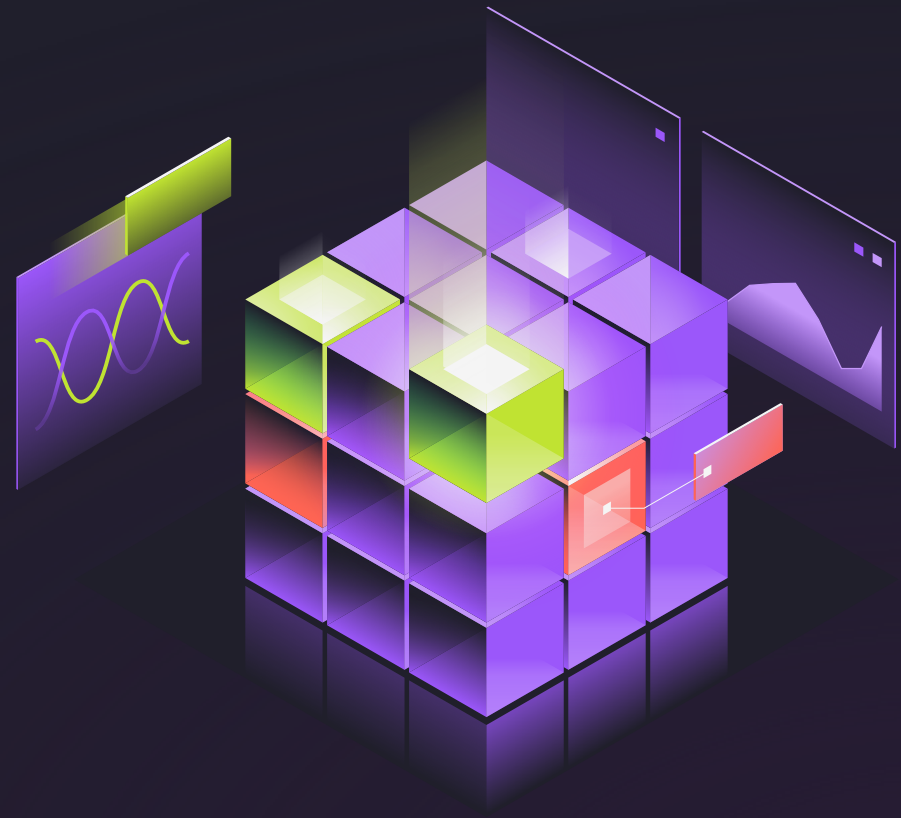
earthmover

virtual chunks: shortcut to an ai-friendly scientific data catalog



NOAA Data Community Forum

Tom Nicholas

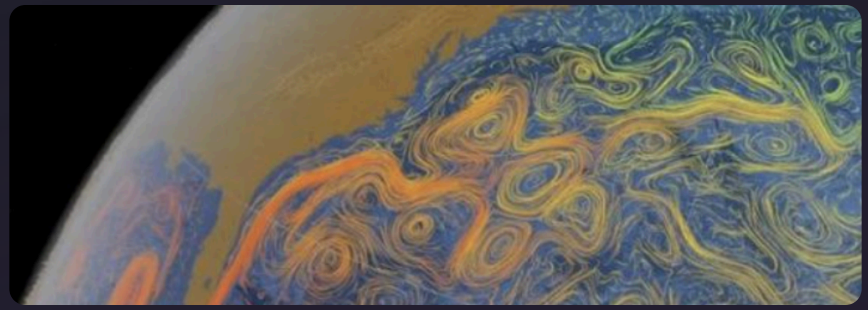
ABOUT ME



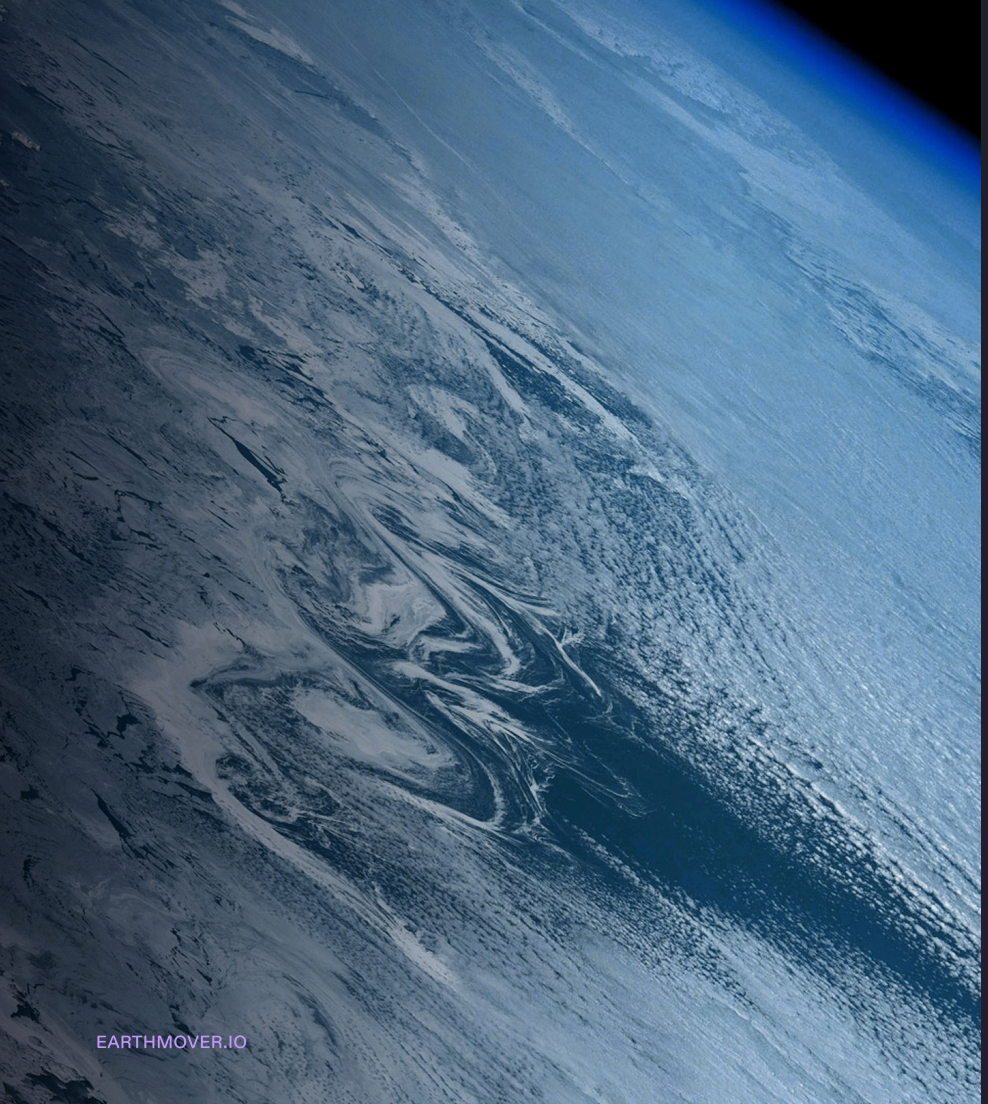
Tom Nicholas, PhD

FORWARD ENGINEER

- Ex-scientist (plasma physics, then oceanography)
- Open source maintainer
  - Xarray
  - VirtualiZarr
  - Icechunk
  - Zarr specification



OPEN SOURCE CONTRIBUTIONS



EARTHMOVER.IO

earthmover

Our Mission

To empower **people** to use **scientific data** to solve humanity's **greatest challenges.**

- **Public benefit corporation**, founded 2022 by the co-founders of **Pangeo**
- A team of **ex-scientists** and industry veterans
- The people behind **Xarray, Zarr, VirtualiZarr and Icechunk**
- "Open core": community open source + a managed cloud data platform (**Arraylake**)

ALREADY AT NOAA

who is using earthmover?



open source

- **NOAA Fisheries:** Eli Holmes and OceanHackWeek
- **NOAA MDL:** the `grib2io` package supports Icechunk
- **NOAA OWP:** IceFabric
- **NOAA EMC:** GEODE chose Icechunk as backend
- **NERACOOS/GMRI:** created Virtual Icechunk of OISST

commercial

- **NWS · CIRRUS:** Icechunk will be a core operational data format.
- **OAR · CEFI:** managed data platform for the Changing Ecosystems and Fisheries Initiative, on top of NODD data.
- **OCIO:** enterprise catalog deployment across line offices
- **GFDL:** SPEAR on Earthmover platform.

TODAY'S TALK

outline

- **Problem:** scientific data stuck in files
- **Solution**
  - Part 1: **virtual chunks**
  - Part 2: **metadata alongside the data**
  - Part 3: **AI-friendly, dataset-level catalog**
- **In production:** GRIB, CEFI, CIRRUS
- **Future work:** catalog everything, cloud-native compute

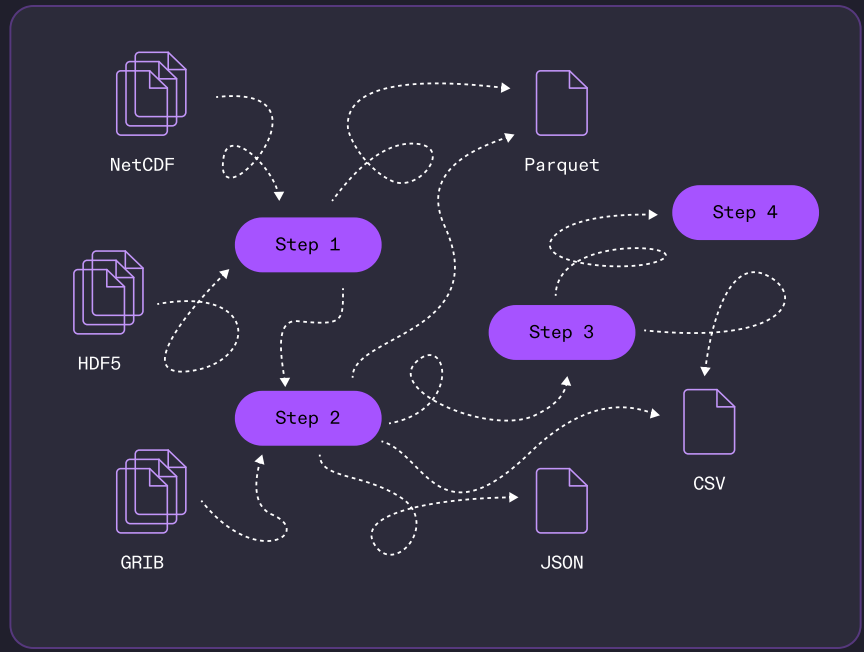
THE PROBLEM

scientific data is stuck in files

LIFT-AND-SHIFT

buckets of files

- Moving archives to the cloud was a **lift-and-shift**: same files, new location
- Open data programs like **NODD** put petabytes in public buckets
- But the bucket is still a black box: you can list filenames, not ask about the data
- Every user has to reverse-engineer the structure
- Means incredibly fragile pipelines



> "I want `OR_ABI-L2-CMIPF-M6C13_G16_s2021063….nc`"
>
> **- said no one ever.**

> "I want **GOES-16 band 13 over Gulf of Mexico, 2021**"
>
> **- someone probably.**

Scientists don't care about files. They care about **datasets** and **queries**.

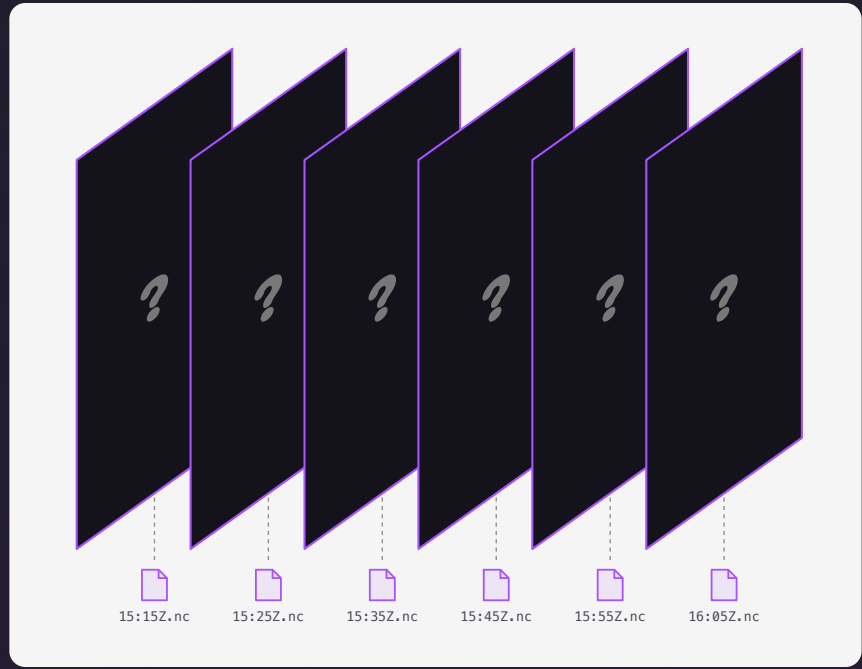
Increasingly so do **AI agents**.

THE PROBLEM

where is the data you want?

- First find the bucket containing GOES-16
- Consists of 300k NetCDF files 😱
- Every chunk is a different location in each file
- Queries must search every file, every time

Slow and error-prone, for humans and agents.





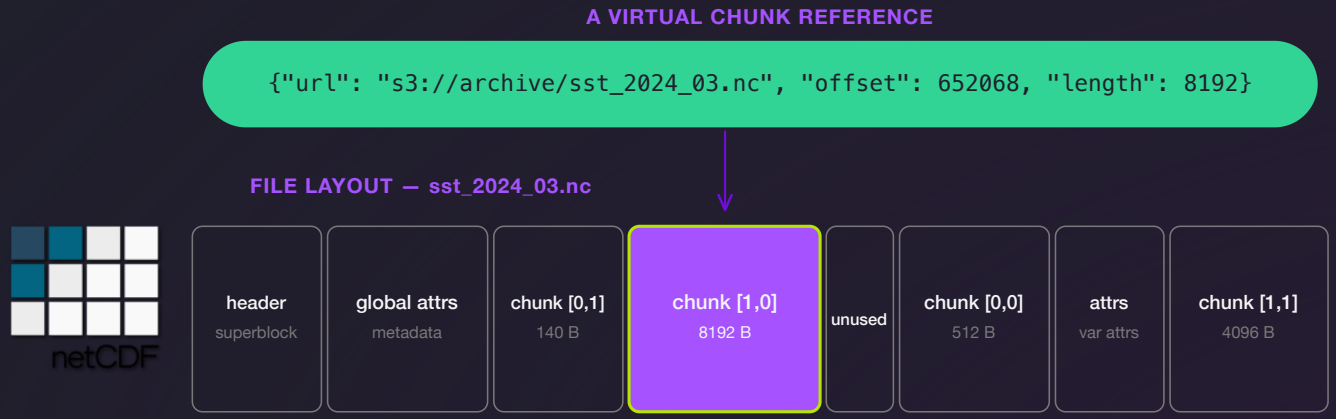
SOLUTION: PART 1

virtual chunks

VIRTUAL CHUNK

virtual chunks are references to existing data

- Object storage makes it easy to **access** data from anywhere, but hard to **change** it. Incentive to **reference existing data.**
- A **virtual chunk reference** is a byte-range pointer to bytes that already exist somewhere else, e.g. in a NetCDF file.



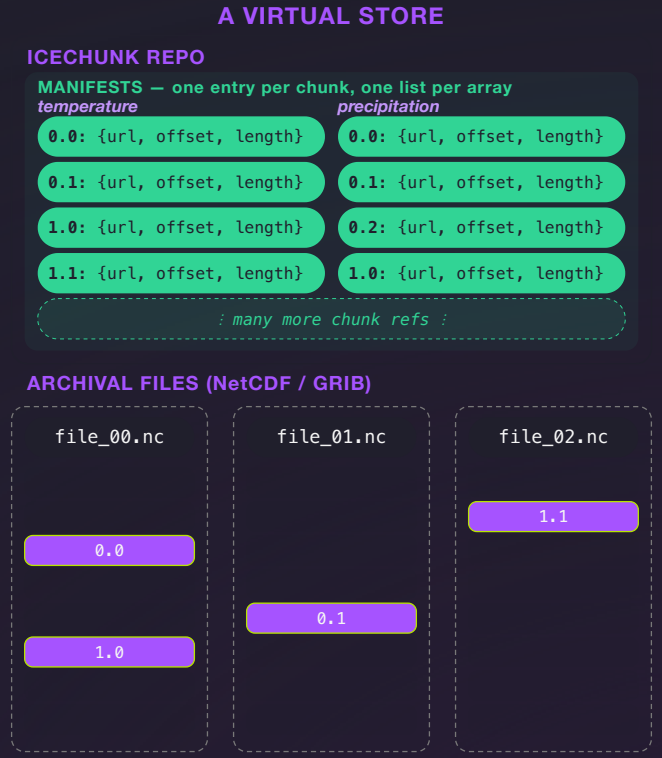
VIRTUAL STORE

an index of many virtual refs

Many scientific formats (NetCDF, HDF5, GRIB, TIFF…) are really just **chunks of bytes** that map onto Zarr's data model.

A **virtual store** is a repository whose manifests point into many archival files.

- Looks like an ordinary Zarr store
- **No bytes copied:** just references + metadata
- Thousands of files become **one dataset**

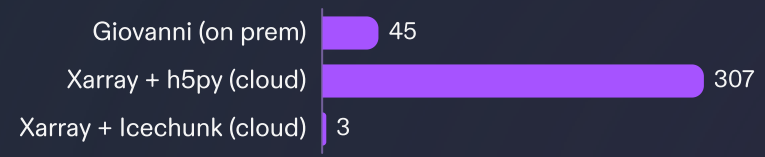


EFFICIENT QUERIES

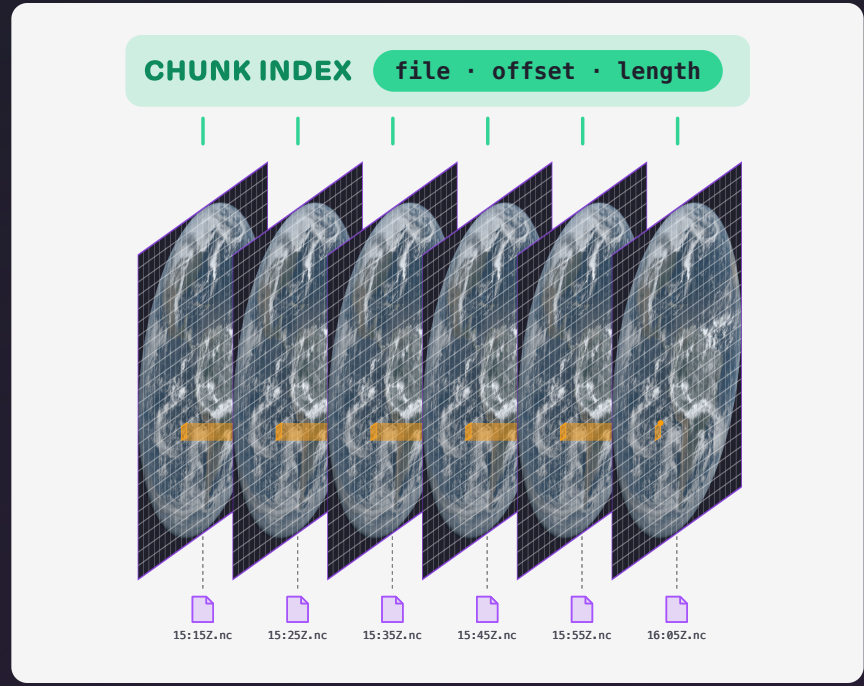
index tells you where the data you want is

- A **chunk index** records every chunk: which file, what offset
- A query reads the **small index**, not the files
- Fetch **only the chunks you need**

Enormously faster: NASA IMERG timeseries (seconds)



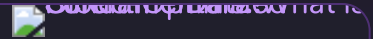
earthmover.io/blog/nasa-icechunk



CLOUD-OPTIMIZED

cloud-optimize without copying

- **Cloud-optimized** means (1) fetching **metadata cheaply**, (2) fetching **minimum data** needed for a query
- The **chunk** is the minimum unit of fetching
- So **creating the index = cloud-optimizing!**
- We can index the chunks **where they already sit**, inside the archival files
- i.e. no second copy: **avoid duplicating PBs** of binary chunk data!



BLOG POST

Fundamentals: What is Cloud-Optimized Scientific Data?

earthmover earthmover.io



BLOG POST

Old format, no problem!: Cloud-optimizing the GOES-16 archive as Virtual Zarr

earthmover earthmover.io

AGILITY

sidestep inertia of existing systems

Building virtual references only requires reading files you already have access to.

- **Data producer moves slowly** → *you don't need permission* to create a virtual store!





SOLUTION: PART 2

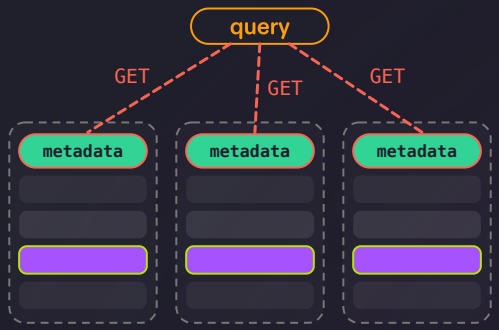
metadata alongside the data

METADATA ALONGSIDE DATA

keep the metadata close, but not too close

Where should we store the index/metadata?

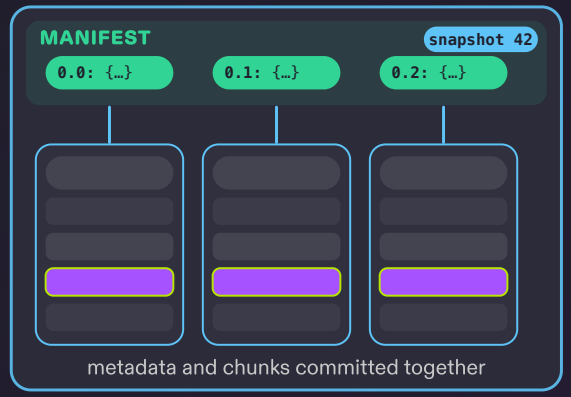
IN THE FILES



open every file just to read its metadata

✗ inefficient

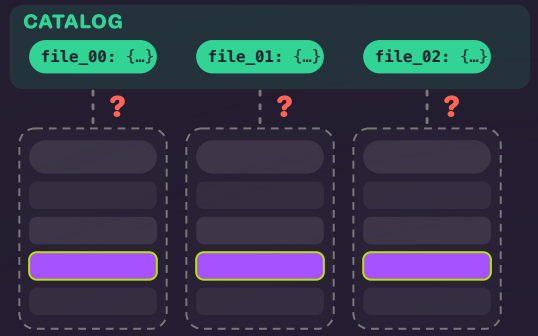
ALONGSIDE THE CHUNKS



metadata and chunks committed together

✓ efficient, scalable, consistent

IN A SEPARATE CATALOG



a second system that drifts out of sync

✗ disconnected, scales poorly

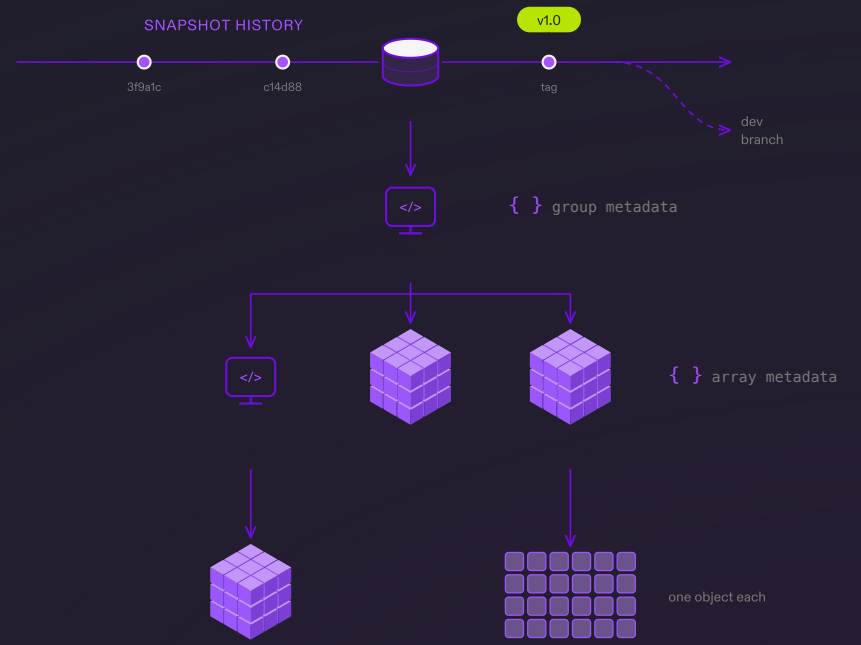
ICECHUNK

zarr for production

A specialized **Zarr Store** with a manifest layer that provides:

- **Virtual chunks:** pointers into existing files
- **Metadata corrections** without rewriting data
- **Safe, cheap updates:** ACID transactions
- **Version control:** snapshots, branches, tags
- **Open source** (Apache 2.0), **serverless**

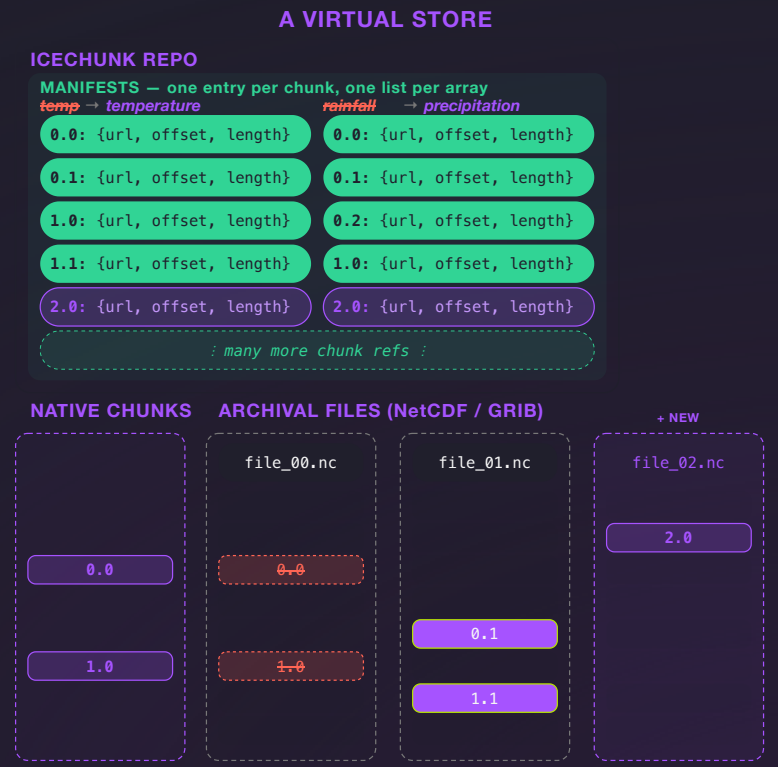
https://icechunk.io



CONSISTENCY

fix problems without rewriting archives

- **Metadata inconsistencies** → the new store can arbitrarily rewrite metadata
- **Bad files** → corrupt files can have chunks selectively overwritten
- **Updating operational data** → atomic commits, including virtually ingesting new files



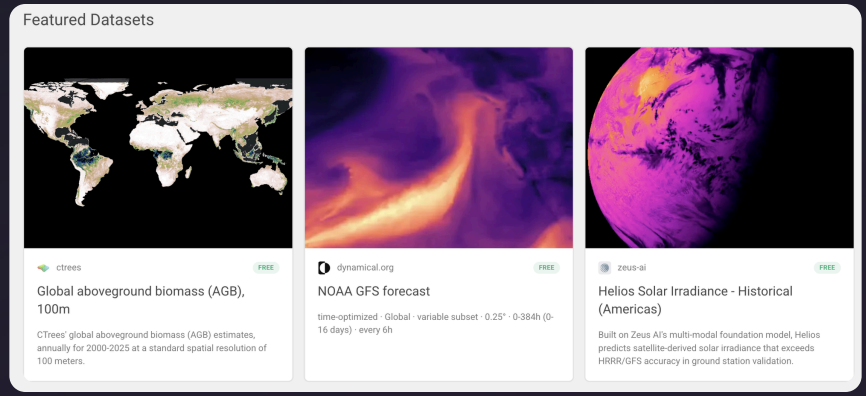
SOLUTION: PART 3

ai-friendly dataset-level catalog

DATASET-LEVEL CATALOG

catalog repos not files

- Group many files into one versioned **repository**
- Your catalog now tracks **O(100s) of repos,** not **O(billions) of files**
- Enforce **access control** at this level
  - Private / embargoed data
  - Access metrics
  - Multi-tenant, data exchanges



ANNOUNCEMENT

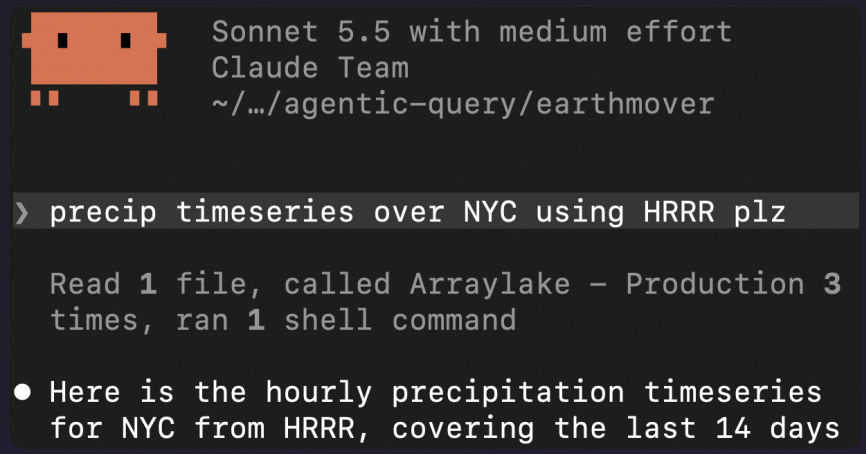
Filtered Subscriptions: Fine-Grained Cloud-Native Data Sharing

earthmover earthmover.io

AGENT-FRIENDLY

agents like dataset-level metadata

- Dataset-level metadata is **accessible,** so agents can find and read it
- Works well even **without an MCP** (since agents know how to write Xarray & Zarr queries)
- Works even better **with an MCP**





ARRAYLAKE ARCHITECTURE

cloud data warehouse, built for scientific data models

		Modern data stack (tabular)	Cloud-native scientific stack (arrays)
APPS	User Applications	Business Applications	Science Projects
PLATFORM	Query & Visualization Services	Tableau · Looker · Power BI	Flux
PLATFORM	Data Catalog & Governance	Snowflake · Delta Lake	Arraylake
OPEN SOURCE	Analysis & Compute	Pandas / DuckDB · Polars	Xarray / ???
OPEN SOURCE	Storage Engine	Iceberg · Delta Tables	Icechunk
OPEN SOURCE	Format Compatibility	DB "external tables"	Virtual Ingestion
CLOUD	Object Storage	aws ·  ·  · CLOUDFLARE · generic S3-compatible	

EARTHMOVER.IO



SUCCESS STORIES

from theory to production

CASE STUDY 1

virtualize all noaa gribs

GRIB is a notoriously fiddly format to work with. Format-specific parsers make virtualizing it tractable.

- **NOAA NBM**, virtualized via "Gribberish" package. No one correct way to map GRIB → Zarr, so uses flexibility of VirtualiZarr to handle the different data model
- **NOAA HRRR**, virtualized by **dynamical**. 14PB, ~1 billion GRIB messages, 150+ variables!
- also NOAA GFS, GEFS, REFS, ...



BLOG POST

Virtually Gribberish: Bringing Icechunk clarity to GRIB archives

earthmover earthmover.io

GUEST POST

Having it all: virtual and materialized data products

earthmover earthmover.io

CASE STUDY 2

noaa cefi

- The CEFI Portal serves regional MOM6/COBALT ocean-climate model output: hindcasts, forecasts and projections
- Data was already on NODD, but as **flat per-variable NetCDF files**
- Icechunk + Arraylake: **one versioned catalog**, model releases as tags, Xarray-native access

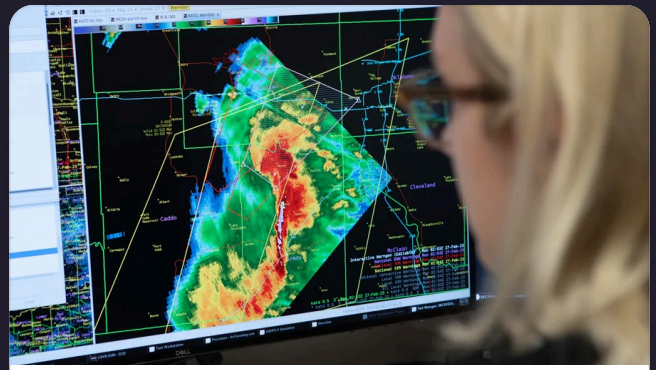


From a **download portal** to a **production data platform**

CASE STUDY 3

nws cirrus

- The National Weather Service is building **CIRRUS** with Booz Allen Hamilton
- Icechunk selected as a core format for operational forecast data



ANNOUNCEMENT

Icechunk Adopted by the National Weather Service: Earthmover Joins Booz Allen on NWS CIRRUS

June 4, 2026



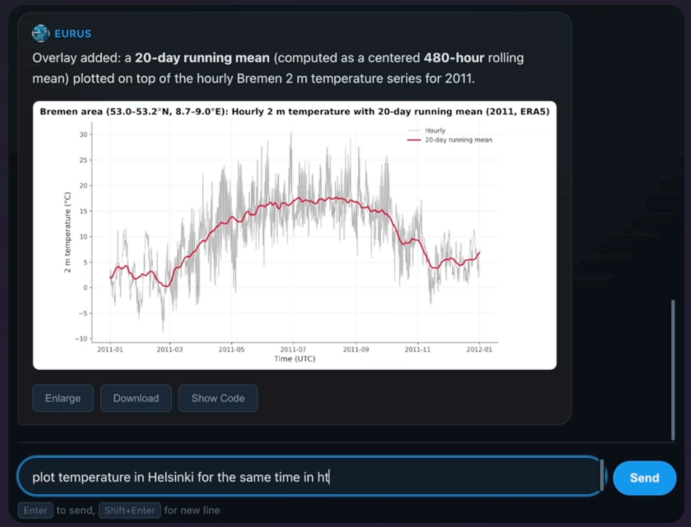
FUTURE WORK

where next?

CATALOG EVERYTHING

one ai-friendly catalog spanning all of noaa

- **Generalize virtualization:** more formats, fewer special cases
- **AI-powered ingestion:** agents write the dataset-specific parsing, with supervision
  - Try the icechunk-datacube-ingestion skill
- **Level 2 data collections:** swaths and scenes, not just gridded cubes
- **Station / radar / satellite data:** prove the same architecture works beyond model output (the OCIO work)



CLOUD-NATIVE COMPUTE

compute next to the data

Zax is a modern, pure-Rust query engine for NetCDF-style data, built directly on Icechunk.

- **Schema-first** and **lazy by default**
- **Chunk-aware execution** with bounded memory
- SQL on array data via **DataFusion**
- **Client / server:** develop locally, execute remotely at scale



pandas : polars :: xarray : zax

```
ds = zax.open_icechunk(repo, branch="main")
ds_region = ds[['temperature']].sel(
    lat=slice(45, 60), lon=slice(0, 90)
)
fragment = ds_region.collect()  # bounded memory
```

CONCLUSION

chunks need catalogs

- Object storage: **cheap targeted reads, cheap metadata, slow scans, expensive edits**
- Naturally leads to **versioned manifests** over **dataset-level repos**
- **Arraylake** catalogs science data, agents like it
- **Virtual chunks** are a useful shortcut: no migration, no permission needed
- Soon: **cloud-native compute** from first principles





BONUS SLIDES

in case there’s time (or questions)

OBJECTION

"but i like downloading single files"

- **Unit of storage ≠ unit of consumption**
- In tabular databases you consume Arrow or CSV, not the raw SQLite file
- In an array database (i.e. Icechunk) you export via OpenDAP or other services
- A file that never existed on disk: **the perfect file for your immediate need**

KERCHUNK VS VIRTUALIZARR

manifest formats side-by-side

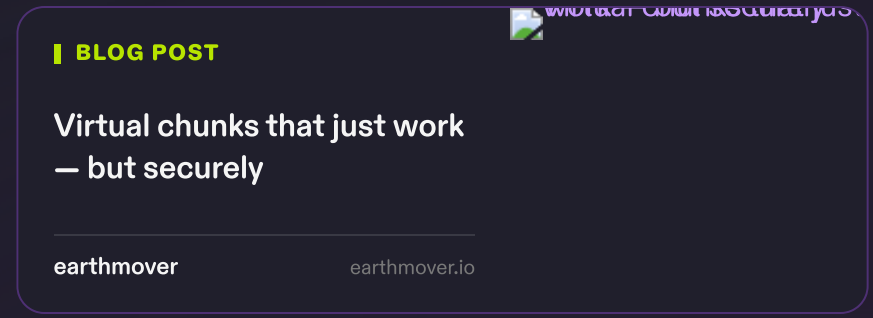
	Icechunk	VirtualiZarr	Kerchunk	DMR++	LanceDB	Iceberg
Logical data model	Zarr (but flexible)	Zarr	Zarr (but flexible)	HDF5?	tabular	tabular
Mixed chunk types	✅	✅ (per-array)	✅ (not native)	❌	✅	?
Efficient at scale	✅	✅	❌ (JSON/dict), ✅ (Parquet)	❌	✅	✅
Prefix management	✅ (VCCs)	✅ (registry)	❌	❌	✅	?
Runtime I/O layer	✅	✅ (obstore)	✅ (fsspec)	✅ (Hyrax)	✅	✅
Serverless data access	✅	✅	✅	❌	✅	✅
Change detection	✅	n/a	❌	❌	✅	✅
Transactions / versioning	✅	n/a	❌	❌	✅	✅
Serialized format	IC flatbuffers	n/a	JSON / Parquet	XML	lance	parquet

SECURITY

virtual chunks cross trust boundaries

Virtual chunks reference **arbitrary locations** containing **arbitrary data.** That has real security implications.

- **Exfiltration:** a malicious manifest could reference `file:///bitcoin-wallet`, then just ask you to "load this Icechunk repo"
- **Overexposure:** a misconfigured prefix can expose an entire private bucket
- **Tampering:** archival files updated in place invalidate assumptions baked into a manifest
- Fix: a **trusted server** validates chunk storage locations and dispenses credentials only for buckets a user is



OBJECTION

"but what about a data lake / data mesh?"

- Your data is **highly structured** (dimensions, coordinates, units)
- Your data is **too big** for schema-on-read to work
- You haven't even got the **warehouse-like** part yet: a governed, queryable catalog



earthmover.io