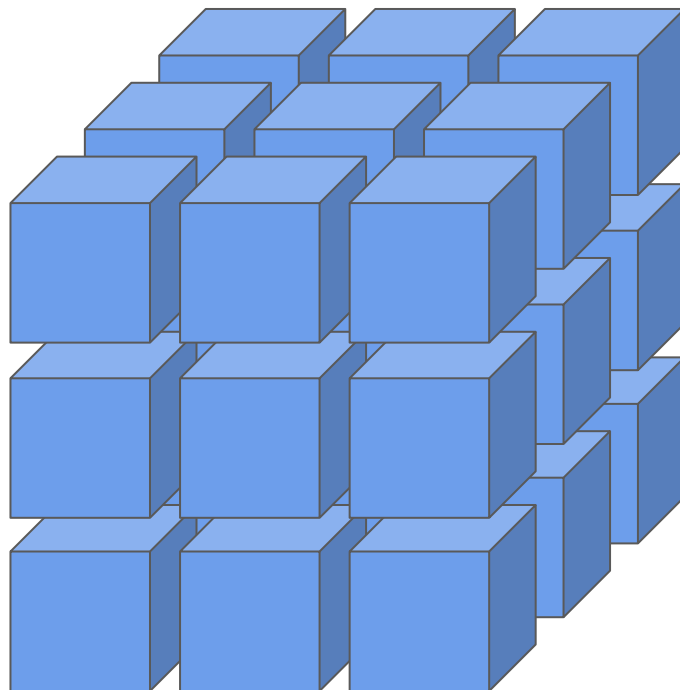
Chunks

$32 \times 32 \times 32$  vx  
= 32kB for uint8



Many 32kB files  
don't play well with  
current file systems

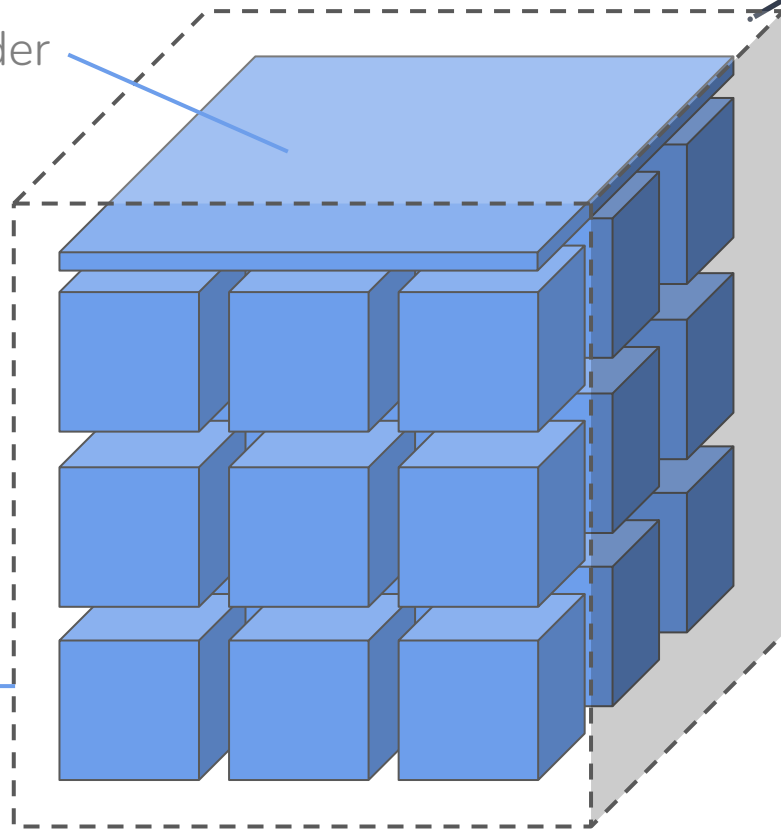
1TB = 33.554.432 files



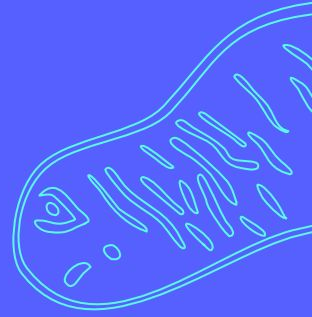
Idea: Put many  
chunks in a larger  
file (shard)

Shard file  
 $32 \times 32 \times 32$  chunks  
= 1GB for uint8

Header







What about interoperability?



WEBKNOSSOS Wrap

N5

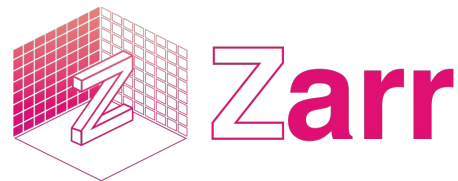
Neuroglancer precomputed

BossDB

Zarr

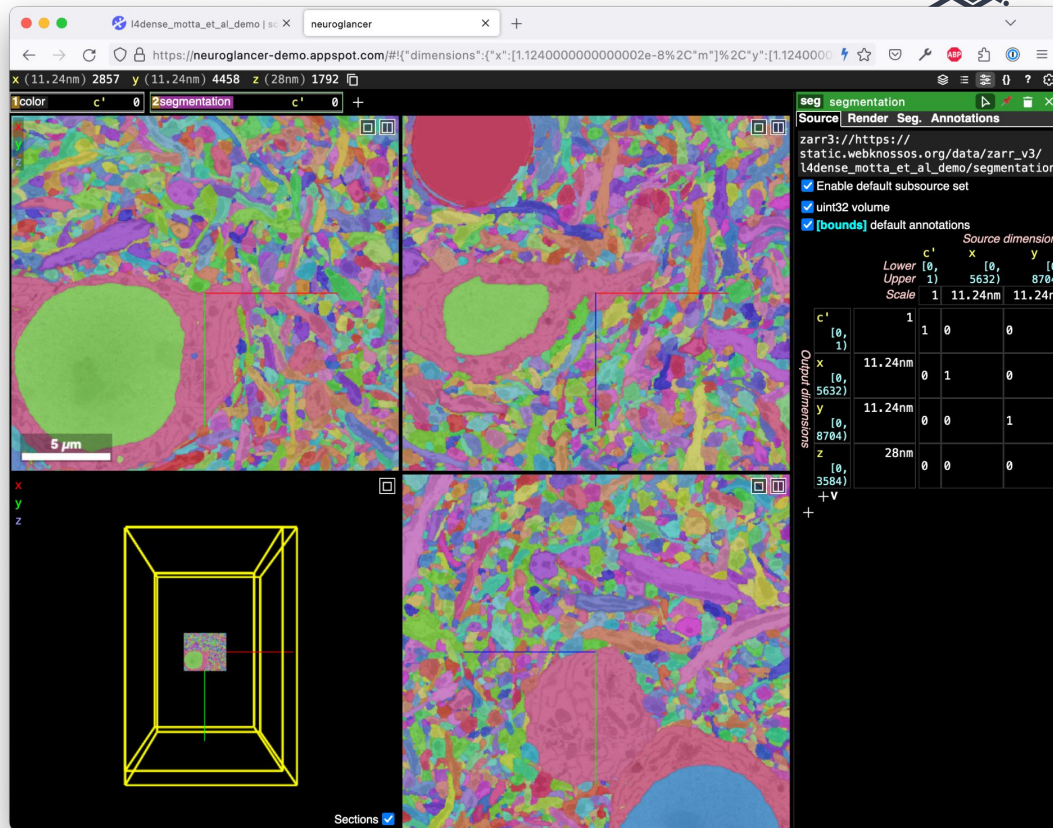
HDF5

# FILE FORMATS



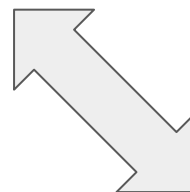
A next-generation  
file format (NGFF)

Broad support  
from applications





Integration with  
OMERO



## Data streaming for scripting

```
import webknossos as wk

with wk.webknossos_context(token="..."):
    dataset = wk.Dataset.remote_open("4496763.zarr")
    layer = dataset.get_color_layer()
    mag = layer.get_mag(1)

    data = mag.read(
        absolute_offset=(0, 0, 0),
        size=(1024, 1024, 25),
    )

run_analysis(data)
```



But, OME-Zarr  
doesn't have  
sharding

The screenshot shows a web browser window displaying a forum post on the image.sc website. The browser's address bar shows the URL <https://forum.image.sc/t/sharding-support-in-ome-zarr>. The forum post is titled "Sharding support in ome-zarr" and is categorized under "Development" by user "ome-ngff". The post is authored by "Norman Rzepka" from the "Community Forum Team" and is dated "Jul 2021". The content of the post discusses the concept of sharding in the context of 3D image data, explaining that it involves co-locating multiple chunks in a single file. It provides an example of a 3D image with a chunk size of  $32^3$  and lists several chunk coordinates:  $\{0, 31\} \cdot \{0, 31\} \cdot \{0, 31\}$ ,  $\{0, 31\} \cdot \{0, 31\} \cdot \{32, 63\}$ , and  $\{0, 31\} \cdot \{0, 31\} \cdot \{32, 63\}$ . The post explains that this configuration allows for fast access to randomly-shaped portions of the data, which is compatible with modern cluster file systems. It also mentions that with uncompressed data, individual chunks can be accessed directly, and with Morton-ordering, data locality can be optimized in n-D. The post concludes by stating that reading compressed data requires reading a header first, and writing compressed data is more complicated but feasible with the right chunk and shard sizes. The post is the first of seven in a thread, as indicated by the "1 / 7" at the bottom right.

The ZEP process  
didn't exist and  
Zarr v3 was  
still in under  
development

Zarr core specification (version 3.0)

**Specification URI:**  
<https://zarr-specs.readthedocs.io/en/latest/v3/core/v3.0.html>

**Editors:**

- Alistair Miles (@alimanfoo), Wellcome Sanger Institute
- Jonathan Striebel (@jstriebel), Scalable Minds
- Jeremy Maitin-Shepard (@jbms), Google

**Corresponding ZEP:**  
ZEP0001 — Zarr specification version 3

Copyright 2019–Present Zarr core development team. This work is licensed under a [Creative Commons Attribution 3.0 Unported License](#).

---

**Status of this document**

ZEP0001 was accepted on May 15th, 2023 via [zarr-developers/zarr-specs#227](#).


**Introduction**

This specification defines a format for multidimensional array data. This type of data is common in scientific and numerical computing applications. Many domains face computational challenges as increasingly large volumes of

# Adding sharding as a codec to Zarr

Sharding codec (version 1.0) — X +

← → ↻ 🔒 <https://zarr-specs.readthedocs.io/en/latest/v3/co> 📄 🔍 ☆ 📧 ⬇️ 🌐 📄 🛡️

☰  ☰

## Sharding codec (version 1.0)

**Specification URI:**  
<https://zarr-specs.readthedocs.io/en/latest/v3/codecs/sharding-indexed/v1.0.html>

**Editors:**

- Jonathan Striebel (@jstriebel), Scalable Minds
- Norman Rzepka (@normanrz), Scalable Minds
- Jeremy Maitin-Shepard (@jbms), Google

**Corresponding ZEP:**  
ZEP0002 — Sharding codec

Copyright 2022–Present [Zarr core development team](#). This work is licensed under a [Creative Commons Attribution 3.0 Unported License](#).

---

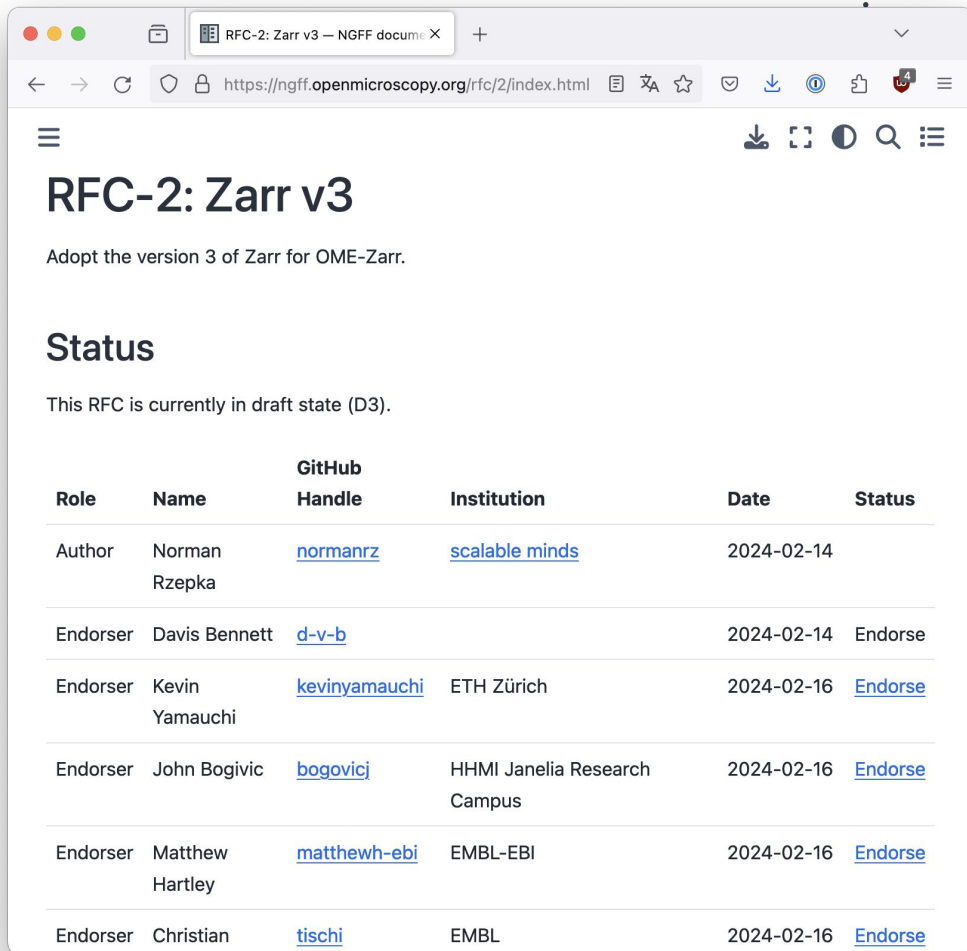
### Status of this document

ZEP0002 was accepted on **November 1st, 2023** via [zarr-developers/zarr-specs#254](#).

### Motivation

In many cases, it becomes inefficient or impractical to store a large number of chunks as separate files or objects due to the design constraints of the underlying storage. For example, the file block size and maximum inode

# Adopt sharding and Zarr v3 in OME-Zarr



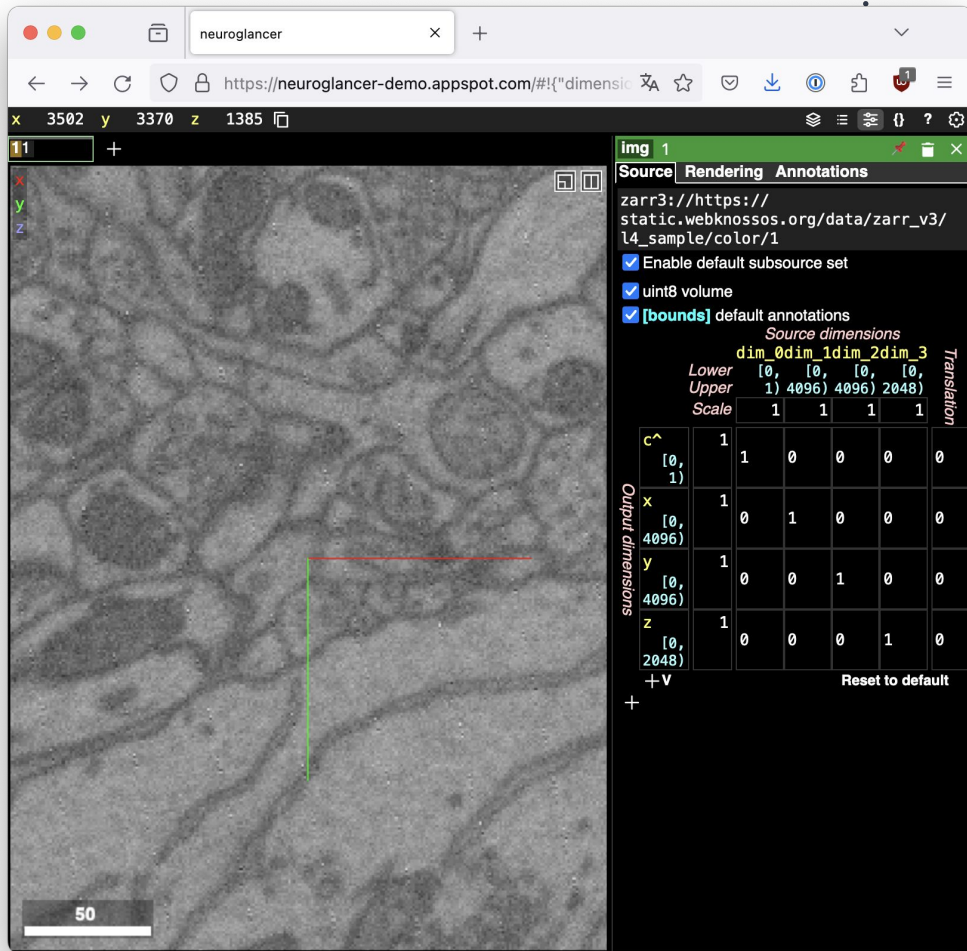
The screenshot shows a web browser window displaying the RFC-2: Zarr v3 page. The page title is "RFC-2: Zarr v3" and the URL is "https://ngff.openmicroscopy.org/rfc/2/index.html". The page content includes a heading "RFC-2: Zarr v3", a sub-heading "Adopt the version 3 of Zarr for OME-Zarr.", and a section titled "Status" with the text "This RFC is currently in draft state (D3)". Below this is a table listing the roles, names, GitHub handles, institutions, dates, and statuses of the individuals involved in the RFC process.

Role	Name	GitHub Handle	Institution	Date	Status
Author	Norman Rzepka	<a href="#">normanz</a>	<a href="#">scalable minds</a>	2024-02-14	
Endorser	Davis Bennett	<a href="#">d-v-b</a>		2024-02-14	Endorse
Endorser	Kevin Yamauchi	<a href="#">kevinyamauchi</a>	ETH Zürich	2024-02-16	<a href="#">Endorse</a>
Endorser	John Bogovic	<a href="#">bogovicj</a>	HHMI Janelia Research Campus	2024-02-16	<a href="#">Endorse</a>
Endorser	Matthew Hartley	<a href="#">matthewh-ebi</a>	EMBL-EBI	2024-02-16	<a href="#">Endorse</a>
Endorser	Christian	<a href="#">tisch</a>	EMBL	2024-02-16	<a href="#">Endorse</a>

Specification

Implementation

# Visualize with neuroglancer





# Visualize and annotate with WEBKNOSSOS

The screenshot shows a GitHub pull request page for the repository 'scalableminds/webknossos'. The pull request title is 'Implement Zarr v3 and sharding codec #7079'. It is marked as 'Merged' and was merged by 'frcroth' into the 'master' branch from the 'zarr3' branch on June 5, 2023. The pull request has 34 commits, 1 check, and 25 files changed, with a net change of +1,027 lines and -58 lines. The pull request is assigned to 'frcroth' and has reviewers 'normanz' and 'fm3'. The pull request description includes a URL for a deployed dev instance, steps to test, and questions.

**URL of deployed dev instance (used for testing):**

- <https://zarr.webknossos.xyz>

**Steps to test:**

- Explore and view an unsharded and a sharded zarr v3 dataset.

**Questions**

- Name classes zarr3 or zarrV3? -> zarr3
- Storage transformers: There are currently no storage transformers proposed (or I didn't find them). Should this implementation already allow for these, or is that not relevant? -> Not relevant
- Are there any more codecs that work like sharding

# Read/write with zarrita

The screenshot shows the GitHub repository page for 'scalableminds/zarrita'. The browser address bar shows 'https://github.com/scalableminds/zarrita'. The repository is public and has 18 stars, 6 forks, and 8 watchers. The repository contains several files and folders, including a workflow file, test files, and documentation. The 'About' section on the right indicates that no description, website, or topics are provided.

File/Folder	Commit Message	Commit Date
<a href="#">.github/workflows</a>	add manual trigger ...	2 weeks ago
<a href="#">tests</a>	added test for shard...	4 months ago
<a href="#">zarrita</a>	added test for shard...	4 months ago
<a href="#">.gitignore</a>	sharding fixes	last year
<a href="#">BENCHMARK.md</a>	updated benchmarks	10 months ago
<a href="#">LICENSE</a>	Update LICENSE	last year
<a href="#">README.md</a>	Rename endian cod...	9 months ago
<a href="#">check.sh</a>	refactor runtime_co...	10 months ago
<a href="#">poetry.lock</a>	update upath	10 months ago

**About**

No description, website, or topics provided.

- Readme
- MIT license
- Activity
- Custom properties
- 18 stars
- 8 watching
- 6 forks

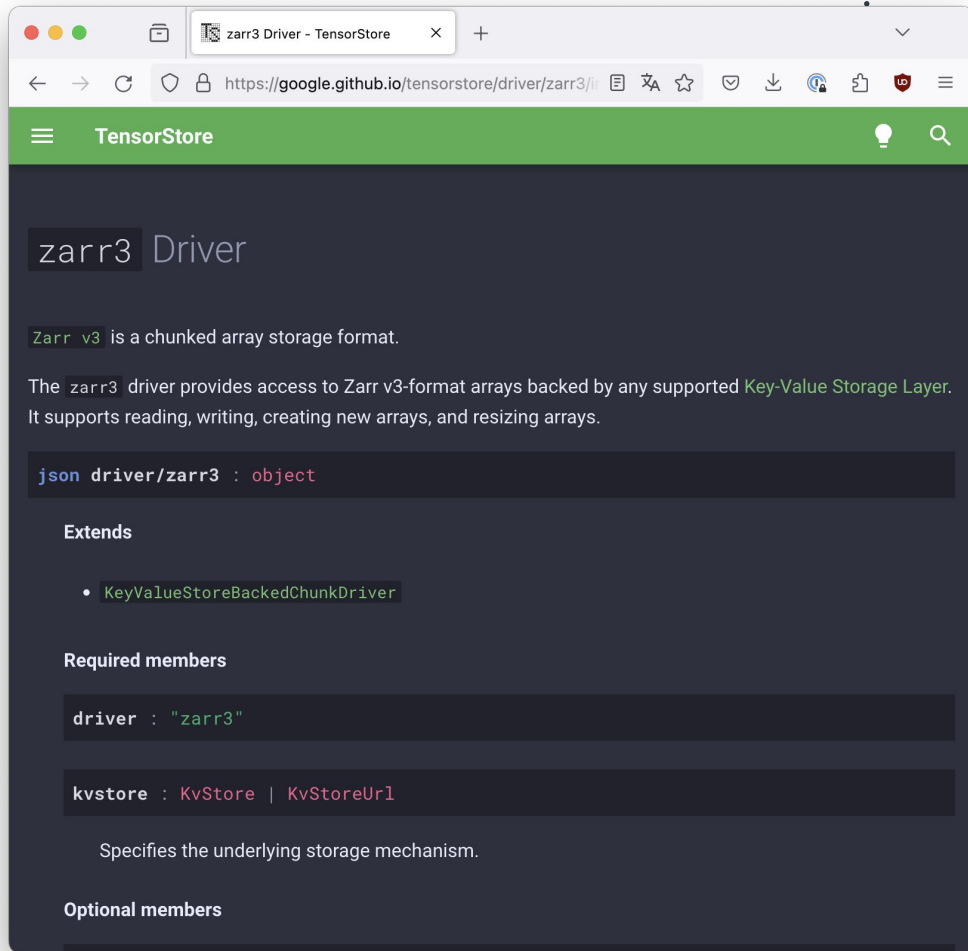
Report repository

**Releases**

1 tags

[Create a new release](#)

# Read/write with tensorstore



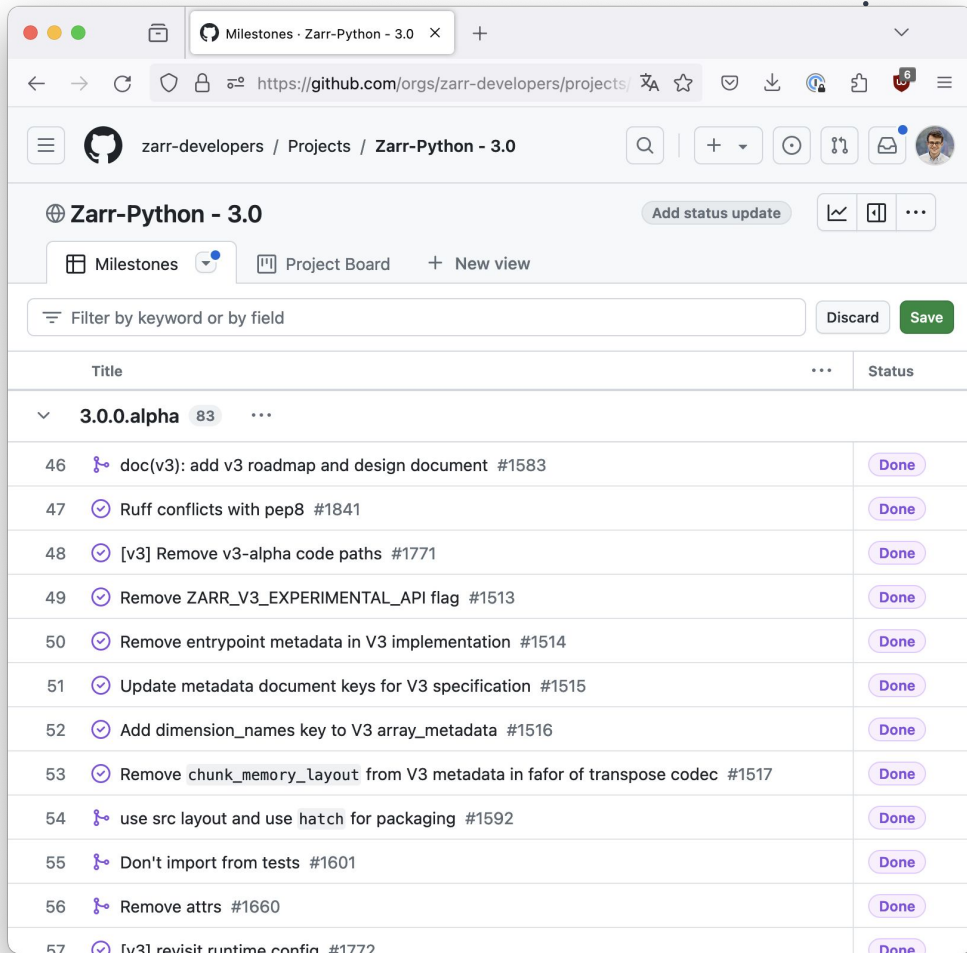
The screenshot shows a web browser window displaying the documentation for the `zarr3 Driver` on the TensorStore website. The page title is "zarr3 Driver". The main text states: "Zarr v3 is a chunked array storage format. The `zarr3` driver provides access to Zarr v3-format arrays backed by any supported [Key-Value Storage Layer](#). It supports reading, writing, creating new arrays, and resizing arrays." Below this, there is a code block showing the JSON representation of the driver: 

```
json driver/zarr3 : object
```

. The page also lists "Extends" as `KeyValueStoreBackedChunkDriver`. Under "Required members", there are two entries: `driver` with the value `"zarr3"`, and `kvstore` with the type `KvStore | KvStoreUrl`. A description for `kvstore` says "Specifies the underlying storage mechanism." The "Optional members" section is currently empty.

# zarr-python 3.0

## Release scheduled for June



Milestones · Zarr-Python - 3.0

zarr-developers / Projects / Zarr-Python - 3.0

### Zarr-Python - 3.0

Add status update

Milestones Project Board + New view

Filter by keyword or by field Discard Save

Title	Status
3.0.0.alpha 83	
46 doc(v3): add v3 roadmap and design document #1583	Done
47 Ruff conflicts with pep8 #1841	Done
48 [v3] Remove v3-alpha code paths #1771	Done
49 Remove ZARR_V3_EXPERIMENTAL_API flag #1513	Done
50 Remove entrypoint metadata in V3 implementation #1514	Done
51 Update metadata document keys for V3 specification #1515	Done
52 Add dimension_names key to V3 array_metadata #1516	Done
53 Remove chunk_memory_layout from V3 metadata in fafor of transpose codec #1517	Done
54 use src layout and use hatch for packaging #1592	Done
55 Don't import from tests #1601	Done
56 Remove attrs #1660	Done
57 [v3] revisit runtime config #1772	Done

zarr-java and  
n5-zarr for  
Fiji, OMERO,  
bf2raw etc.

zarr-developers / zarr-java

Code Issues Pull requests Actions Projects Wiki Security

zarr-java Public

Edit Pins Unwatch 13 Fork 2 Star 3

main Go to file Code

normanrz throw an exception... 850de50 · 2 months ago

📁 .github/workflows	ci	last year
📁 src	throw an exception ...	2 months ago
📄 .gitignore	refactoring	last year
📄 LICENSE	init	last year
📄 README.md	com.scalableminds ...	7 months ago
📄 pom.xml	com.scalableminds ...	7 months ago

README Code of conduct MIT license

Early preview of zarr-java

About

Java implementation of the Zarr Specification

- Readme
- MIT license
- Code of conduct
- Activity
- Custom properties

3 stars  
13 watching  
2 forks

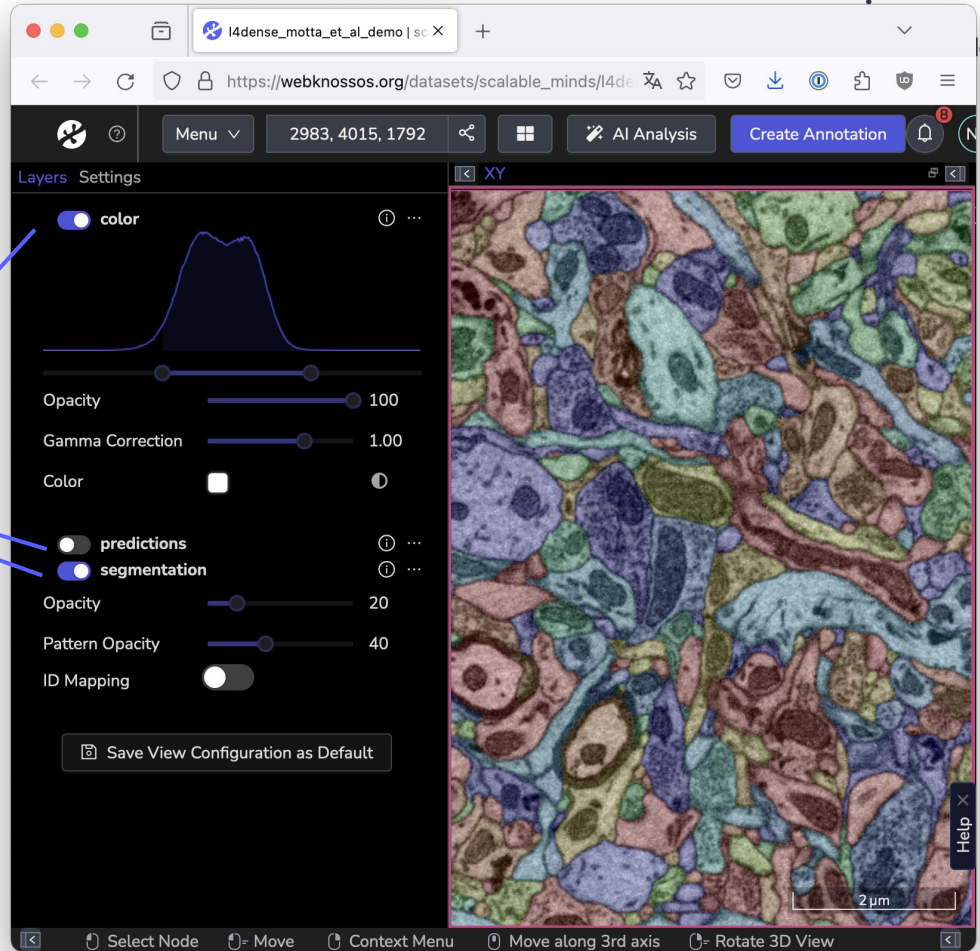
Releases

No releases published  
[Create a new release](#)

What is next?



Collections of images



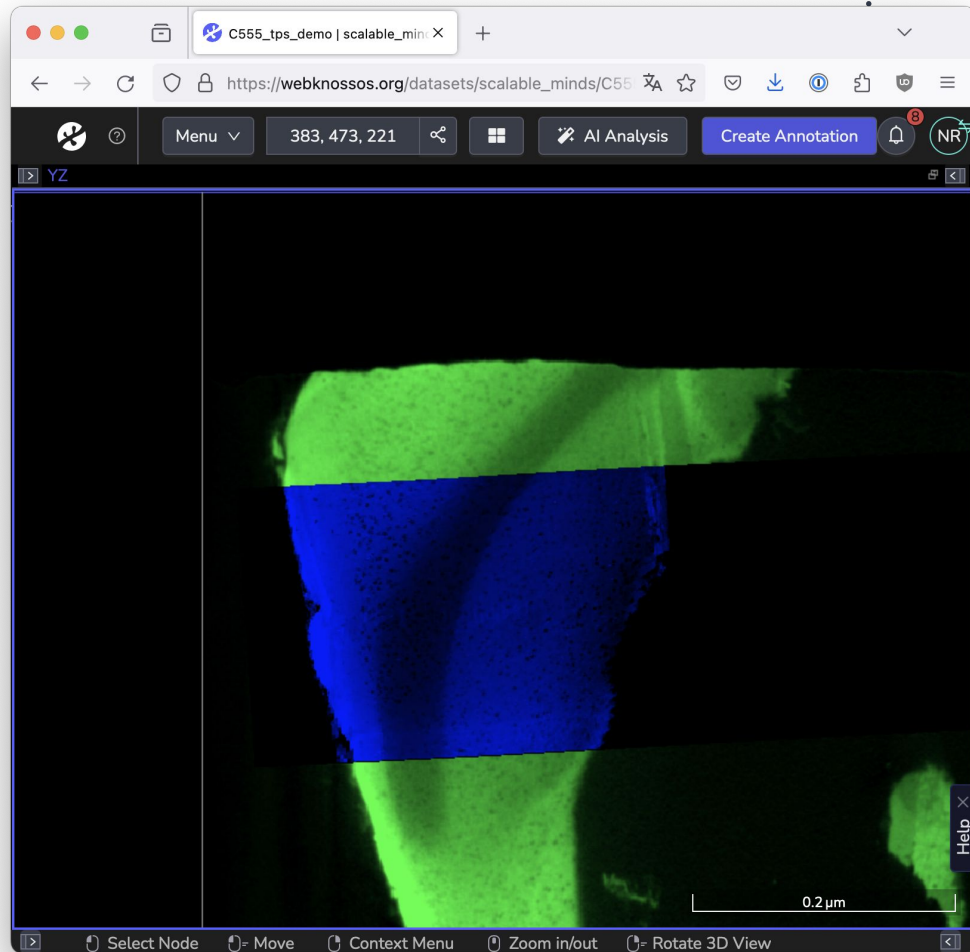
# Collections of images

```

...: {
  "https://ngff.openmicroscopy.org/0.6": { // versioned namespace
    collection: { // just one collection per json file
      name: "jrc_hela-1",
      members: [
        {
          type: "image", // "image" or "collection"; "image" is default
          path: "./fibsem-uint8", // relative path or absolute URL
          name: "fibsem",
          attributes: { // image-specific user-defined metadata
            rendering: { // not standardized
              min: 123,
              max: 230,
              inverted: true,
            },
          },
        },
        {
          path: "./endo_pred",
          name: "endo_pred",
          attributes: {
            rendering: { // not standardized
              disabled: true,
            },
          },
        },
        {
          path: "./endo_seg",
          name: "endo_seg",
          attributes: {},
        },
      ],
      attributes: { // collection-wide user-defined metadata
        rendering: {
          position: [1728, 1273, 1263],
          zoom: 2.3,
        },
      },
    },
  },
}

```

# Correlative visualization and annotation





~~OME-NGFF~~



OME-Zarr 1.0

Thank you!



@webknossos



webknossos@mstdn.science

<https://webknossos.org>

<https://scalableminds.com>