

slides: bit.ly/zarr-evolution-scipy-2023



Chunked, Compressed, & Cloud-native
N-dimensional arrays

Maintenance and Evolution of Zarr

Josh Moore

Zarr Steering Council
@notjustmoore



@zarr_dev

slides:
bit.ly/zarr-evolution-scipy-2023



Sanket Verma

Zarr Community Manager
@MSanKeys963

Josh Moore

Zarr Steering Council

@joshmoore@fediscience.org

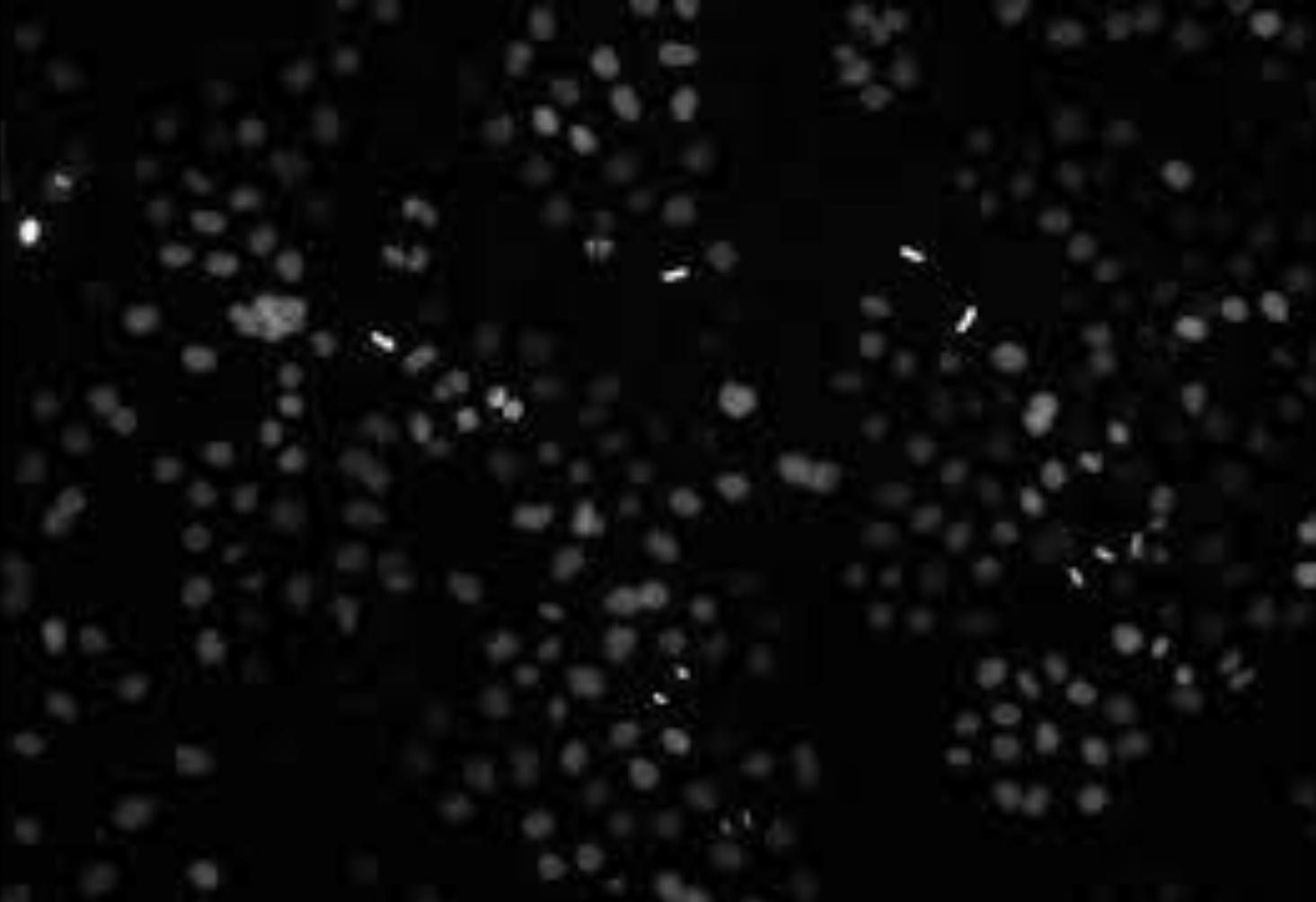


@zarr@fosstodon.org

slides:
bit.ly/zarr-evolution-scipy-2023



So what's my problem?



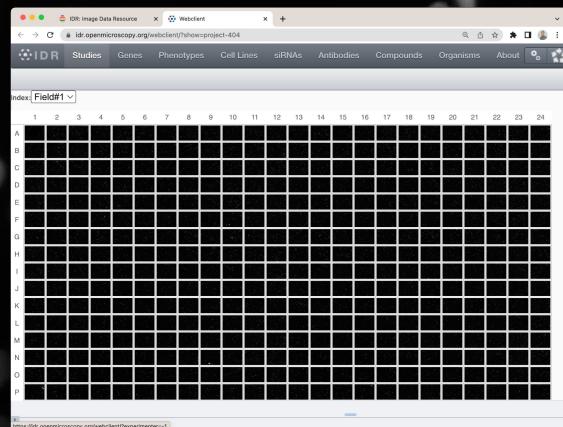
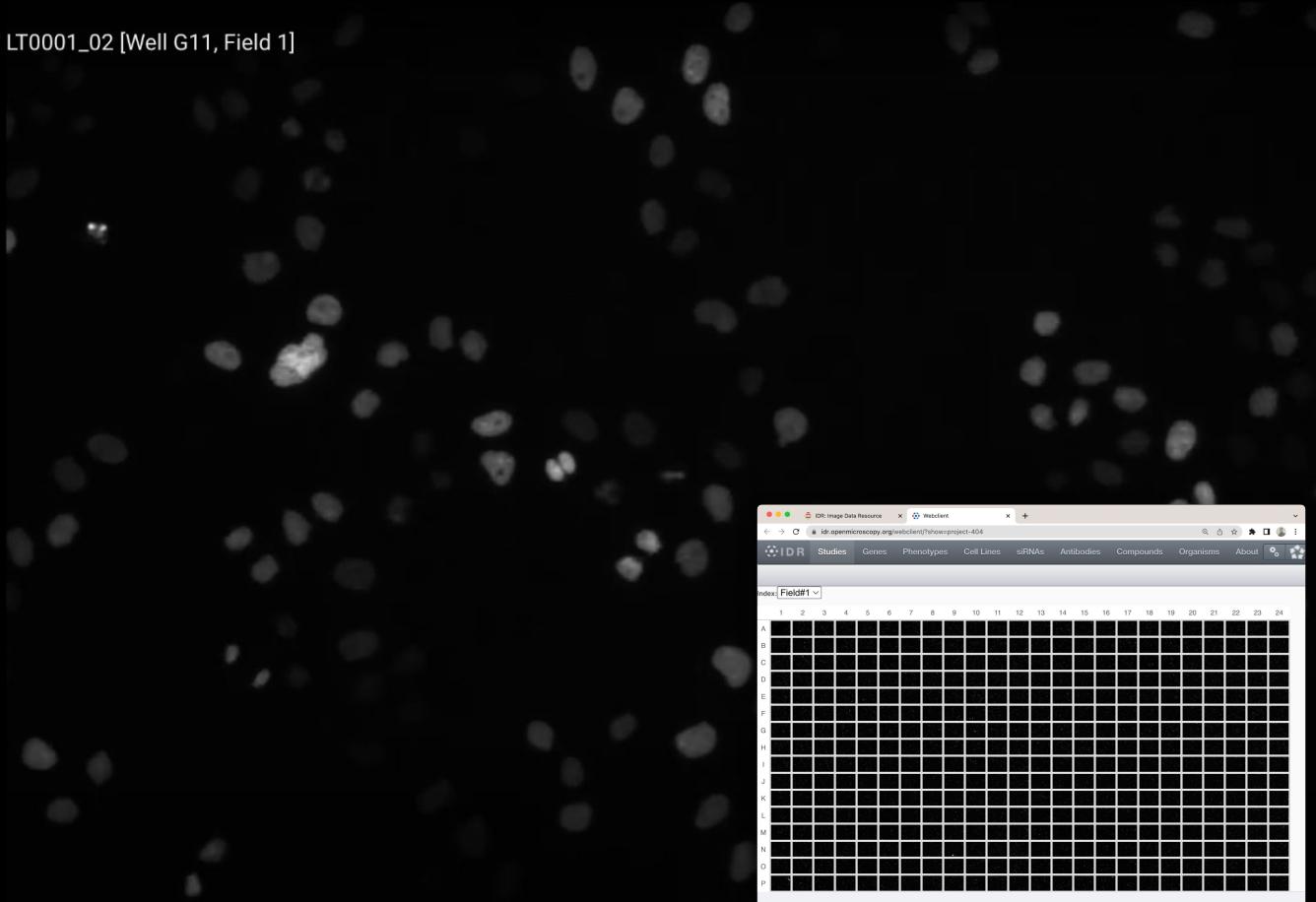
Mitocheck screen (~2003) **25TB, 200K videos** Credit: Neumann *et al.*, IDR (idr0013)

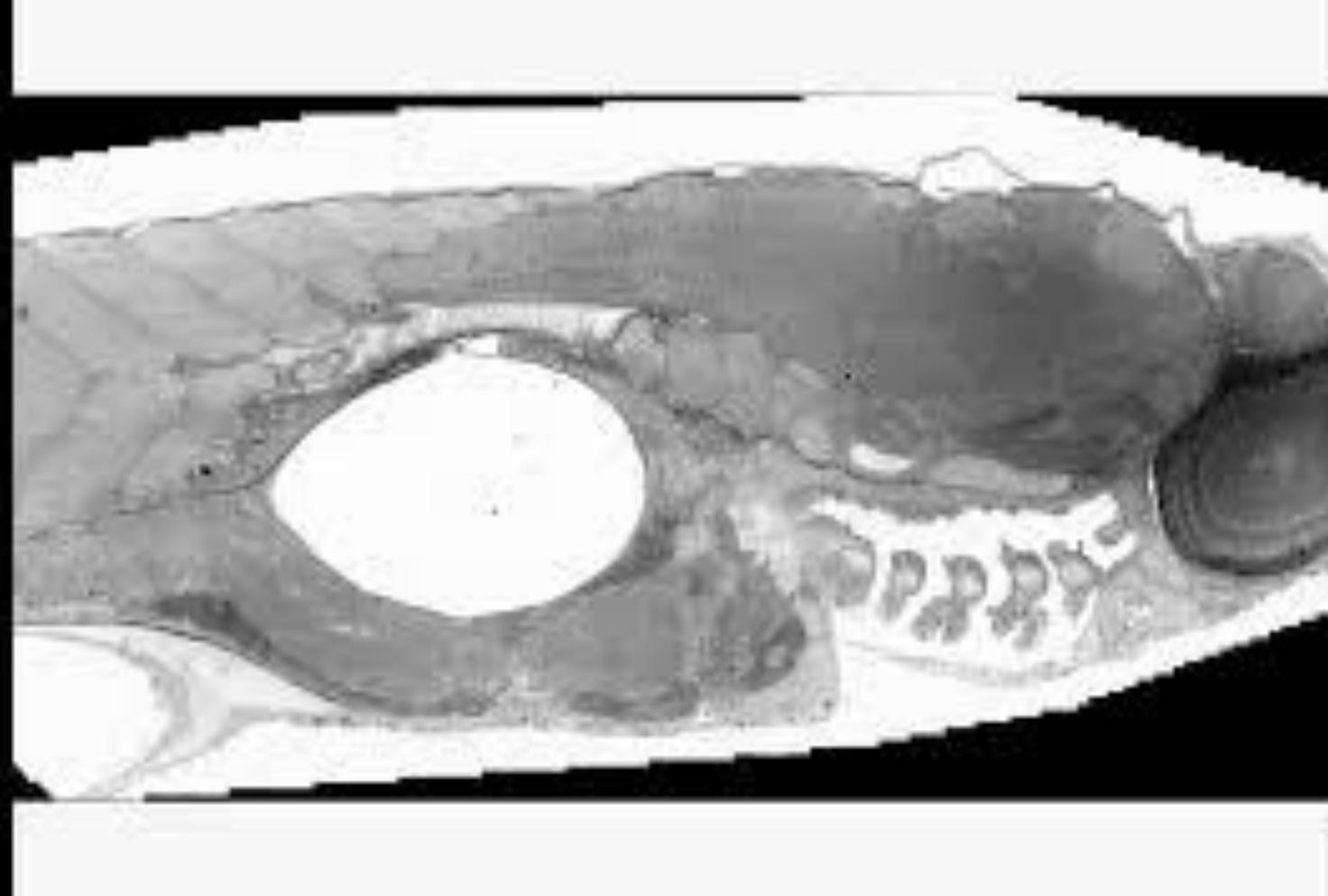


idr0013: LT0001_02 [Well G11, Field 1]



Copy link

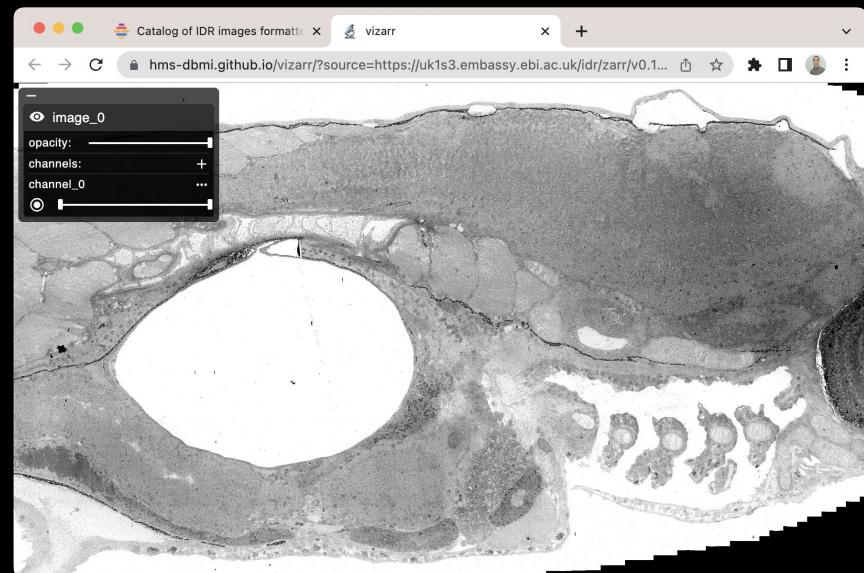




Zebrafish EM (~2012) **500 GB, 1 frame** Credit: Faas *et al.*, IDR (idr0053)

Catalog of IDR images formatted for OME-NGFF

Sample ID	Image Preview	File Name	Dimensions	Pixel Type	Resolution (nm)	Image Count	Frame Count	Volume (GB)	Format
0.1		1885619.zarr	30 30 571	1 1	30 30	1	1	XYZCT	
0.1		4007801.zarr	2169 2048 988	2 532	2169 2048	2	532	XYZCT	
0.1		4495402.zarr	921600 380928 1	1 1	921600 380928	1	1	XYZCT	
0.1		6001237.zarr	1024 1024 39	4 1	1024 1024	4	1	XYZCT	
0.1		6001238.zarr	1024 1024 27	4 1	1024 1024	4	1	XYZCT	

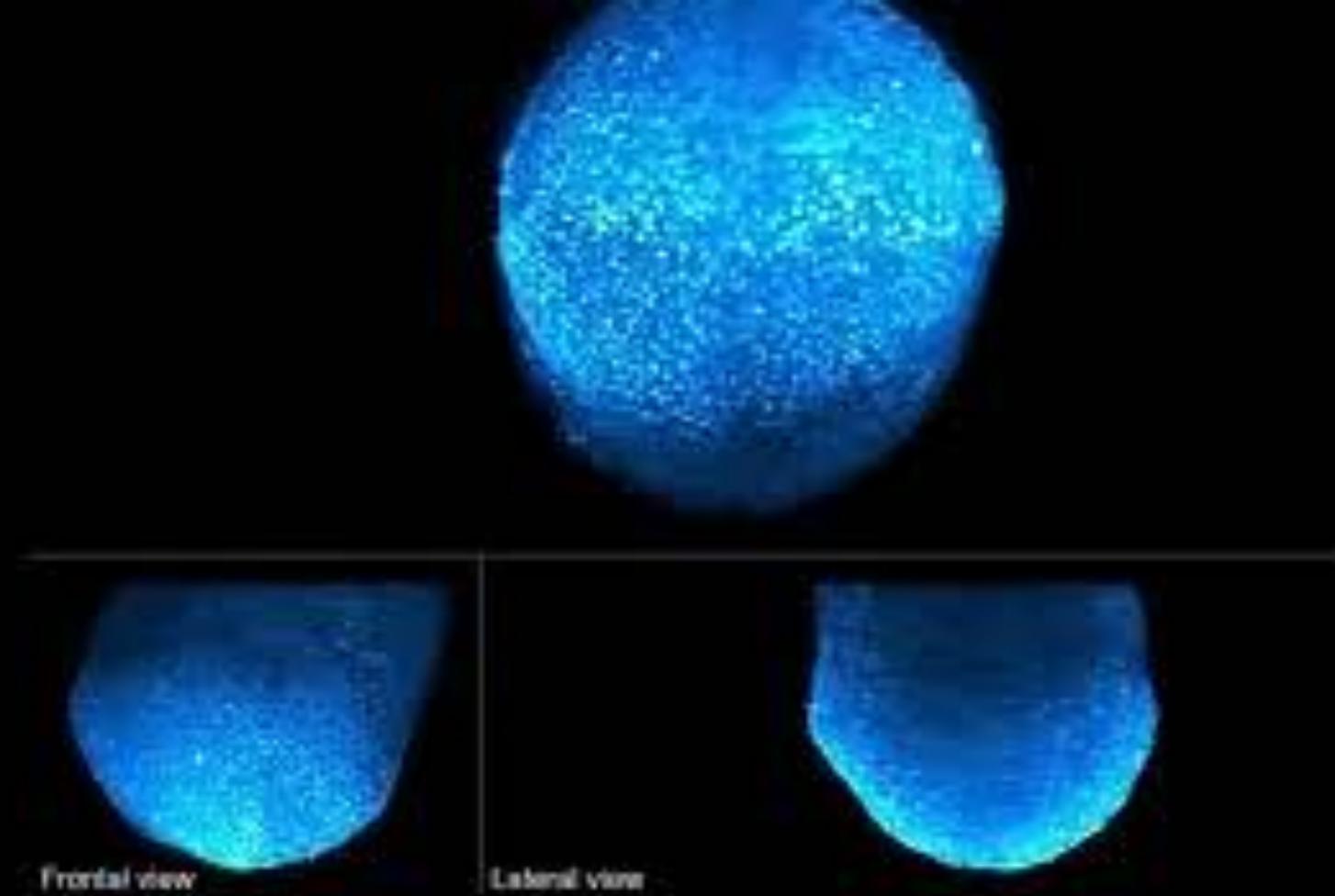


<https://idr.github.io/ome-ngff-samples/>

<https://github.com/hms-dbmi/vizarr>

<https://www.buymeacoffee.com/manzt>





Mouse embryo (~2018) **5 TB, 1 volume** Credit: McDole *et al.* *Cell* (10.1016/j.cell.2018.09.031)



TBD **10 PB, "1 brain"** Credit: <https://knowyourmeme.com/>

<https://planetarycomputer.microsoft.com/catalog?filter=zarr>

Microsoft | Planetary Computer

Data Catalog

The Planetary Computer Data Catalog includes petabytes of environmental monitoring data, in consistent, analysis-ready formats. All of the datasets below can be accessed via Azure Blob Storage, and can be used by developers whether you're working within or outside of our Planetary Computer Hub.

[zarr](#)

Datasets matching "zarr"

- ERAS - PDS**
A comprehensive reanalysis, which assimilates as many observations as possible in the upper air and near surface.
- CMIP5 | ECAWRF | Precipitation | Temperature | Reanalysis | Weather**
- CIL Global DOWNSCALED Projections for Climate Impacts Research (CC-0-1.0)**
Climate Impact Lab Global DOWNSCALED Projections for Climate Impacts Research (CC-0-1.0)
- CMIP6 | Climate Impact Lab | Rhodium Group | Precipitation | Temperature**
- CIL Global DOWNSCALED Projections for Climate Impacts Research (CC-BY-4.0)**
Climate Impact Lab Global DOWNSCALED Projections for Climate Impacts Research (CC-BY-4.0)
- CMIP6 | Climate Impact Lab | Rhodium Group | Precipitation | Temperature**
- Daymet Annual Puerto Rico**

Earth

<https://vortexfd.com/blog/zarr-the-new-format-to-store-4d-wind-flow-blocks>

VORTEX

ZARR FILES

The use of zarr files to store and study 4D wind flow BLOCKS

Marta Gil-Bardaji

Results

netCDF vs Zarr Format

netCDF is a standard for storing and sharing scientific data, and is the most common format for scientific data. Zarr files are a compressed, hierarchical, multi-dimensional array format designed to store large amounts of data in a compressed, efficient, and flexible way. Zarr files are designed to be used in both formats, and Zarr files can be converted to netCDF files.

Current atmospheric modeling is technically capable of generating historical wind data time series over a region for several heights. This 4-dimensional grid-based time-series dataset is what we internally name BLOCKS at Vortex and will allow wind engineers to study the wind flow in great detail and use it as input for complex calibration, wakes, and...

Wind

<https://bgc-jena.mpg.de/en/bgc/gallery>

LOCAL CLIMICS

Seasfire Project: Seafire is a 3D climate model for sea ice and sea surface temperature. It is a high-resolution, multi-dimensional array dataset that is designed to be used in both formats, and Zarr files can be converted to netCDF files.

Seafire Project: Chunking one variable: <https://zenodo.org/record/7108392>

© Lazaro Alonso

Seasfire Project: Seafire is a 3D climate model for sea ice and sea surface temperature. It is a high-resolution, multi-dimensional array dataset that is designed to be used in both formats, and Zarr files can be converted to netCDF files.

Seafire Project: Seafire is a 3D climate model for sea ice and sea surface temperature. It is a high-resolution, multi-dimensional array dataset that is designed to be used in both formats, and Zarr files can be converted to netCDF files.

Fire

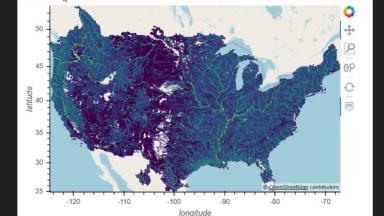
<https://discourse.pangeo.io/t/the-national-water-model-reanalysis-zarr-dataset-on-aws/18>

PANGEO

The National Water Model Reanalysis Zarr dataset on AWS

 rsignell May 21

The National Water Model Reanalysis v2.0 is a 26 year simulation of 2.7 million rivers in the US at hourly intervals. The data was delivered as part of the NOAA Big Data Program to AWS as 227,000+ hourly NetCDF files.



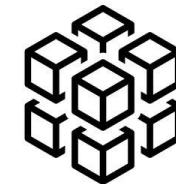
I downloaded (!) and then converted the streamflow files from the reanalysis to a single Zarr dataset with chunks that had a dimension of 100 in the time dimension to facilitate the extraction of time series data. I used `rechunker` 4, and to deal with potential input data problems, I looped through the data in month-long chunks, writing and then appending to Zarr at the end of every month. This way I could correct issues with the input data (missing data and bad time star 1/8

(Water, etc.)

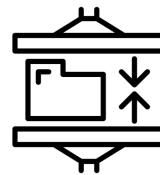
Benefits of Zarr



Distributed & Cloud Storage



Chunked Storage

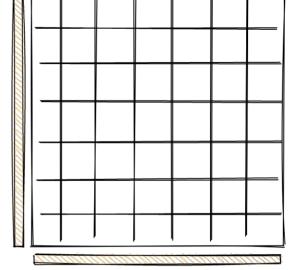
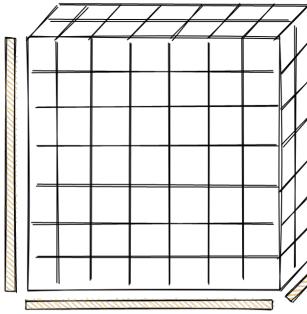


Built-In Compression

How Zarr works?

How Zarr works?

Arrays are container of items of the same data-type & size (in bits). The number of dimensions and items in container are described by the shape.

		
shape	(7,)	(7,7)
# dimensions	1D	2D
# items	7	$7 * 7$

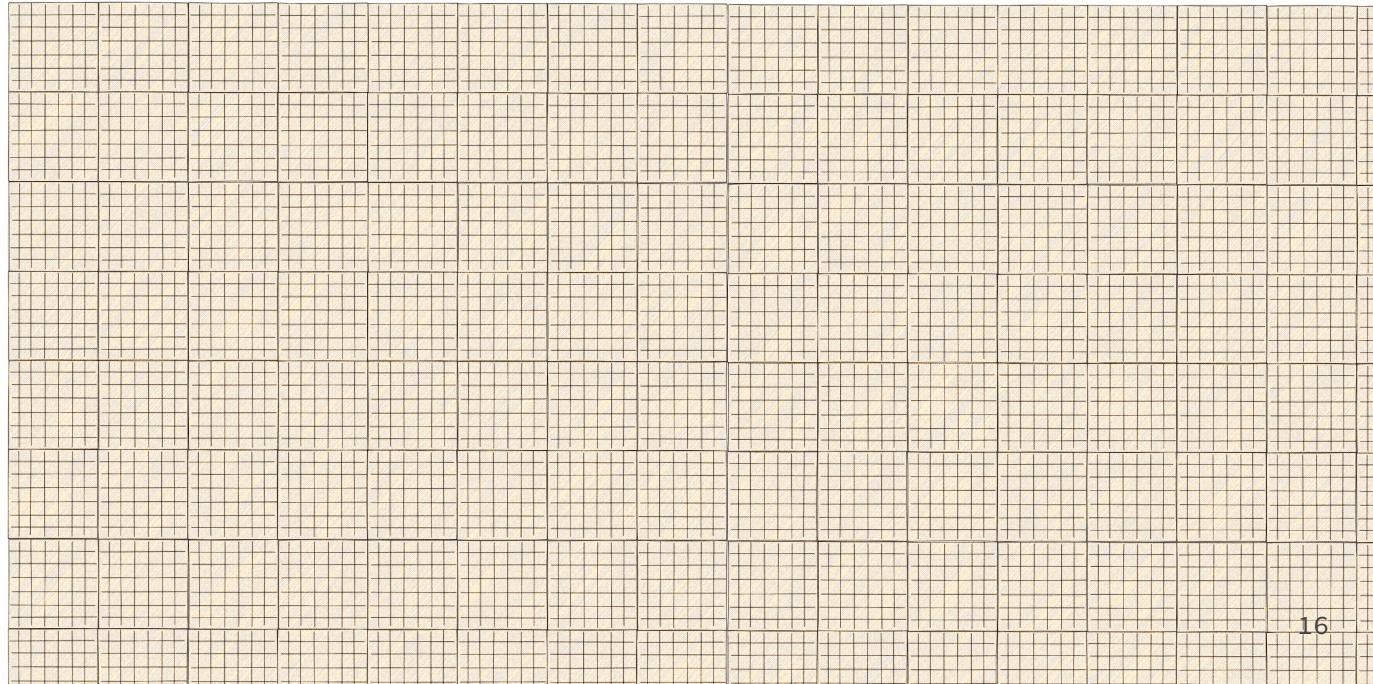
How Zarr works?

So take the arrays you know and love?



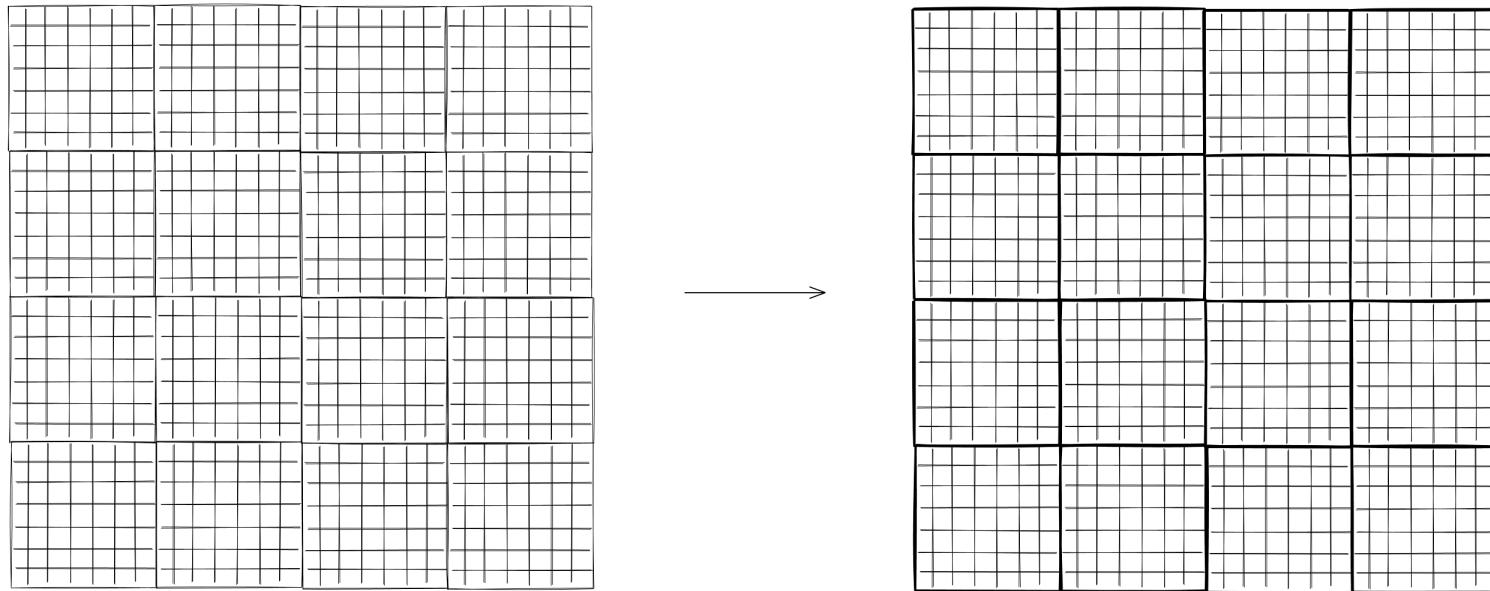
How Zarr works?

What if the data is too big to fit in memory?



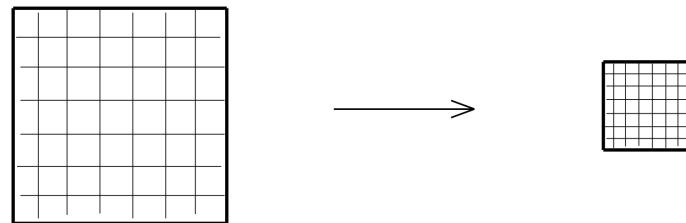
How Zarr works?

Divide array into chunks (Chunking)



How Zarr works?

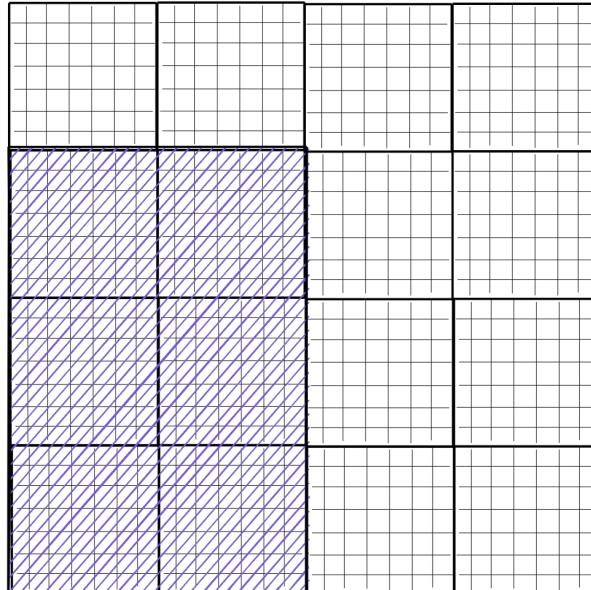
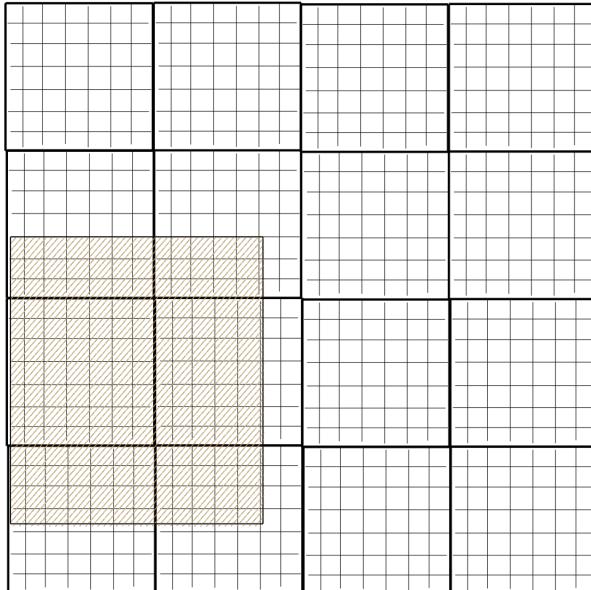
Compress each chunk



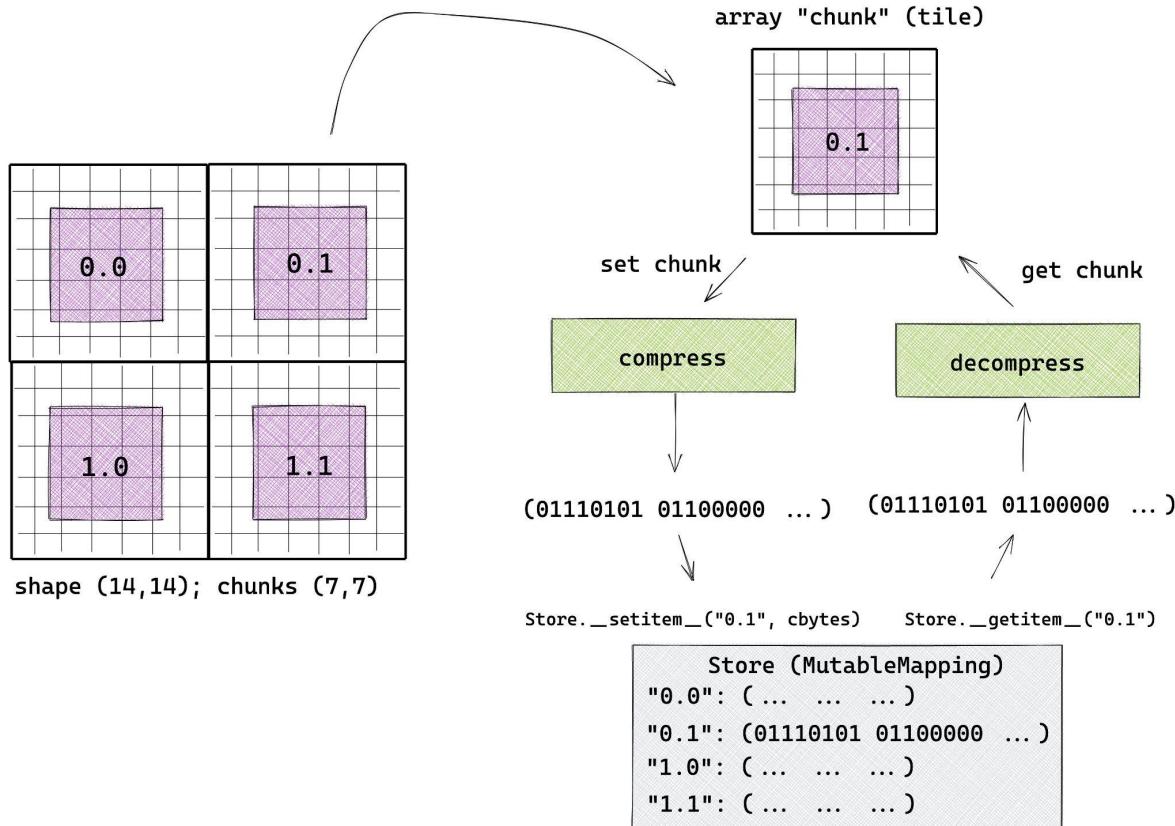
Over 20 supported compressors (BLOSC, Zstd, Zlib etc)

How Zarr works?

Retrieve chunks only when needed

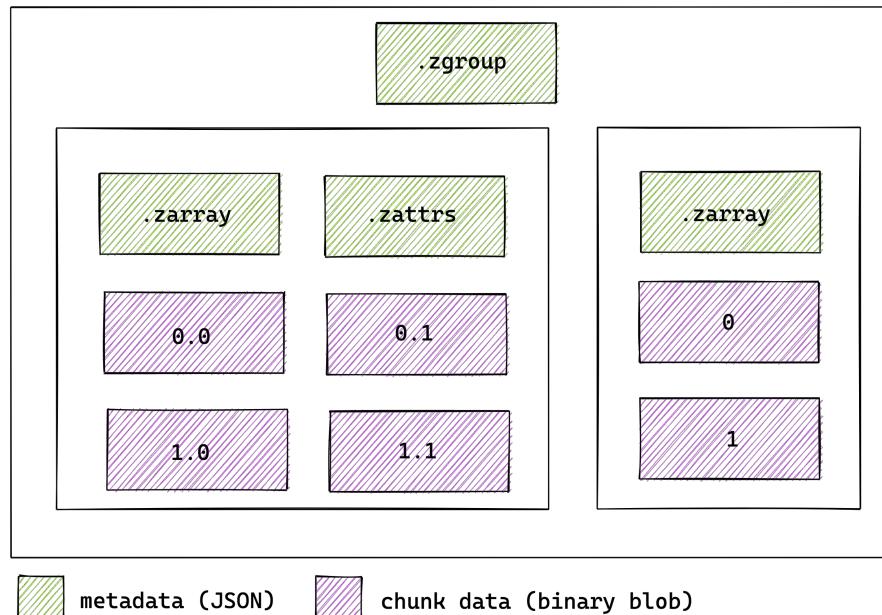


How Zarr works?



How Zarr works?

Multiple arrays can be organised in hierarchies of groups



da.from_array & xr.open_zarr

Integrated methods through the PyData ecosystem

```
import dask.array as da
import zarr

# set up input
store = ... # some Zarr store
root = zarr.group(store)
big = root['big']
big = da.from_array(big)

# define computation
output = big * 42 + ...

# if output is small, compute to memory
o = output.compute()

# if output is big, compute and write directly to Zarr
da.to_zarr(output, store, component='output')
```

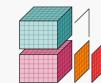


```
In [8]: if pangeo=='ESIP-AWS-S3':
    import s3fs
    fs = s3fs.S3FileSystem(anon=True)
    map = s3fs.S3Map('esip-pangeo/pangeo/adcirc/ike', s3=fs)

In [9]: ds = xr.open_zarr(map)

In [10]: ds['zeta']

Out[10]: <xarray.DataArray 'zeta' (time: 720, node: 9228245)>
dask.array<shape=(720, 9228245), dtype=float64, chunksize=(10, 141973)>
Coordinates:
  * time      (time) datetime64[ns] 2008-09-05T12:00:00 ... 2008-09-10T11:50:00
    x        (node) float64 dask.array<shape=(9228245,), chunksize=(141973,)>
    y        (node) float64 dask.array<shape=(9228245,), chunksize=(141973,)>
Dimensions without coordinates: node
Attributes:
  location:      node
  long_name:    water surface elevation above geoid
  mesh:          adcirc_mesh
  standard_name: sea_surface_height_above_geoid
  units:          m
```



Credit: Alistair Miles

<https://zarr.dev/slides/scipy-2019.html#/8/1>

Credit: Richard Signell

<https://www.mdpi.com/2077-1312/7/4/110>

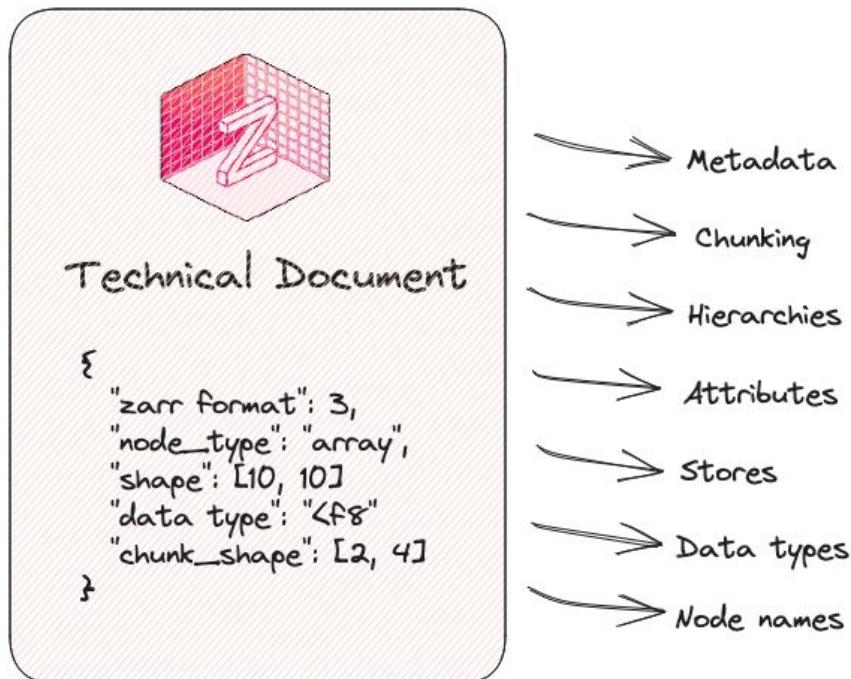
Zarr Specification

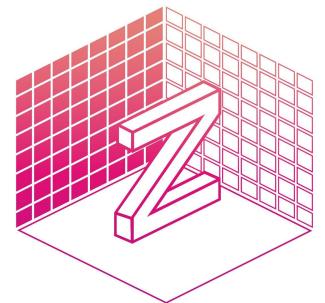
V1 → V2 → V3



<https://zarr-specs.readthedocs.io/>

Zarr Specification





Zarr



Timeline

First commit

2015

2019

SciPy (Alistair Miles)
Work on V3 begins

2021

<https://zarr.dev>
New Logo

CZI Grant #1
NumFOCUS
Steering Council

2020

2022

CZI Grant #2
Community Manager
Enhancement Process

2023

ZEPs

SciPy!

Zarr Community

ZARR OSS



AND SPECIFICATION

USERS & DEVS



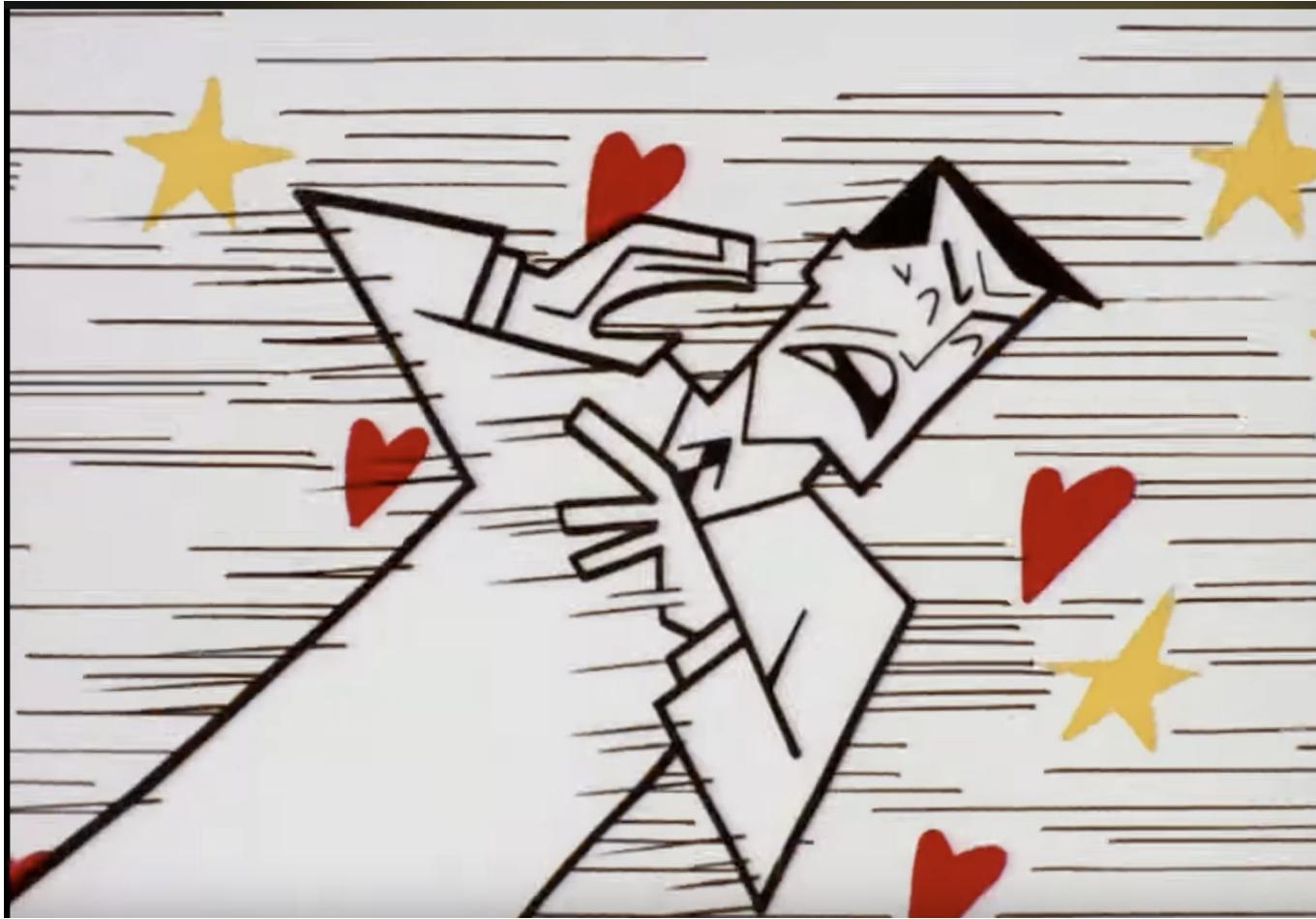
CONTRIBUTORS & MAINTAINERS





EASY TO USE

HACKABLE AND AGNOSTIC



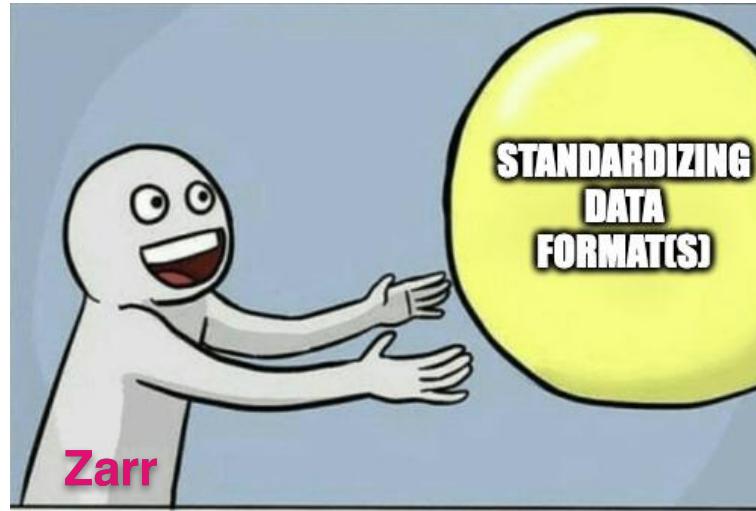






We have a large and diverse active community!

But... 



We needed a
structured way to
solicit and ***process***
the feedback!

zarr.dev/zeps



Zarr

Search ZEP

Zarr Homepage

active ZEPs / ZEP0000

home

active ZEPs

ZEP0000

draft ZEPs

template

implementations council

ZEP meetings

join the community

ZEP 0 — Purpose and process

Author: Sanket Verma (@MSanKeys963), Zarr

Email address: svsanketverma5@gmail.com

Status: Active

Type: Process

Created: 2022-14-03

Discussion: <https://github.com/zarr-developers/governance/pull/16>

What is ZEP?

ZEP stands for Zarr Enhancement Proposal. A ZEP is a design document providing information to the Zarr community, describing a modification or enhancement of the Zarr specification, a new feature for its processes or environment. The ZEP should provide specific proposed changes to the Zarr specification and a narrative rationale for the specification changes.

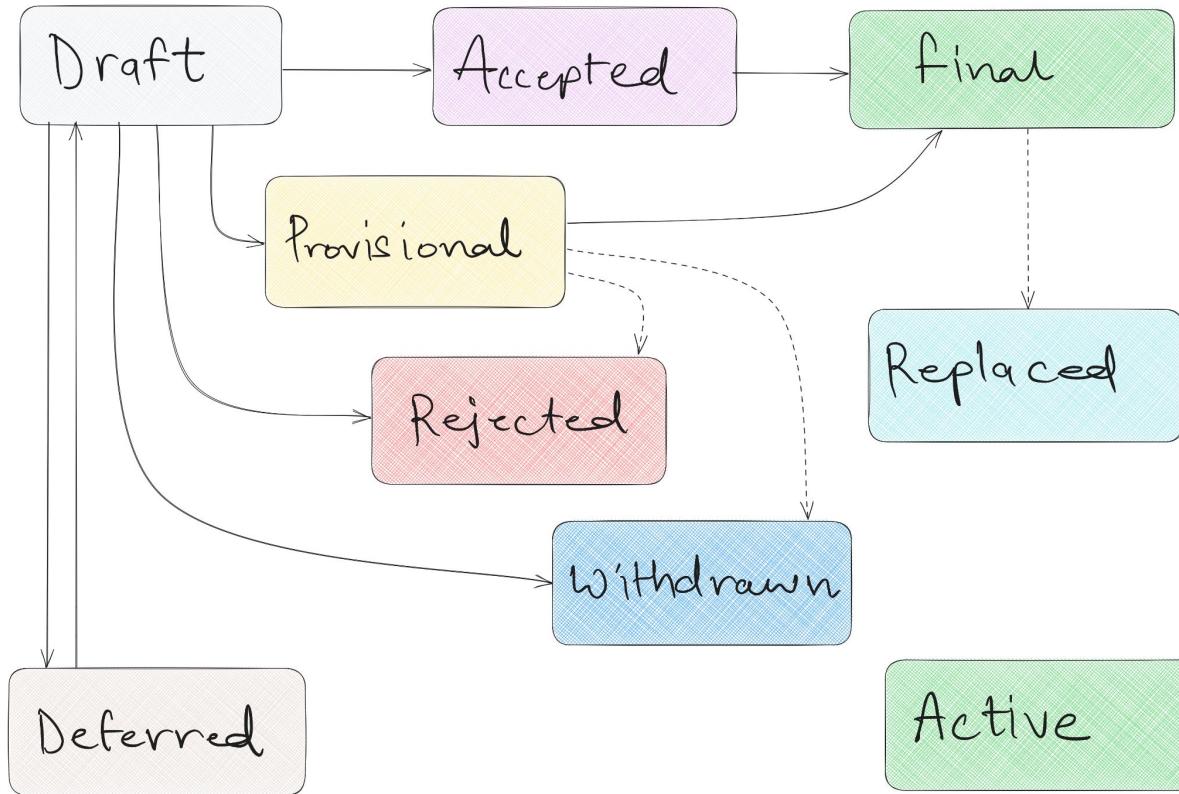
We intend ZEPs to be the primary mechanism for evolving the spec, collecting community input on major issues and documenting the design decision that has gone into Zarr. In addition, the ZEP author is responsible for building consensus within the community and documenting dissenting opinions.

Because the ZEPs are maintained as text files in a versioned repository, their revision history is the historical record of the feature proposal.

WHERE:

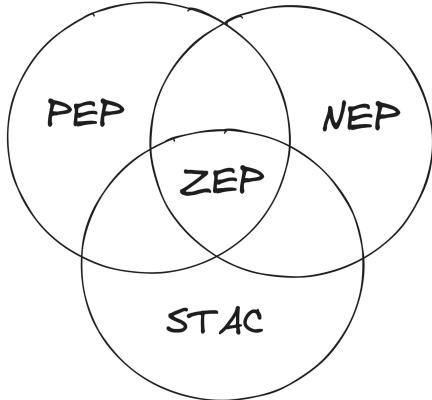
- Developers refer to contributors and maintainers of the project
- User(s) refers to an individual or group of individuals or the broader community using the project in any way.

ZEP Flowchart ↗↖↔



How we did it?





Lots of Reading



Previous Experience



Understanding the needs of the community

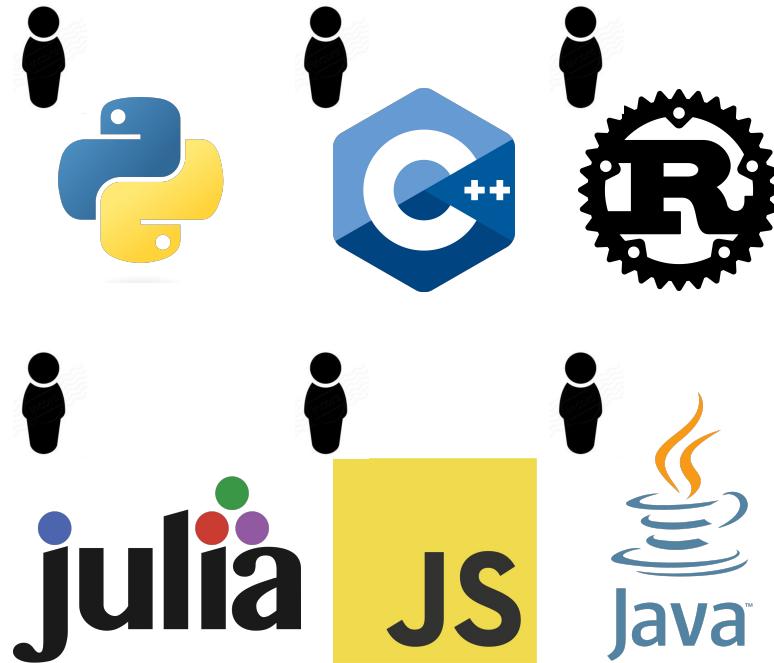
Braindump



ZEP Inception Blog post



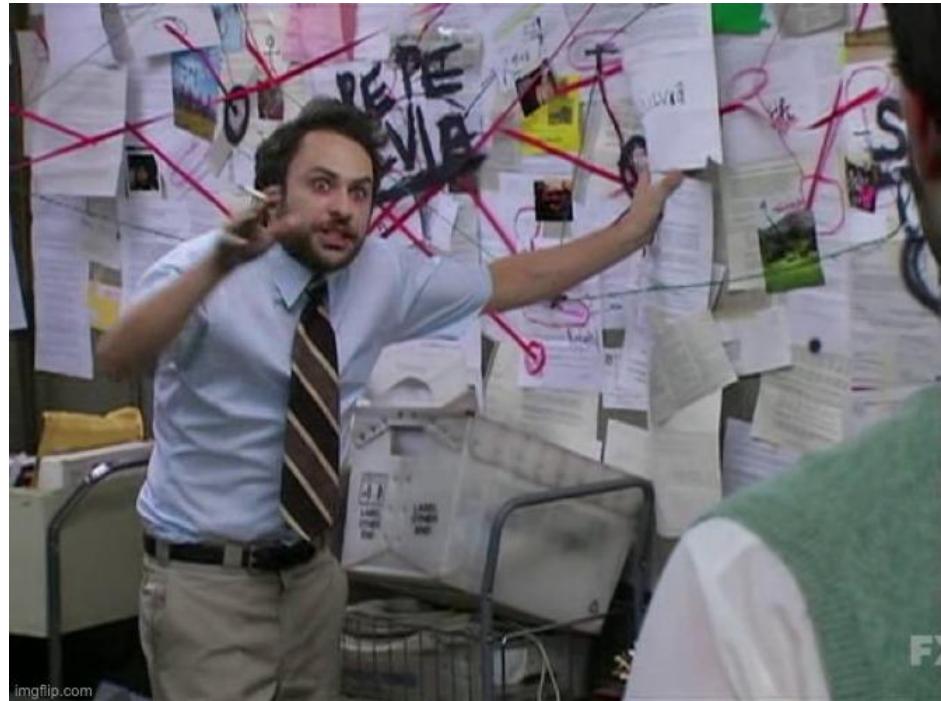
How do we adopt a ZEP?





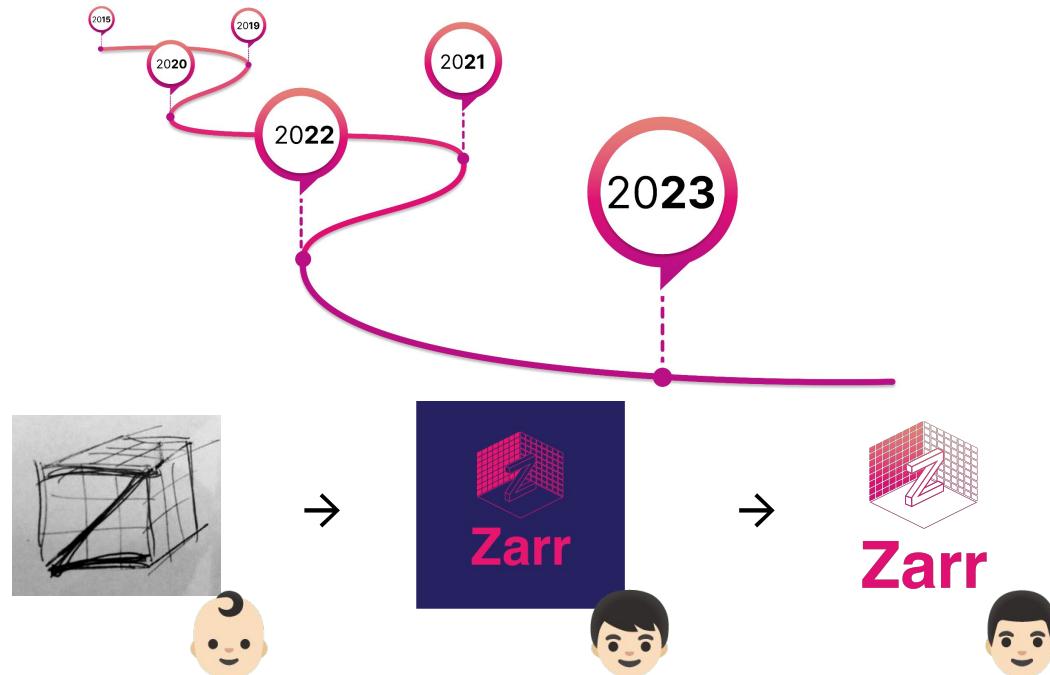


Work in progress!



imgflip.com

Evolution



zarr.dev/implementations



Zarr

chunked, compressed, N-dimensional arrays

Documentation

Contribute

Python Tutorial



Content

ABOUT

Description
Applications
Features
Sponsorship
Videos

SUBPAGES

Adopters
Blog
Community
Conventions
Datasets
Implementations
Office Hours
Slides
Specification
ZEPs

Zarr Implementations

Zarr is a data storage format based on an open-source [specification](#), making implementations across several languages possible. It is used in various domains, including geospatial, bio-imaging, genomics, data science, and HPC. 

Implementations are listed (in alphabetical order) as follows:

C	C++	Java	Javascript	Julia	Python	R	Rust
NetCDF-C	GDAL	JZarr	Zarr.js	Zarr.jl	Zarr-Python	Rarr	Rust-N5
		Tensorstore	N5-Zarr	Zarr.js	Zarrita	Zarr	
	Xtensor-Zarr	NetCDF-Java					
			Z5				

zarr.dev/adopters

arr Adopters

If you're using Zarr in any way and would like to be added on this page, please drop your logo and blurb [here](#).

anks to the amazing community, Zarr is widely adopted and used by these groups. Here are the logos (in alphabetical order):

carbon)plan

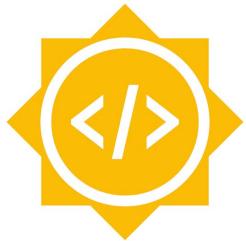
Zarr is used by [CarbonPlan](#) as a storage format for analysis and visualization of climate data.

 COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

 COLUMBIA CLIMATE SCHOOL
LAMONT-DOHERTY EARTH OBSERVATORY



/zarr-developers/{gsoc,outreachy}



Google Summer of Code

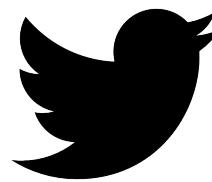
WHAT DID WE LEARN?



Lessons

- Creative ways to foster the community & OSS 
- Establishing trust is critical for standardization 
- Everyone on the same page - difficult but achievable 
- Be considerate to everyone! 

Thank you!



@zarr_dev