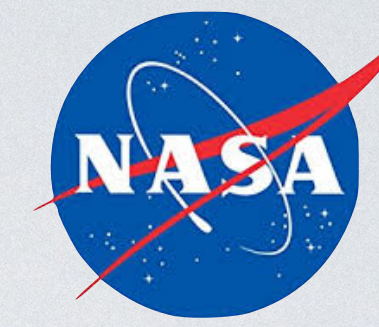




PennState
Eberly College of Science



SWIFT, ACROSS AND VO

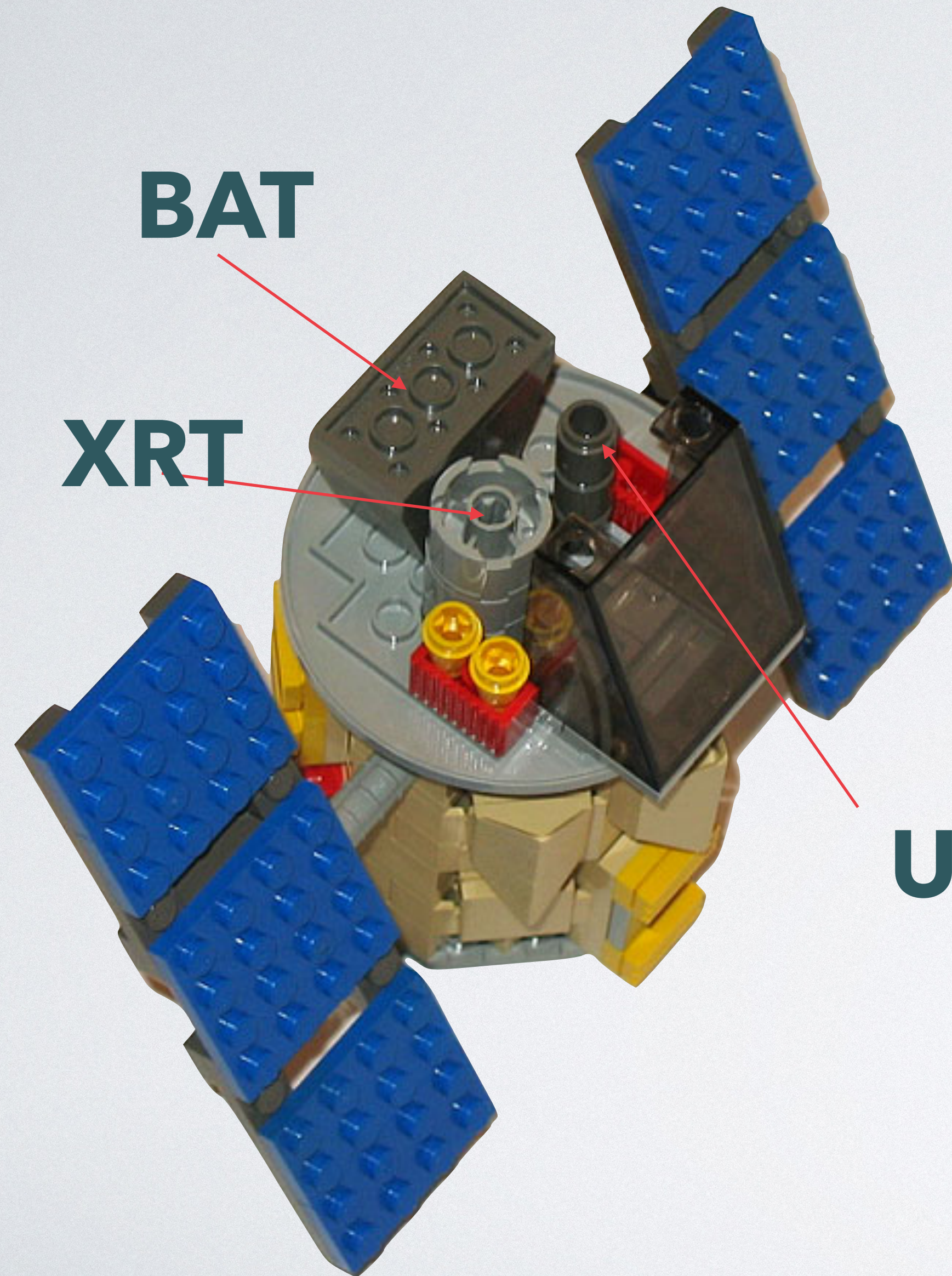
JAMIE A. KENNEA (Penn State)

Science Operations and X-ray Telescope Teams Lead

NASA Neil Gehrels Swift Observatory

Lead Scientist, NASA ACROSS Initiative

NEIL GEHRELS SWIFT OBSERVATORY



BAT

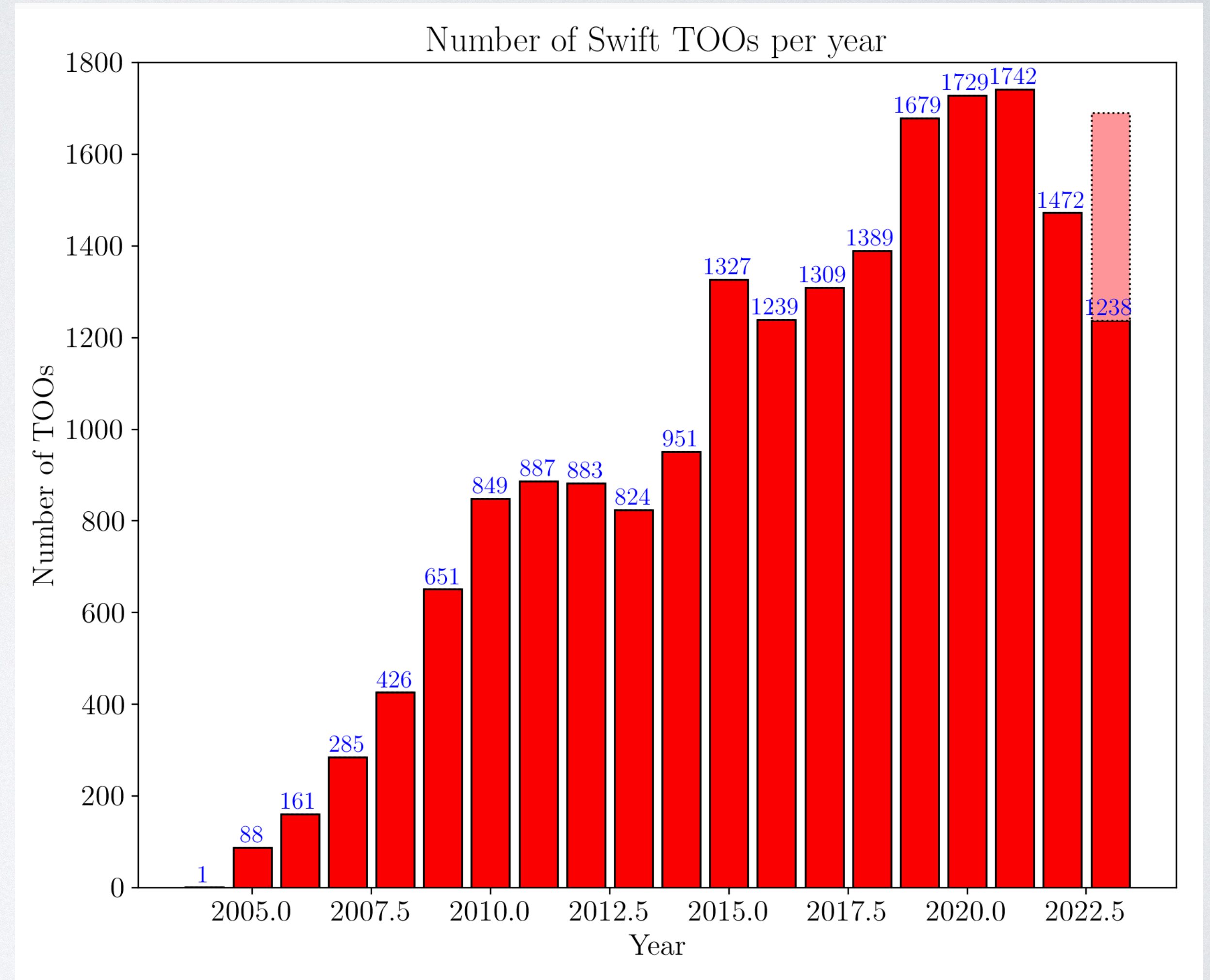
XRT

UVOT

- **Multi-wavelength observations** - Space unique Hard X-ray/ X-ray / UV all in one package, simultaneously.
- **Transient discovery** - BAT is a hard X-ray transient all-sky (in a day) monitor.
- **Broad Science Portfolio** - Swift observes >100 targets per day, with a broad range of science topics.
- **Rapid slewing** - gets you to a GRB fast. Also allows for **very high efficiency** of operations (~72%, despite being in LEO and spending time in SAA). Also allows **time domain astrophysics** due to ability to perform high cadence high sensitivity monitoring.
- **Constantly evolving ground and onboard software** - we don't stand still evolving the operations concept.
- **Open Target of Opportunity (TOO) program with open data program** - Our TOO program is extremely open, with low rejection rates, and our data is made public ASAP.

SWIFT RECEIVES A LOT OF TOOS

- 2021 - received 1742 TOOs in 1 year.
2023: ~1700 predicted.
- TOOs performed on "best effort" basis.
- The advent of large surveys such as ZTF and the upcoming Vera Rubin lead to development of an API to submit TOOs for Swift.
- Also we wanted to make Swift responsive to triggers from other GRB missions and also GW, so programatic submission of TOOs was a necessary step for this.



HOW TO SUBMIT A TOO

Using the TOO Web Page

The screenshot shows the PennState Mission Operations Center for Swift website. The page title is "Swift Target-of-Opportunity Request". The navigation menu includes Home, Target of Opportunity, Observations, BAT, XRT, and UVOT. The main content area contains instructions for filling out the form and a "TOO Form" section. The form is divided into two parts: "Object Information" and "Type or Classification".

Object Information

Enter object name, RA and Dec in decimal degrees or sexagesimal formats. Note that sexagesimal must be in hours for RA and will be converted to decimal degrees when you validate. To check target visibility, please use the Long Term or Short Term (high resolution) target visibility calculators.

Object Name:

Resolve coordinates

Right Ascension (J2000):

Declination (J2000):

Position Error (90% confidence - arcminutes):

Type or Classification

Using the Swift TOO API

The screenshot shows a Jupyter Notebook titled "Swift TOO submission Example". The notebook contains a Python script that demonstrates how to use the Swift TOO API to submit a request. The script includes comments explaining the process and the output of the code.

```
swift_too module

Swift_TOO example - Swift Target of Opportunity Submission Example

API version = 1.2, swifttools version = 2.3

Author: Jamie A. Kennea (Penn State)

The Swift_TOO class is used to submit Target of Opportunity Requests for the Swift mission. Before this required manual filling out of a web based form, but given the requirements to trigger Swift using algorithmically determined criteria, this can now be automated through the Swift TOO API, and this class. Here we give a simple example of how to submit a TOO request with this class. Note that we will enable debug mode here, so that the submission will actually complete, but importantly it will not submit an actual Swift TOO!

In [1]: from swift_too import TOO, Resolve

We start by initializing the class and giving our username and shared secret. These can be set up on the Swift TOO website. After you log in, you will find your shared secret, and can modify it as necessary under the Update Account Info link. Note that unlike other TOO API classes, you cannot use anonymous login to submit a TOO request.

In [2]: username = "myuser"
        shared_secret = "mysecret"

OK let's set up the Swift_TOO request (in this case we'll use the swifttools 2.3 shorthand, TOO). Also we're setting debug mode here. Note that I'm not passing my shared_secret here, because my computer supports keyring. This records your shared_secret the first time that you use it, so it's not necessary to include it in later requests.

In [3]: too = TOO()
        too.username = username
        # too.shared_secret = shared_secret
        too.debug = True

OK, so what are we going to observe, let's start with a name, oh say, SMC X-3, that's a cool object. However, darn, I can't remember the coordinates, but thankfully we have a class called Swift_Resolve (we'll use Resolve for short) for looking these up.

In [4]: too.source_name = "SMC X-3"
        res = Resolve(name=too.source_name)
        print(f"RA/Dec (J2000) = {res.ra:.4f}, {res.dec:.4f}")
        print(f"SkyCoord = {res.skycoord}")

RA/Dec (J2000) = 13.0234, -72.4345
SkyCoord = <SkyCoord (FK5: equinox=J2000.000): (ra, dec) in deg
(13.02343917, -72.43450833)>

Swift_Resolve reports back ra and dec, but it also reports back as a SkyCoord using the skycoord property if you have astropy installed. Swift_TOO can take a SkyCoord directly.

In [5]: too.skycoord = res.skycoord

If you use a SkyCoord, it means you can also use more other coordinate systems or formats, rather than just J2000 decimal degrees. Internally all Swift TOOs are stored as decimal RA/Dec in J2000, as this is what Swift uses. Let's check what the values are.

In [6]: print(f"RA/Dec (J2000) = {too.ra:.4f}, {too.dec:.4f}")
```

<https://www.swift.psu.edu/toop>

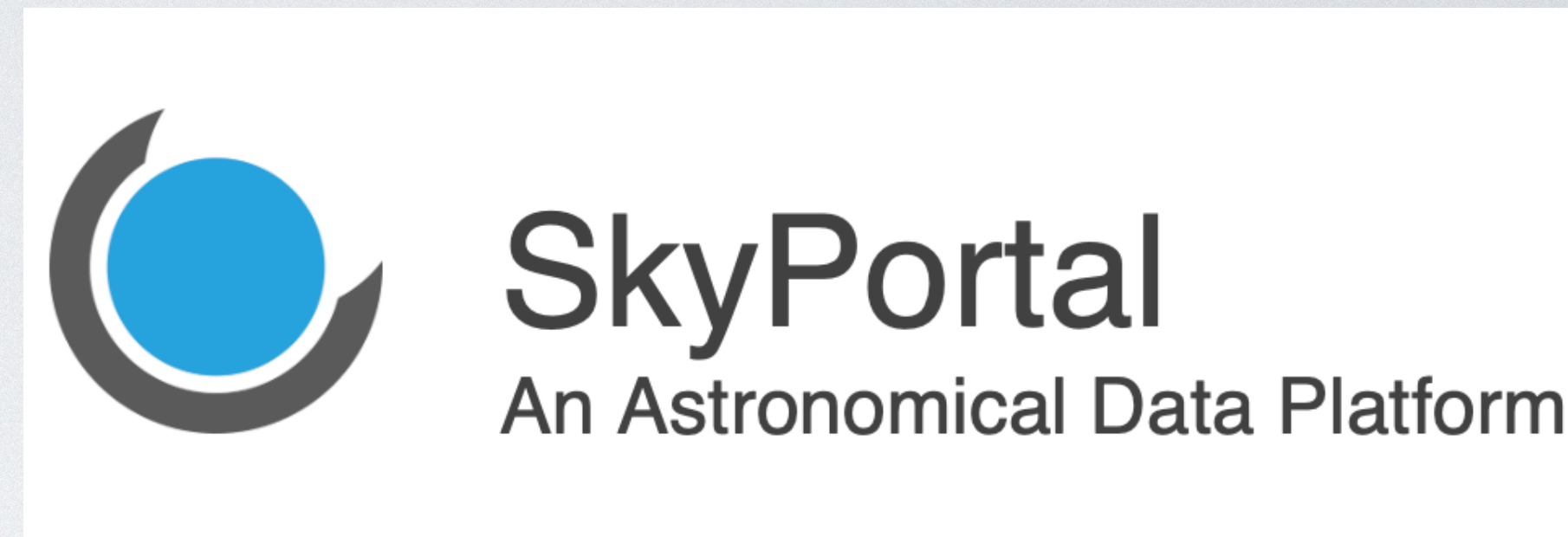
(This is just an API client)

```
# pip install swifttools
```

https://www.swift.psu.edu/too_api

SWIFT TOO API

- Python / JSON based API with two main servers:
 - University of Leicester - X-ray Telescope Analysis API
 - Penn State Swift Science Operations Team API
- TOO API started off as just an API to submit TOOs but soon broadened to:
 - Fetching Observation Plan and Performed Observations
 - Instrument configuration APIs (e.g. convert UVOT hex modes into
 - Object Visibility Calculator
 - API for downloading data from Swift data from the three data centers in the USA (HEASARC + Swift quicklook), UK and Italy.
- As of last night, Penn State side of API has processed more 15 million API requests since launching in August 2020.
- Swift API has been **wildly successful**.



LCO's TOM Toolkit and SkyPortal have Swift TOO capabilities built in now thanks to the Swift TOO API integration.



SWIFT AND VO

- I am a co-author of the ObsLocTAP and ObsVisSAP (AKA ObjObsSAP) protocols.
- Swift implemented ObsVisSAP protocol, simply as a WSGI application that calls the swifttools `VisQuery` python class and converts JSON output to VO format.
 - This implementation can be seen in action in the TOBY application: <http://integral.esa.int/toby>
- Swift does not currently implement ObsLocTap at the Science Operations Center.

VISIBILITY AND SCHEDULE

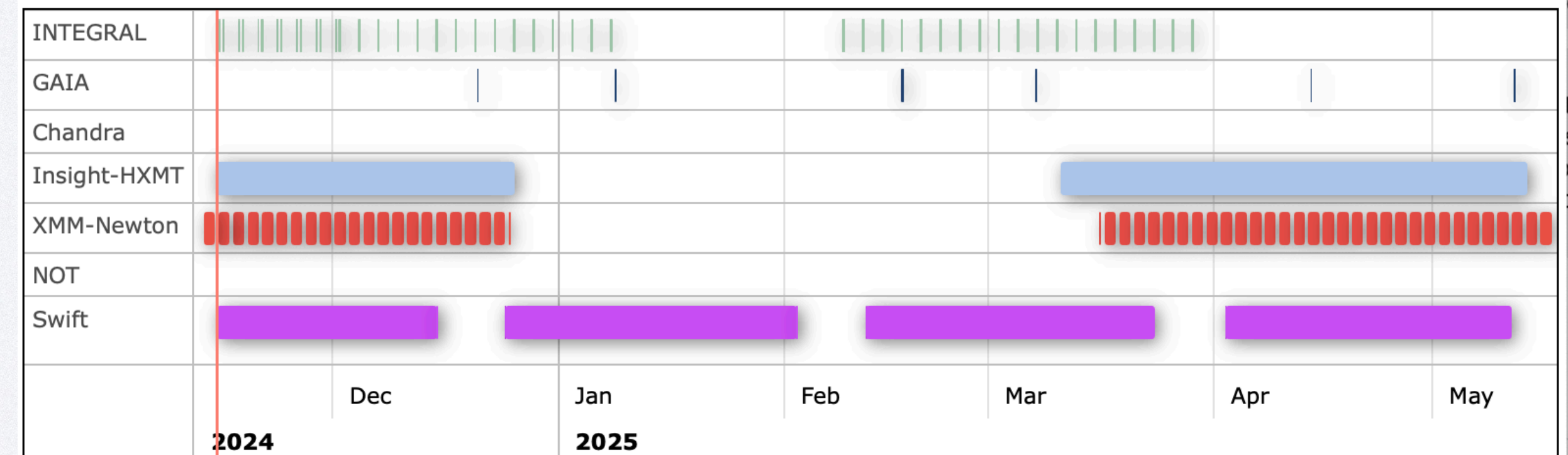
Source name

Coordinates (RA, DEC) in degrees *

Start and End (UTC) *

Start and End (MJD)

Visibility



WHY NO SWIFT OBSLOCTAP?

- Swift API ObsQuery and PlanQuery provides similar functionality, with Swift specific search parameters.
 - Should be possible to build support for a simple cone search version of ObsLocTAP on top of this like done for ObsVisSAP.
 - We haven't seen any demand for this feature from the community, so the resources haven't be put into it. Swift is highly cost constrained and low staffed.
- HEASARC should be able to provide this feature for data that has been sent to the archive. Caveat here is that Swift API data is updated in near realtime, HEASARC is longer latency.
- ADQL queries are complex to set up and APIs that process user submitted SQL queries caused Penn State IT Security to red flag it and clearly state "you cannot do this".
- Generally there's an issue where small missions like Swift do an outsized amount of stuff compared to their funding level.
 - Implementing things like the Swift API are out of working hours tasks for us.
 - Implementing VO standards will only be done if it can compellingly be shown to be wanted and **needed**. No funding available to do this.



ACROSS

ENABLING TIME DOMAIN AND MULTI-MESSENGER ASTROPHYSICS



Jamie Kennea (Penn State), ACROSS Lead Scientist

Brian Humensky, Physics of the Cosmos Chief Scientist
Chris Roberts, TDAMM Study/ACROSS Manager

Core Team:

- Dan Kocevski, Michelle Hui (Marshall)
- Tom Barclay, Christina Hedges, Tyler Pritchard, Kirill Vorobyev, Samuel Wyatt (Goddard)



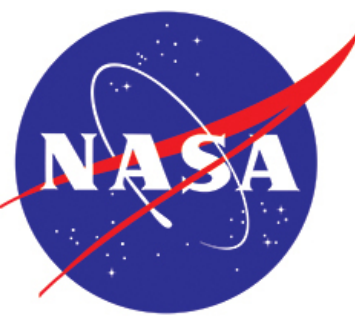
Background

- **The PhysCOS Time-Domain and Multi-Messenger (TDAMM) Initiative responds to a top priority of the Astro2020 decadal report recommendation and has been tasked with:**
 1. Organizing or supporting **TDAMM workshops**,
 2. Conducting a three-year **TDAMM Study** investigating policy, processes and technical coordination mechanisms to enable TDAMM science, and
 3. Recommending one or more potential implementations for enabling TDAMM science support.

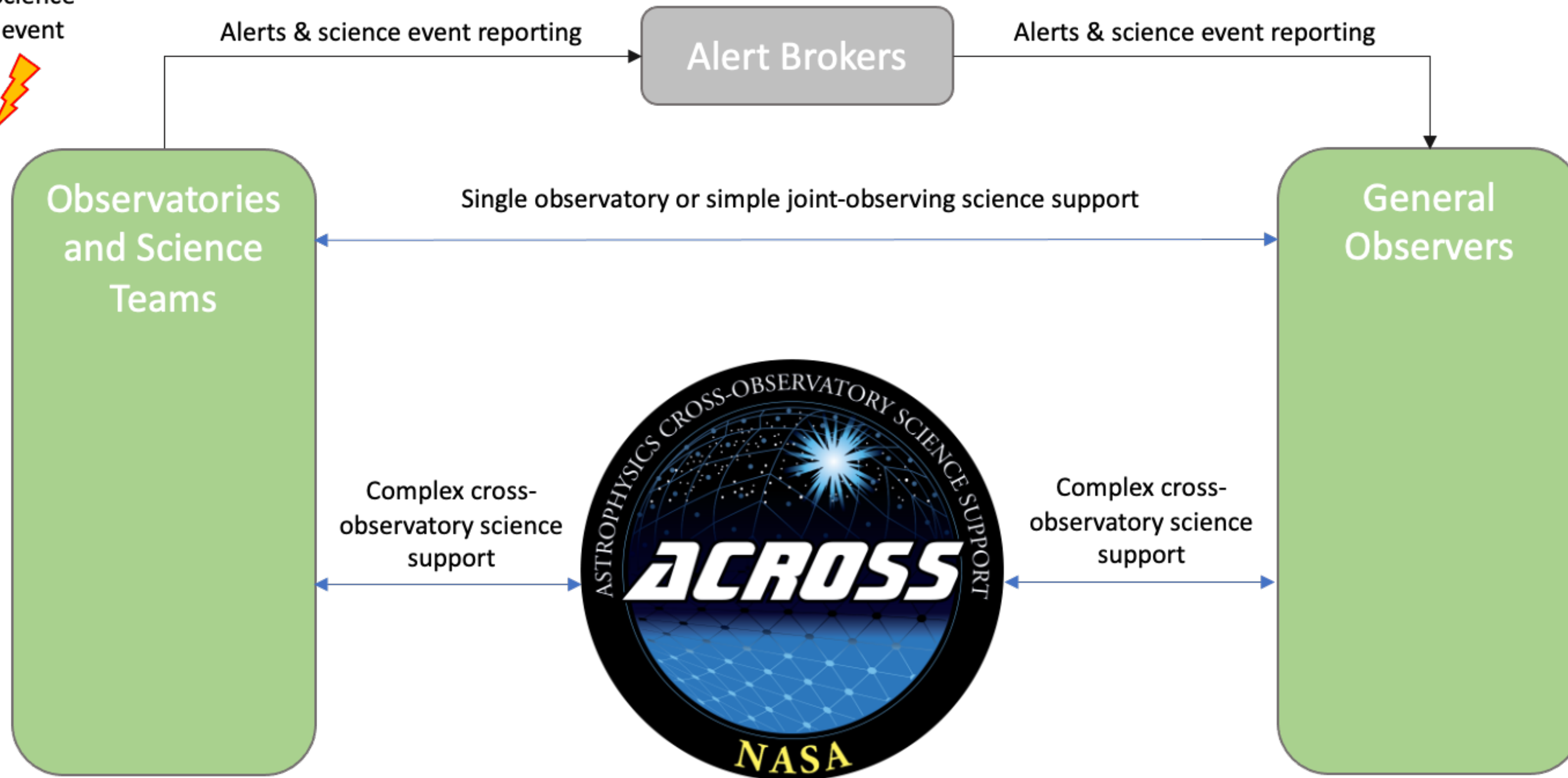
- **The Astrophysics Cross-Observatory Science Support (ACROSS) pilot project is an outcome of the first year of the TDAMM study, which identified needs for:**
 1. Software & data systems to facilitate TDAMM science workflows,
 2. TDAMM help desk to provide expertise & facilitate coordination, and
 3. TDAMM community grant program to incentivize scientific innovation.



High-Level Architecture: Future-State Context Diagram



Science event



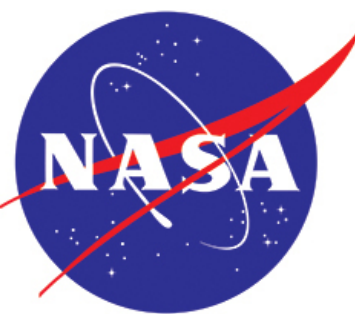
Key

Deciders

Science Support Infrastructure



High-Level Architecture: Future-State Context Diagram



Observatory Workflow Status Dashboard and API

- Live dashboards of NASA observatory status parameters relevant for TDAMM science planning & execution (e.g. observing plans).
- API access to easily incorporate into Observer and Operations Team planning & execution tools.

Science event



Observatory and Science Team

Complex cross-observatory science support



Complex cross-observatory science support

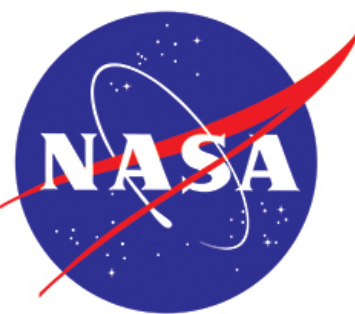
Key

Deciders

Science Support Infrastructure



High-Level Architecture: Future-State Context Diagram



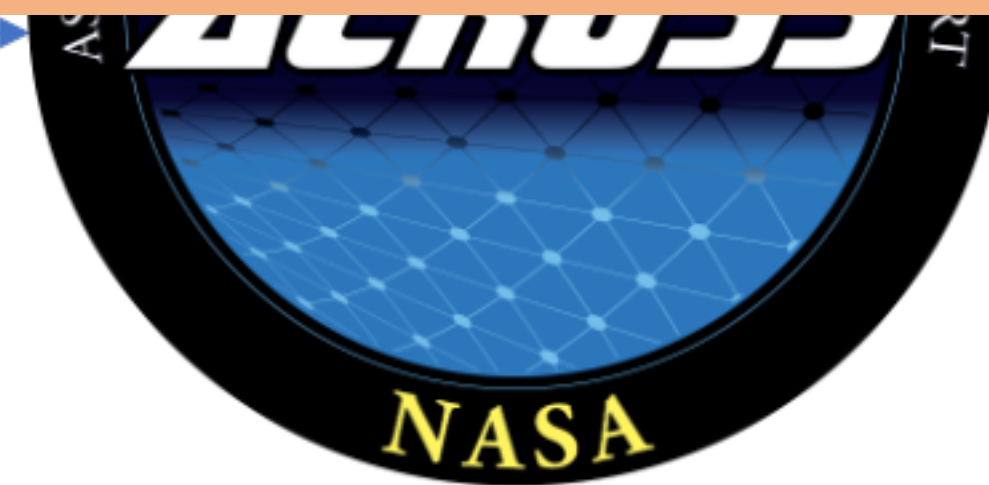
Science event



Observa
and Sci
Team

TDAMM Toolkit

- A collection of open-source software products that can be instantiated to streamline, standardize, and automate TDAMM planning workflows.
- Deployed locally by Observers and Operations Teams, with or without support from ACROSS Subject Matter Experts.
- Future TDAMM grant program will incentivize community contributions to the toolkit.



Key

Deciders

Science Support Infrastructure



Science event



Observa
and Sci
Team

TDAMM Web Services

- Accessed through our portal
- Organizes and displays status data feeds
- Services are cloud-hosted, with human and machine interfaces, and provide:
 1. Science Situational Awareness Multi-observatory follow-up planning & feasibility analysis tools (REST API + **VO Interfaces**).
 2. Follow-up hub for, e.g., ToO requests
 3. Follow-up decision support & recommendations

Key

Deciders

Science Support Infrastructure

NASA



ACROSS and VO

- ACROSS aims to make cross coordination of observations easier for NASA missions and beyond, to enable TDAMM science.
- Baked into ACROSS is support for VO standards, therefore implementations of ObsLocTAP and ObjObsSAP will be provided by ACROSS's for all supported NASA astrophysics missions.
- ACROSS also leverages existing VO infrastructure. Therefore if your mission already provides ObsLocTAP and ObjObsSAP VO, the ACROSS will use those in favor of local implementations.
- As we intend to support every NASA mission, and also ground based telescopes and missions from other. Therefore ACROSS should broaden the availability of certain VO protocols for missions.
- ACROSS tools will be open-source and provided to all future missions, so even small low cost missions (e.g. cubesats) can implement these interfaces without overly burdening the teams.
- ACROSS should lead to more missions having available implementations of VO ObjObsSAP and ObsLocTAP.