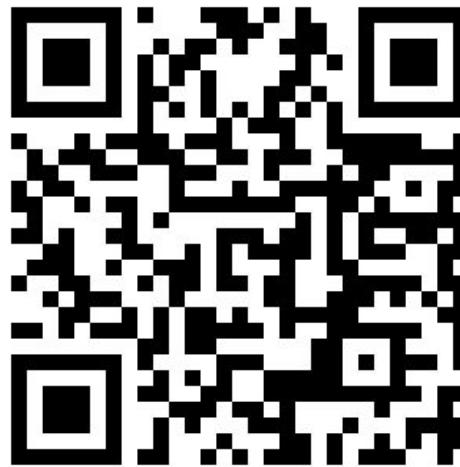


Chunked, Compressed, & Cloud-native
N-dimensional arrays

Exploring Zarr: From Fundamentals to V3.0 & Beyond

Sanket Verma

Community and OSS @ Zarr



@MSanKeys963

Slides

https://bit.ly/explore_zarr_v3



Slides & Notebook

https://bit.ly/explore_v3_more



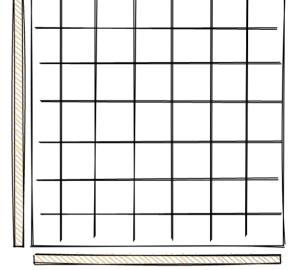
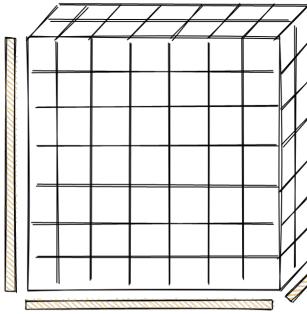
What I'll be talking about?

- What is Zarr?
- What's new in the Zarr Spec V3?
- 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.

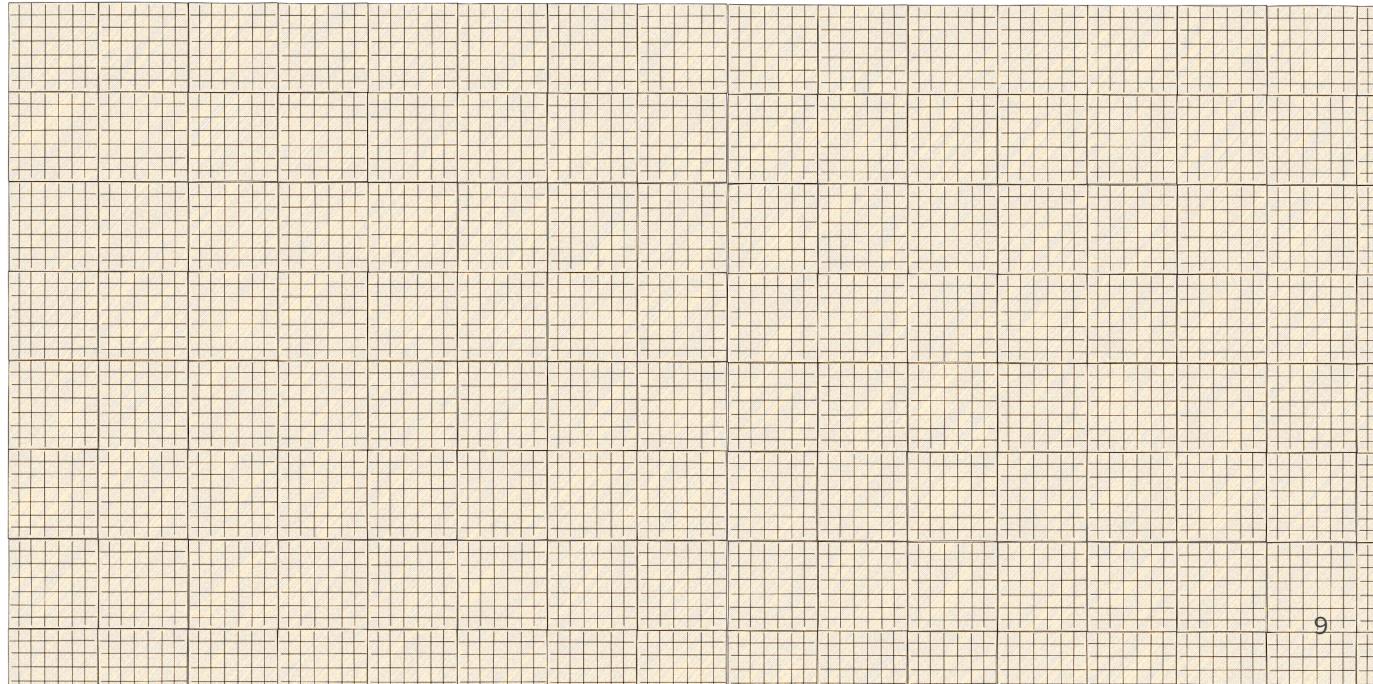
		
shape	(7,)	(7,7)
# dimensions	1D	2D
# items	7	$7 * 7$

How Zarr works?



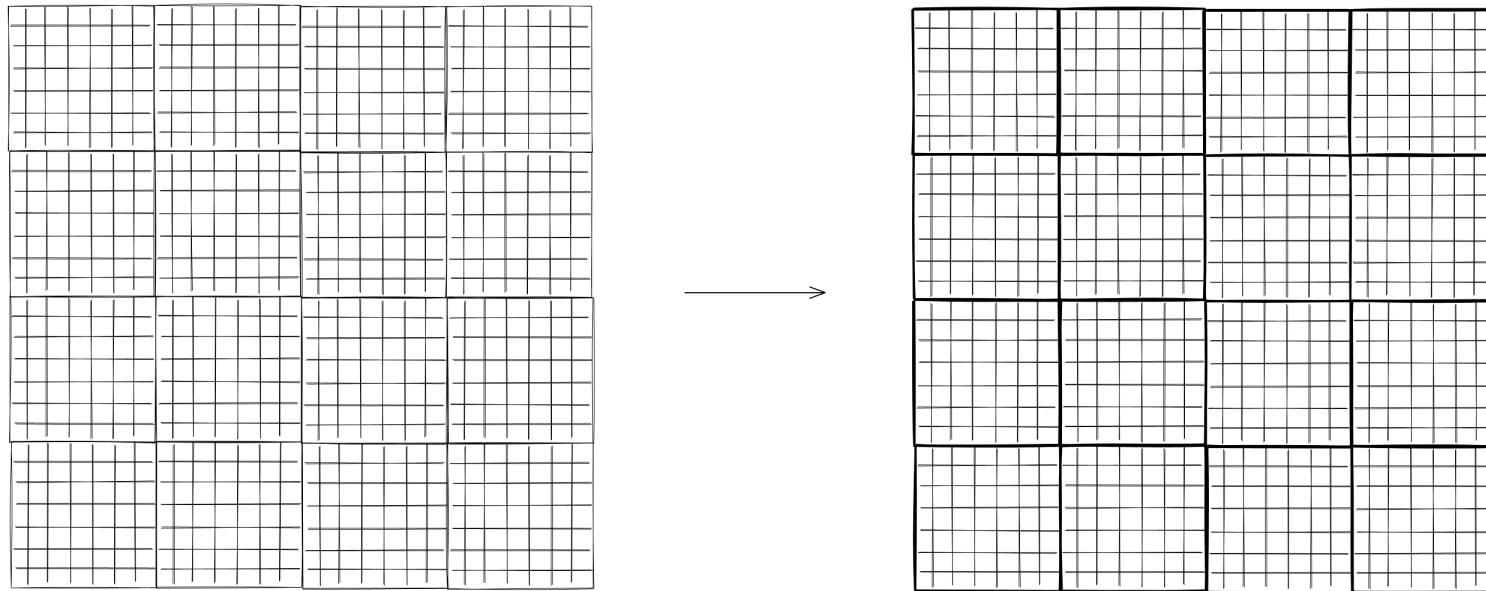
How Zarr works?

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



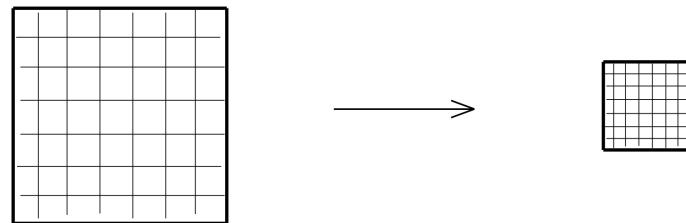
How Zarr works?

Divide array into chunks (Chunking)



How Zarr works?

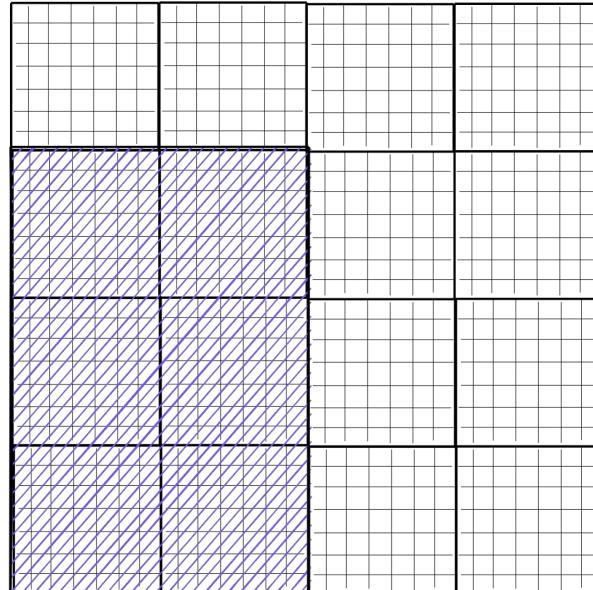
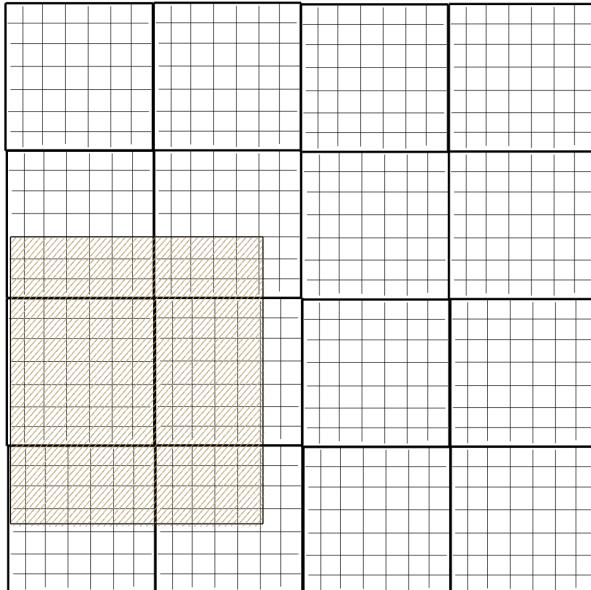
Compress each chunk



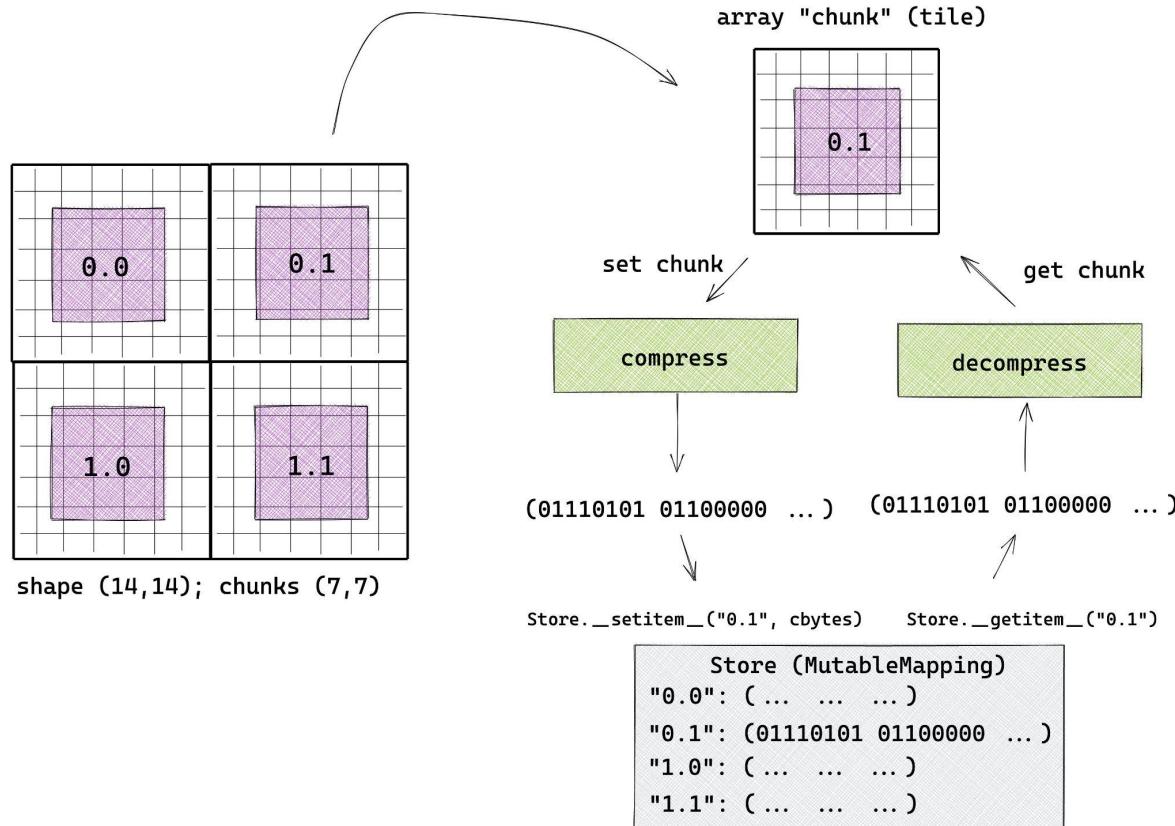
Over 20 supported compressors (BLOSC, Zstd, Zlib etc)

How Zarr works?

Retrieve chunks only when needed

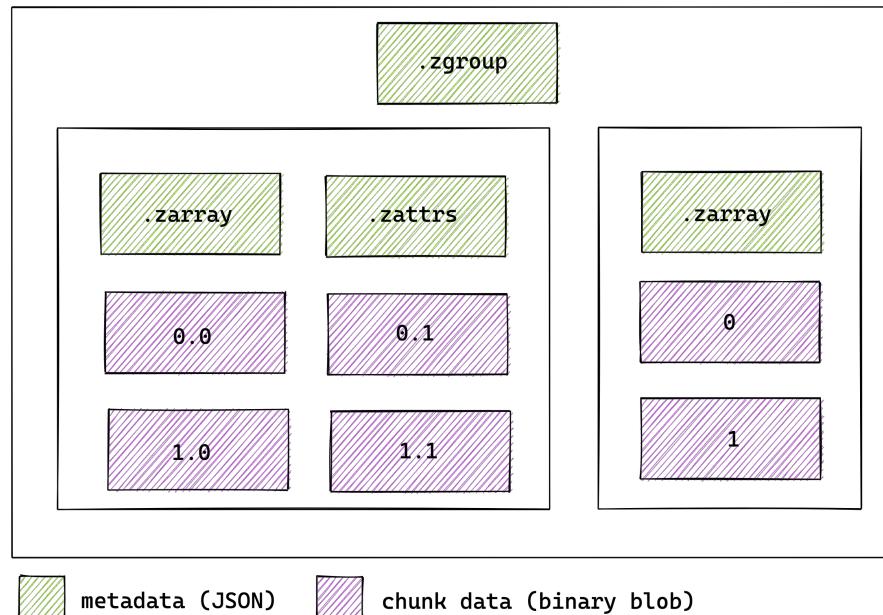


How Zarr works?



How Zarr works?

Multiple arrays can be organised in hierarchies of groups



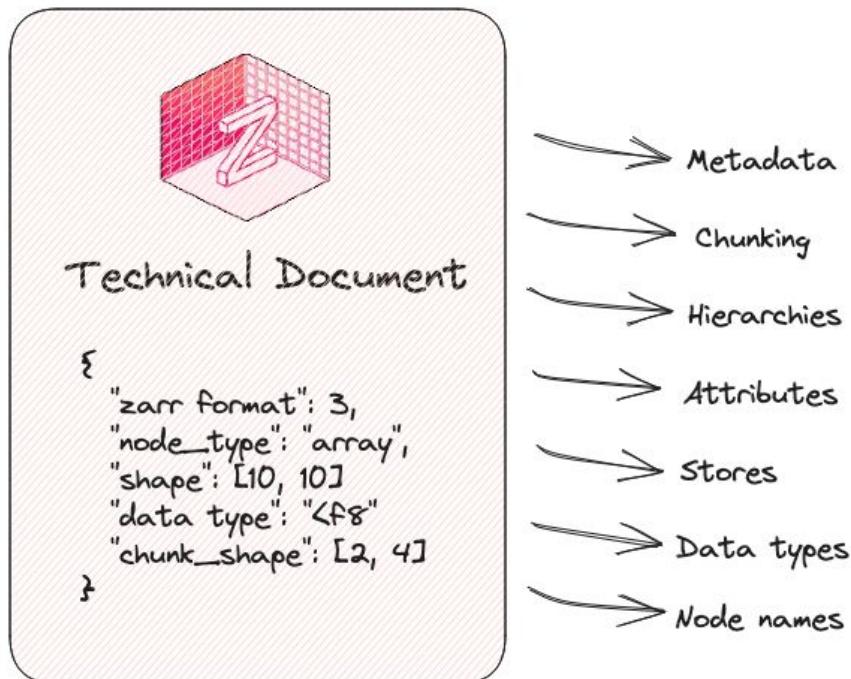
Zarr Specification

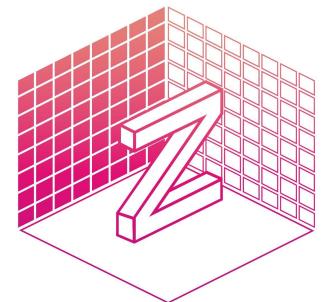
V1 → V2 → V3



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

Zarr Specification



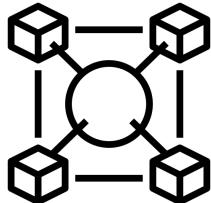


Zarr



Why did we work on V3 Specification?

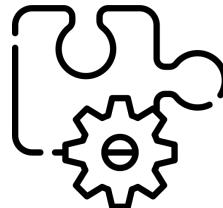
The Motivation



Interoperability



High-latency storage

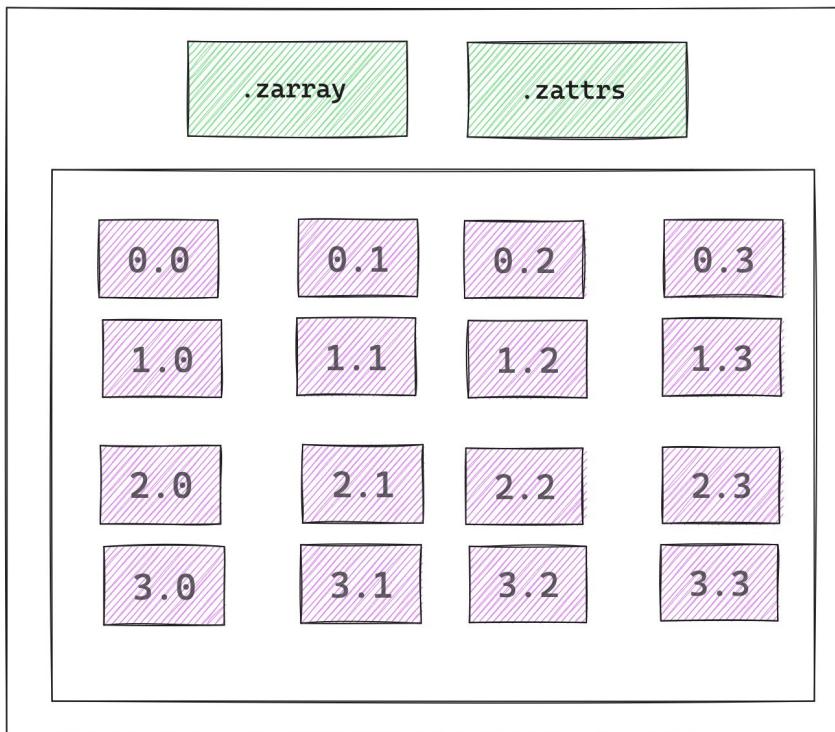


Extensibility

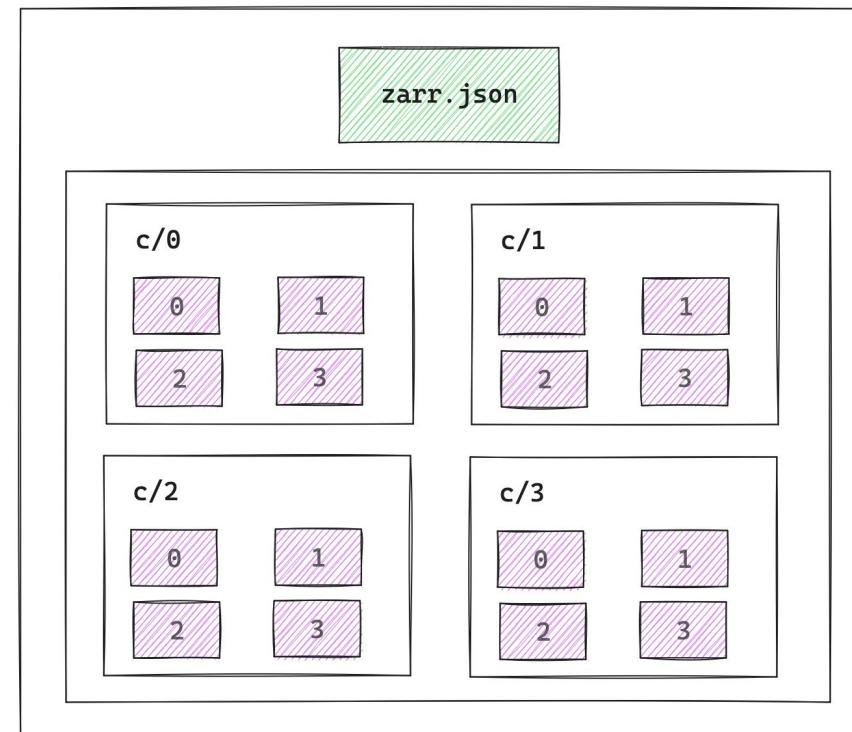
Major design updates

Restructured the
JSON metadata

Zarr Arrays → V2 vs. V3

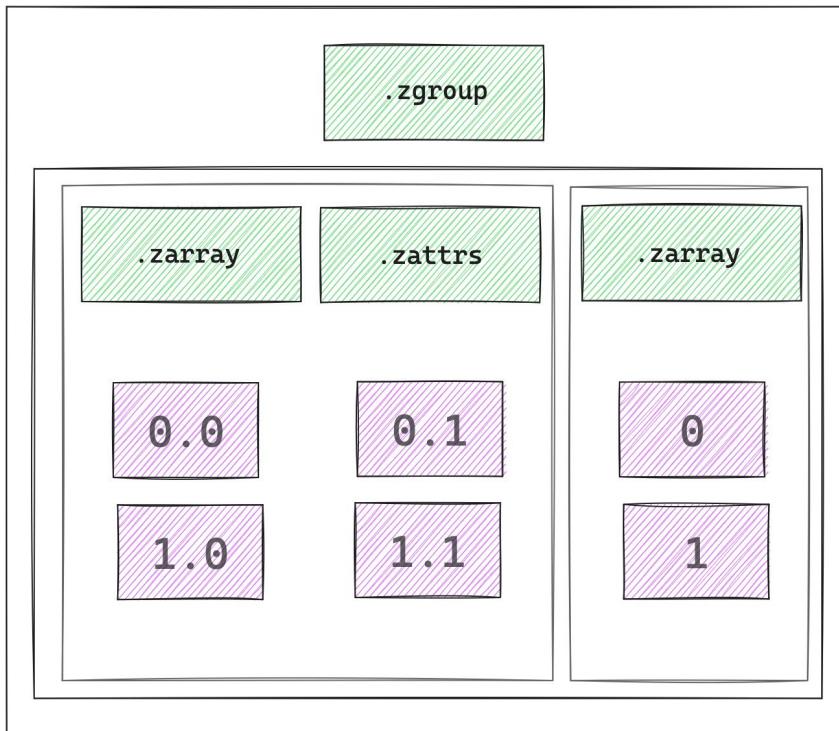


V2

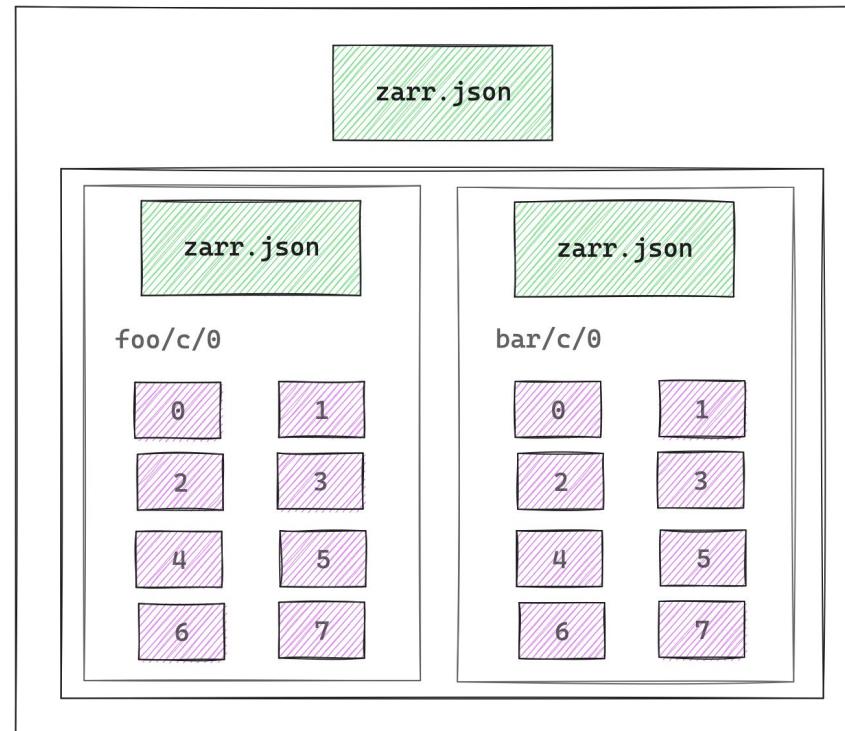


V3

Zarr Groups → V2 vs. V3



V2



V3

Explicit support for
features via defined
extension mechanism

Extension points

Different types of extensions can exist and they can be grouped as follows:

level	extension	metadata
array	data type	<code>data_type</code>
array	chunk grid	<code>chunk_grid</code>
array	chunk key encoding	<code>chunk_key_encoding</code>
array	codecs	<code>codecs</code>
array	storage transformer	<code>storage_transformers</code>

V3 Metadata

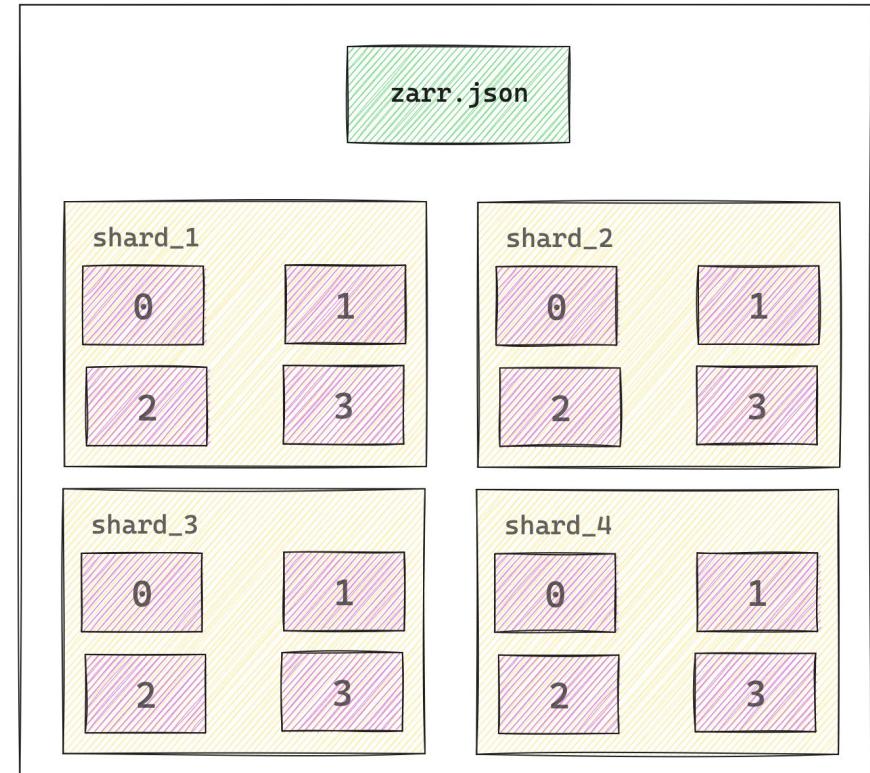
```
{  
    "zarr_format": 3,  
    "node_type": "array",  
    "shape": [10000, 1000],  
    "dimension_names": ["rows", "columns"],  
    "data_type": "float64", ←  
    "chunk_grid": { ←  
        "name": "regular",  
        "configuration": {  
            "chunk_shape": [1000, 100]  
        }  
    },  
    "chunk_key_encoding": { ←  
        "name": "default",  
        "configuration": {  
            "separator": "/"  
        }  
    },  
    "codecs": [{ ←  
        "name": "gzip",  
        "configuration": {  
            "level": 1  
        }  
    }],  
    "fill_value": "NaN",  
    "attributes": {  
        "foo": 42,  
        "bar": "apples",  
        "baz": [1, 2, 3, 4]  
    }  
}
```

V3 Metadata w/ Extension points

```
{  
    "zarr_format": 3,  
    "node_type": "array",  
    "shape": [10000, 1000],  
    "data_type": {  
        "name": "datetime",  
        "configuration": {  
            "unit": "ns"  
        }  
    },  
    "chunk_grid": {  
        "name": "regular",  
        "configuration": {  
            "chunk_shape": [1000, 100]  
        }  
    },  
    "chunk_key_encoding": {  
        "name": "default",  
        "configuration": {  
            "separator": "/"  
        }  
    },  
    "codecs": [{  
        "name": "gzip",  
        "configuration": {  
            "level": 1  
        }  
    }],  
    "fill_value": null,  
}
```

Sharding Codec

Via Extension Mechanism



chunk → compressible unit



shard → storage unit

V3 Metadata w/ Sharding Codec

```
{  
    "zarr_format": 3,  
    "node_type": "array",  
    "shape": [10000, 1000],  
    "data_type": {  
        "name": "datetime",  
        "configuration": {  
            "unit": "ns"  
        }  
    },  
    "chunk_grid": {  
        "name": "regular",  
        "configuration": {  
            "chunk_shape": [1000, 100]  
        }  
    },  
    "chunk_key_encoding": {  
        "name": "default",  
        "configuration": {  
            "separator": "/"  
        }  
    },  
    "codecs": [{  
        "name": "gzip",  
        "configuration": {  
            "level": 1  
        }  
    }],  
    "fill_value": null,  
}
```

This field gets modified

V3 Metadata w/ Sharding Codec

```
{  
  "codecs": [  
    {  
      "name": "sharding_indexed"  
      "configuration": {  
        "chunk_shape": [32, 32],  
        "codecs": [  
          {  
            "name": "bytes",  
            "configuration": {  
              "endian": "little",  
            }  
          },  
          {  
            "name": "gzip",  
            "configuration": {  
              "level": 1  
            }  
          }  
        ],  
        "index_codecs": [  
          {  
            "name": "bytes",  
            "configuration": {  
              "endian": "little",  
            }  
          },  
          { "name": "crc32c" }  
        ],  
        "index_location": "end"  
      }  
    }  
  ]  
}
```

More extensions coming soon...

 Zarr

Search ZEP

Zarr Homepage

home

accepted ZEPs

active ZEPs

draft ZEPs

ZEP0003

ZEP0004

ZEP0005

template

implementations council

ZEP meetings

join the community

Draft ZEPs

Shows the list of Draft ZEPs.

TABLE OF CONTENTS

- [ZEP0003](#) ←
- [ZEP0004](#)
- [ZEP0005](#)

https://zarr.dev/zeps/draft_zeps/

ZEP0003 - Variable Chunking



Zarr

Search ZEP

Zarr Homepage

[draft ZEPs](#) / ZEP0003

home

accepted ZEPs

active ZEPs

draft ZEPs

ZEP0003

ZEP0004

ZEP0005

template

implementations council

ZEP meetings

join the community

ZEP 3 – Variable chunking

Authors:

- Martin Durant (@martindurant), Anaconda, Inc.
- Isaac Virshup (@ivirshup), Helmholtz Munich

Status: Draft

Type: Specification

Created: 2022-10-17

Discussion: <https://github.com/orgs/zarr-developers/discussions/52>

<https://zarr.dev/zeps/draft/ZEP0003>

Comparison with Zarr V2 Spec

- `dtype` renamed to `data_type`
- `chunks` replaced with `chunk_grid`
- `dimension_separator` replaced with `chunk_key_encoding`
- separator has changed from `.` to `/`
- `filters` and `compressor` combined into `codecs` field

Zarr Community

ZARR OSS



AND SPECIFICATION

USERS & DEVS



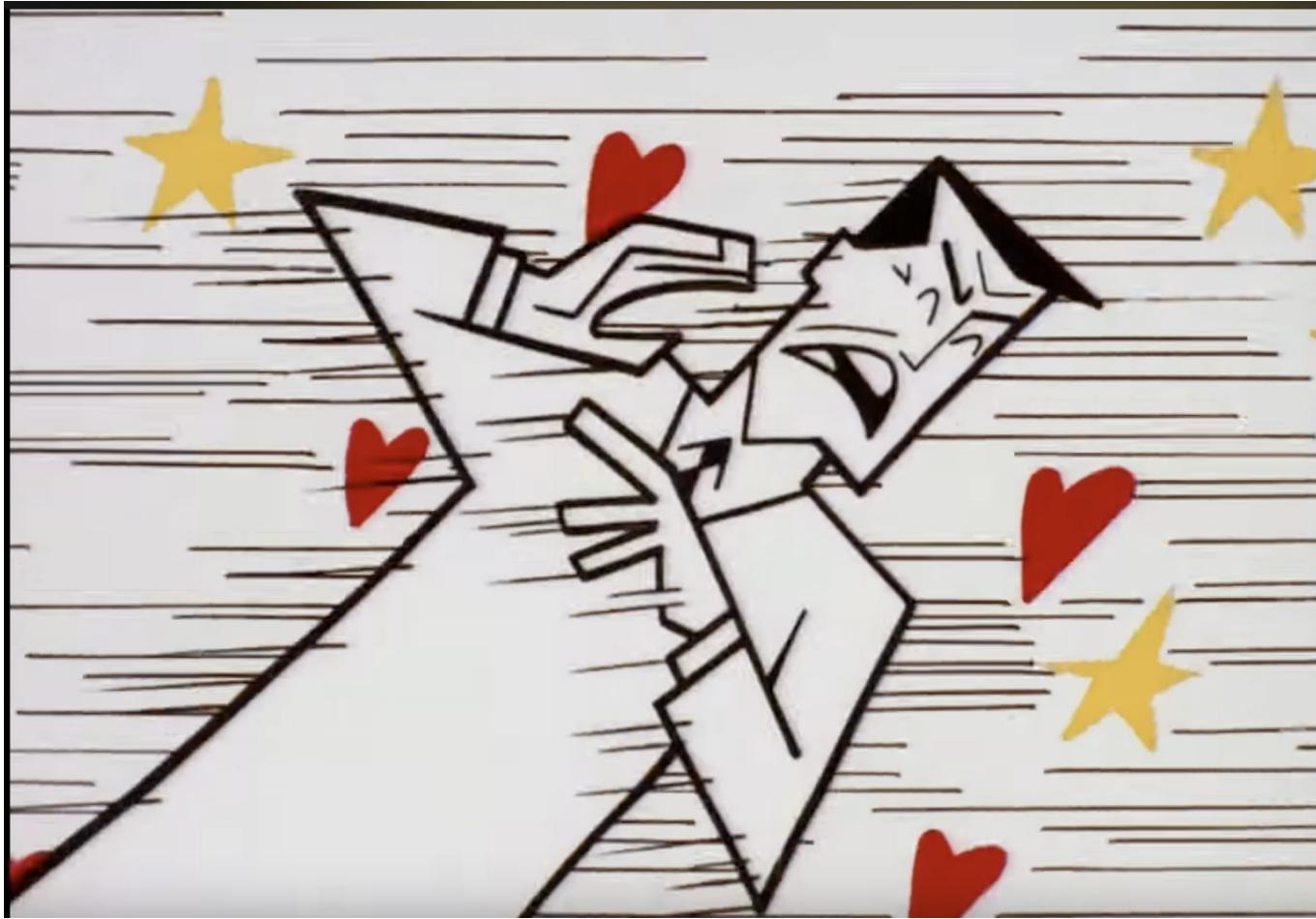
CONTRIBUTORS & MAINTAINERS





EASY TO USE

HACKABLE AND AGNOSTIC



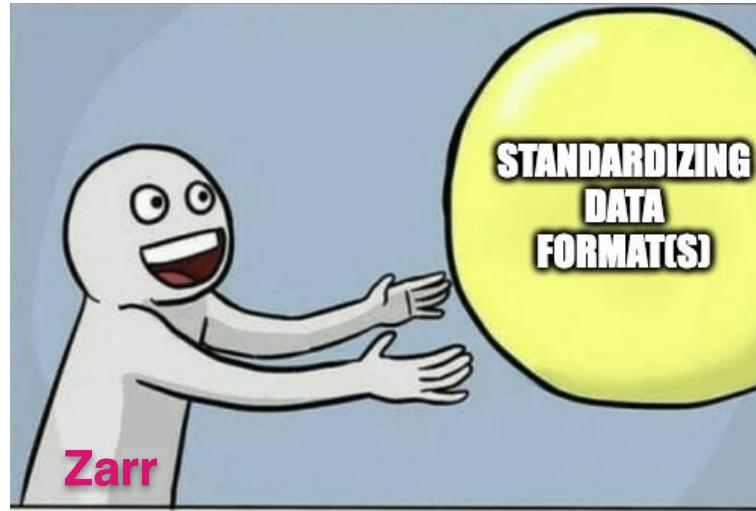






We have a large and diverse active community!

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.

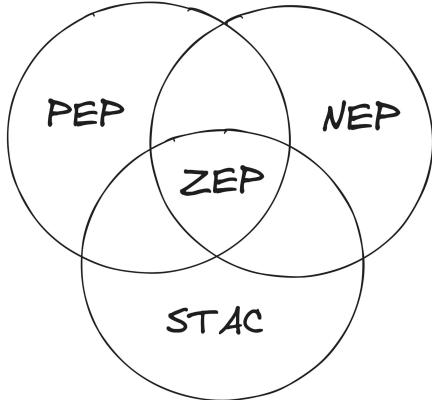
Because the ZEPs are maintained as text files in a versioned repository, their revision history is the historical record of the feature proposal.

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.

How we did it?

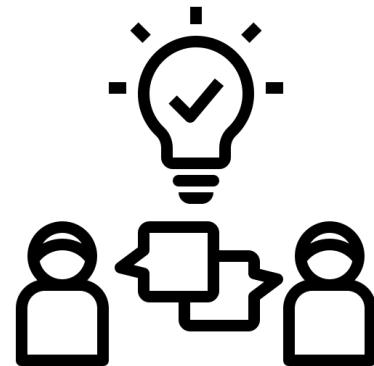




Lots of Reading



Previous Experience

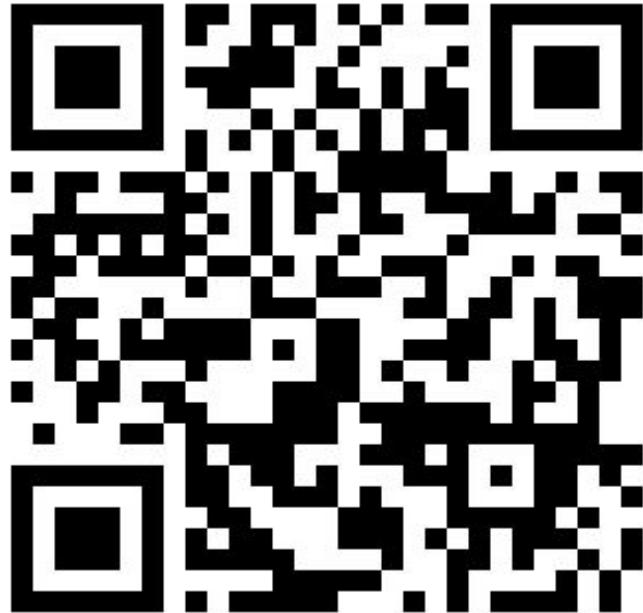


Understanding the needs of the community

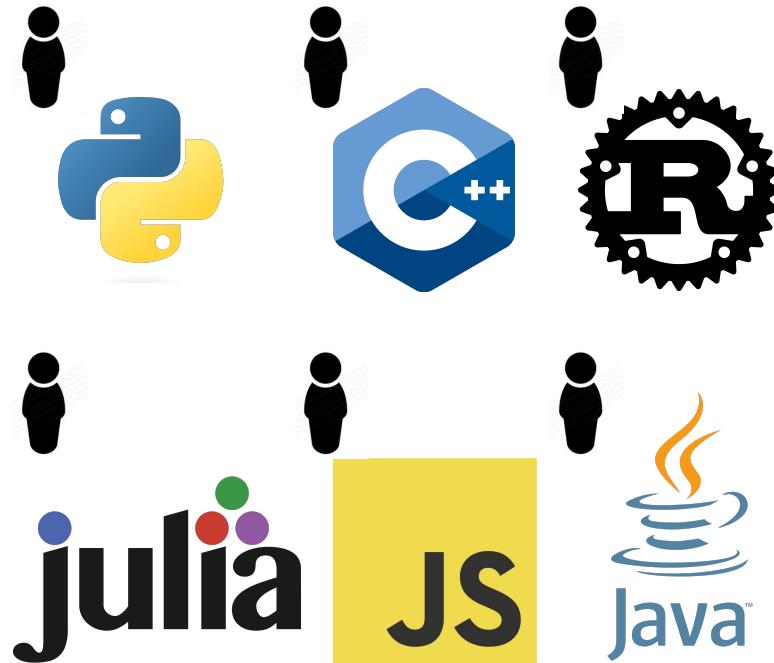
Braindump



ZEP Inception Blog post



How do we adopt a ZEP?





Demo

Creating and opening

- Zarr arrays and groups
- Sharded arrays



How does V3 data looks like?



WEBKNOSSOS

Citation:

Motta et al., Science 2019

Maintained by:  /normanrz

(Norman Rzepka)



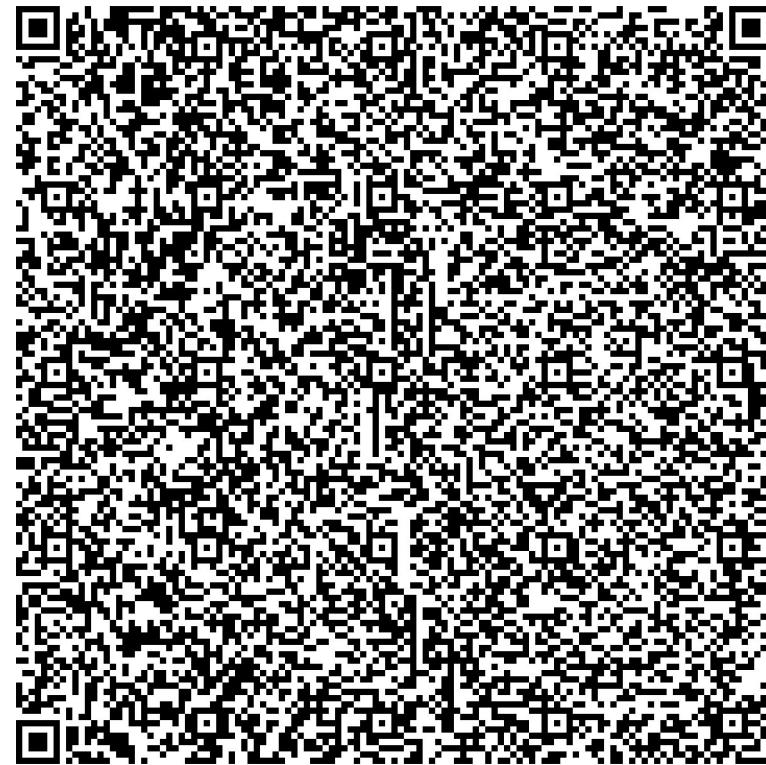
<https://webknossos.org/links/NxMfc4B65CmTVGhr>



<https://github.com/google/neuroglancer>

Maintained by:  /jbms

(Jeremy Maitin-Shephard)



[Neuroglancer visualiser](#)

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

Zarr Public Calendar

Heute April 2024 ▾

Drucken Woche Monat Terminübersicht ▾

So	Mo	Di	Mi	Do	Fr	Sa
31	1. Apr.	2	3	4	5	6
			17:00 GeoZarr Bi-we 18:00 ZEP Meeting 20:00 Zarr Communi 19:00 Zarr-Python Bi			
7	8	9	10	11	12	13
			18:00 Zarr-Python Re 18:00 Zarr Steering I 20:00 Zarr Office Hor			
14	15	16	17	18	19	20
			17:00 GeoZarr Bi-we 18:00 ZEP Meeting 20:00 Zarr Communi			
21	22	23	24	25	26	27
			18:00 Zarr-Python Re 20:00 Zarr Office Hor			
28	29	30	1. Mai	2	3	4
			17:00 GeoZarr Bi-we 18:00 ZEP Meeting 20:00 Zarr Communi			

Terminanzeige in der Zeitzone: Mitteleuropäische Zeit - Berlin

+ Google Kalender

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

Thank you!

