

The Link between STAC and Open Science

Why a data producer should publish a catalog

Who is in this room

STAC is maintained by a seven-person steering committee. Most of us are here this week, and several are speaking today.

Matthias Mohr (<i>chair</i>)	moreGeo
---------------------------------------	---------

Alex Leith	Auspatious
-------------------	------------

Emmanuel Mathot	Development Seed
------------------------	------------------

Julia Signell	Development Seed
----------------------	------------------

Matthew Hanson	LGND AI
-----------------------	---------

Michelle Roby	Radiant Earth
----------------------	---------------

Pete Gadomski	Development Seed
----------------------	------------------

If your metadata cannot be crawled and indexed, your data *might as well not exist* .

If you need to download files, you *will not be able to scale processing* .

If data provenance cannot be determined, your research *cannot be reproduced* .

I wrote this in 2020. Six years on, I would not change a word of it.

What open science asks for

FAIR. Four words, from a 2016 paper by Wilkinson and about fifty other people. Every open data policy is written in this language.

F Findable. A permanent ID, good metadata, indexed where people look

A Accessible. You can fetch it with that ID, over an open protocol

I Interoperable. One tool can read two datasets

R Reusable. A licence, community standards, and **provenance**

Two parts get read wrong. **A1.1 says the protocol is open — not the data.** A restricted product can still be FAIR. And **A2 says the metadata has to outlive the data.**

This talk is about F and R. Accessible is the cloud-native part, and that is mostly done. Interoperable is what STAC already gives you.

1 Scientists are messy

Data provenance is a luxury

Publishing the paper is the goal. Provenance does not help you publish, so it is the first thing cut.

- ◆ **There was never a standard way to record how a result was made**, so it did not get recorded
- ◆ **The tools did not exist**. Writing down the inputs, the software and the parameters was manual work, every time
- ◆ **Nothing checked**. No reviewer asked for it, no journal required it, no archive rejected a product without it

Thirty years on, that is still the honest description of most labs.

Trust

Every result you use, you decide whether to trust it. That used to be a judgement about people and institutions. AI takes those away.

- ◆ **A model output looks the same whether it is right or wrong.** You cannot tell by looking at it
- ◆ **The volume removed the human check.** Nobody reviews ten thousand derived products, so a bad input is no longer one bad result
- ◆ **Model output becomes the next model's input.** Errors compound, and nothing marks which is which

When you cannot inspect the result, and you cannot ask the person who made it, **provenance is what is left.** It is the only thing that scales as fast as the output does.

A model trained on Sentinel-2 before January 2022 and applied after it saw two different datasets. Nothing in the metadata says so.

Most published analysis cannot be re-run

Two peer-reviewed audits, both of Earth science.

4 of
360 Hydrology papers that reproduced from what the authors published. **89% included no instructions at all**

6 in 100 Remote sensing and machine learning papers that shared their code. 185 reviewed. 44% shared no data either

The second study is this field. Remote sensing, with machine learning on top, which is most of what gets published now.

Stagge et al. 2019, Scientific Data · Ren et al. 2022, Applied Energy

A DOI names a collection, not your scenes

Both agencies version their archives, and they do it properly.

- ◆ **Landsat** — one DOI per sensor, per level, per collection
- ◆ **Sentinel-2** — Collection 0 runs to baseline 03.01. Collection 1 starts at 05.XX. Separate DOIs

That is the right thing to do. It still does not say which scenes you used. And it leaves a gap in the middle:

- ◆ **Baseline 04.00, January 2022.** Every reflectance value in the archive moved by -1000 . That data sits between the two collections
- ◆ Catalogs disagreed about what to do. Earth Search left the offset in. Earth Engine harmonised it. **Same scene, different pixels**
- ◆ Copernicus started deleting those products in August 2025

2 What STAC already does

Three things people use STAC for

Discovery	<i>Find</i> the data. Space, time, collection — plus content extensions that add searchable fields. Electro-Optical, SAR, View
Usage	<i>Open</i> the data. The assets, plus usage extensions that describe it well enough to read without opening it. Projection, Raster, Storage
Workflows	<i>Chain</i> the processing. STAC in, STAC out. The Items are the interface between steps

Discovery is Findable. Usage is Accessible. **Workflows are where Reusable comes from**, and that is the part still being built.

Nothing is read until you ask

A search result is not a list of files to fetch. It is enough to plan the whole computation before touching any of them.

Discover The search returns the Items. Which scenes, and the URL of every file

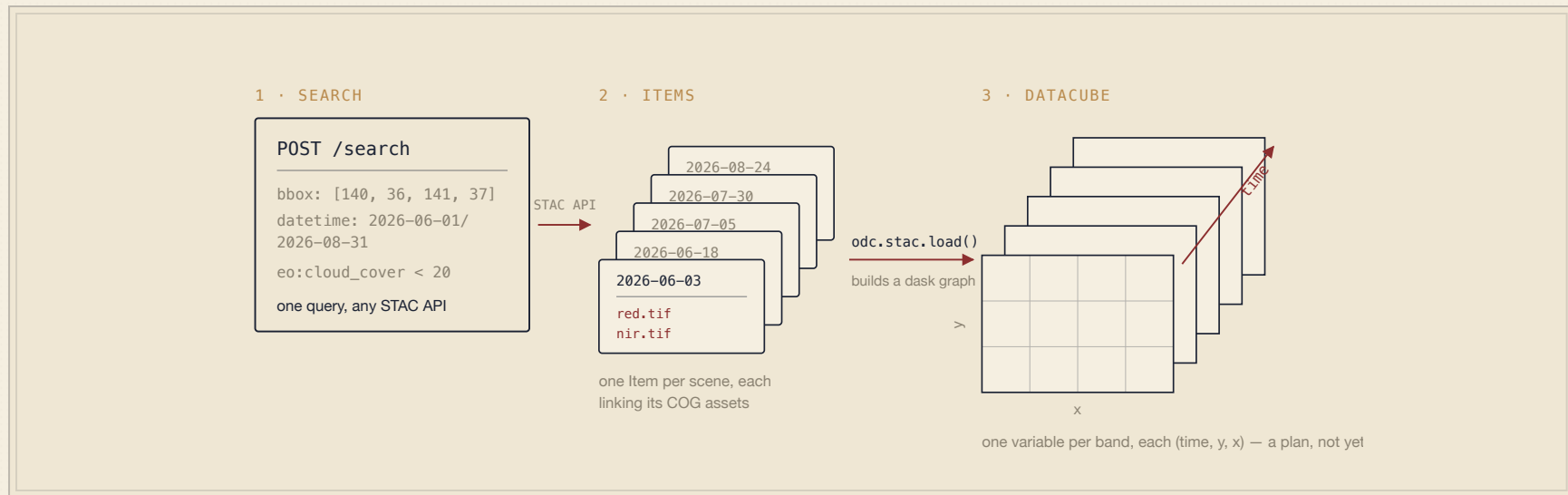
Describe `proj:` carries the CRS, shape and transform. The band metadata carries data type and nodata. **The shape of the cube is known from metadata alone**

Defer `chunks=` builds a Dask graph. A cube spanning terabytes exists as a plan, not as pixels

Read Only when you compute, and only the tiles you need

This is what the extensions are for. Without `proj:`, the loader has to open every file to find out where it is. Which is the download you were avoiding.

One search, one cube

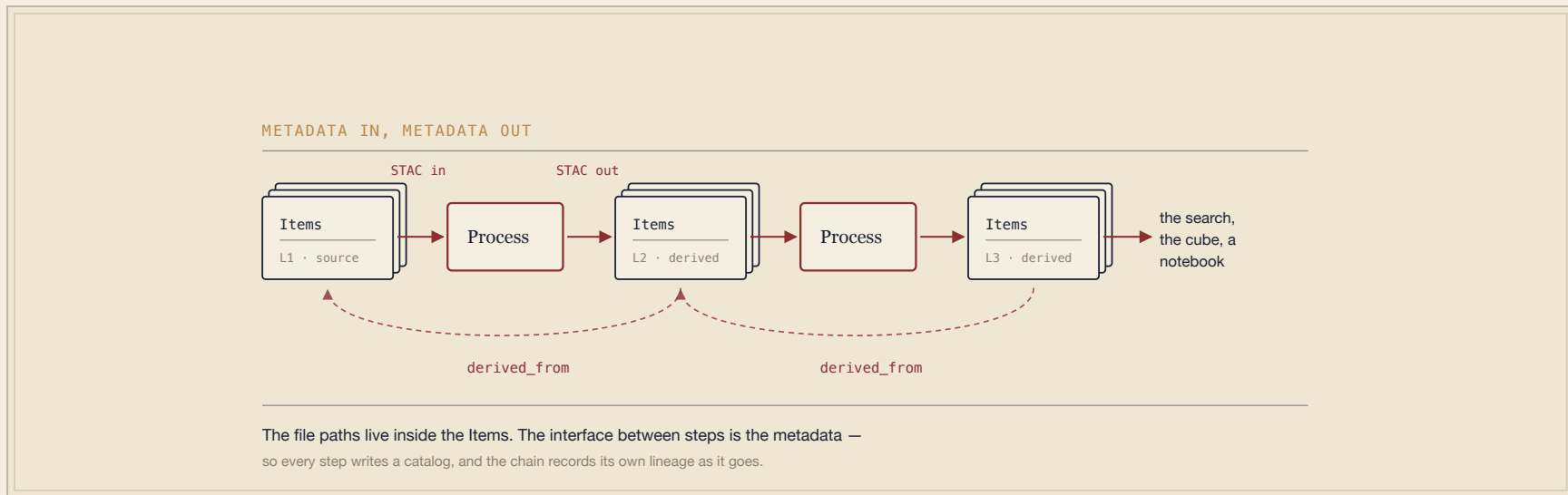


From one search to one cube

The query goes to any STAC API and comes back as one Item per scene. odc-stac stacks those Items into an xarray cube. Nothing is downloaded.

3 Metadata in, metadata out

Every step writes a catalog



Metadata in, metadata out

A processing step takes Items in and puts Items out. Each output carries a `derived_from` link back to what made it, so the chain writes its own history as it runs.

The Processing extension

What a step can record about itself, beyond the `derived_from` link.

<code>processing:software</code>	The libraries that ran, and their versions
<code>processing:datetime</code>	When the run happened
<code>processing:level</code>	L1, L2, L3
<code>processing:version</code>	The processing baseline. For the Sentinel missions this is exactly the 04.00 that moved every value
<code>processing:lineage</code>	What was done. Free text

If producers had been writing `processing:version` , the baseline change would have been visible in the metadata instead of in everyone's results.

Write the catalog as you write the data

Do not catalog afterwards. The processing should write the STAC, at the same moment it writes the pixels.

- ◆ **You know things now that you cannot recover later.** The software version. The inputs. The parameters. Which run this was
- ◆ **A separate cataloguing job drifts.** It runs on its own schedule. It misses files. In the end it disagrees with the archive
- ◆ **Going back later is the expensive way.** It is also the usual way

If the processing writes the catalog, the catalog cannot go out of date. It is part of the run, not a description of it.

What STAC does not give you yet

Being accurate about how far the specification goes today.

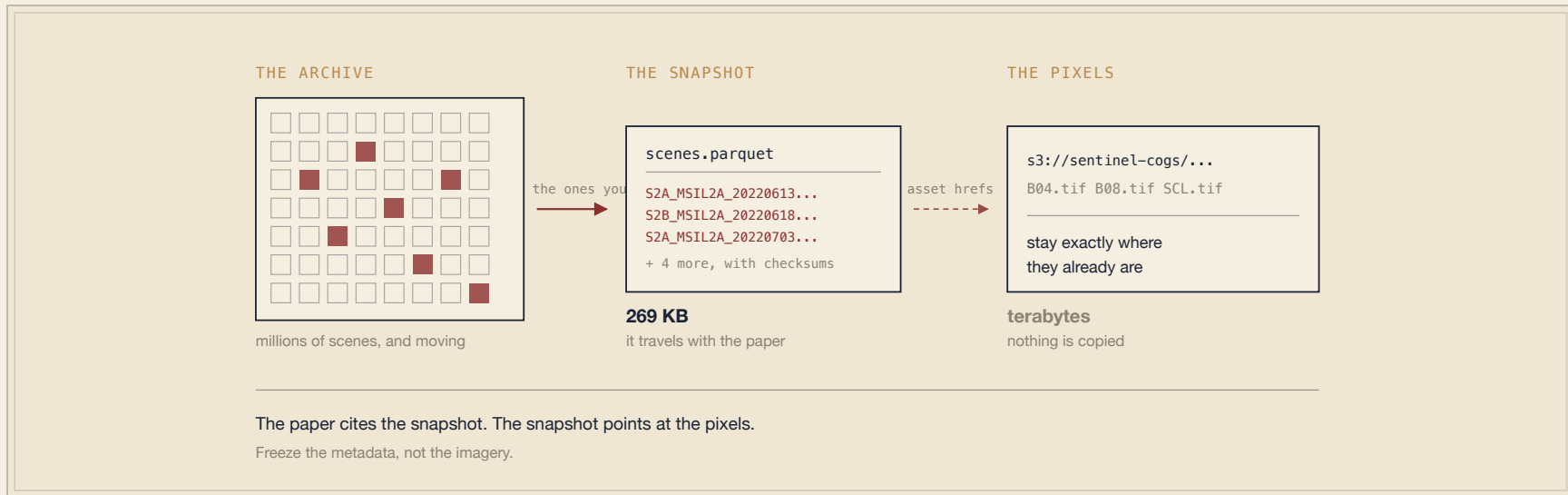
- ◆ `derived_from` is a link you have to write. Emitting STAC from every step records nothing on its own
- ◆ The Processing extension carries the rest, and it is at Candidate maturity. `processing:lineage` is free text
- ◆ There is no provenance extension and no lineage extension

The best practices are usually quoted as "a full provenance model is far beyond the scope of STAC." The rest of that sentence matters:

| *...and the goal is to align with any good independent spec that comes along for that.*

That is a deferral, not a refusal — and the specs already exist. W3C PROV has been a Recommendation since 2013. ISO 19115 has a lineage model. Nobody has done the work of bridging one of them to STAC.

The snapshot



A stac-geoparquet of the scenes you used

Write out the Items your analysis actually used. It is a copy of the metadata, not the data, so it is small enough to publish beside the paper.

Recommendations

If you produce data	Write STAC at every level — raw, L1, L2. Put <code>derived_from</code> on each output. Put <code>file:checksum</code> on the assets. Never reuse a product identifier
----------------------------	---

If you use data	Publish a stac-geoparquet of the scenes you used, beside the paper and its DOI
------------------------	--

The first one is what makes the second one worth anything. A snapshot of a catalog with no lineage and no checksums records what you meant, not what you got.

That is why a data producer should publish a catalog. Not so the data can be found. So that somebody else's result can be checked.

How to get involved

You do not need permission, and you do not need to be a member. You only need to turn up.

The calls	Every other Monday, 11:00 US Eastern — the middle of the night here. And an Asia-Oceania call, first Friday of the month, 12:00 Japan time , which exists so this region does not have to do that
-----------	--

The channels	<code>#stac</code> on the Cloud-Native Geospatial Slack, and the <code>stac-community</code> Google Group
--------------	---

Your own extension	An extension can live in your own organisation. You do not need our permission to write one
--------------------	---

This week	Most of the people who maintain this specification are in this building
-----------	---

What helps most is telling us what does not work. **We only find a gap when somebody reports it.** We are here until Thursday, and we should use the time.

Thank you