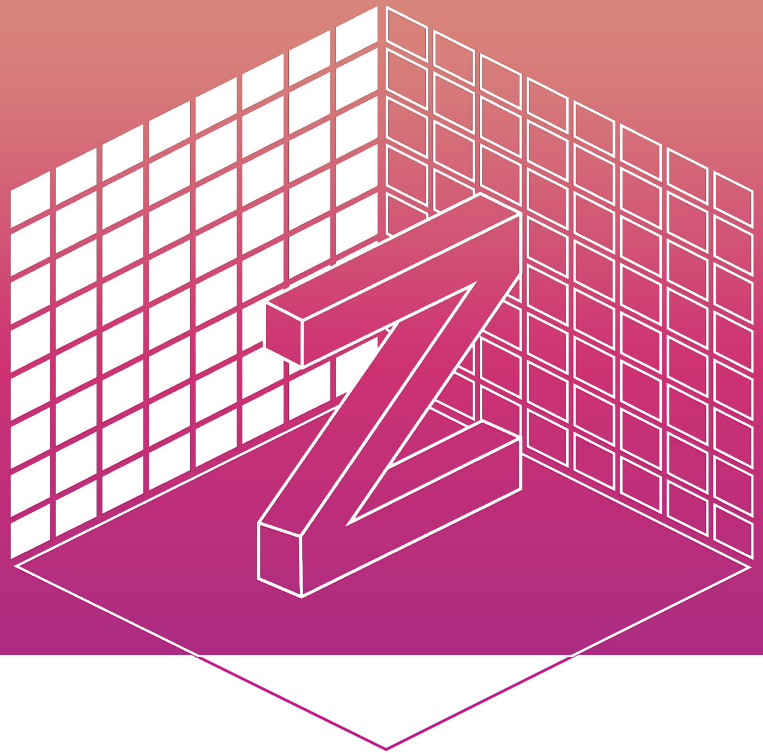


Chunked, Compressed, & Cloud-native
N-dimensional arrays

FOSS United x ILUG-D Meetup
18th May, 2024 @ IFF

Sanket Verma

Community and OSS @ Zarr

@MSanKeys963

Slides

https://bit.ly/zarr_foss_ilugd

GitHub Repository

https://bit.ly/foss_ilugd_more

What I'll be talking about?

→ What is Zarr?

→ How is Zarr different?

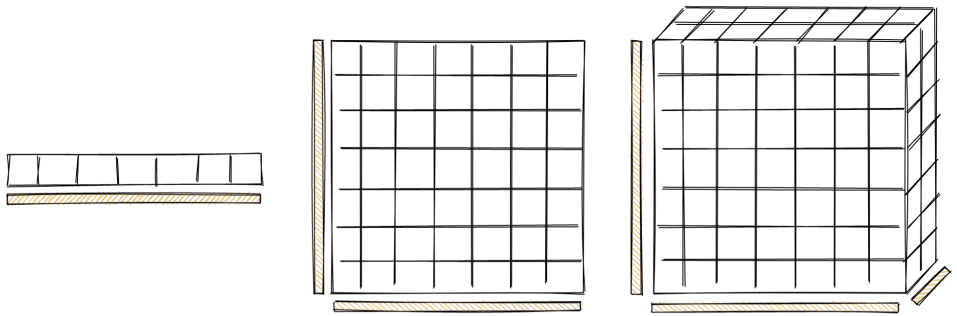
→ What is the Zarr Community and ZEPs?

→ Demo

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)	(2,7,7)
# dimensions	1D	2D	3D
# items	7	7 * 7	7 * 7 * 2

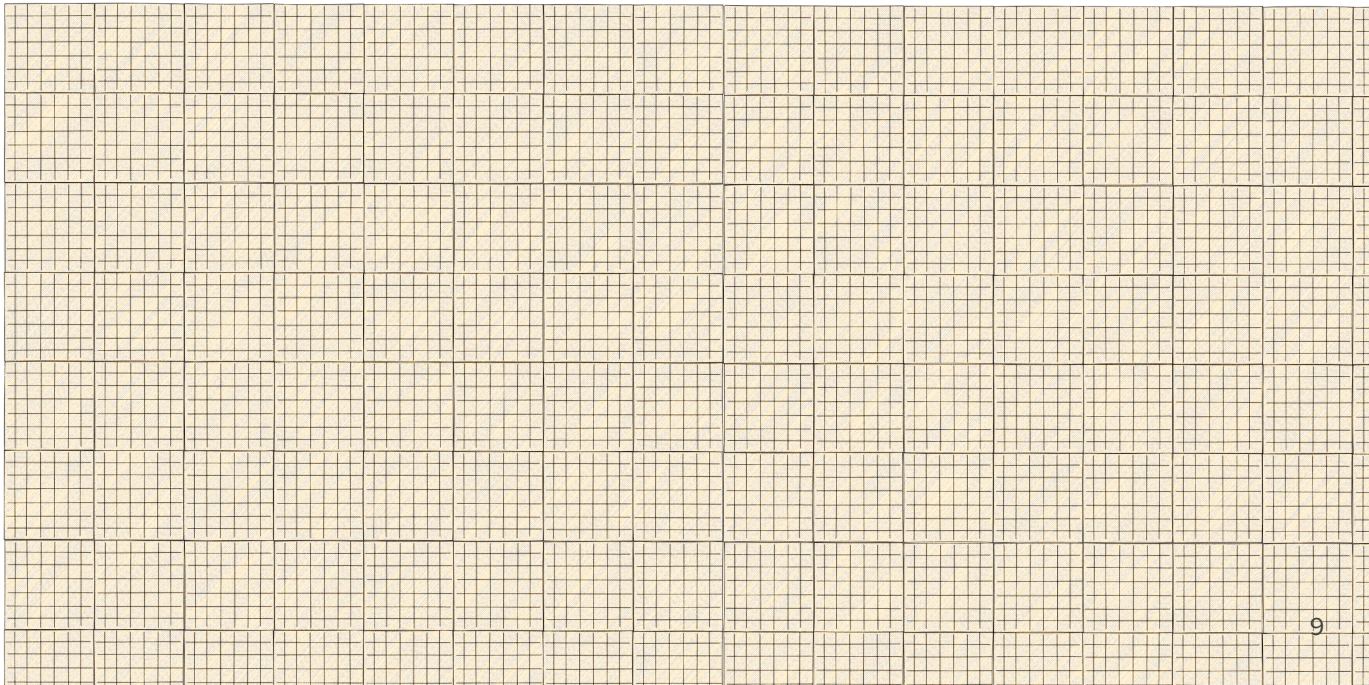
Credit: @trevmanz

How Zarr works?

Credit: @trevmanz

How Zarr works?

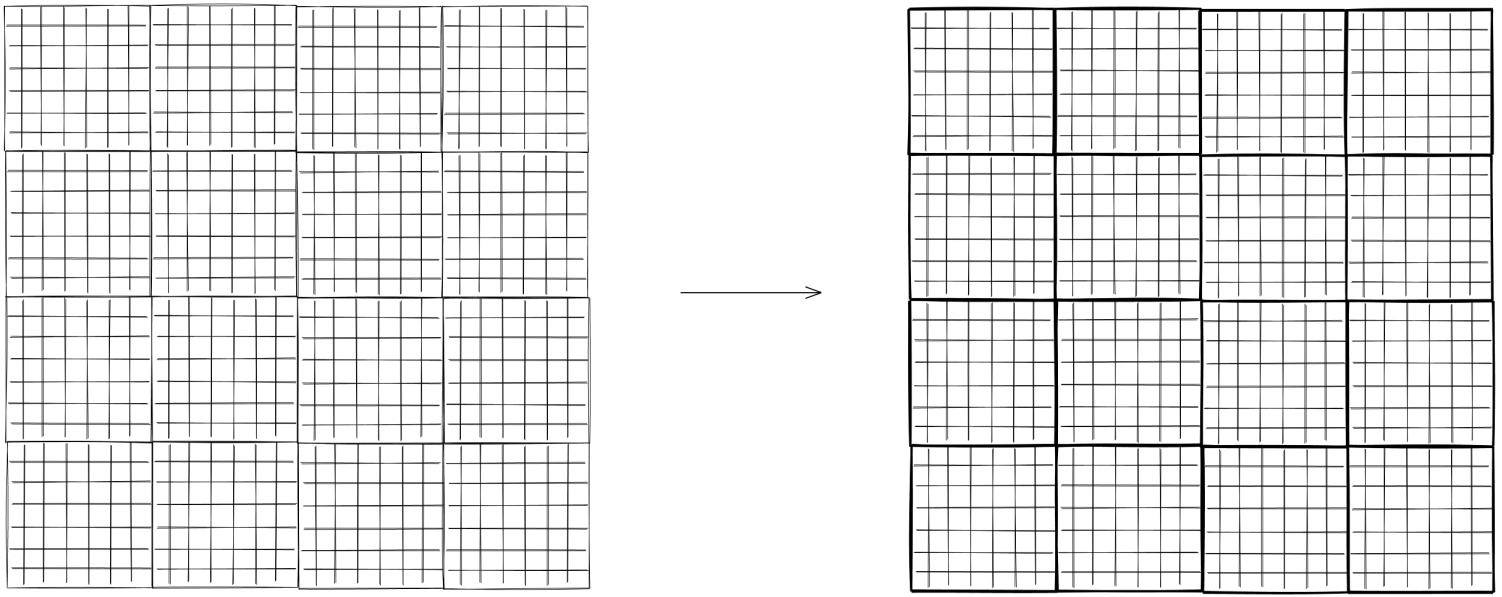
What if the data is too big to fit in memory?



Credit: @trevmanz

How Zarr works?

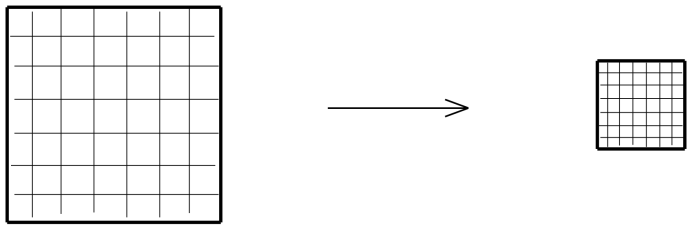
Divide array into chunks (Chunking)



Credit: @trevmanz

How Zarr works?

Compress each chunk

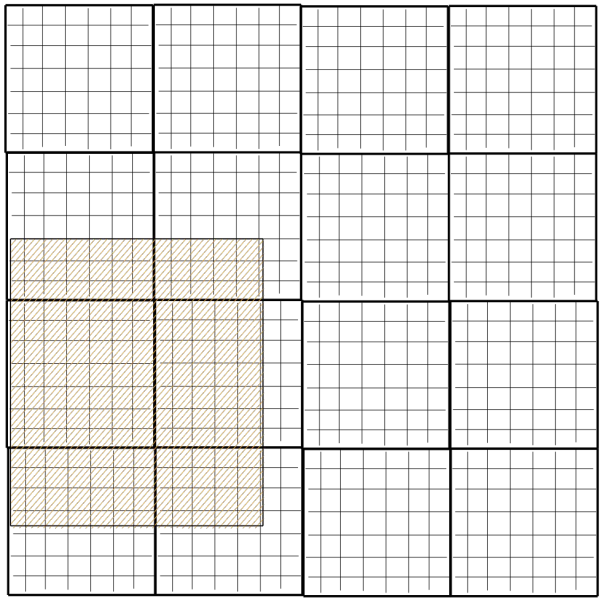


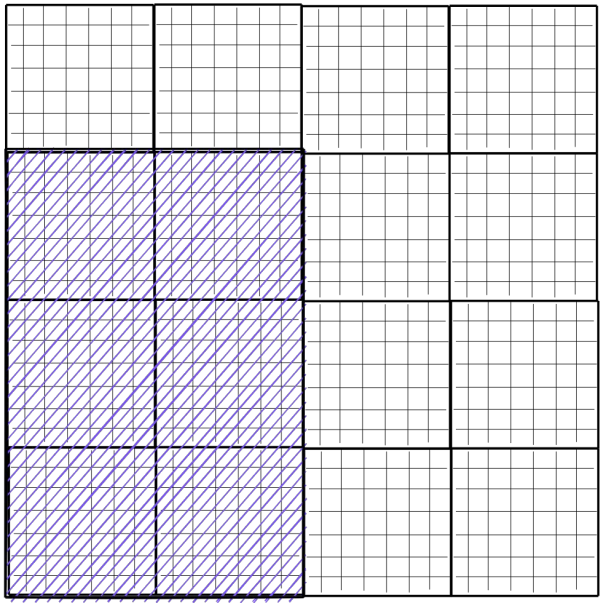
Over 20 supported compressors (BLOSC, Zstd, Zlib etc)

Credit: @trevmanz

How Zarr works?

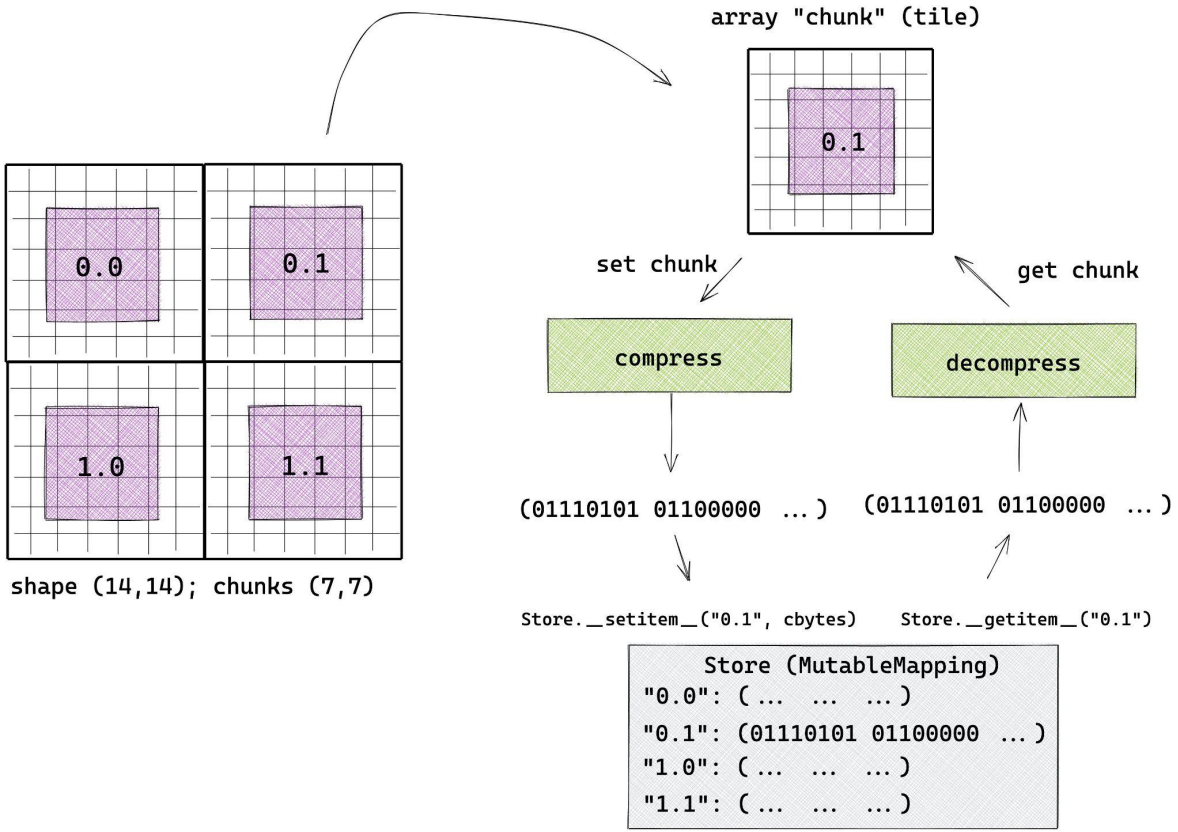
Retrieve chunks only when needed





Credit: @trevmanz

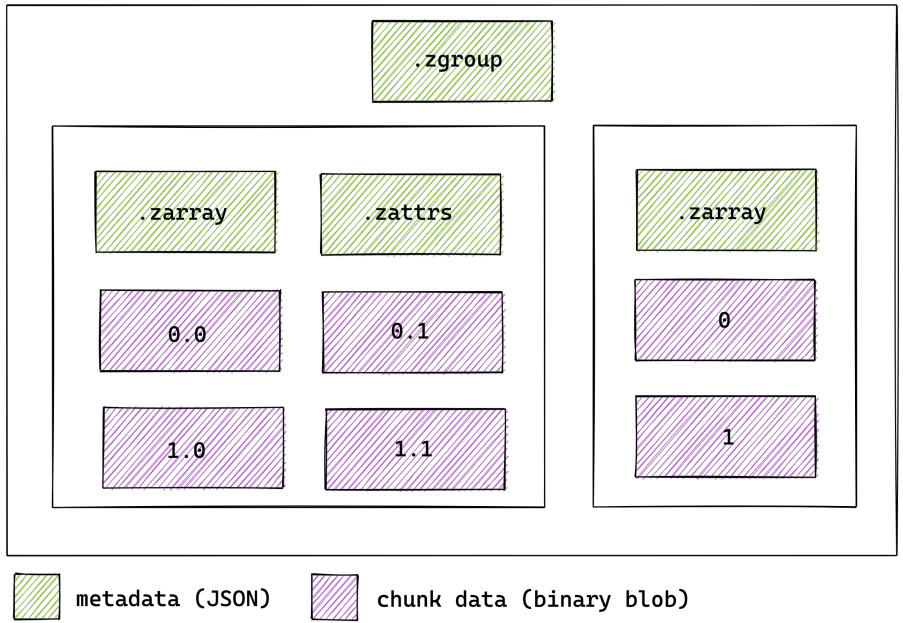
How Zarr works?



Credit: @trevmanz

How Zarr works?

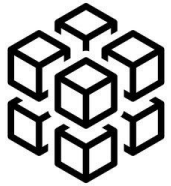
Multiple arrays can be organised in hierarchies of groups



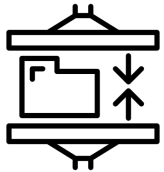
Credit: @trevmanz

Benefits of Zarr

Distributed & Cloud Storage

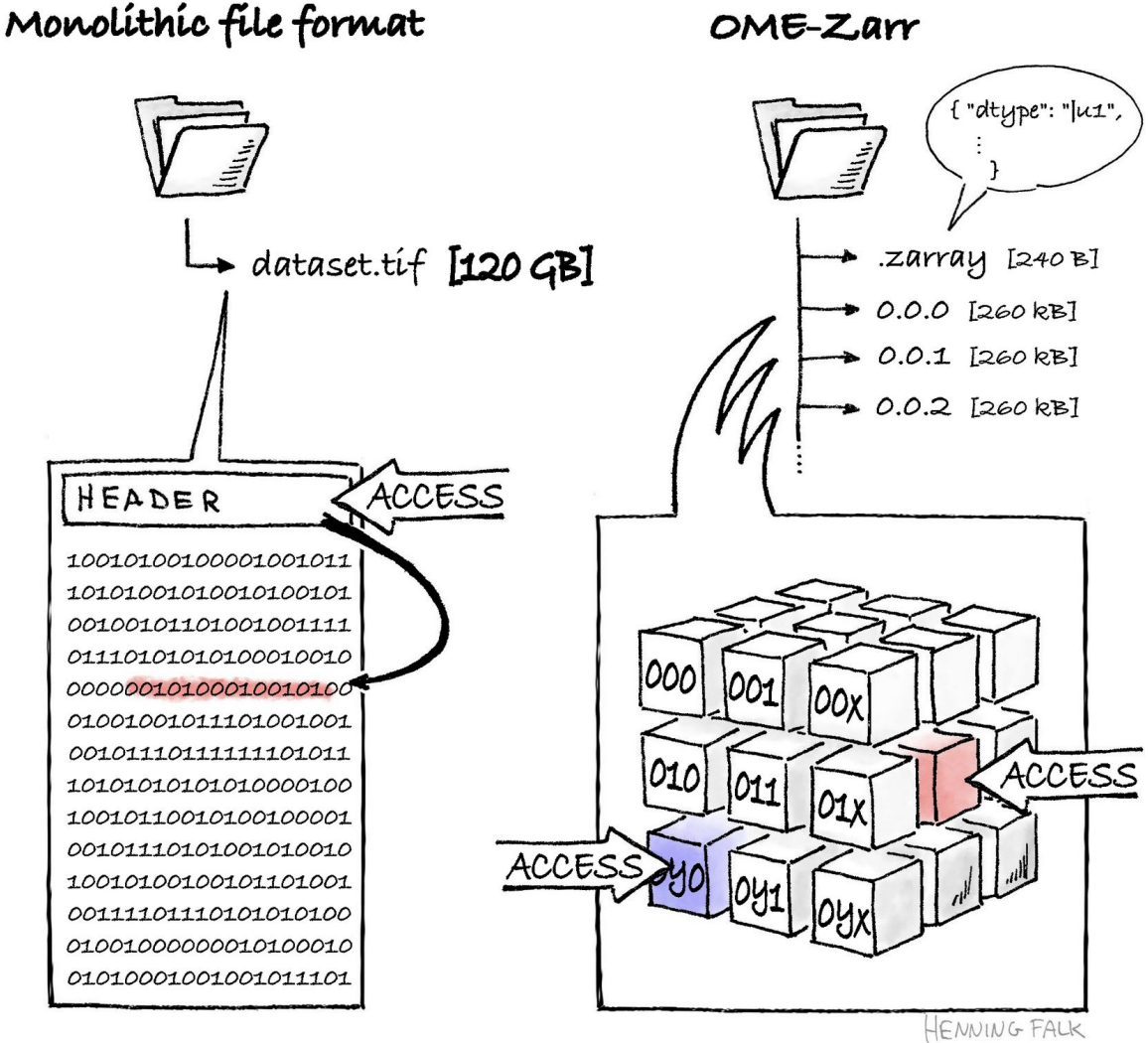


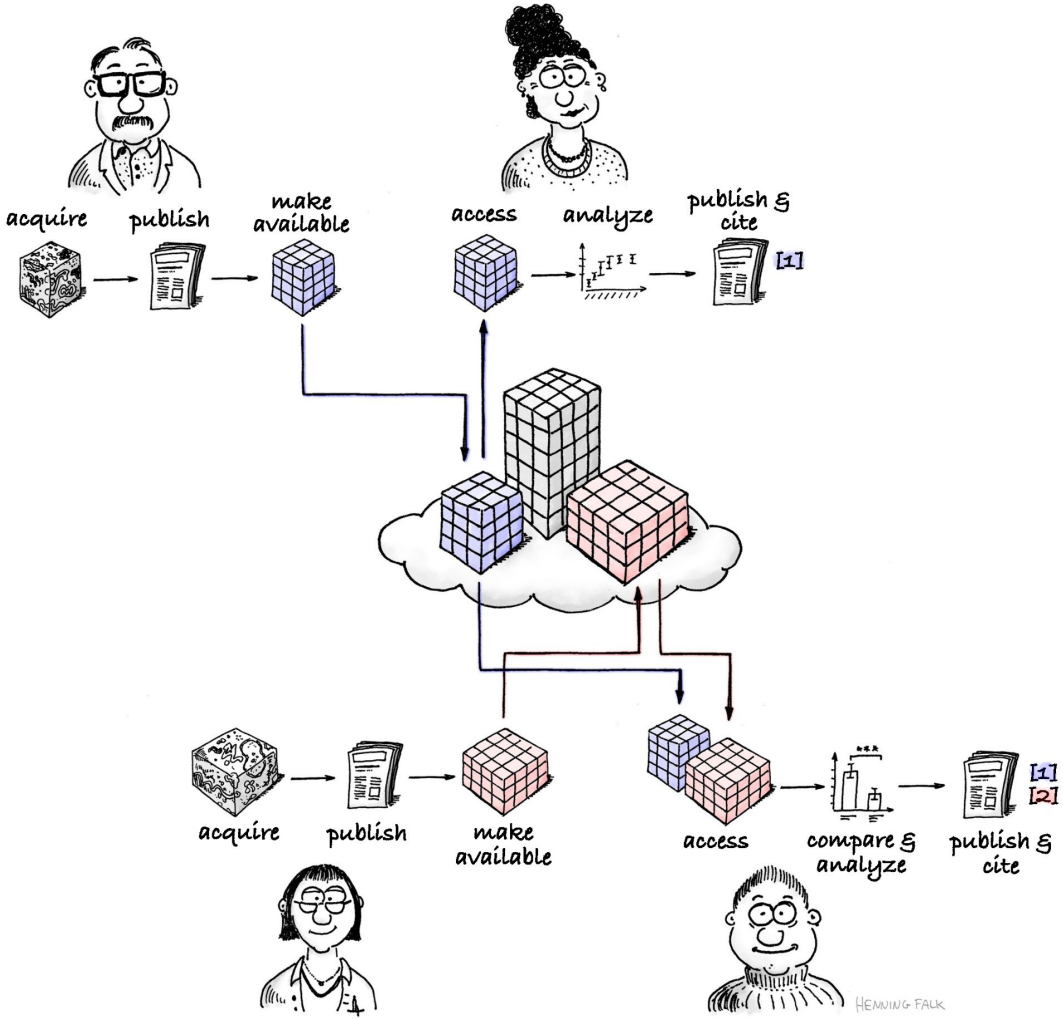
Chunked Storage

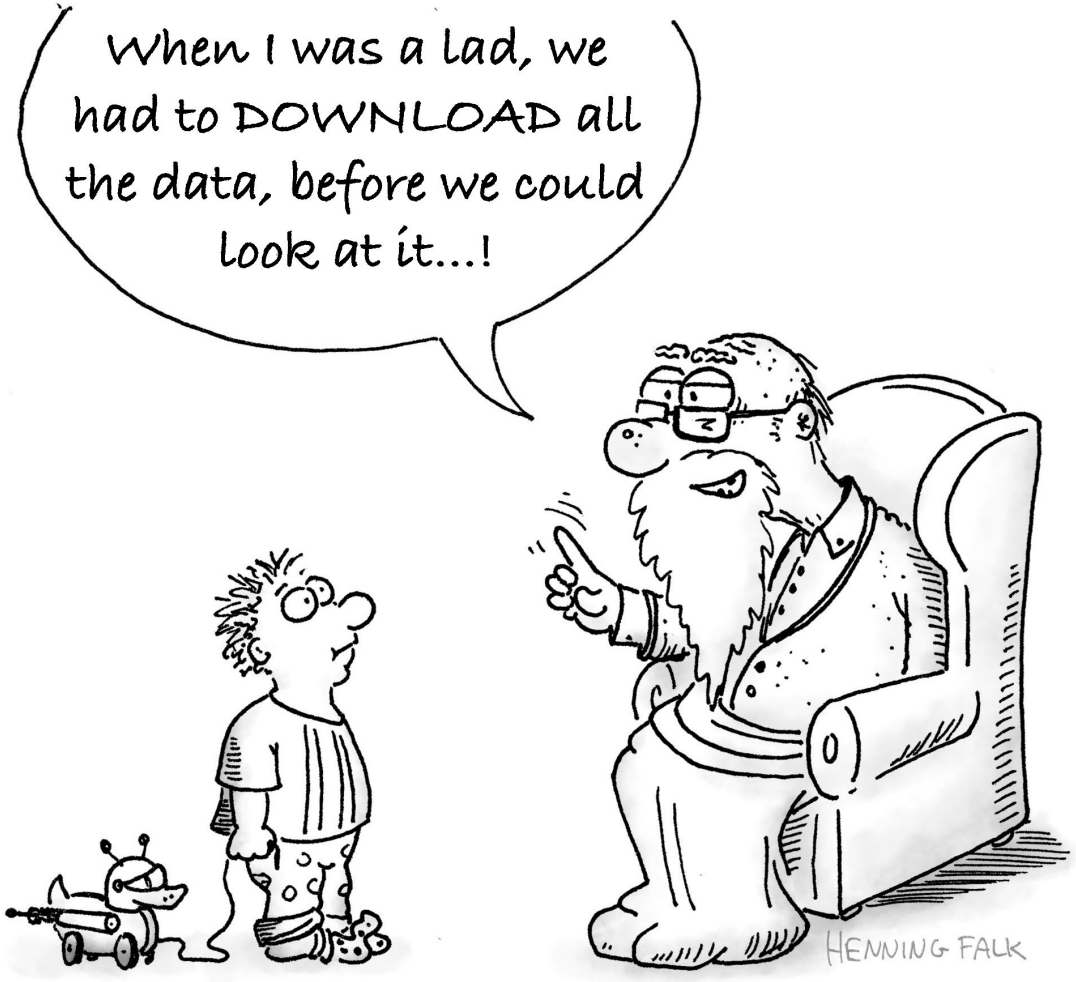


Built-In Compression

How Zarr is different?


Monolithic file format
OME-Zarr
dataset.tif [120 GB]
HEADER
ACCESS
1001010010000100101 1
1010100101001010010 1
0010010110100100111 1
0111010101010001001 0
0000001010001001010 0
0100100101110100100 1
0010111011111110101 1
1010101010101000010 0
1001011001010010000 1
0010111010100101001 0
1001010010010110100 1
0011110111010101010 0
0100100000010100010
0101000100100101110 1
{ "dtype": "|u1",
}
.zarray [240 B]
0.0.0 [260 kB]
0.0.1 [260 kB]
0.0.2 [260 kB]
000
001
00X
010
011
01X
ACCESS
ACCESS
0y0
0y1
0yX
HENNING FALK


acquire
publish
make available
access
analyze
publish & cite
[1]
acquire
publish
make available
access
compare & analyze
publish & cite
[1]
[2]
HENNING FALK


When I was a lad, we had to DOWNLOAD all the data, before we could look at it...!
HENNING FALK

More illustrations

Check here 👉

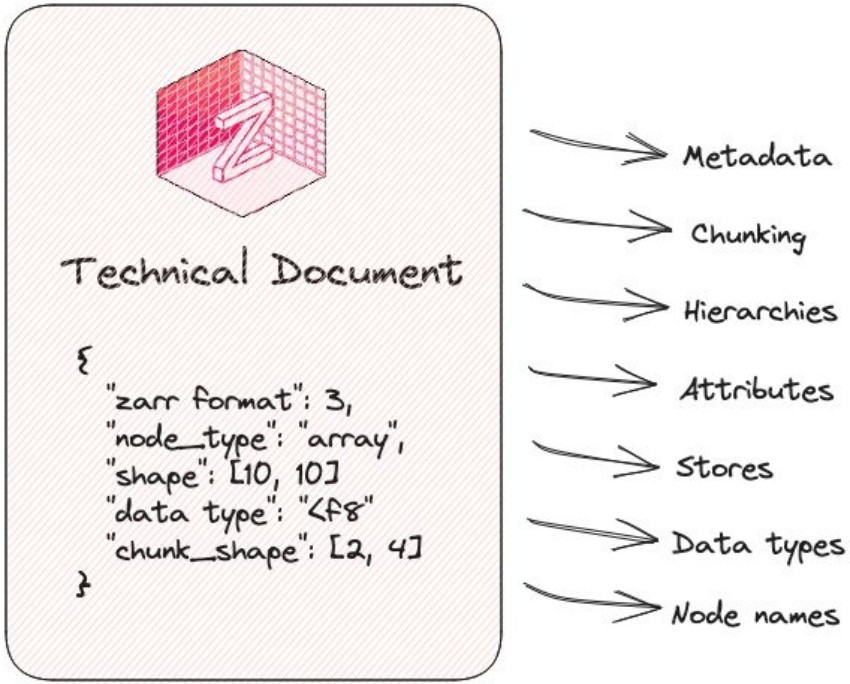
Zarr Specification

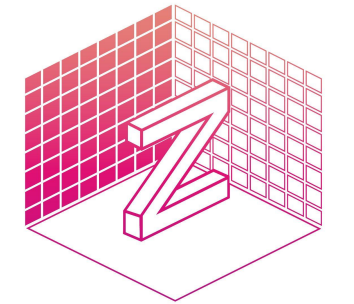
Zarr Specification

V1 → V2 → V3

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

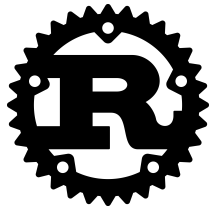
Zarr Specification





Zarr

↔

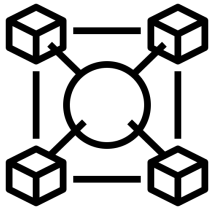


julia

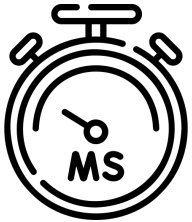
JS

Why did we work on V3 Specification?

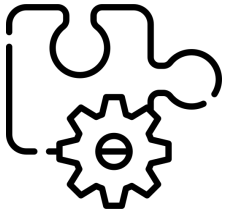
The Motivation



Interoperability



High-latency storage

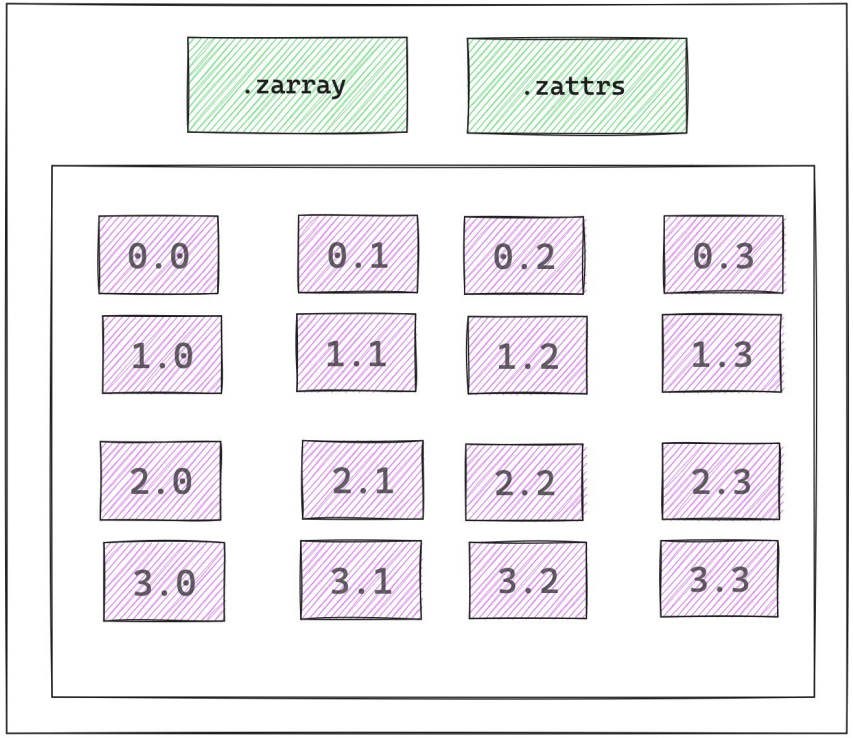


Extensibility

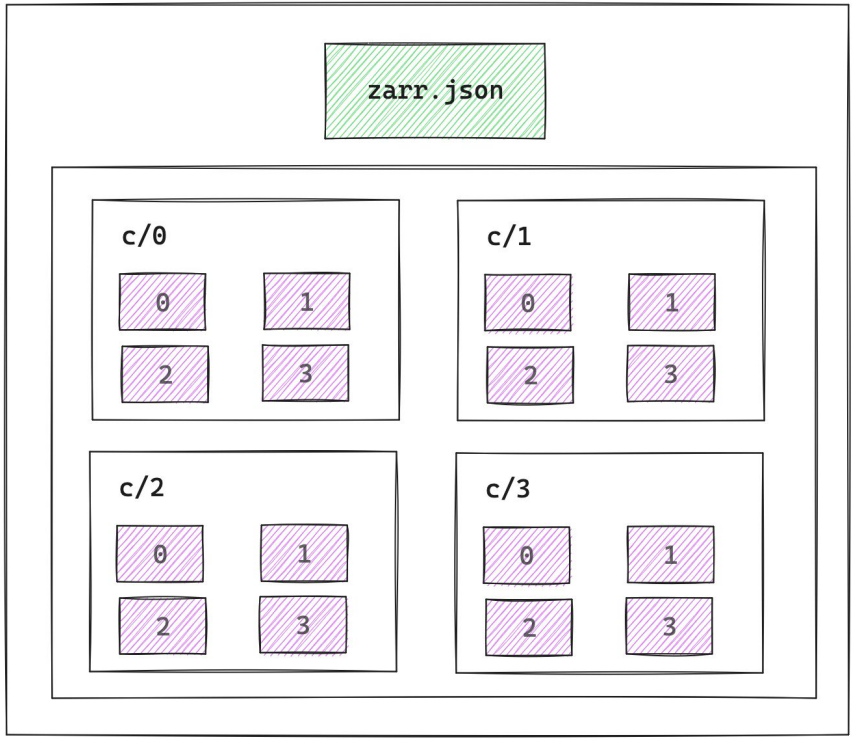
Major design updates

Restructured the JSON metadata

Zarr Arrays → V2 vs. V3

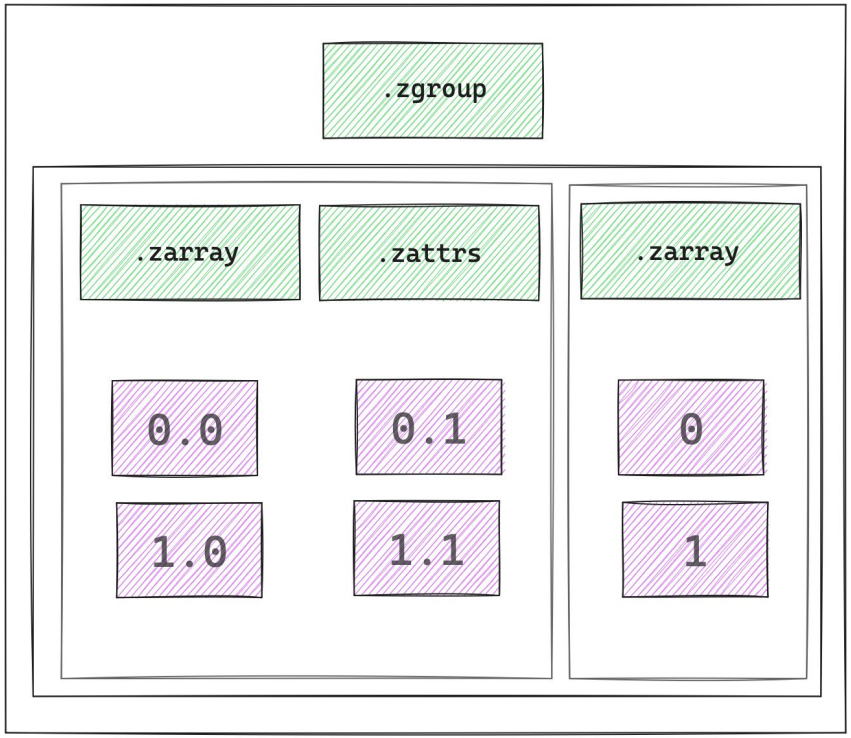


V2

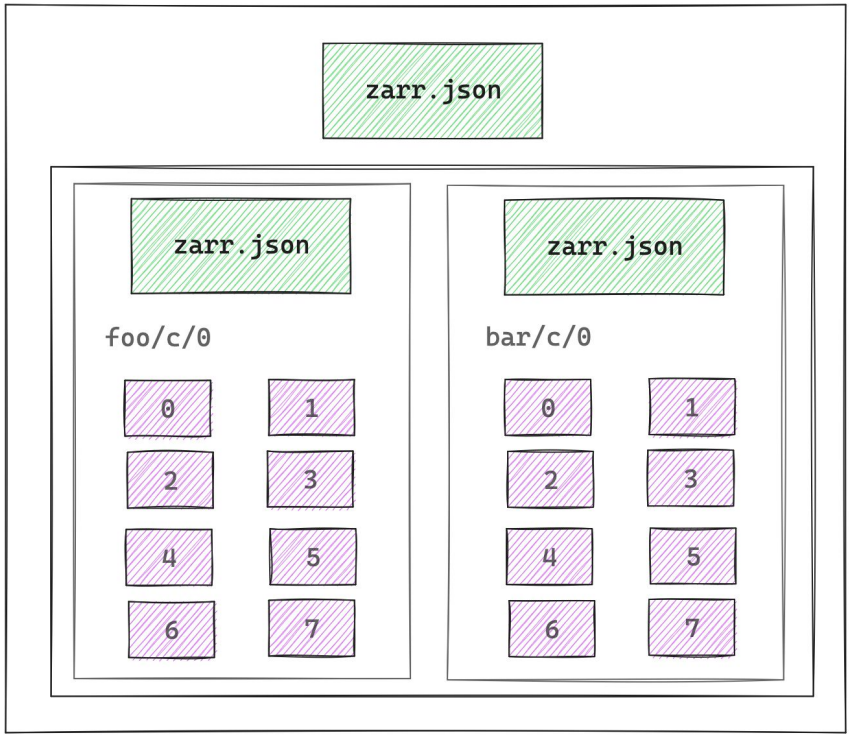


V3

Zarr Groups → V2 vs. V3



V2



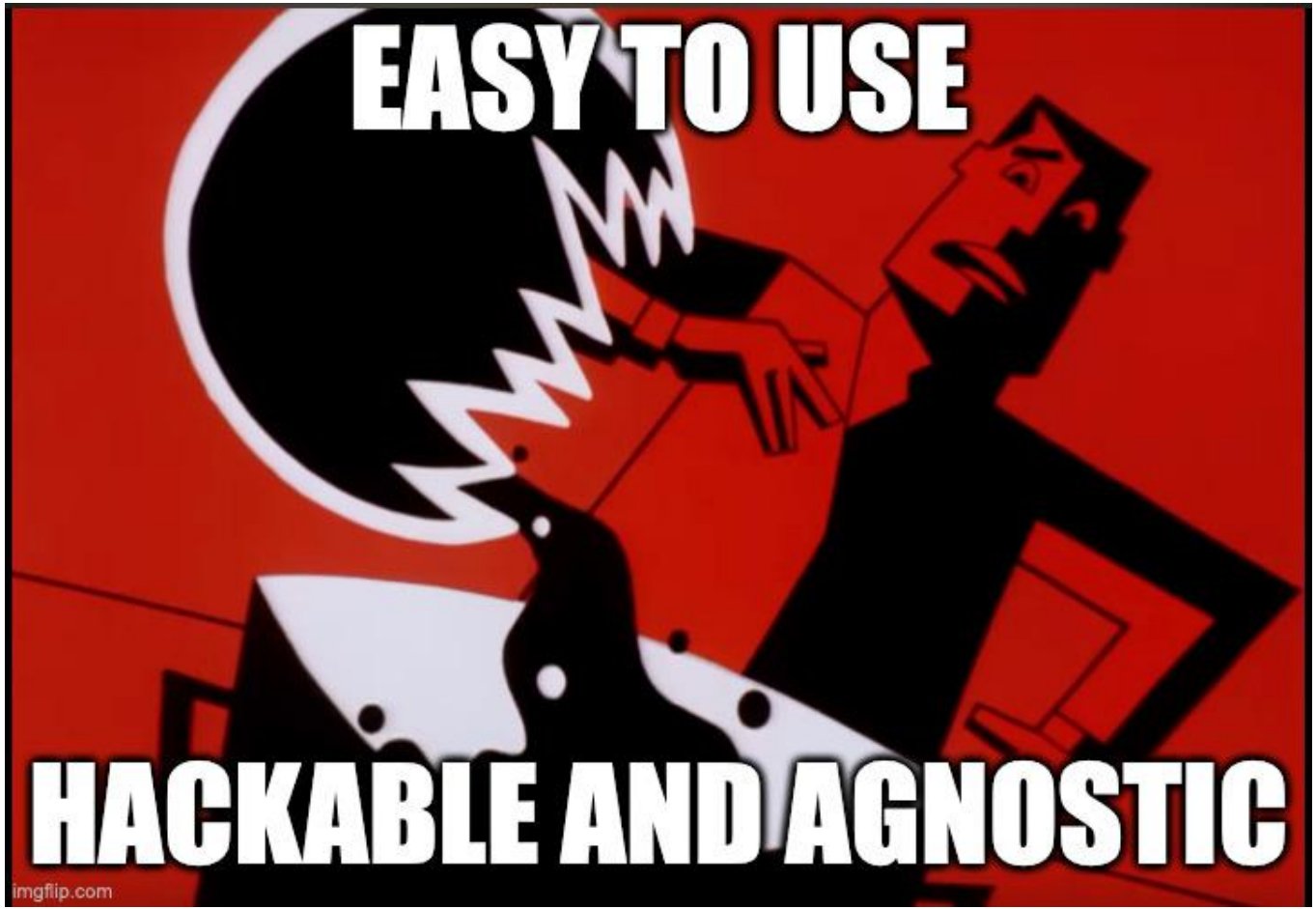
V3

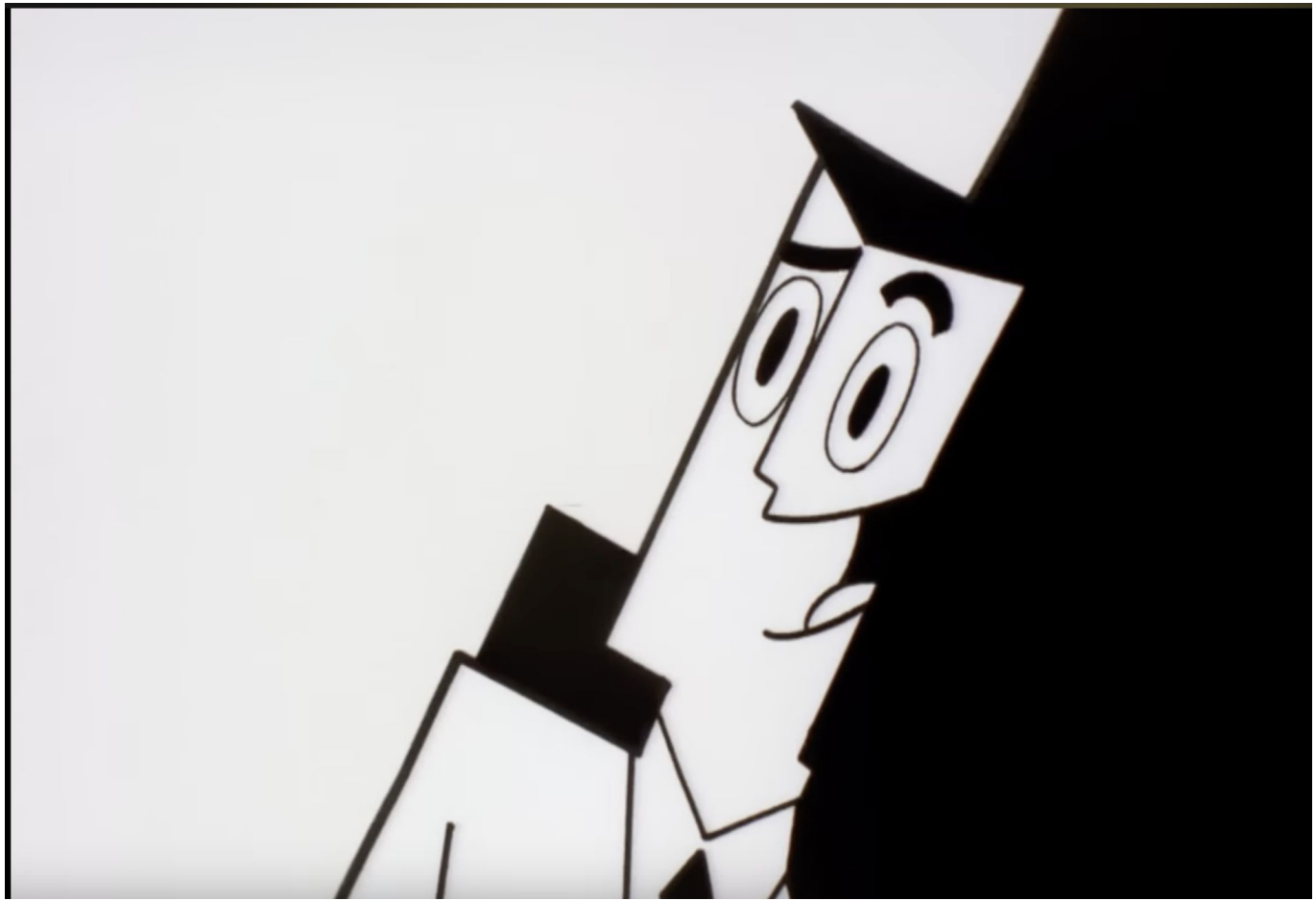
Zarr Community


ZARR OSS
SUGAR
AND SPECIFICATION
imgflip.com


USERS & DEVS
CONTRIBUTORS & MAINTAINERS
imgflip.com


GOVERNANCE AND COUNCIL
imgflip.com


EASY TO USE
HACKABLE AND AGNOSTIC
imgflip.com




Zarr

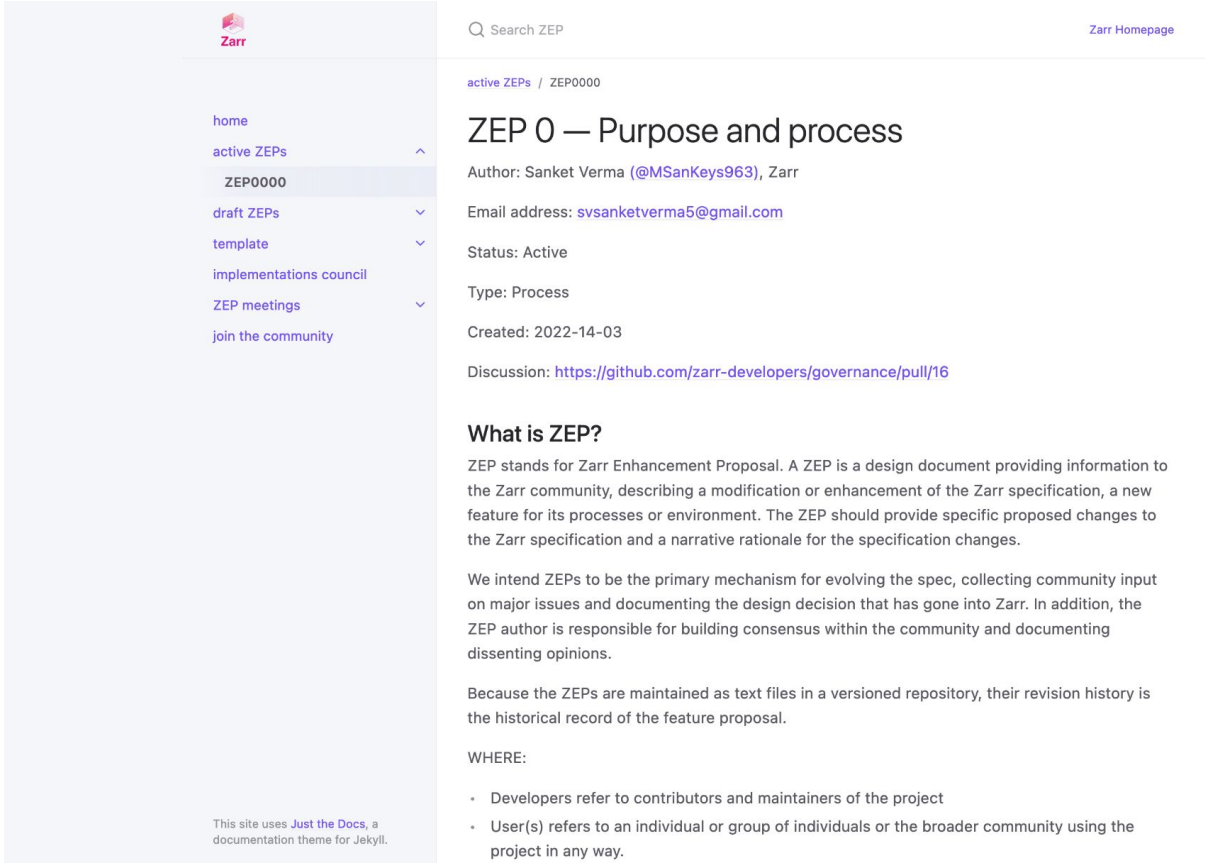
We have a large and diverse active community!

But...👀

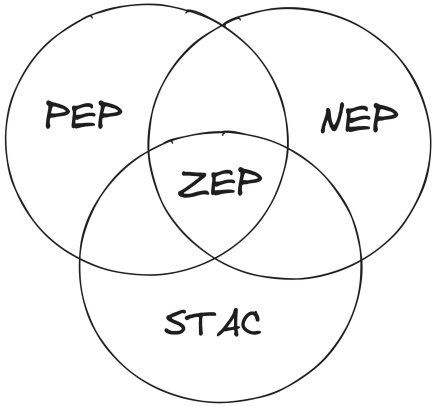

STANDARDIZING DATA FORMAT(S)
Zarr
STANDARDIZING DATA FORMAT(S)
ACHIEVING CONSENSUS
Zarr
imgflip.com

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

zarr.dev/zeps

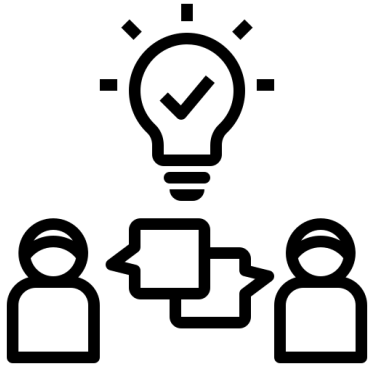


How we did it? 



Lots of Reading

Previous Experience

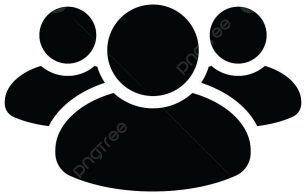


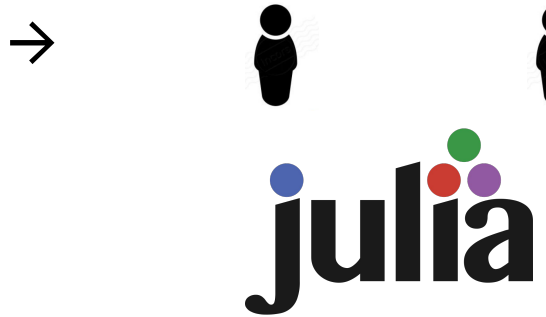
Understanding the needs of the community

Braindump 🧠

ZEP Inception Blog post

How do we adopt a ZEP?

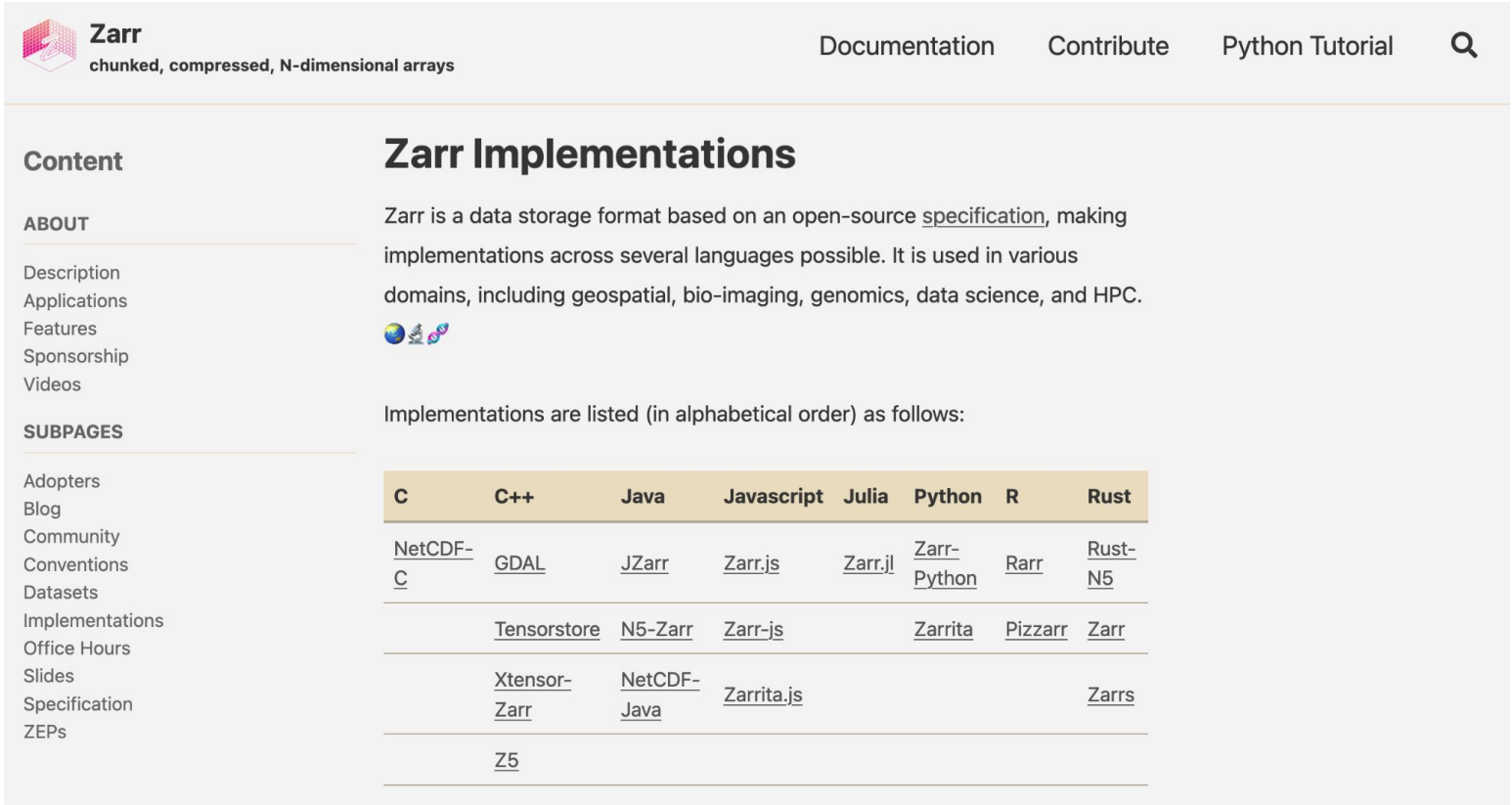



julia
JS
Java


ZARR
COMMUNITY
ZIC
ZSC
ADOPTING A ZEP
imgflip.com

Demo

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	Pizzarr	Zarr
	Xtensor-Zarr	NetCDF-Java	Zarrita.js				Zarrs
	Z5						

zarr.dev/adopters

arr Adopters

If you're using Zarr in any way and would like to be added on this page,
ease drop your logo and blurb here.

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

carbon)plan

Zarr is used by CarbonPlan as a storage format for analysis and visualization of
nate data.

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

COLUMBIA CLIMATE SCHOOL
LAMONT-DOHERTY EARTH OBSERVATORY

How does V3 data looks like?

WEBKNOSSOS

Citation:

Motta et al., Science 2019

Maintained by: /normanrz

(Norman Rzepka)

https://webknossos.org/links/NxMfC4B65CmTVGhr

/neuroglancer

https://github.com/google/neuroglancer

Maintained by: /jbms

(Jeremy Maitin-Shephard)

Neuroglancer visualiser

Join us at community, ZEP, core-dev meetings or office hours!

https://zarr.dev/community-calls/

Thank you! 🙏

/zarr_dev