

FROST: Federated Registry Of Scientific Things

*Tom Nicholas



*tomnicholas1@gmail.com



@TomNicholas

What I *will not* talk about:

- My blog post (<https://hackmd.io/@TomNicholas/H1KzoYrPJe>)

Science needs a social network for sharing big data

Tom Nicholas, 18th Jan 2025

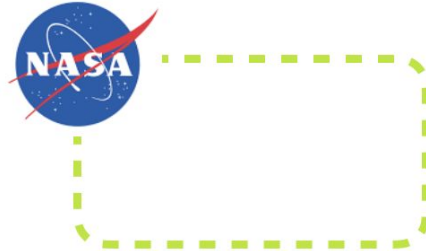
tl;dr: Science is missing a crucial data sharing network. We have the technology - here's what we should build.

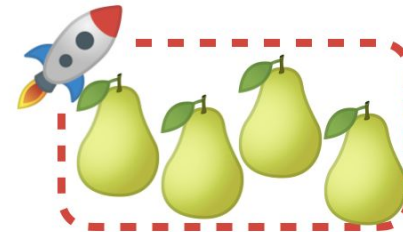
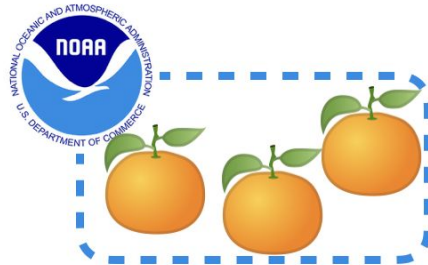
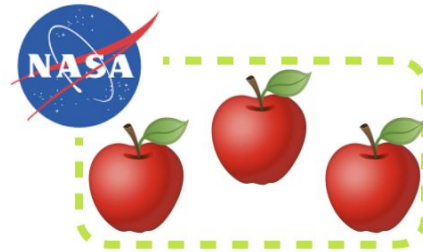
- Instead I'm going to work in the opposite direction
 - From the stack today to the dream

What I *will* talk about:

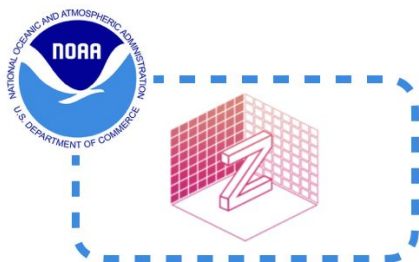
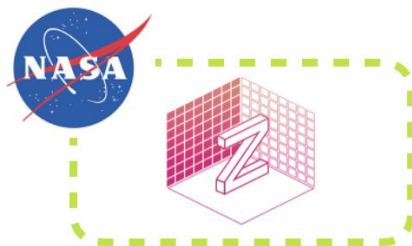
- Version-controlled cloud-native data today
- Connecting the catalogs
- Use cases
- Missing link: FROST
- Design ideas

Today:

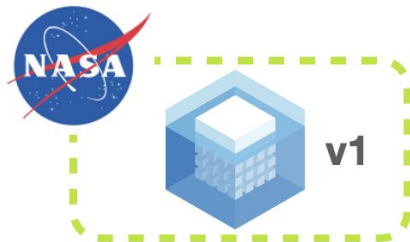




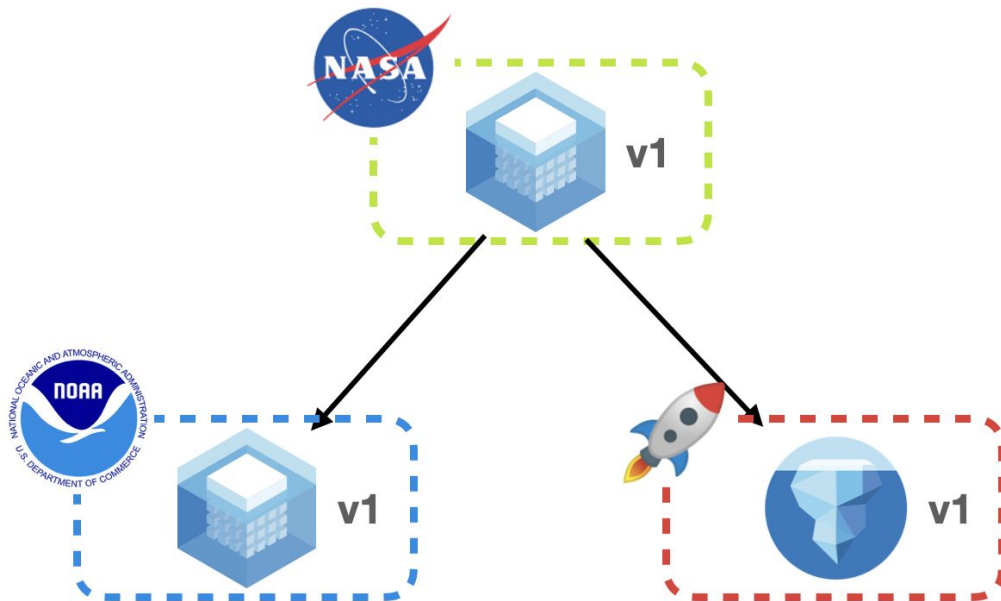
One dataset = URL: s3://some/bucket/data.zarr



One dataset = URL: s3://some/bucket/data.zarr
Version: 4a70ecef278b3sn37

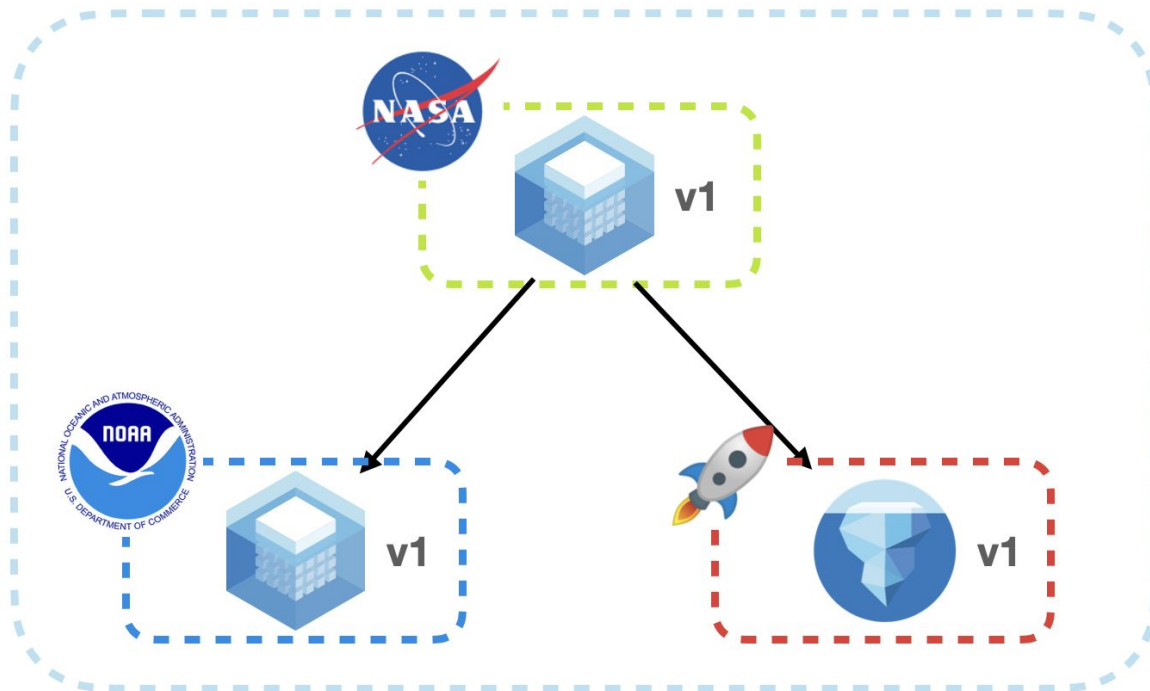


One dataset = URL: s3://some/bucket/data.zarr
Version: 4a70ecef278b3sn37
Dependencies: [(url, hash), ...]

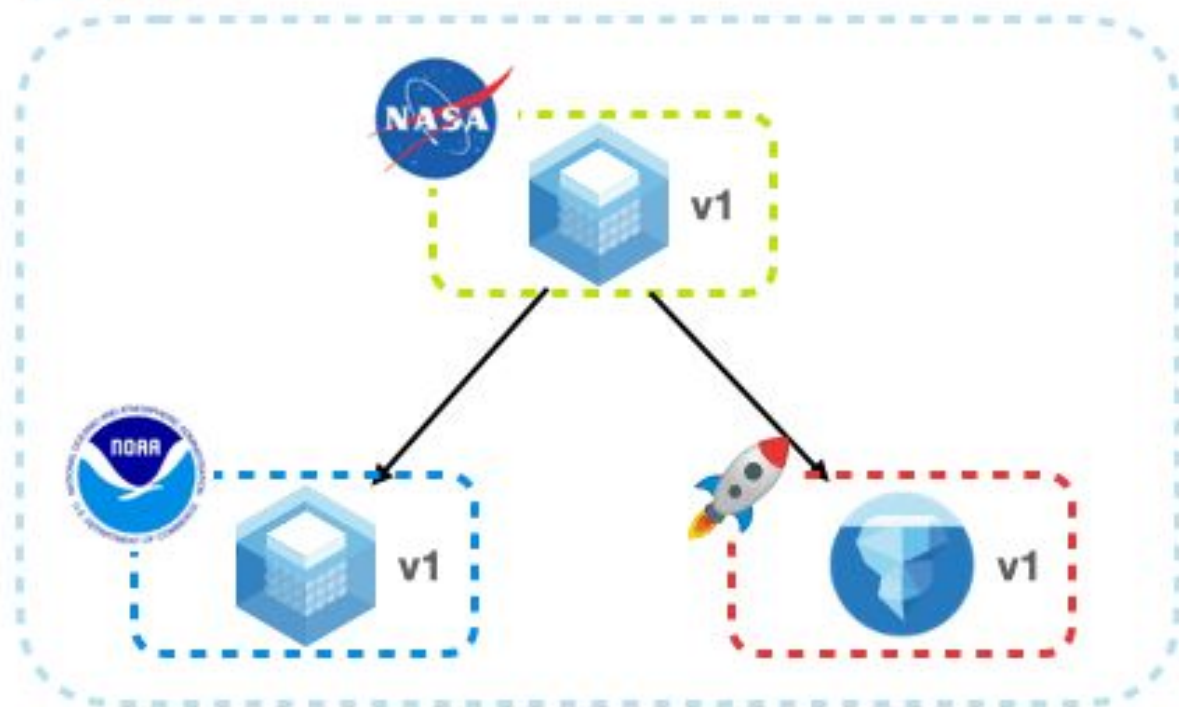


All datasets = {(url, version, dependencies),
(url, version, dependencies),
(url, version, dependencies), ...}

FROST



FROST



Problem: Disconnected catalogs

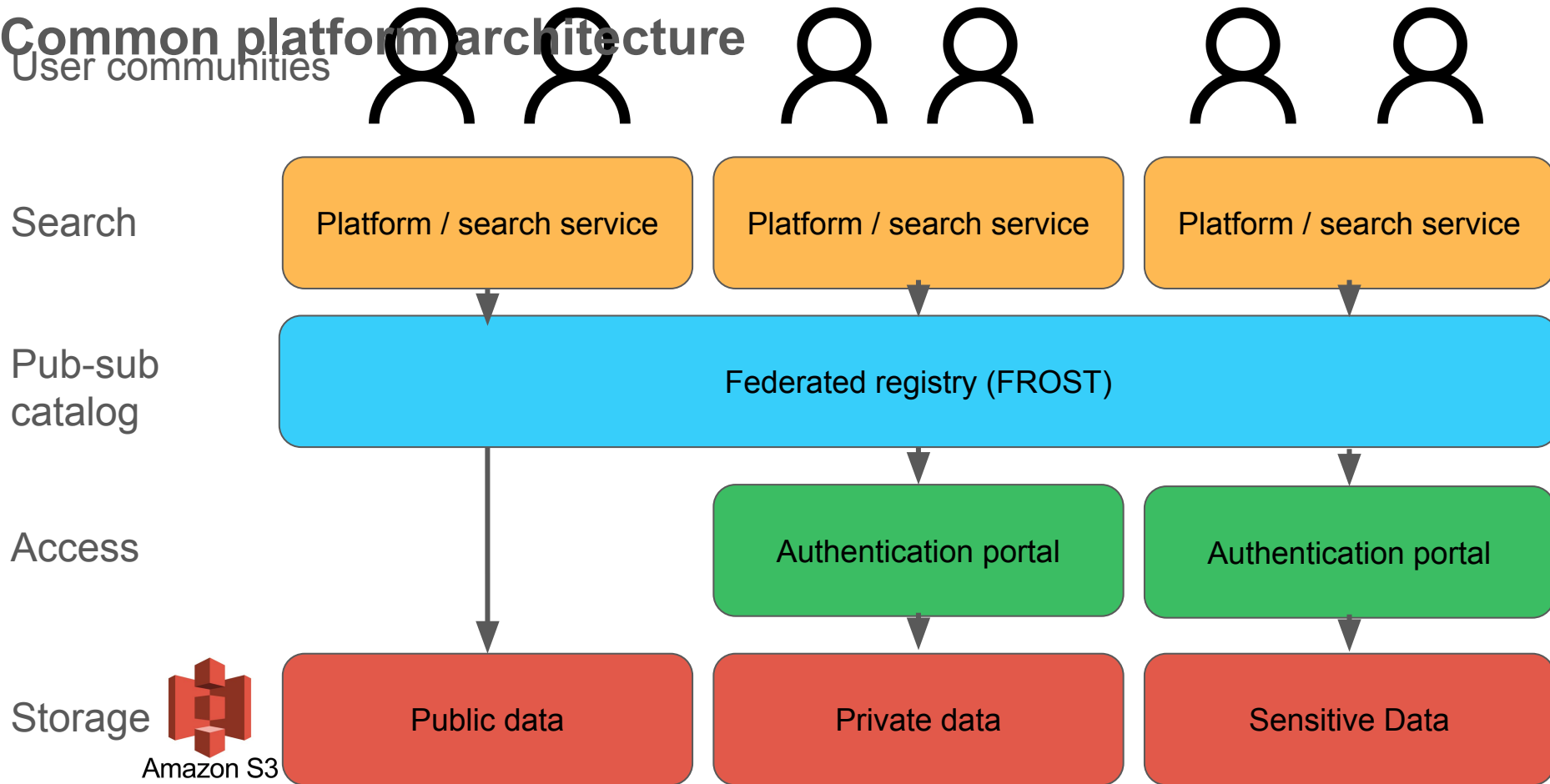
- No cross-org discoverability
- No cross-org update tracking
- Risk of “catalog wars”
- Risk of network lock-in

Solution: Connected catalogs

- Cross-org discoverability
- Cross-org update tracking
- Search entire registry
- Free to move while bringing dependencies

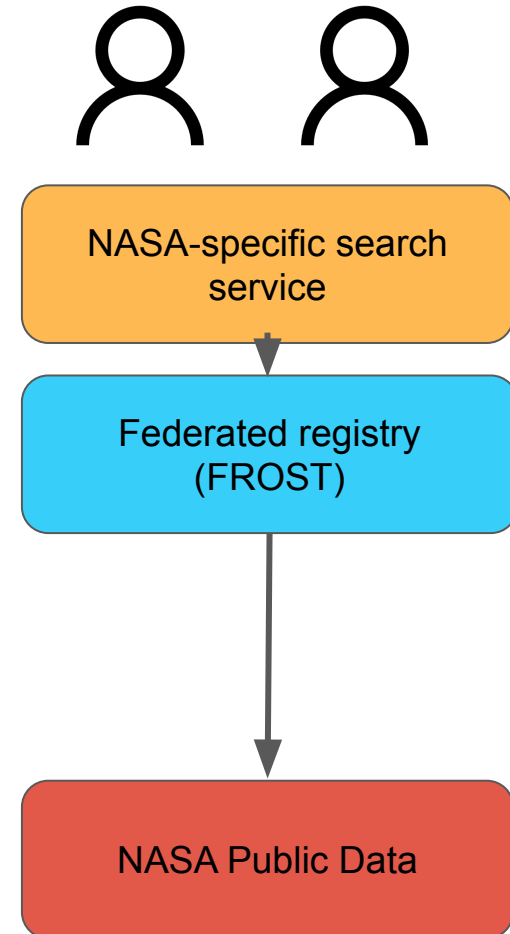
Common platform architecture

User communities



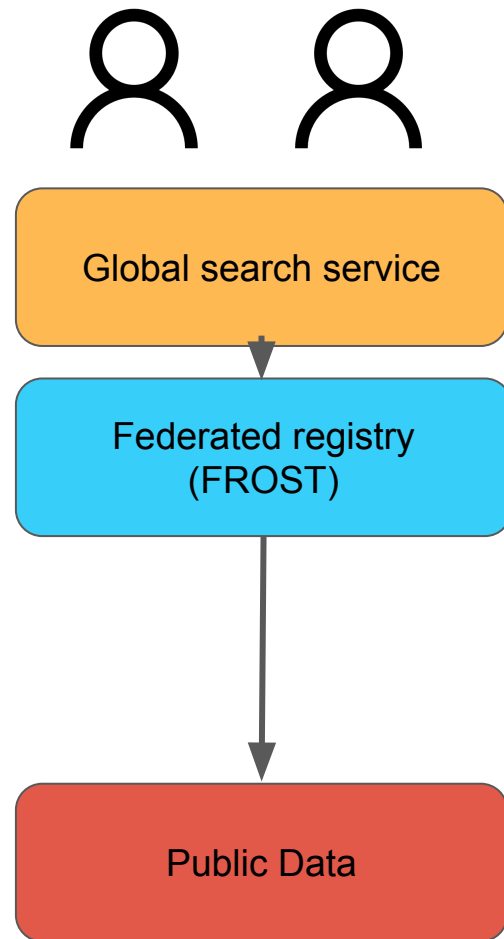
Use cases:

- **Public data catalog** (e.g. NASA)



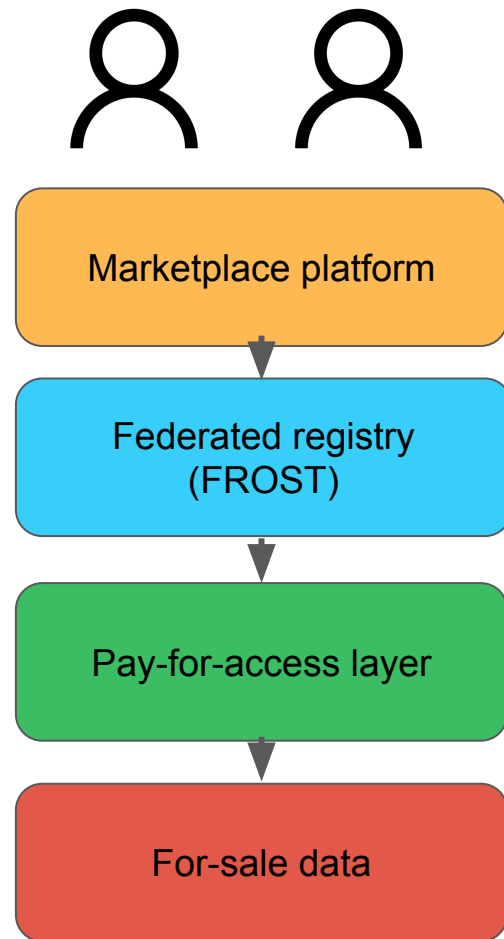
Use cases:

- Public catalog (e.g. NASA)
- **Aggregated search engine** (e.g. Google dataset search)



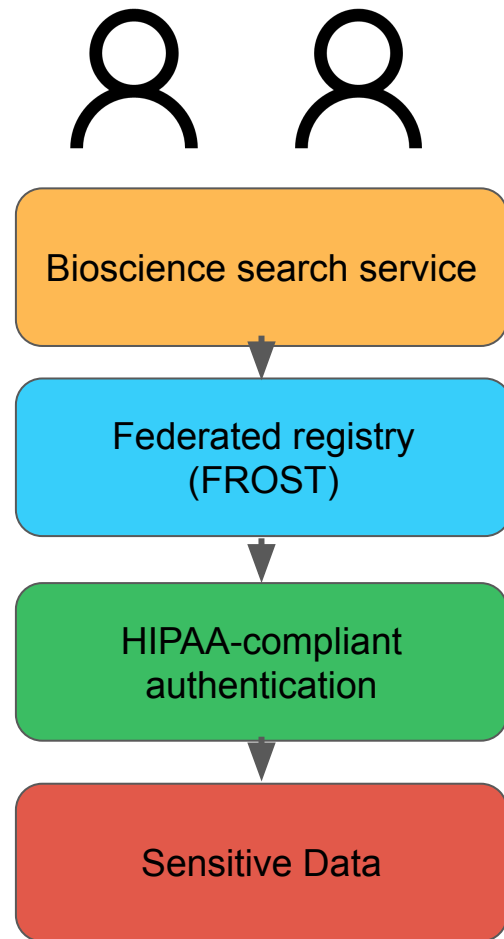
Use cases:

- Public catalog (e.g. NASA)
- Aggregated search engine (e.g. Google dataset search)
- **Data marketplace** (e.g. Source Cooperative)



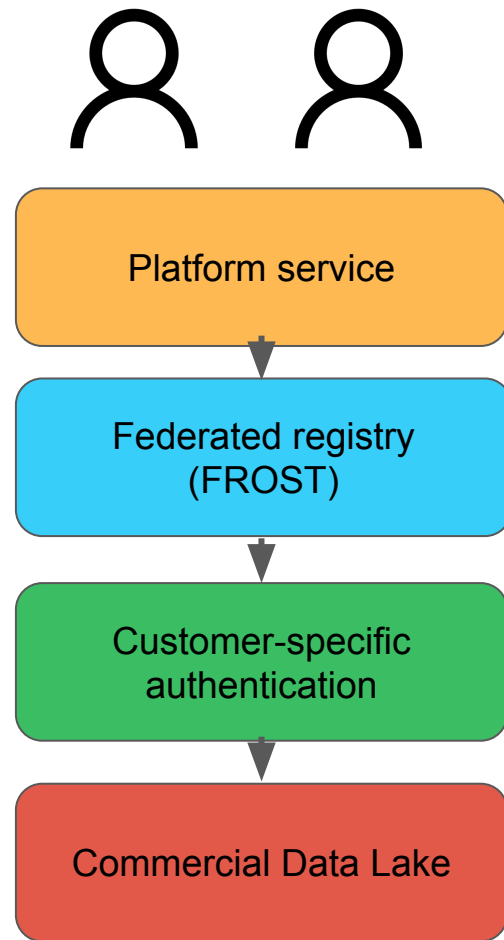
Use cases:

- Public catalog (e.g. NASA)
- Aggregated search engine (e.g. Google dataset search)
- Data marketplace (e.g. Source Cooperative)
- **Federated data analysis** (e.g. lifebit)



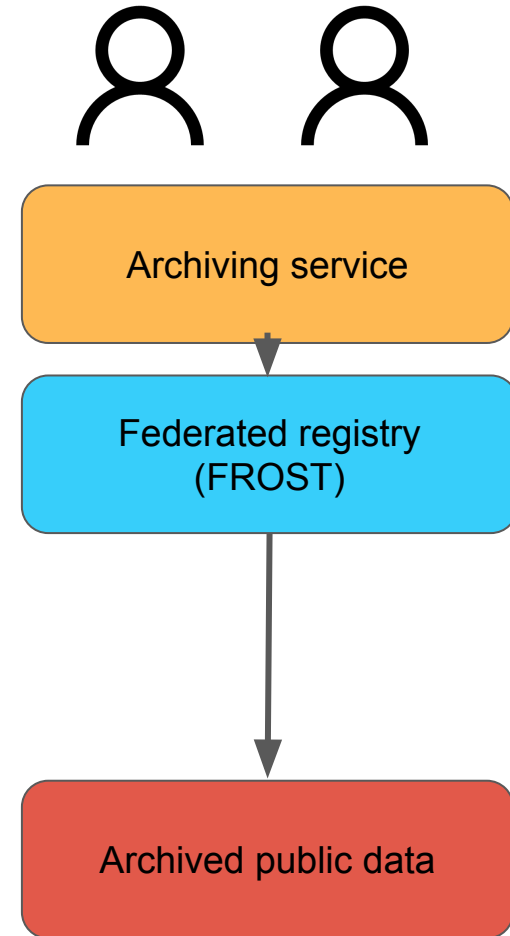
Use cases:

- Public catalog (e.g. NASA)
- Aggregated search engine (e.g. Google dataset search)
- Data marketplace (e.g. Source Cooperative)
- Federated data analysis (e.g. lifebit)
- **Data lake** (e.g. Earthmover's Arraylake)



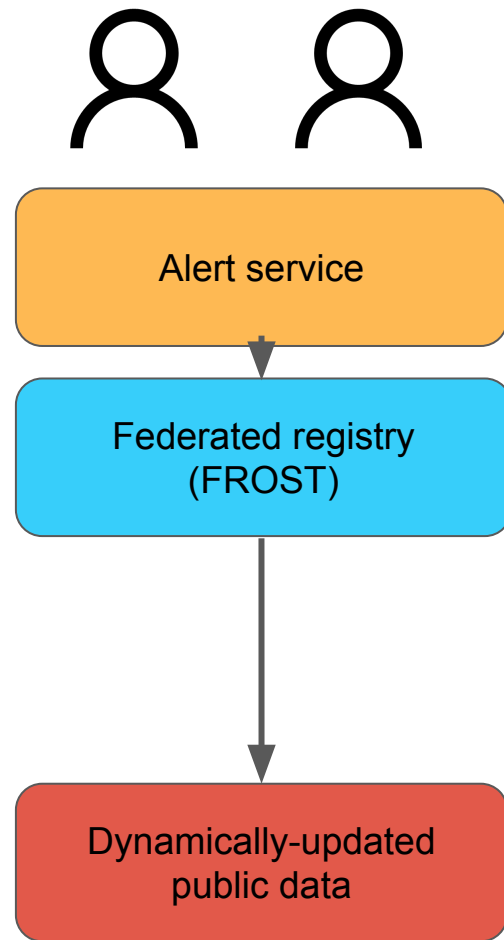
Use cases:

- Public catalog (e.g. NASA)
- Aggregated search engine (e.g. Google dataset search)
- Data marketplace (e.g. Source Cooperative)
- Federated data analysis (e.g. lifebit)
- Data lake (e.g. Earthmover's Arraylake)
- **Backup trawler** (e.g. Wayback Machine)



Use cases:

- Public catalog (e.g. NASA)
- Aggregated search engine (e.g. Google dataset search)
- Data marketplace (e.g. Source Cooperative)
- Federated data analysis (e.g. lifebit)
- Data lake (e.g. Earthmover's Arraylake)
- Backup trawler (e.g. Wayback Machine)
- **Real-time operations** (e.g. WatchDuty)



Missing link: Decentralized pub-sub catalog network

Requirements:

- Defines data catalog entries, updates, and dependencies
- Handle any data at S3-like URL that's version-controlled
- Fast, reliable update notifications
- Federated trust model
- Global search

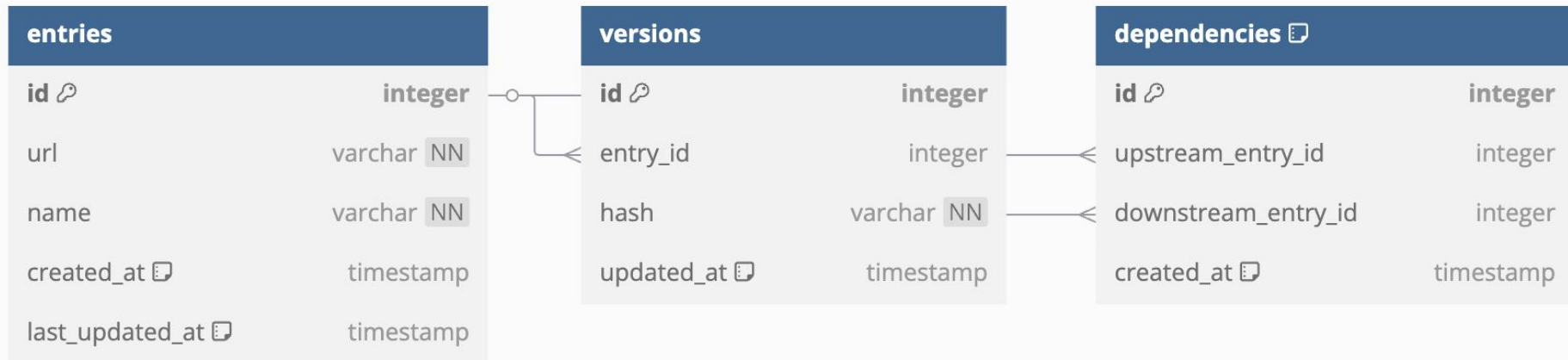
How could we build it?

Design inspirations:

- Pub-sub protocols (e.g. RSS)
- Federated social networks (e.g. Bluesky, Mastodon)
- Database replication (e.g. Raft algorithm)
- Software package registries (e.g. npm)

Design idea 1: Leaderless cross-org database sync'ing

- Schema
 - Entry
 - Version
 - Dependency



Design idea 2: Custom AT protocol lexicon



- Use Bluesky's AT protocol
- Define a custom lexicon
 - `com.frost.dataset.entry`
 - `com.frost.dataset.update`
 - `com.frost.dataset.dependency`
- Publish from your PDS
- Consume via firehose
- Display via feedgen
 - BUT consuming whole firehose forever is basically sync'ing a DB...

Action: FROST working group??

- Do we need a standard / protocol?
- Does this already exist?
- Would have to be a multi-organisation effort

Please indicate interest in participation / following updates here:

<https://tinyurl.com/r76v43u6>



FAQ: Which storage formats?

- Icechunk / Iceberg have nice properties:
 - version-controlled at rest, with a uniquely identifiable address/hash for each commit (i.e. icechunk, git itself)
 - some idea of a diff, potentially one that's small enough to be sent over the network (e.g. git diff, icechunk's ChangeSet)
 - bytes can be pulled out via http range requests to a storage URL

FAQ: Domain-specific features?

- No - github doesn't, why should this?

FAQ: But we surely should verify / police entries?

- No - github doesn't, why should this?

FAQ: Decentralize the storage layer?

- Sure, just make it S3-compatible

FAQ: Why not ActivityPub?

- No global search
- Unreliable update propagation