



An end-to-end time-domain and multi-messenger astronomy platform

Dan Avner, NSF NOIRLab, RAPID

Discovery is no longer the bottleneck

- Millions of alerts per night
- Multi-wavelength follow-up required
- Rapid response is critical



Credit: NSF-DOE Vera C. Rubin Observatory

The workflow is fragmented

Alerts & Brokers

TOM

Telescope

Pipeline

Analysis

Archive

The workflow is fragmented

Alerts & Brokers

TOM

Telescope



Pipeline

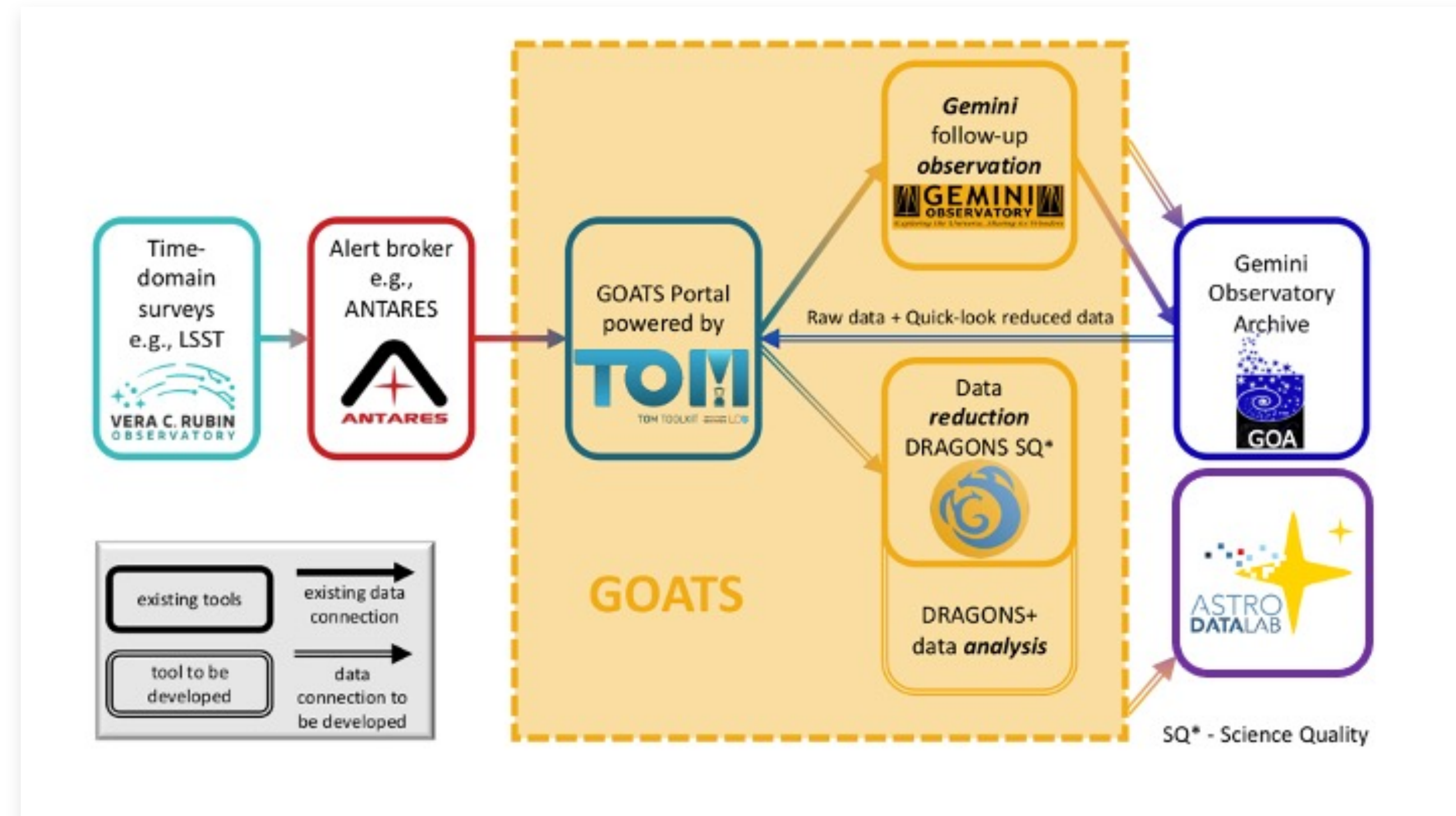
Analysis

Archive

GOATS

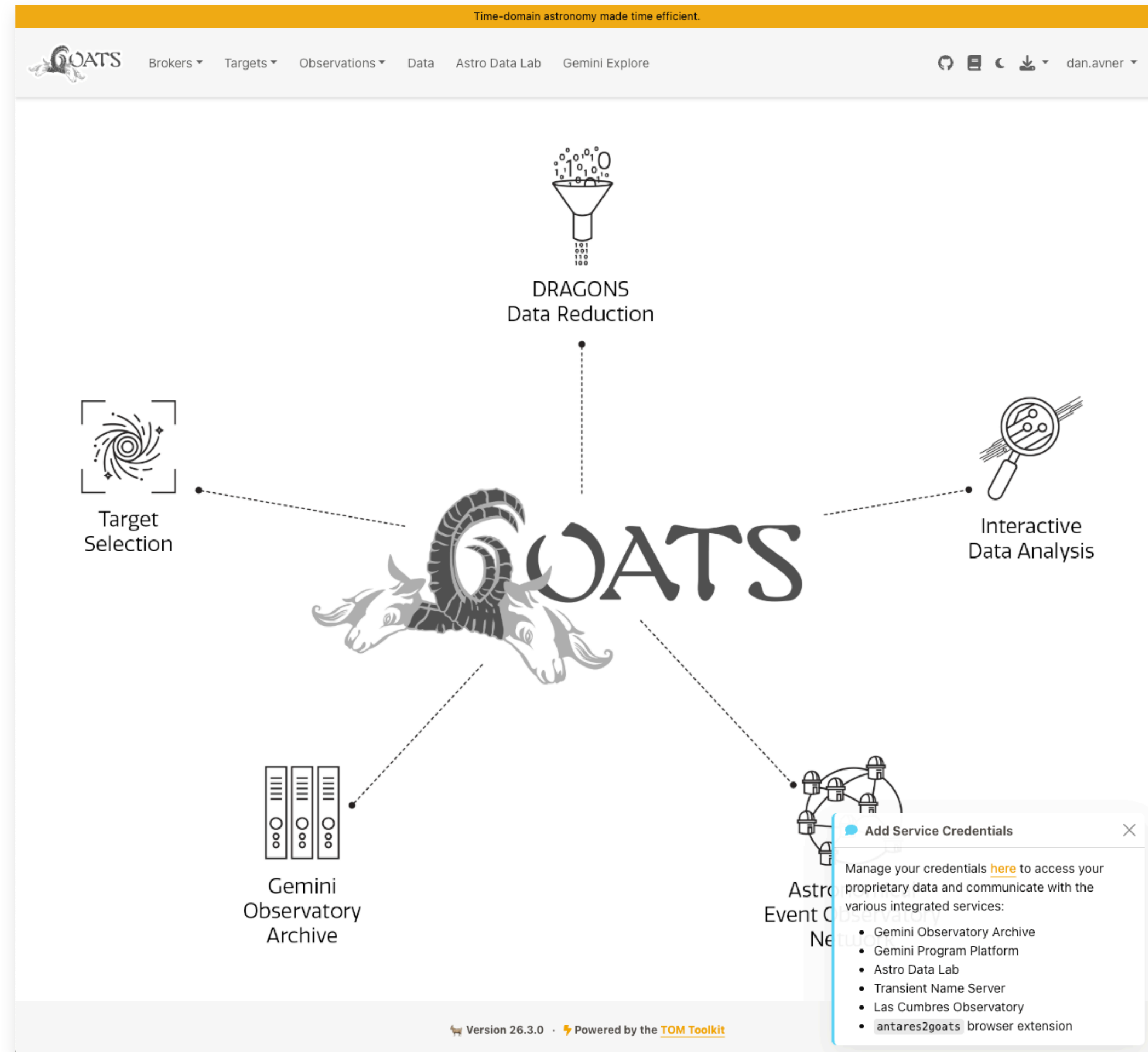
Gemini Observation and Analysis of Targets System

- Local, browser-based application
- No cloud deployment required
- Integrated end-to-end workflow
- Asynchronous data processing
- Unified interface across systems



GOATS

Software architecture



Core Application

- Django application (built on TOMToolkit)
- Python backend

User Interface

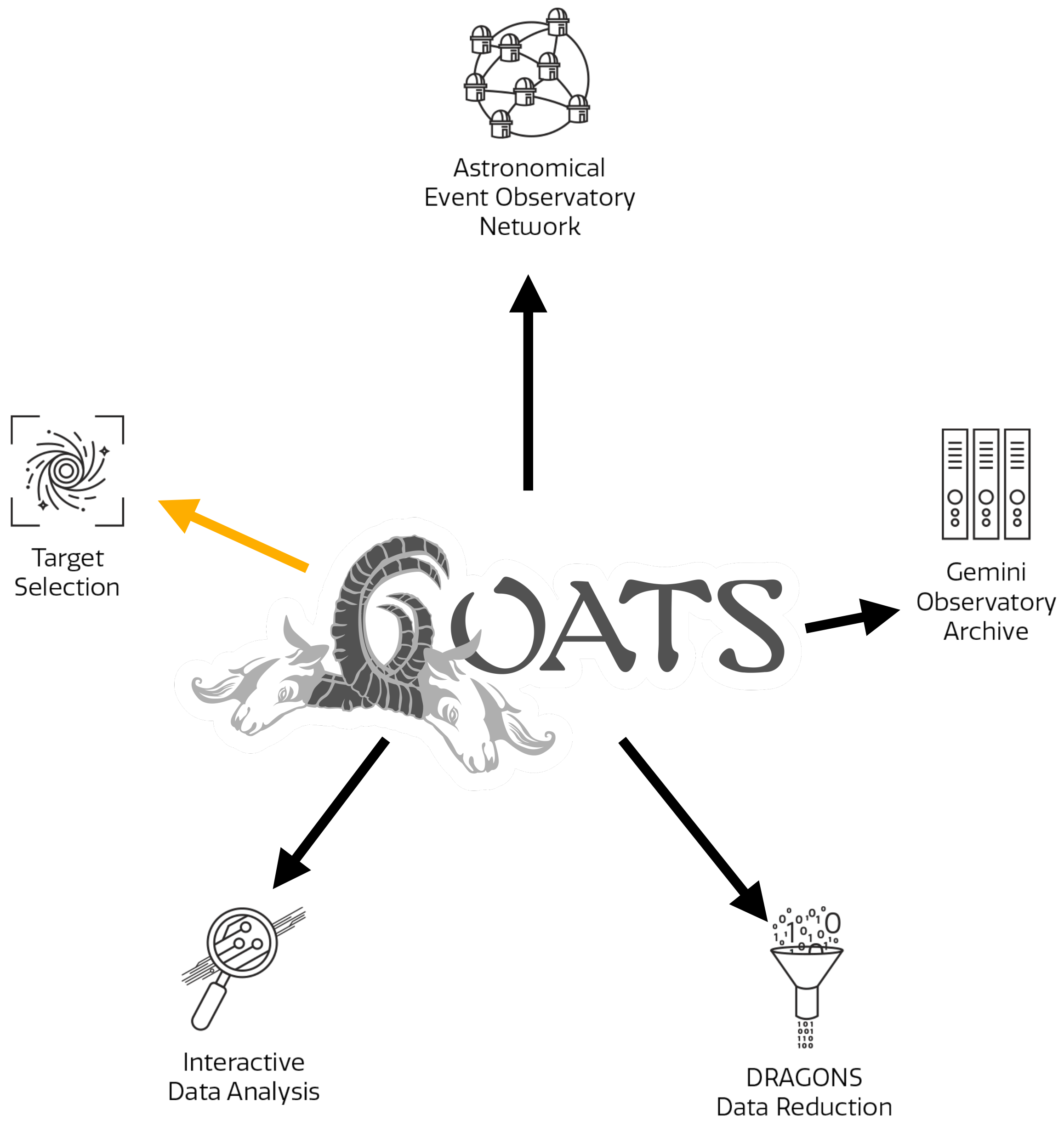
- Browser-based UI (vanilla JS)

Execution Model

- Asynchronous background tasks
- Designed for long-running workflows

Deployment

- Distributed as a Conda package
- Runs locally



antares2goats

From alert to target in one step

- Browse alerts in ANTARES
- Select object of interest
- Send directly to GOATS

- ▶ *No manual data entry*
- ▶ *No context switching*



The screenshot shows the 'antares2goats' application window. It features a dark header with the application name and a close button. The main content area is a light grey form with several sections:

- Token:** A text input field containing a series of dots, indicating a masked token.
- Connection Details:** A section with two radio buttons: 'GOATS Default' (selected) and 'Other'. Below this are two text input fields: 'URL:' with the value 'http://127.0.0.1' and 'Port:' with the value '8000'.
- ANTARES:** A section with two radio buttons: 'Production' (selected) and 'Development'.

At the bottom of the form is a large yellow button labeled 'Save'.

antares2goats

From alert to target in one step

- Browse alerts in ANTARES
- Select object of interest
- Send directly to GOATS

▶ *No manual data entry*

▶ *No context switching*



The screenshot shows the app listing for 'antares2goats' by 'GOATS @Gemini'. It includes a description, a star rating of 5 (2 reviews), and a user count of 1. Below the rating is a star rating interface with five stars and a bar chart showing the distribution of ratings.

antares2goats
by **GOATS @Gemini**

Enhance your GOATS experience by seamlessly integrating ANTARES directly into GOATS with just a single click.

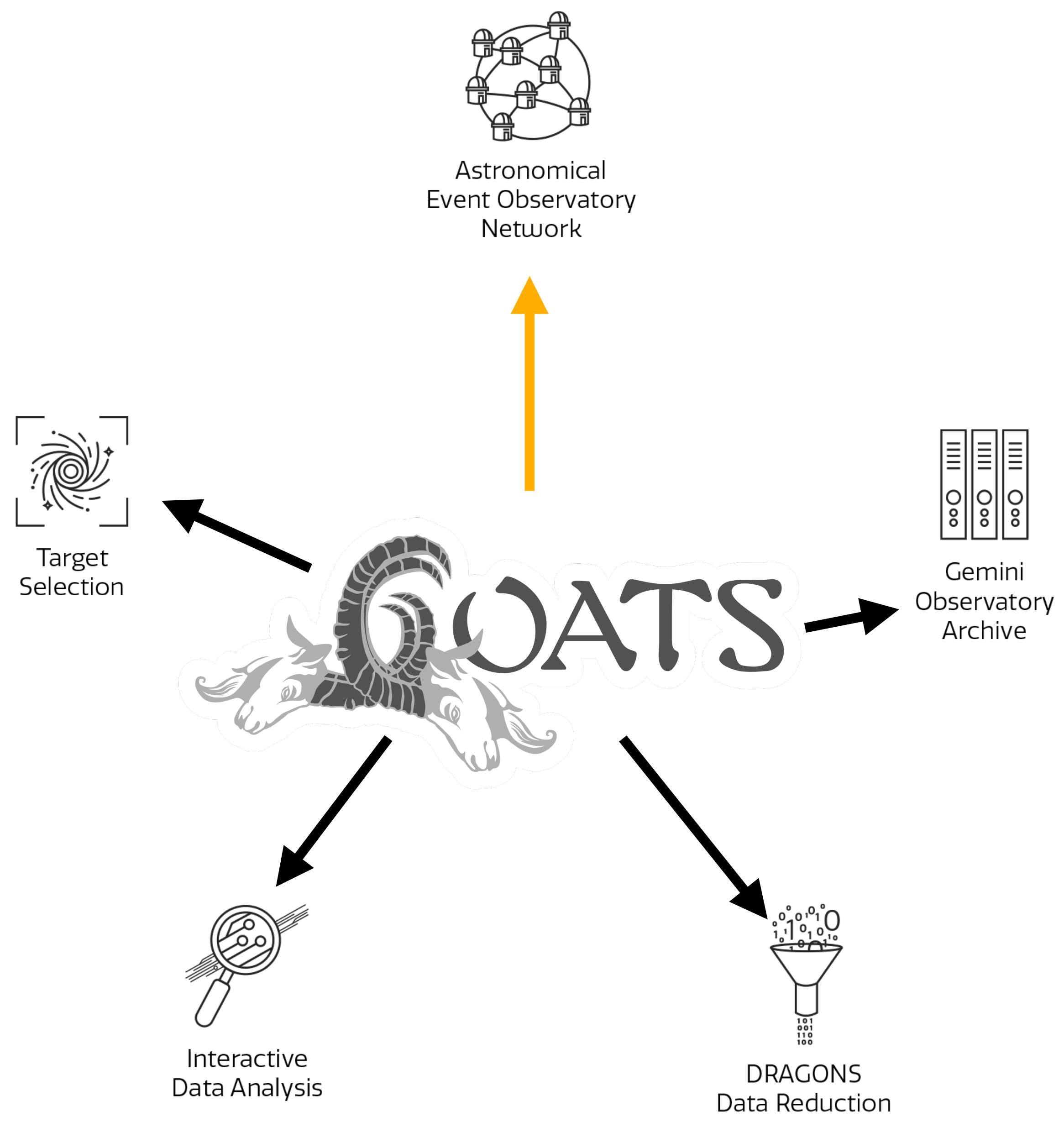
★ 5 (2 reviews) 1 User

Rated 5 by 2 reviewers

Click to rate: ★ ★ ★ ★ ★

5	★	2
4	★	0
3	★	0
2	★	0
1	★	0

[Read all 2 reviews](#)



GOATS & Gemini

GOATS as an interface to GPP

- GPP: next-generation observatory control system
- API-driven ToO observation creation and submission
- GOATS provides a unified user interface

► *Simplifies interaction with complex observatory systems*

Gemini Program Platform

Use the Gemini Program Platform (GPP) to browse your active programs and corresponding observations. Select a program to load its observations and autofill observation details. You can then save the observation on GOATS without changes, update the observation details and resubmit, or create a new observation for a ToO. Any updates or new observations are saved on GOATS automatically upon submission.

Active Programs

G-2026A-0331-D - 2026A Transients

Active Observations: Choose an observation... Approved ToO Configurations: o-4b51 - ToO Target 1

Update On GPP Save To GOATS Create On GPP & Save To GOATS

Details

Brightnesses

Constraint Set

Scheduling Windows

Configuration

Position Angle Mode: Average Parallactic

Instrument: GMOS South Position Angle: 0 deg

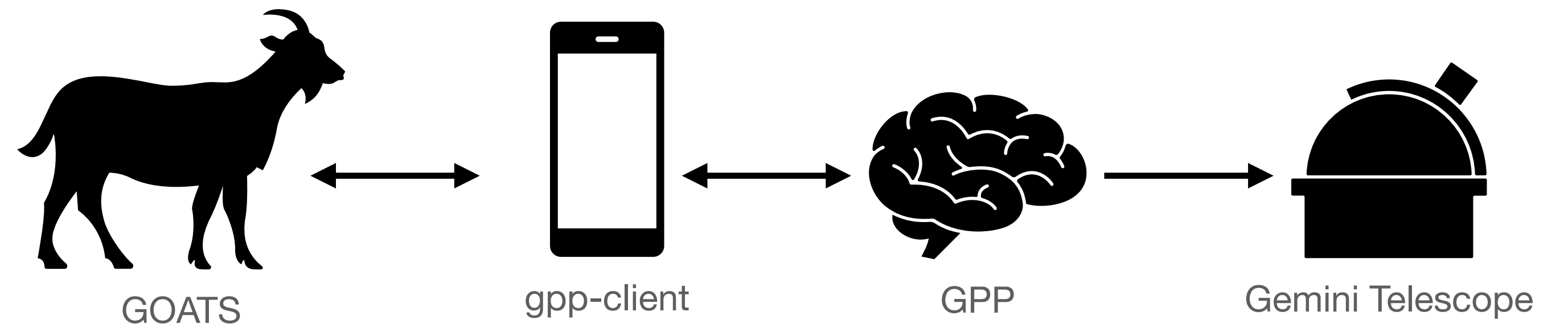
Grating: B480 G5327 Filter:

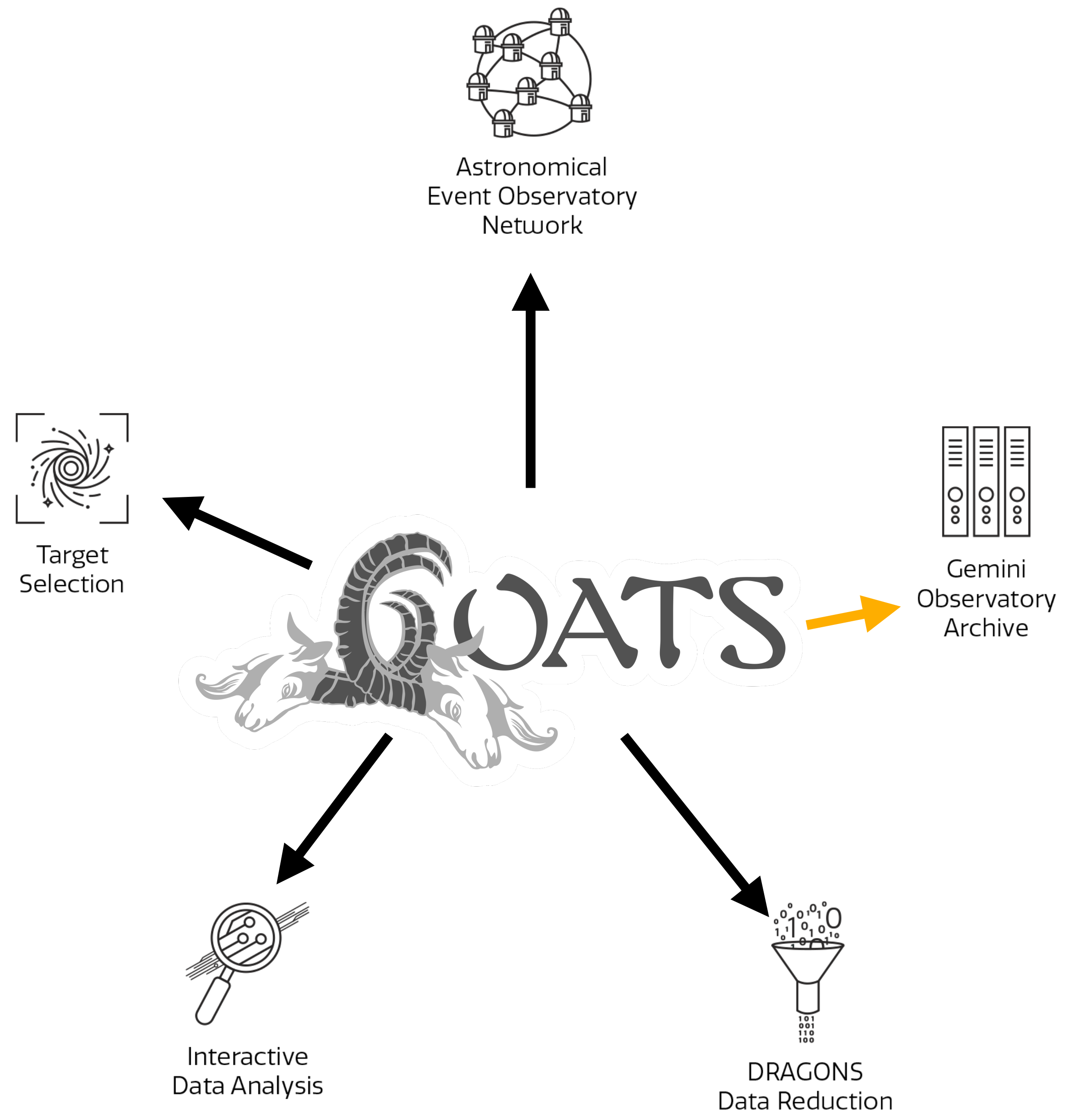
FPU: Longslit 1.00 arcsec Spatial Offsets: 0.00, 15.00, -15.00 arcsec

GPP Client

Programmatic interface to Gemini

- Asynchronous Python API for GPP
- Enables programmatic ToO submission
- Simplifies interaction with observatory system
- Used by GOATS for workflows





GOATS & GOA

Accessing data through GOATS

- Retrieve observation data directly from Gemini Observatory Archive
- Supports public and proprietary data
- Integrated into the follow-up workflow
- No manual interaction with archive interfaces

Retrieve data from the Gemini Observatory Archive (GOA)
This functionality is available once the observation is complete, i.e., the status is updated to 'Observed', 'Ongoing', or 'Completed'

[View at Archive »](#)

Observation Class: Any
Observation Type: Any

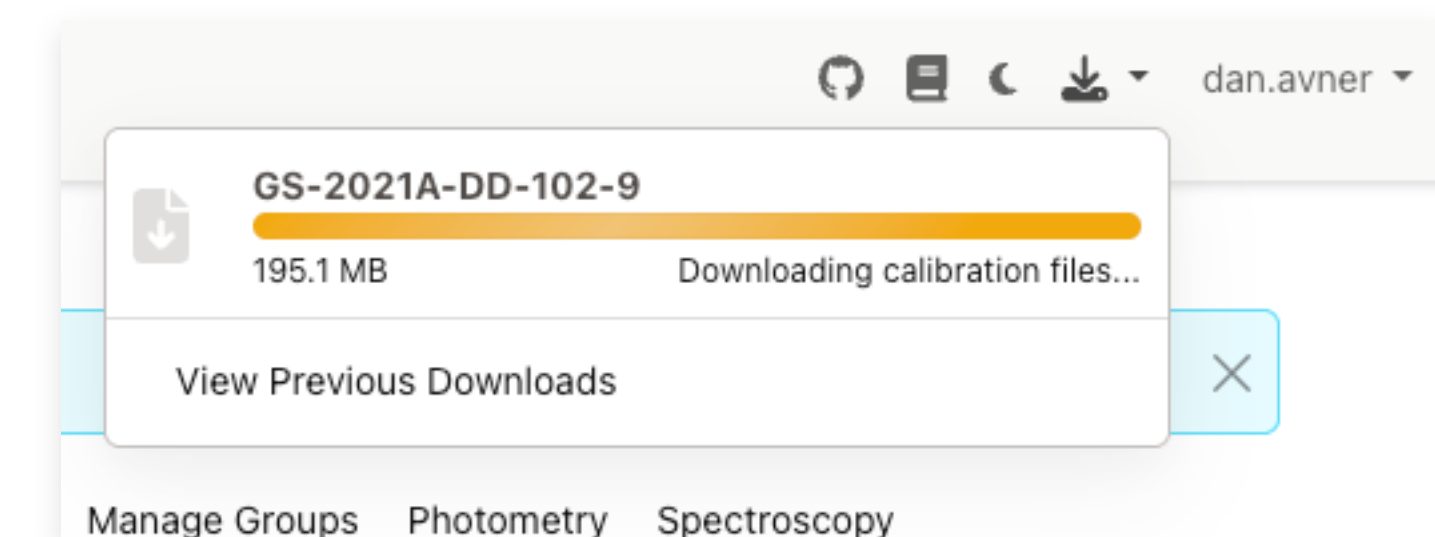
Raw/Reduced: Any (Select data by processing state)
QA State: Not Fail (Quality of results you wish to access)

Filename Prefix: Leave empty for any (Specify the first part of the filename to match by)

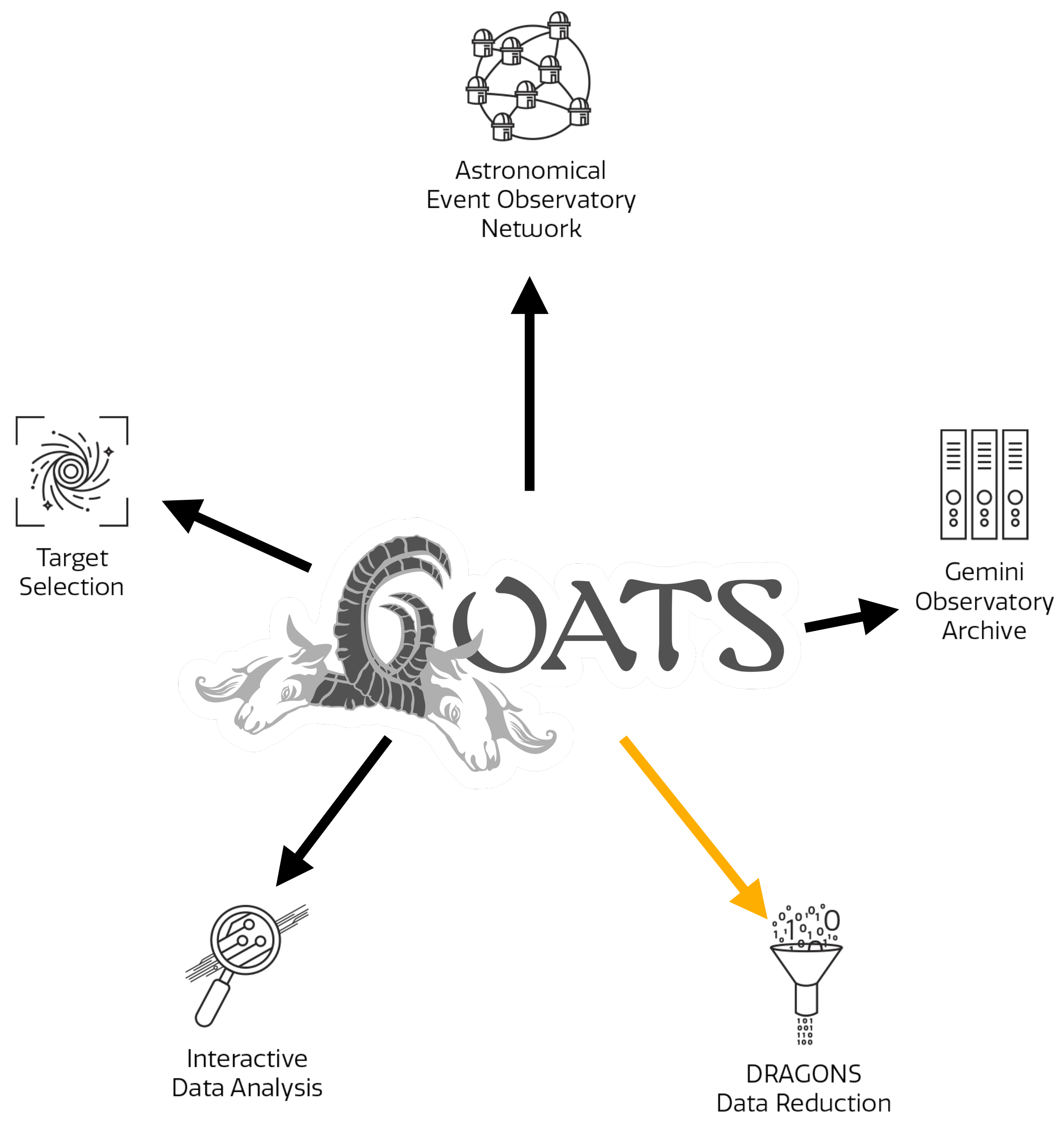
Observation ID: GS-2021A-DD-102-9

Download Associated Calibrations: Yes No Only Calibrations

[Submit](#)



► *Automatically decompresses and organizes data*



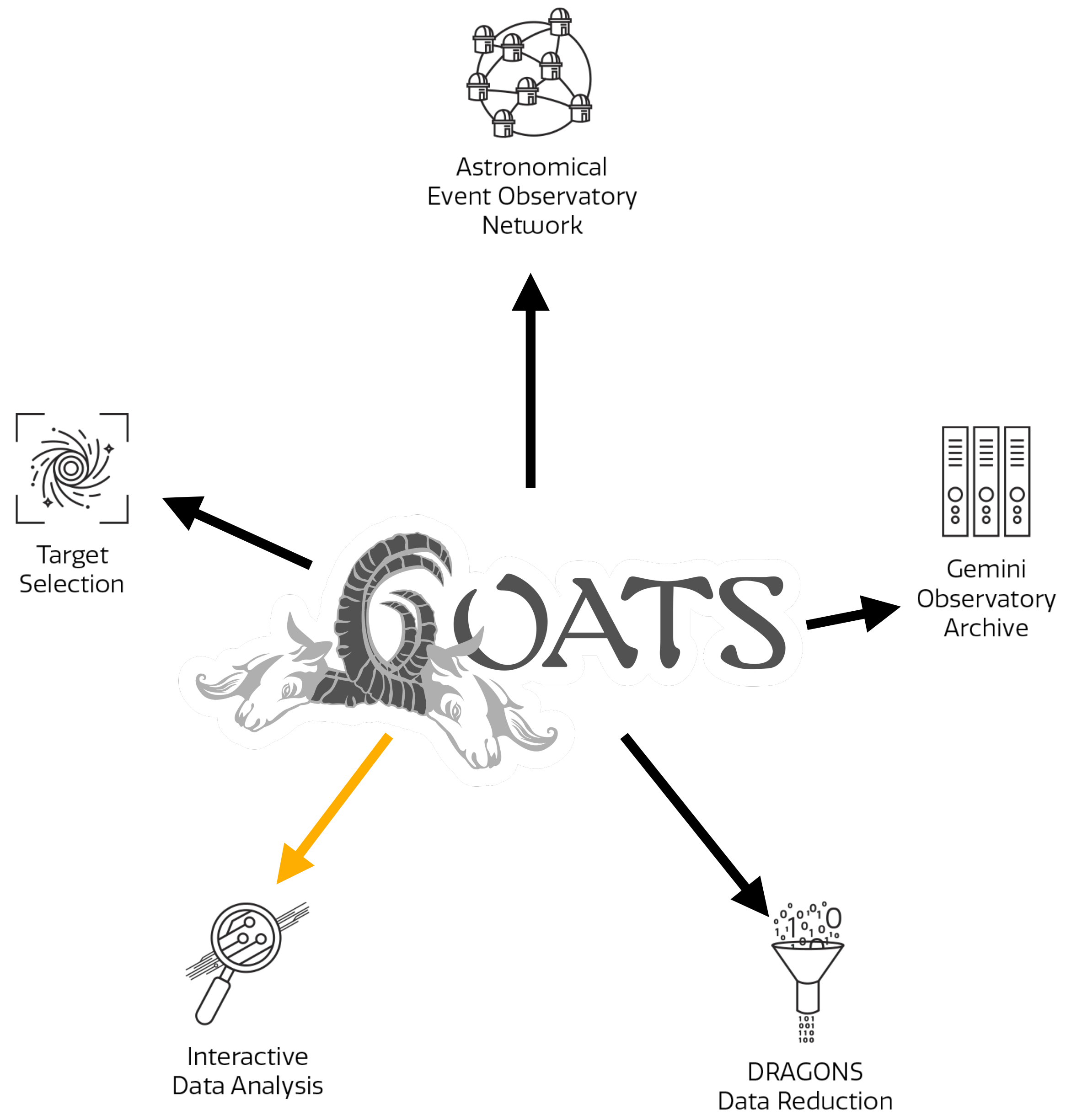
GOATS & DRAGONS

Interactive data reduction within GOATS

- DRAGONS integrated directly into GOATS
- DRAGONS available through a web-based interface
- No need to learn a CLI
- Customizable reduction recipes
- Reproducible reduction runs

The screenshot displays the GOATS web interface for the observation GS-2021A-DD-102-9. The interface is divided into several sections:

- Setup and Manage Reduction Runs:** Shows an existing run (run-20260320182734) with details such as Created (2026-03-20T18:27:34.543109Z), Version (4.1.0), and Directory. A 'Delete' button is present.
- Available Recipes and Files:** Lists available recipes (e.g., 'makeProcessedBias (default)') and files. It includes a 'Filter' section with a text input ('exposure_time > 10 and ut_date < "YYYY-MM-DD"') and checkboxes for 'Use strict filter expression matching' and 'Use all files for observation ID'. An 'Apply Groupings And Filter' button is at the bottom.
- Bias Reduction:** A 'Running' section with 'Start' and 'Stop' buttons. Below it, a 'Modify Recipe' section allows configuration of 'Additional Files', 'Reduction Mode' (set to 'sq (science-quality)'), 'DR Package' (set to 'geminidr'), 'Optional Parameters', 'Calibration Overrides', and 'Output Suffix'. A code editor shows the recipe definition for 'makeProcessedBias(p)'. The log at the bottom shows the execution of primitives like 'addVAR', 'overscanCorrect', and 'subtractOverscan'.
- Calibration Database:** A section for managing files.
- Bias Primitives Documentation:** A sidebar on the right provides documentation for various primitives. It lists categories like 'Adding Primitives', 'Modify Parameters', 'Interactive Mode', 'Reorder Primitives', and 'Remove Primitives'. A list of available primitives and parameters is also shown, including ADUToElectrons, addDQ, addIllumMaskToDQ, addMDF, addObjectMaskToDQ, addToList, addVAR, appendStream, applyDQPlane, associateSky, biasCorrect, checkWCS, clearAllStreams, clearStream, combineSlices, copyInputs, correctBackgroundToReference, darkCorrect, dilateObjectMask, display, fixPixels, flatCorrect, and flushPixels.



Analysis & Data Sharing

From reduced data to science-ready products

- Visualize data products (light curves, spectra)
- Transfer data to Astro Data Lab
- Continue analysis in external science platforms

	Filename	Type
Actions ▾	ANT2020affe44q/GEM/GS-2021A-DD-102-9/S20210220S0409.fits	FITS File
Actions ▾	ANT2020affe44q/GEM/GS-2021A-DD-102-9/S20210207S0130.fits	FITS File
View with JS9	ANT2020affe44q/GEM/GS-2021A-DD-102-9/S20210213S0060.fits	FITS File
Send to Astro Data Lab	ANT2020affe44q/GEM/GS-2021A-DD-102-9/S20210207S0084.fits	FITS File
Actions ▾	ANT2020affe44q/GEM/GS-2021A-DD-102-9/S20210207S0084.fits	FITS File

Observe Observations Manage Data Manage Groups Photometry Spectroscopy

Select and Plot Data Products

Use the table below to select data products. Note that only CSV files are supported. See [Manage Data](#) for an example of the format. Click the **Plot** button to add the selected data product to the plot.

Click to enter Plot title
Click to enter Plot subtitle

Apparent Magnitude

Time (MJD)

Click on labels to edit text, and click directly on the end values of the x and y axes to adjust their ranges.

Type to filter file names...

File Name	Actions
ANT2020affe44q/GEM/GS-2021A-DD-102-9/S20210220S0409.fits	Plot
ANT2020affe44q/GEM/GS-2021A-DD-102-9/S20210207S0130.fits	Plot
ANT2020affe44q/GEM/GS-2021A-DD-102-9/S20210213S0060.fits	Plot
ANT2020affe44q/GEM/GS-2021A-DD-102-9/S20210207S0084.fits	Plot
ANT2020affe44q/GEM/GS-2021A-DD-102-9/S20210205S0079.fits	Plot

**We exercised this entire workflow
end-to-end using Rubin alerts**

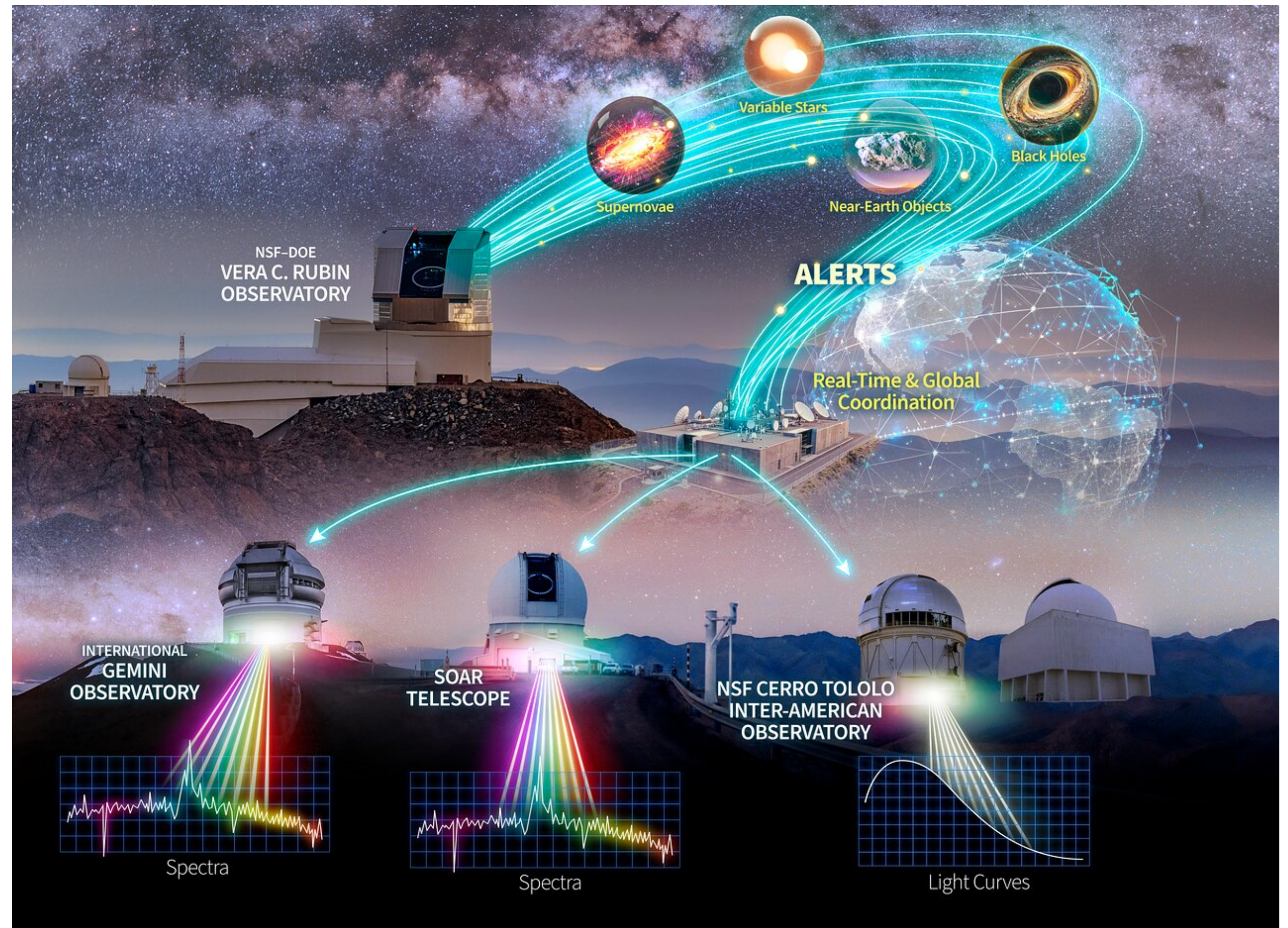
We exercised this entire workflow end-to-end using Rubin alerts



End-to-end Rubin Demonstration

Successful stress test of the GOATS workflow

- Processed alerts from ANTARES
- Triggered follow-up observations (Gemini, SOAR, Blanco)
- Retrieved, reduced, and visualized data
- Demonstrated full end-to-end workflow



► From alert to science products in a single system

Lessons learned

- End-to-end workflows require coordination across teams and systems
- CI is essential for multi-service pipelines
- Dedicated development and sandbox environments are critical
- Full workflow testing is as important as component testing



GOATS / TOMToolkit workshop (2026)

► *Integration is the primary challenge, not individual components*

Discussion

How do we move from fragmented workflows to responsive systems?

- ▶ *How do we reduce the operation friction between alert ingestion and follow-up?*
- ▶ *What are the remaining gaps between systems?*



GOATS documentation