

Alpenhorn: Managing Data Products for the Canadian Hydrogen Intensity Mapping Experiment

Davor Cubranic
University of British Columbia

CHIME: Canadian Hydrogen Intensity Mapping Experiment

Novel Canadian radio telescope

Designed as a cosmology experiment: map redshifted hydrogen gas as a measure of dark energy

Large field of view, bandwidth, and processing power enable additional experiments:

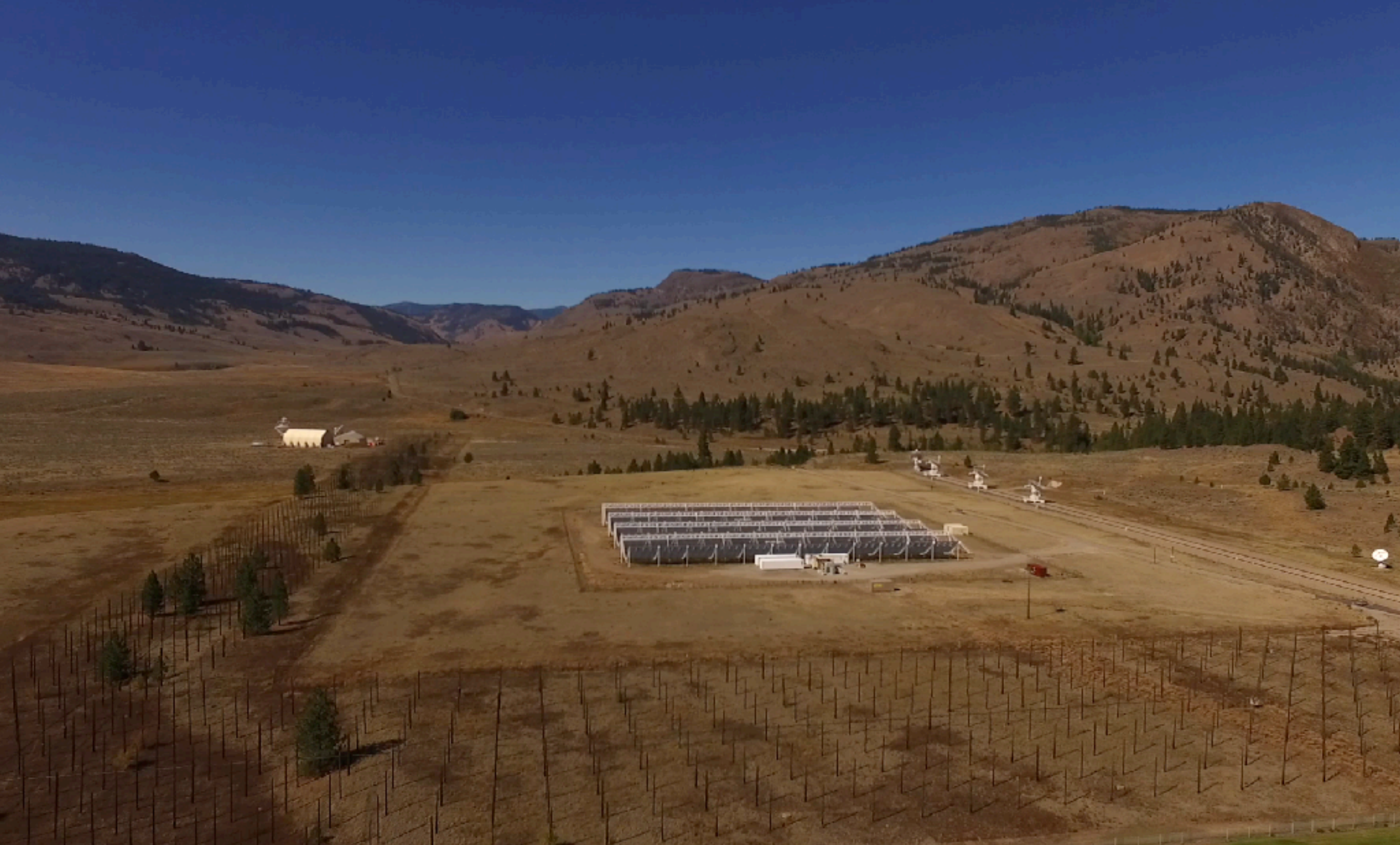
- Pulsar timing survey
- Fast radio burst search

Participating Institutions

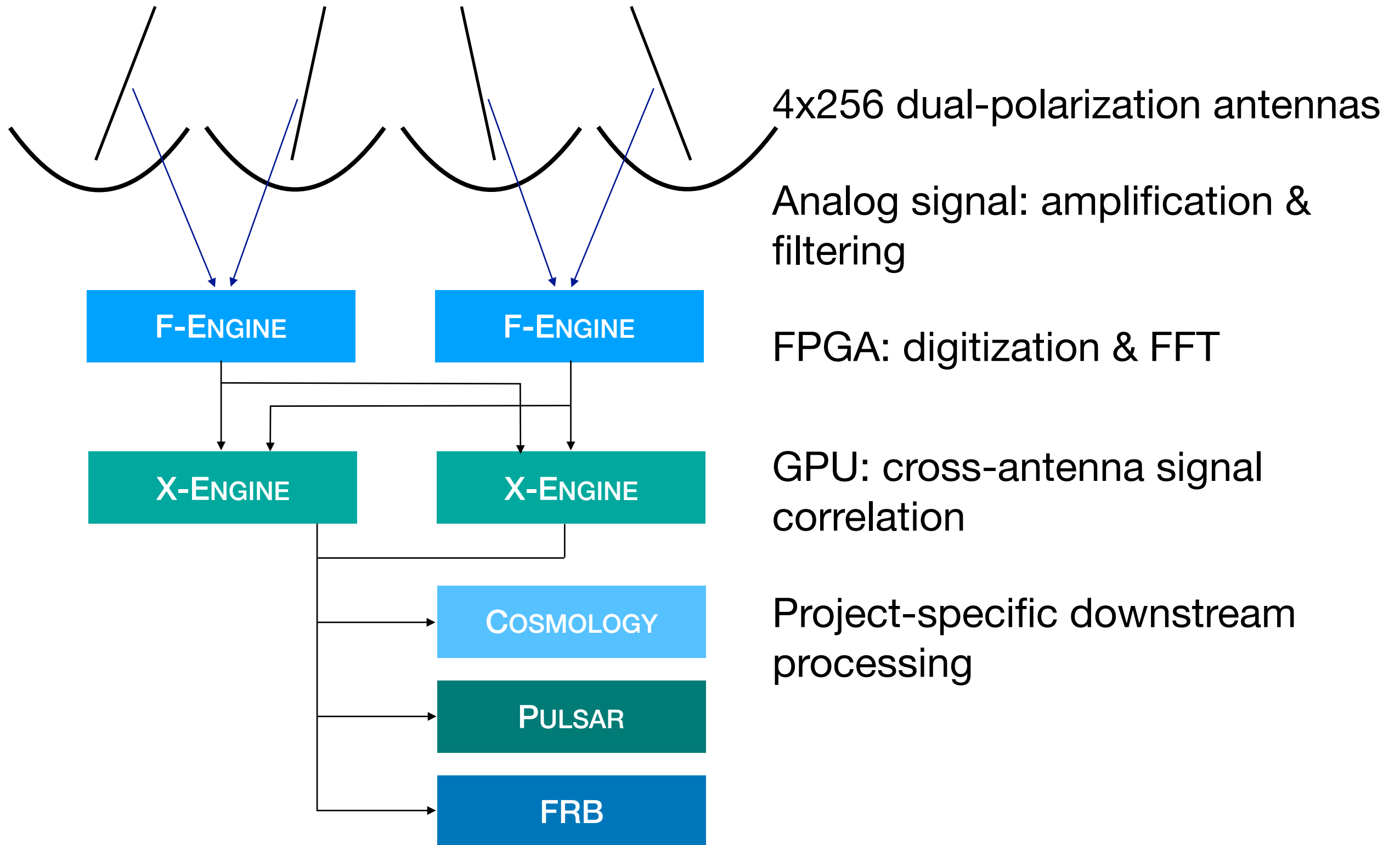
- NRC – Dominion Radio Astrophysical Observatory, Kaleden, BC
- University of British Columbia
- Perimeter Institute, Waterloo
- University of Toronto
- Canadian Institute for Theoretical Astrophysics, Toronto
- McGill University
- National Radio Astronomy Observatory, Charlottesville, Va.
- West Virginia University, Morgantown, WV

CHIME @ DRAO

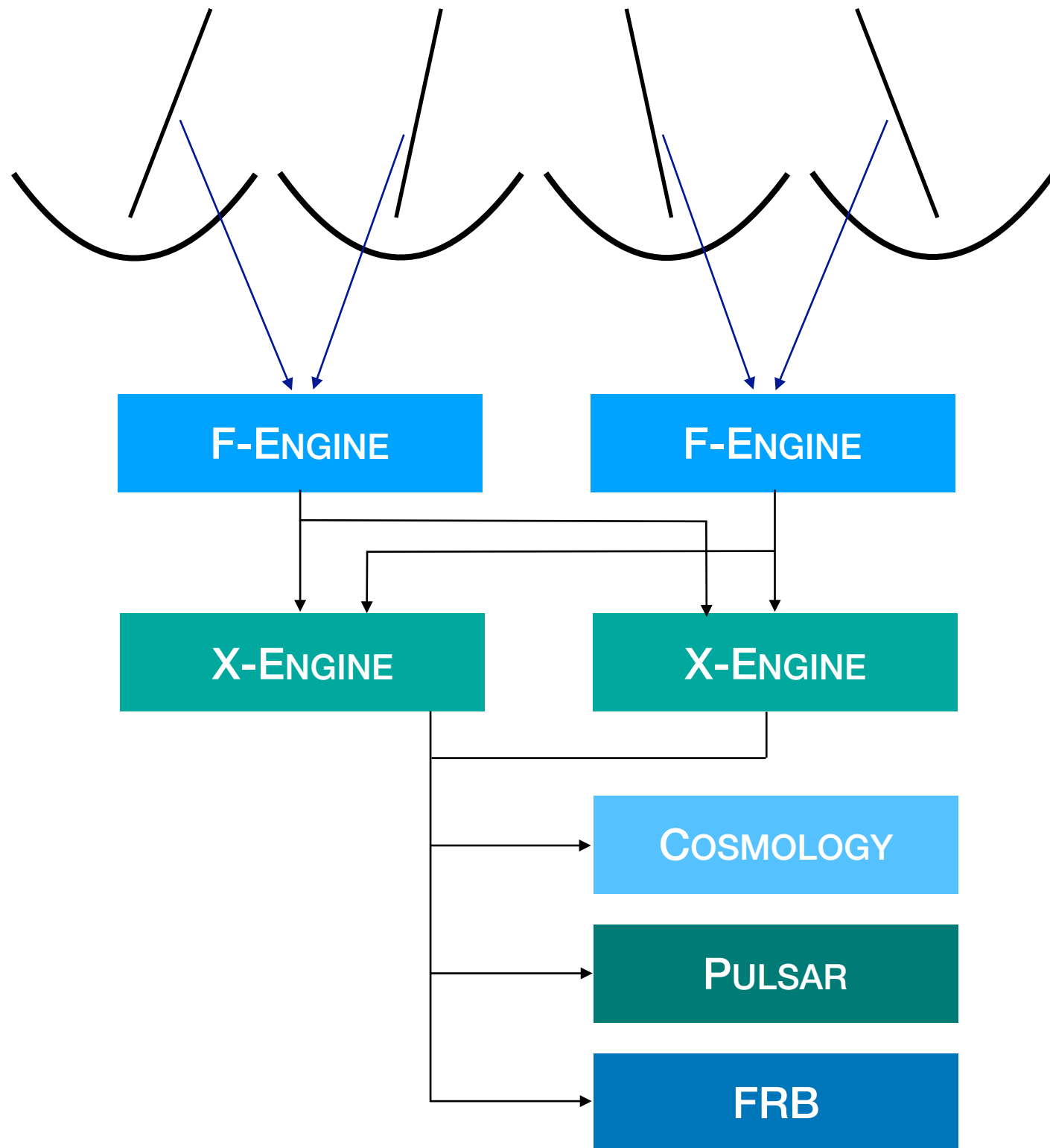




Pipeline



Data Rates



FPGA: 6.5 Tb/s output

GPU: 256 x 25.6 Gb/s input

Cosmology: 2-3 TB/day \approx 0.2 Gb/s

Pulsar: 256x0.25 Gb/s \rightarrow \sim 0.6 Gb/s

FRB: 256x0.55 Gb/s \rightarrow \sim 0.2 Gb/s

1 Gb/s >>> 100 Mb/s



Processing Sites



Managing Data Products

Wide range of data files produced daily

Move data off the location and to the researchers' analysis site(s) safely and reliably

- Replication
- Data integrity checks

Make things findable:

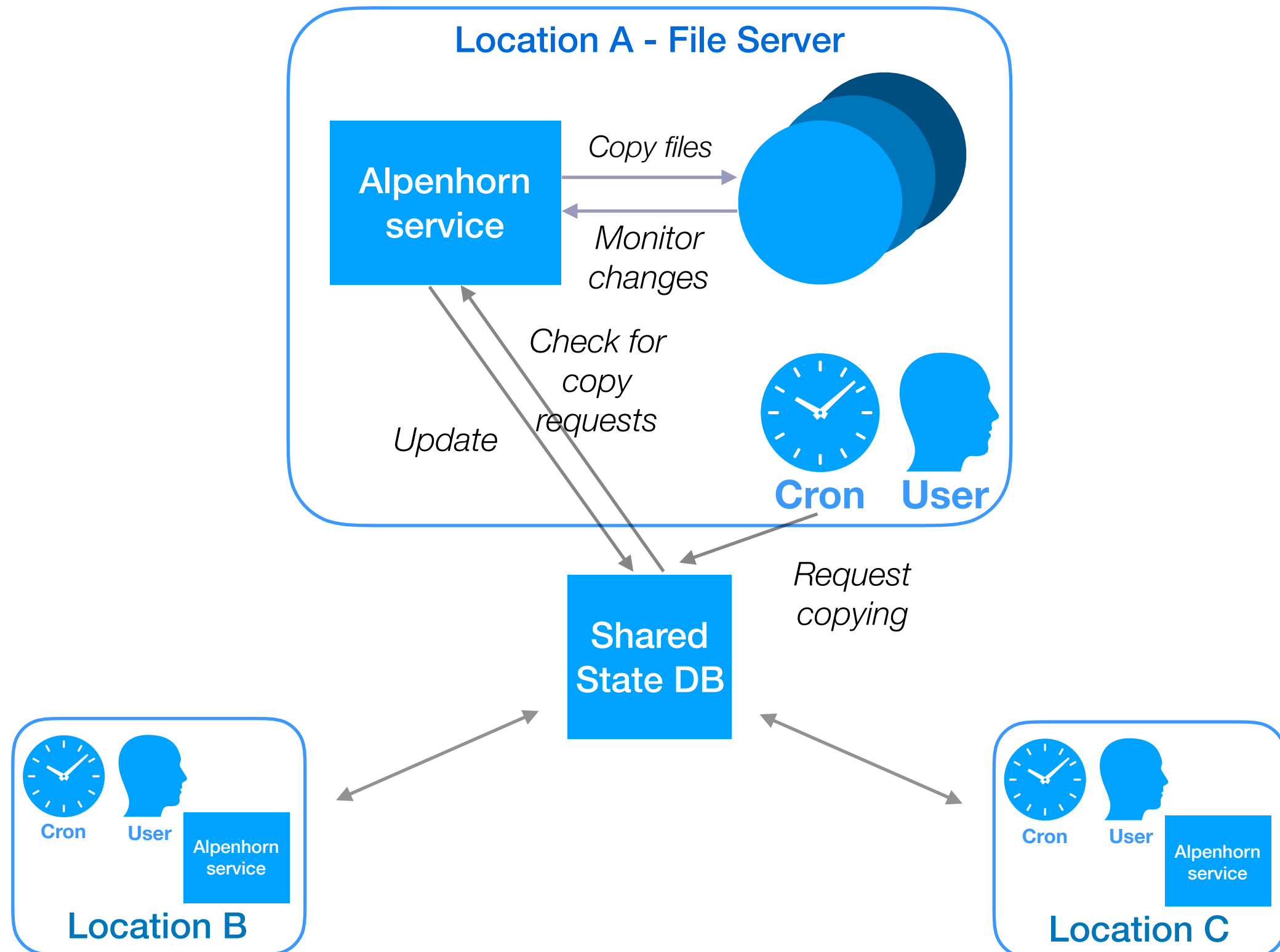
- Where are copies of this file located
- What files have data for X

Keep it simple??

Alpenhorn

- Set of tools for data management and replication
- Developed incrementally by CHIME since ~2013
- Used for the past five years on the CHIME Pathfinder
- Recently extended and generalized to accommodate CHIME FRB and Pulsar projects' data needs

System Architecture



Data Model

Storage:

- Storage node: directory on a host
- Storage group: group of nodes \approx location

Data products:

- Acquisition: uninterrupted collection of data from a single instrument
- Archive file: acquisition component containing data

Data replicas:

- Archive file copy: physical instance of an archive file at a specific location
- Copy request: action of copying an archive file copy to another location

Service

Watches every storage node available on the system for new files matching a registered name pattern

- If new/moved, add archive file copy +[archive file, acquisition] to the database
- If deleted, mark in the database as *absent*
- If a lock file is deleted, process the *locked* file as if new

Periodically:

- Execute archive file copy requests
- Check integrity of suspect files
- Delete unwanted files (*iff* also not-needed)

Transfer Jobs

Moving data between two sites is done with regularly-scheduled “sync” jobs

Request a copy from one storage node all files not available on the destination **storage group**

- The request is executed by the Alpenhorn service that has both source and destination locally *reachable*
- Copy method is configurable (rsync, bbcp, Globus)

In “target” mode, sync copies to a local destination, but deciding what to copy (the “target”) is based on a group that doesn’t have to be local

Demo

Transport Disks

How does Alpenhorn help CHIME manage data offload?

Hot-swap 4-disk enclosures at DRAO and UBC

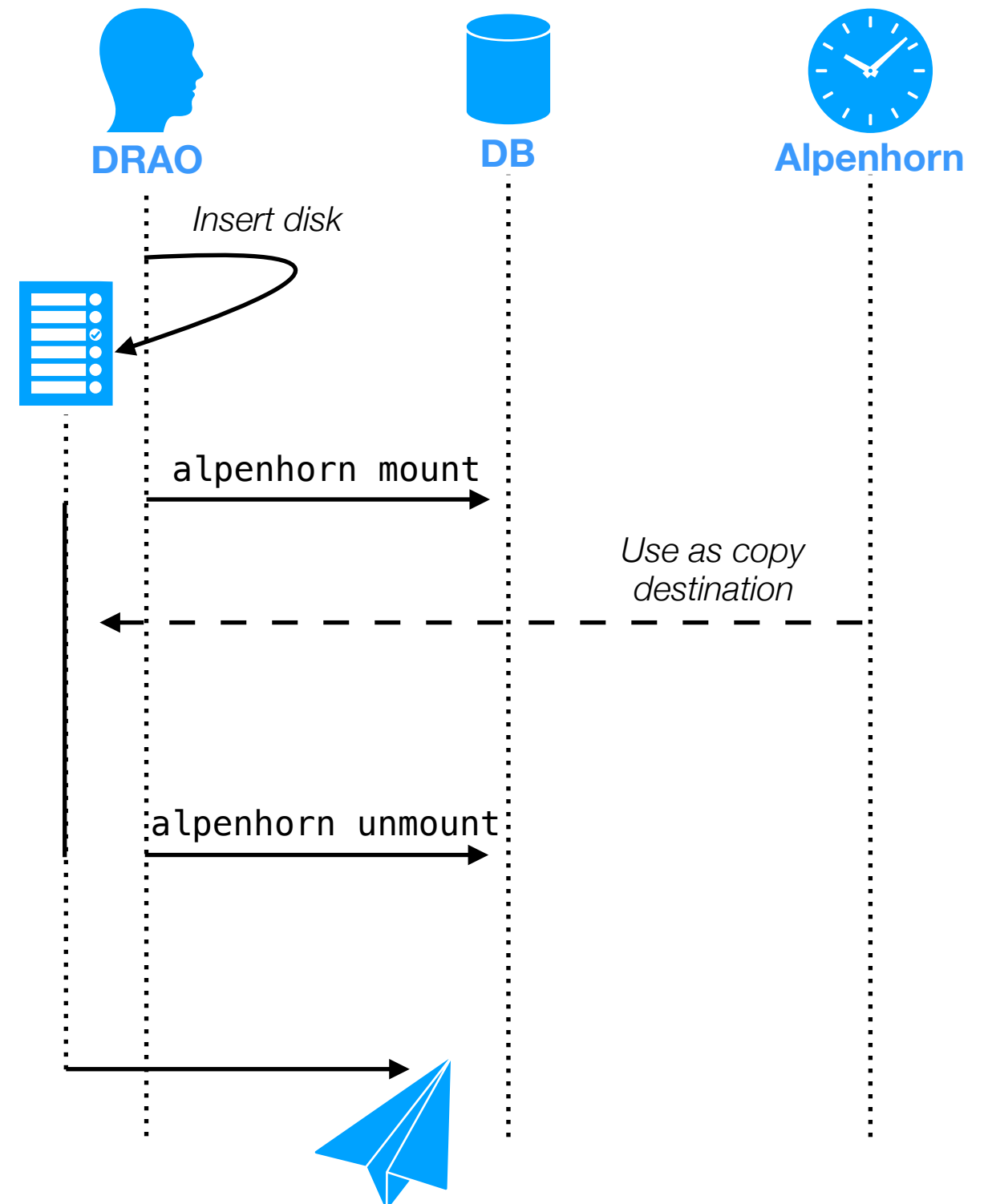
- Enclosure \approx transport storage group
- Individual drives \approx storage node

Cron job at DRAO syncs files to the **transport** group that are on the local **source** and not in the remote **target** group

The Human Interface

The actual workflow for a transport disk:

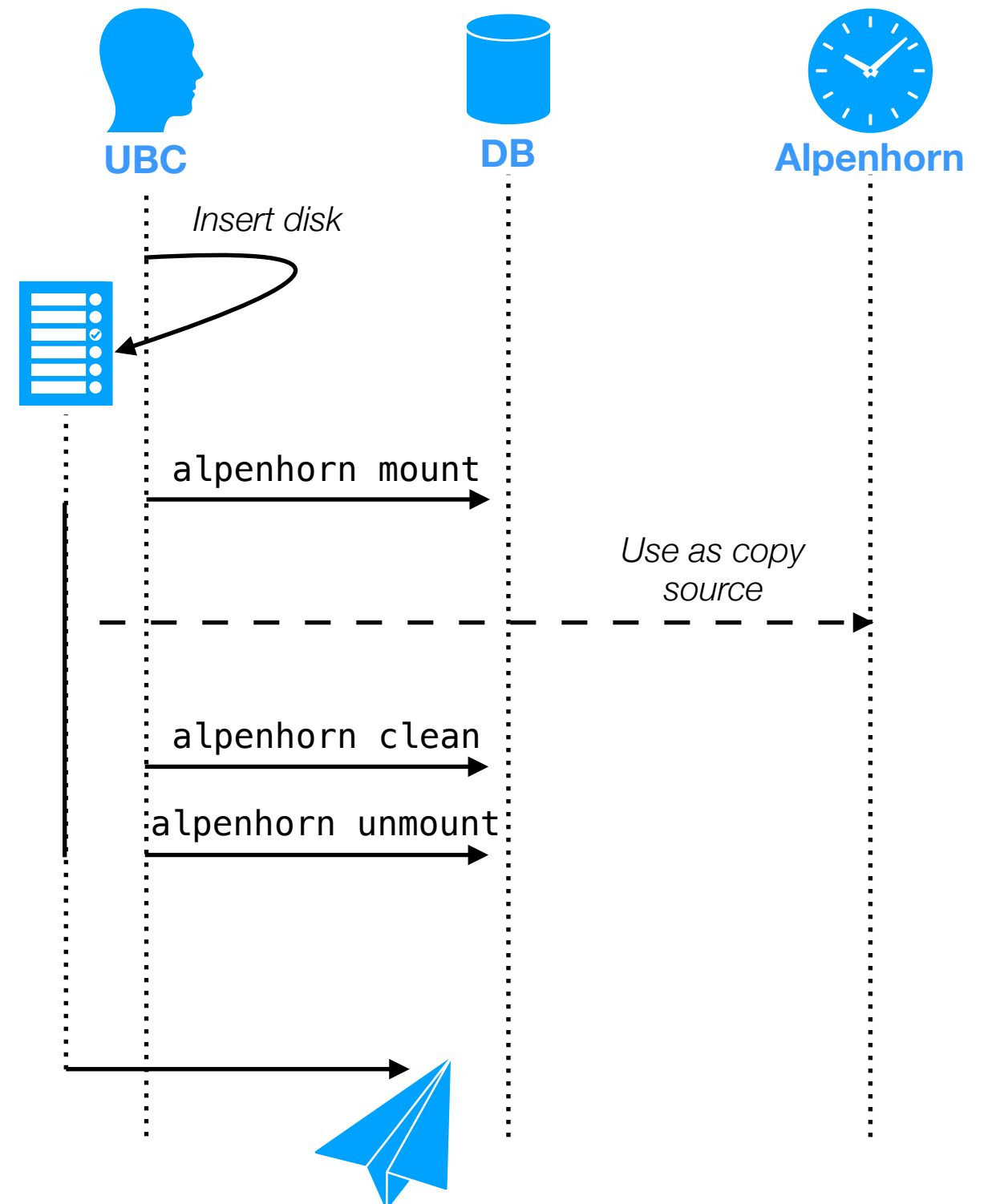
- Site operator at DRAO inserts an empty hard disk into the enclosure and “alpenhorn mounts” it as part of a storage group
- Alpenhorn service at DRAO will automatically use this disk if its group is the destination of a copy request (e.g., issued as part of a cron job’s “sync”)
- When the disk is full, alpenhorn will stop copying to it, and the operator runs “alpenhorn unmount”
- Filled data disk(s) are shipped to UBC



The Human Interface (2)

At the other end...

- UBC operator inserts the full data disks into the enclosure and mounts them as part of the UBC storage group
- Alpenhorn service at UBC registers those files as locally available, and copies them to the local destination if any request is outstanding
- When all files are copied, the UBC operator can “alpenhorn clean” the transport disk and “alpenhorn unmount” it
- Cleaned (empty) data disk(s) are shipped back to DRAO and the process repeats



Demo part 2

Customizing

Acquisitions and archive files have a type

Alpenhorn configuration file specifies the map between pathname patterns and matching type

Built-in “generic” types match using the configured patterns, but don’t keep track of any metadata

Types are dynamically extensible using user-contributed classes

- Must provide a few required callbacks and properties
- Can perform arbitrary processing to extract metadata when called-back on new archive file events
- This metadata usually goes into type-owned tables in the DB

Summary

Alpenhorn is a set of tools for managing an archive of scientific data across multiple sites

Automatically:

- tracks all copies of a single file,
- handles available disk storage on the destination, and
- ensures file integrity and sufficient replication

CLI for cron scripts and interactive use

Written for the CHIME radio-telescope, but includes a framework for user-provided customization

**github.com/
radiocosmology/
alpenhorn**