OPERATION SOFTWARE AND PHILOSOPHY

LAURA ASQUINI

ON BEHALF OF THE SOXS CONSORTIUM

LAURA.ASQUINI@INAF.IT

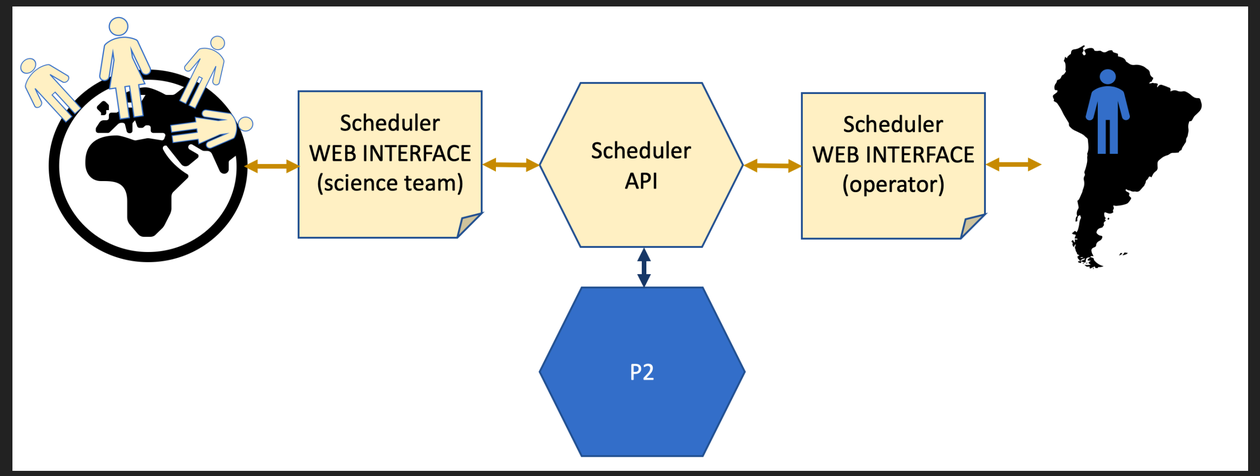
NEAPLES, 25/06/24

OUTLINE

- The envisioned observing night
- How to get your follow up:
    - ★ What targets / Pre-approval
    - ★ Marshall
    - ★ Urgent OBs
- The scheduler:
    - ★ Science Team interface
    - ★ Scheduling algorithm
    - ★ Automated night management

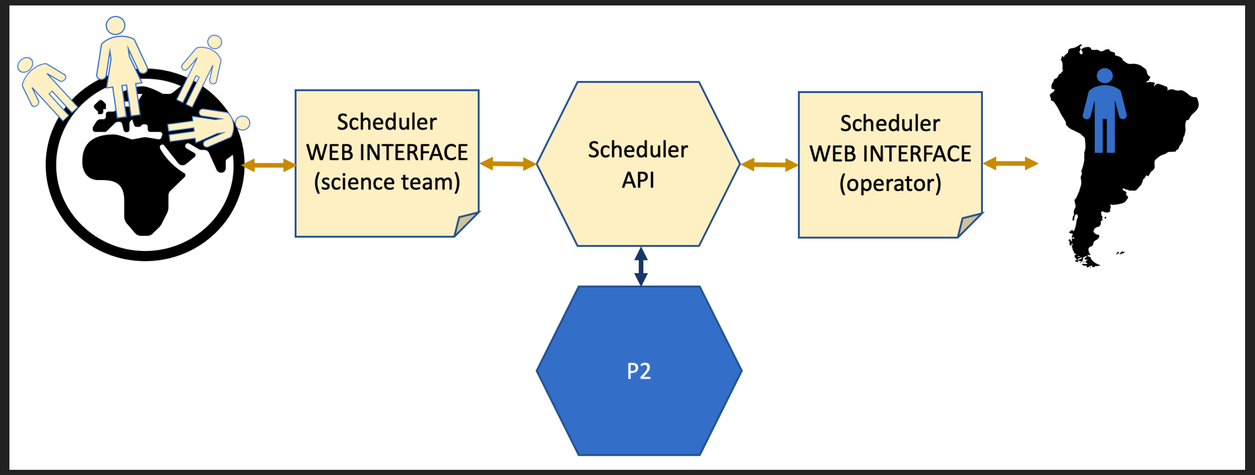
OPERATION REQUIREMENTS – CHILEAN AFTERNOON

- We work on ToOs, schedule decided daily
- No astronomer on the mountain, the scheduler will do all the work
- The Science Team (PIs, aka you) with gather online to approve the proposed schedule



OPERATION REQUIREMENTS – CHILEAN NIGHT

- Somebody will be on call for trouble shooting
- Operations start
- Tio intervenes only if something goes wrong (VES is self updating, BOB autofetch enabled)



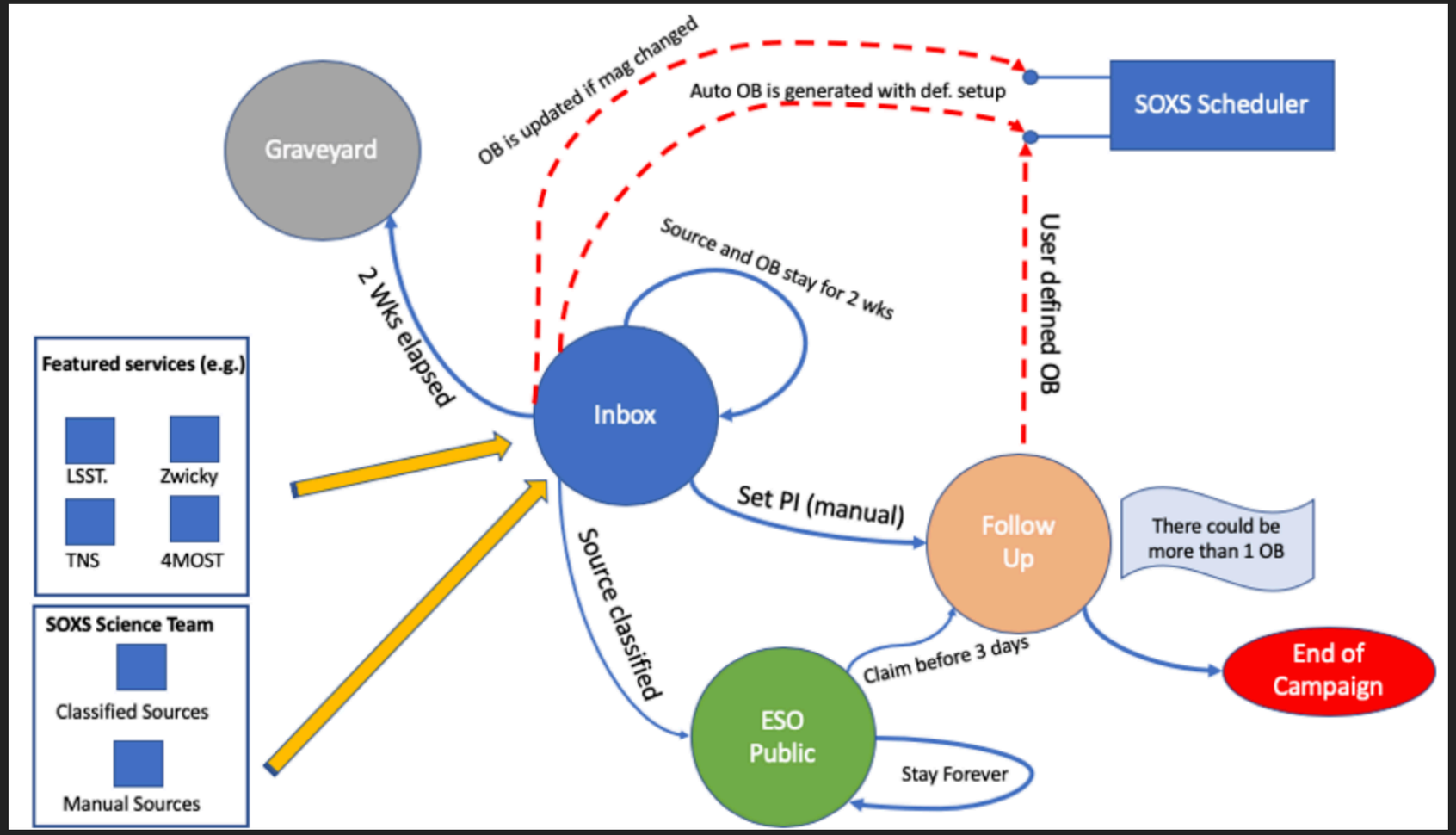
WHAT TARGETS

- I will refer to three categories of targets: Classification, follow-up, urgent
- All of these will fall within our (well defined) science cases

PRE-APPROVAL

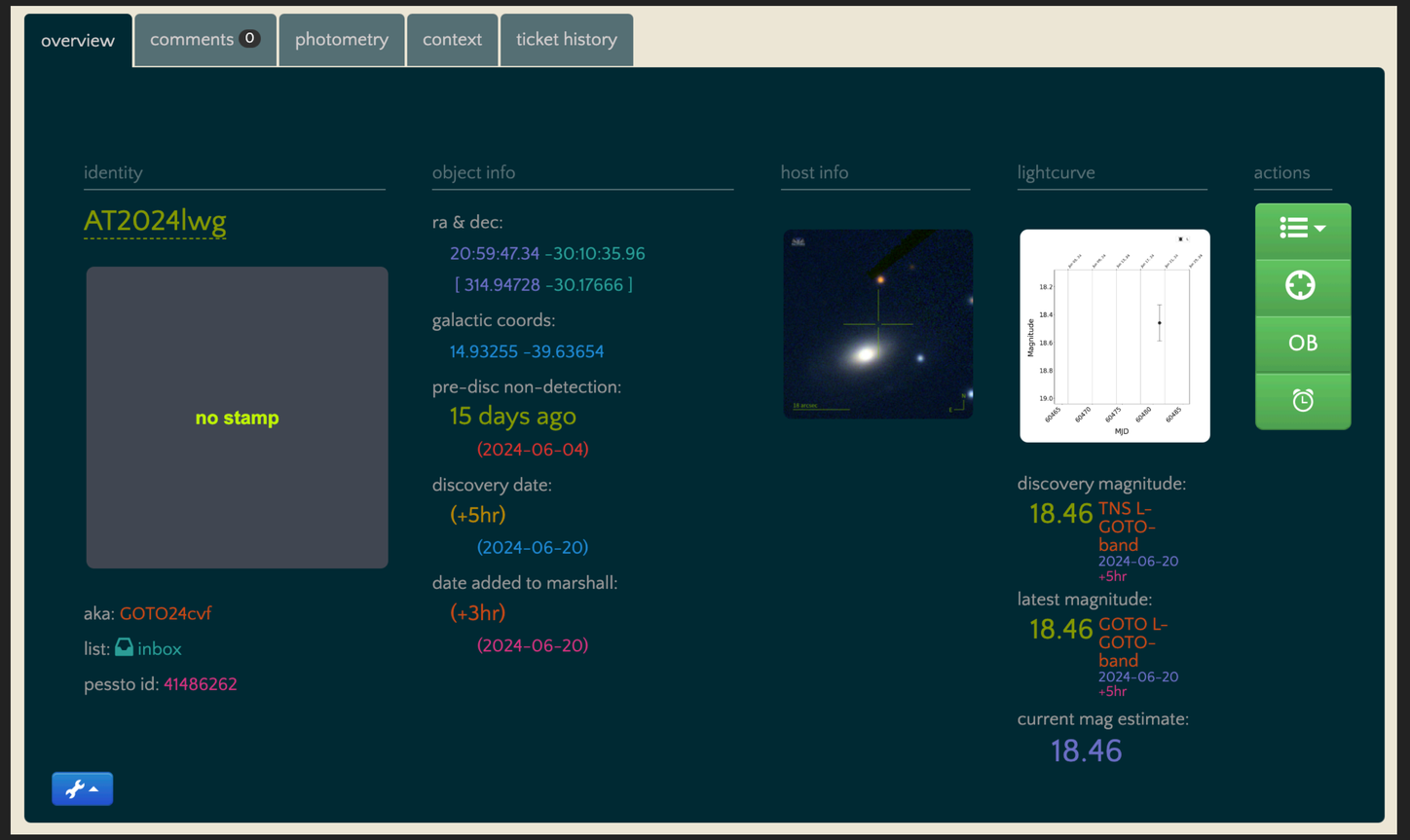
- To request a follow-up observation, PIs will need to submit a mini-proposal to our Scientific Committee
- Swift procedure, following our guidelines (see Sergio's talk)

MARSHALL – SOURCE FLOW



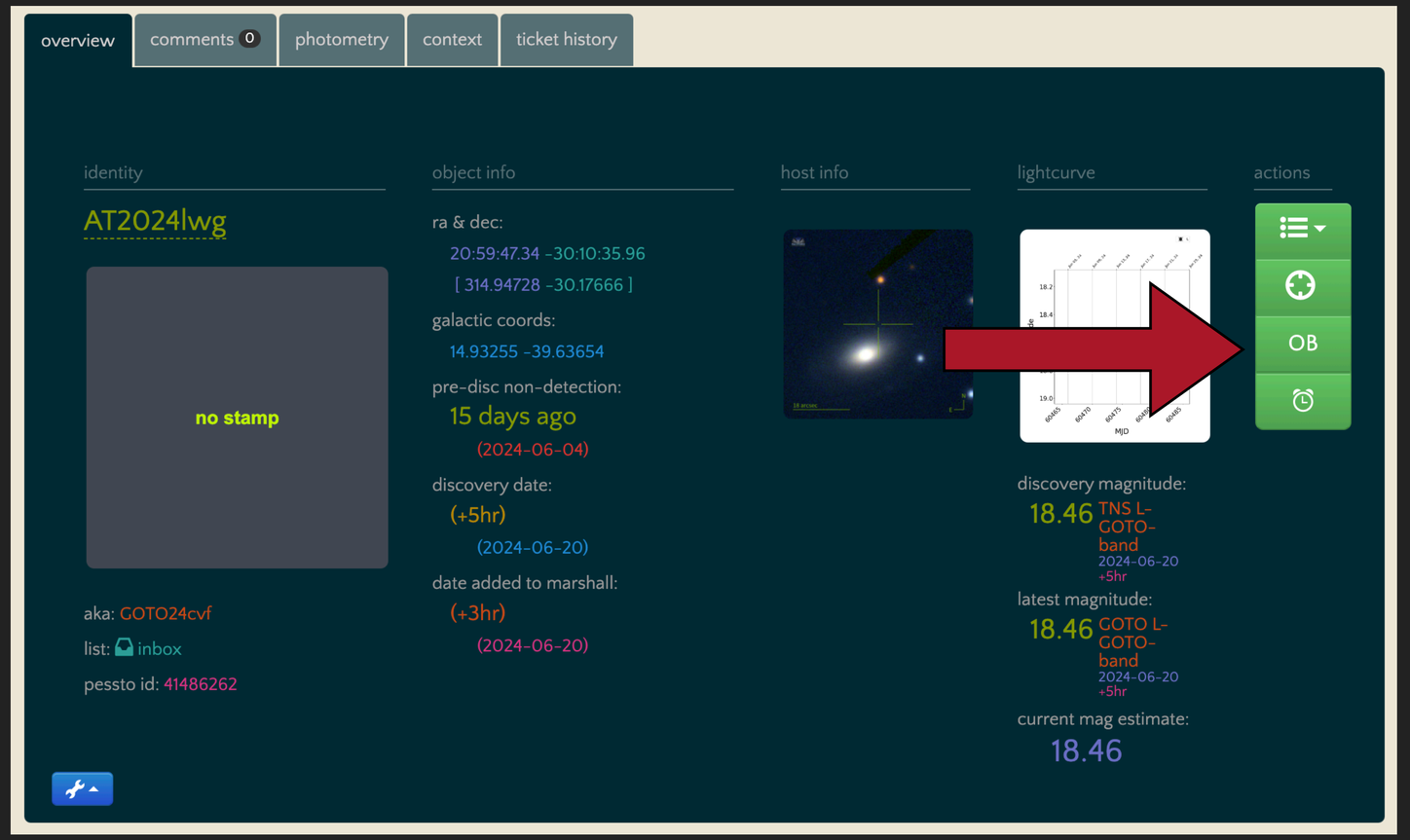
MARSHALL

- The Marshall aggregates all of the info from new/old transients
- Sends Classification OBs/ Updates automatically to the scheduler



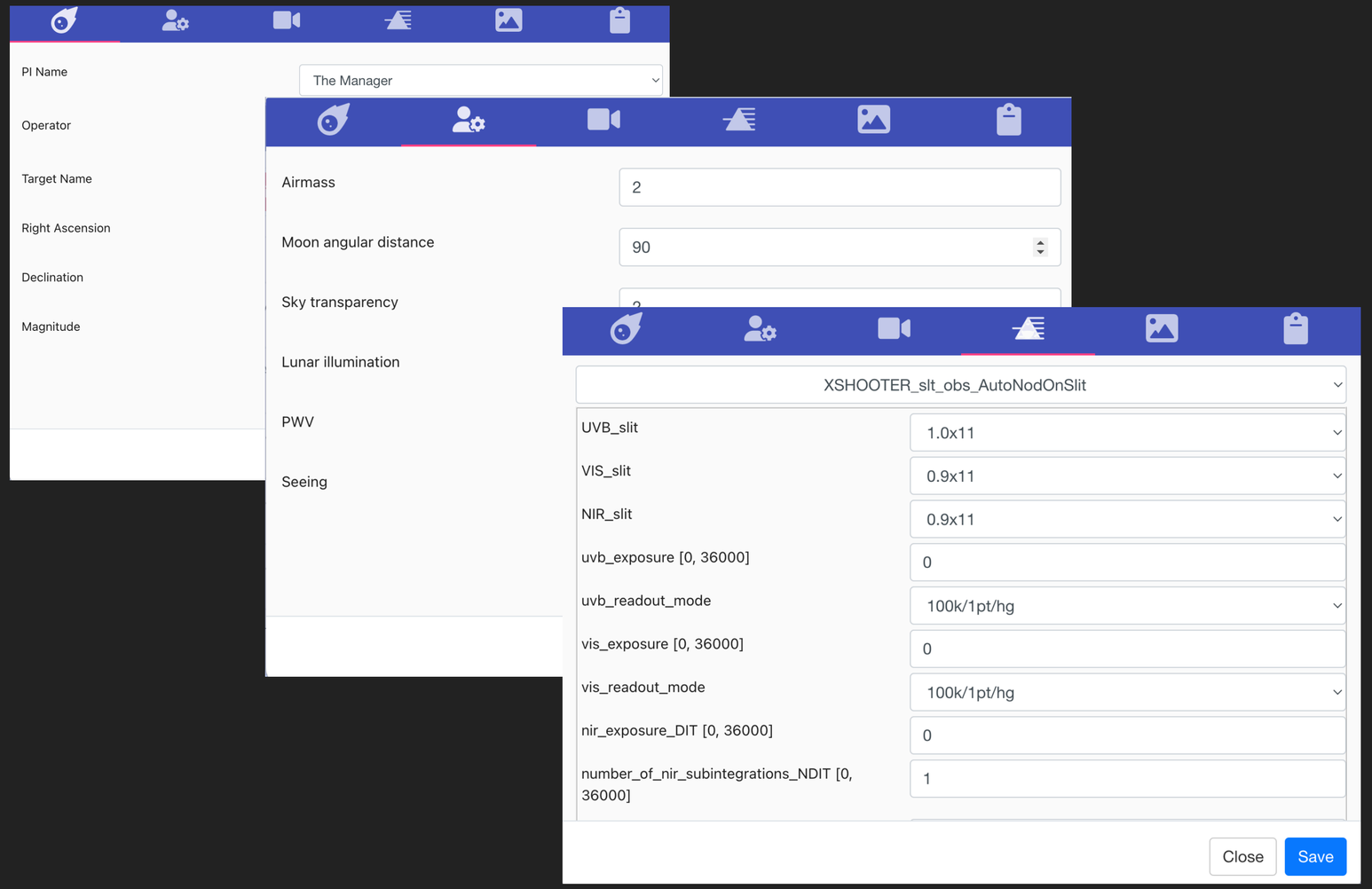
MARSHALL

- The follow up requests will be created from the SOXS Marshall
- Will either create them ex-novo, or claiming a classification target/a known source
- These will be sent to the Scheduler DB (our own)



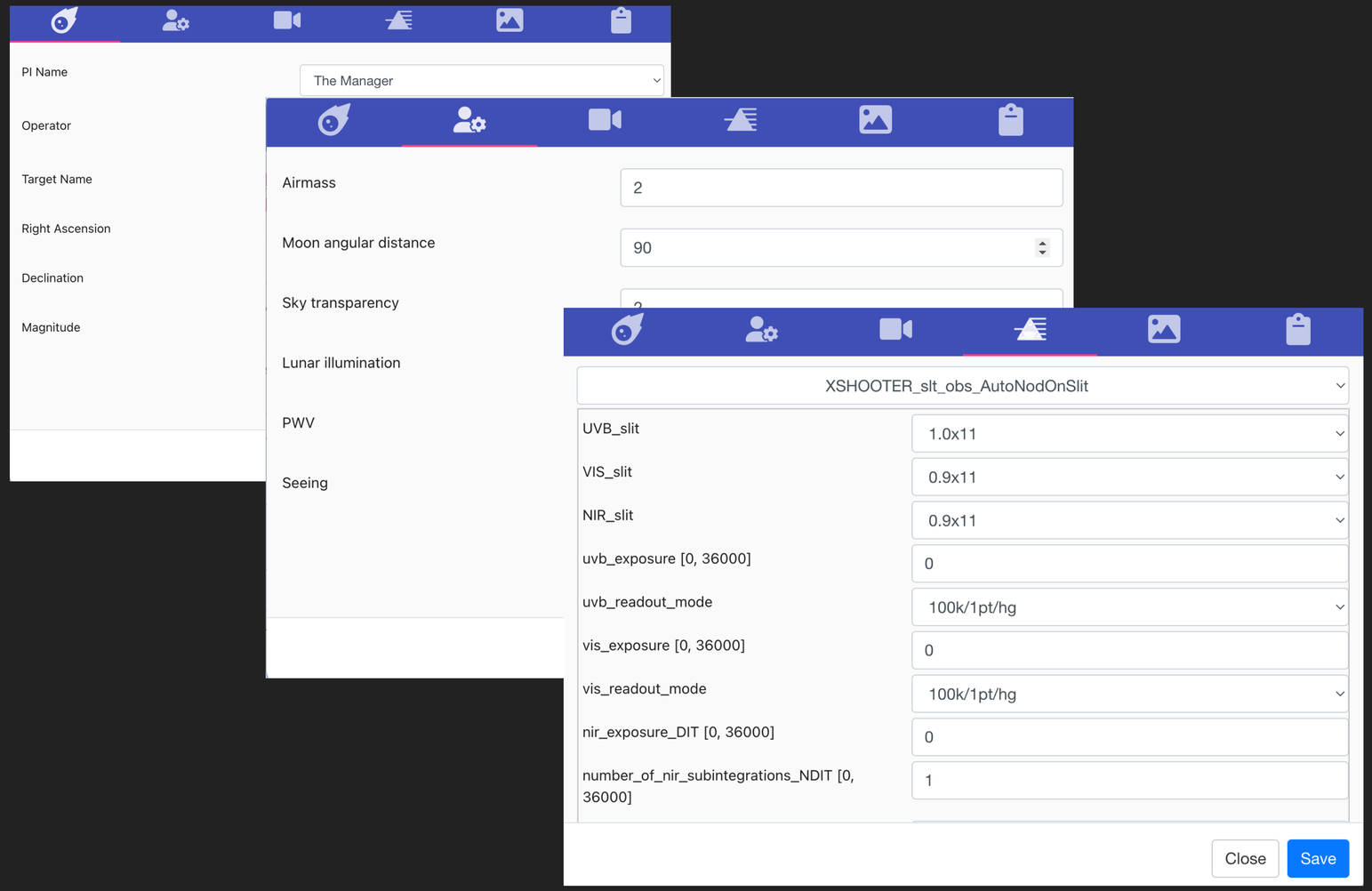
MARSHALL

- The follow up OB is fully customizable (templates, fixed time of observation, to do today)
- Could look something like this

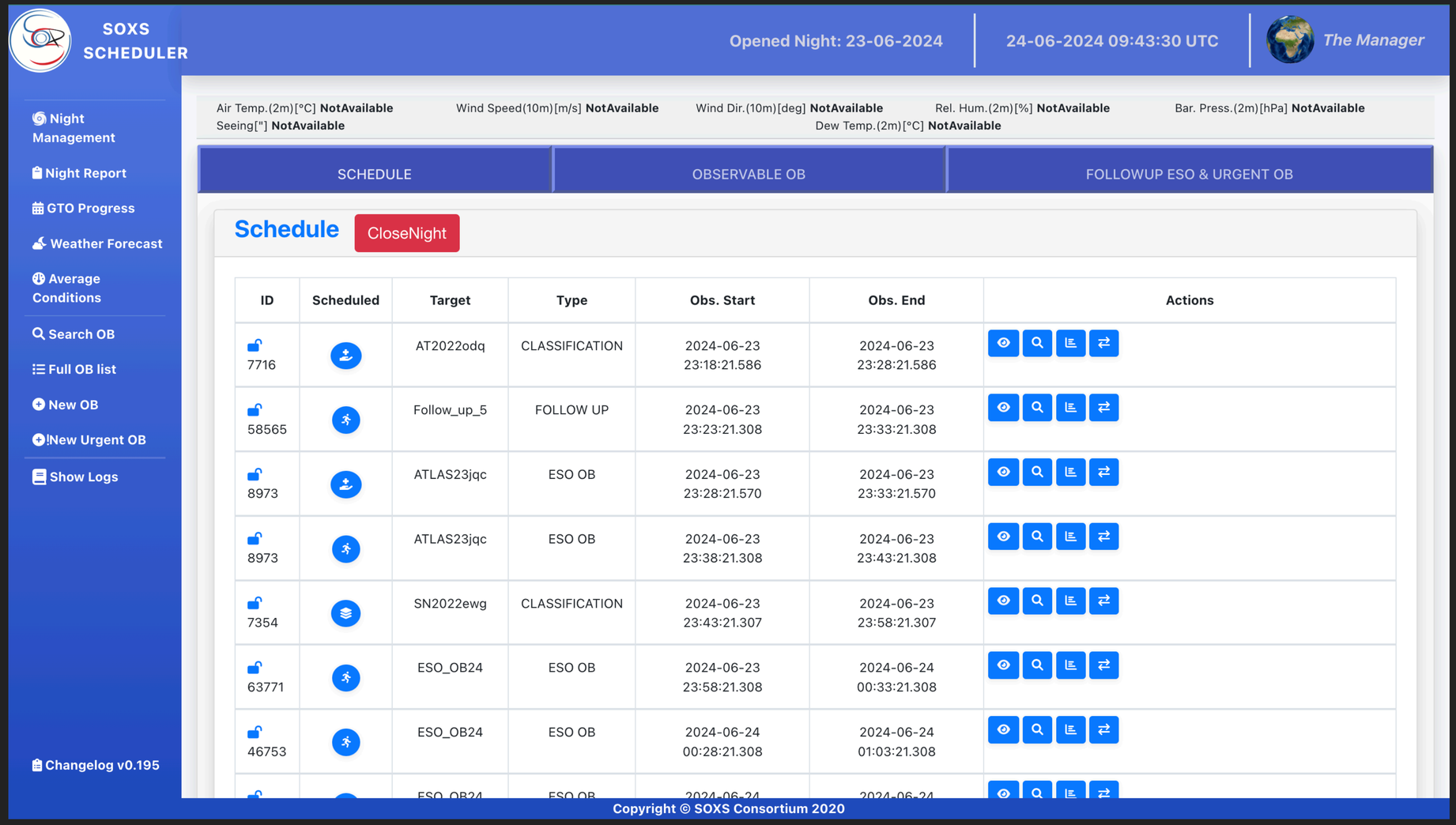


URGENT OBS

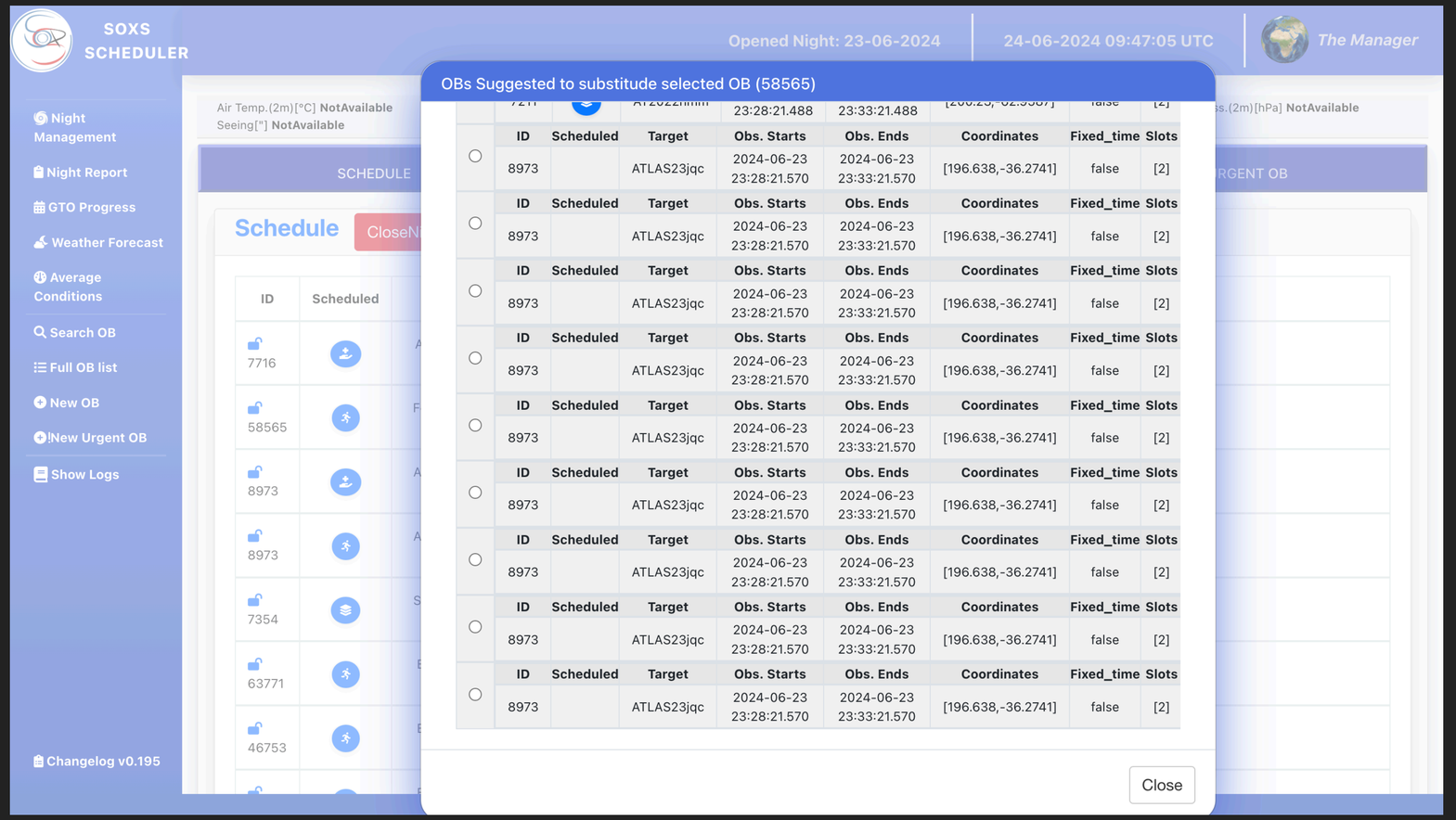
- Directly sent through the Scheduler interface
- Full customization (could actually look like this)
- Top priority target



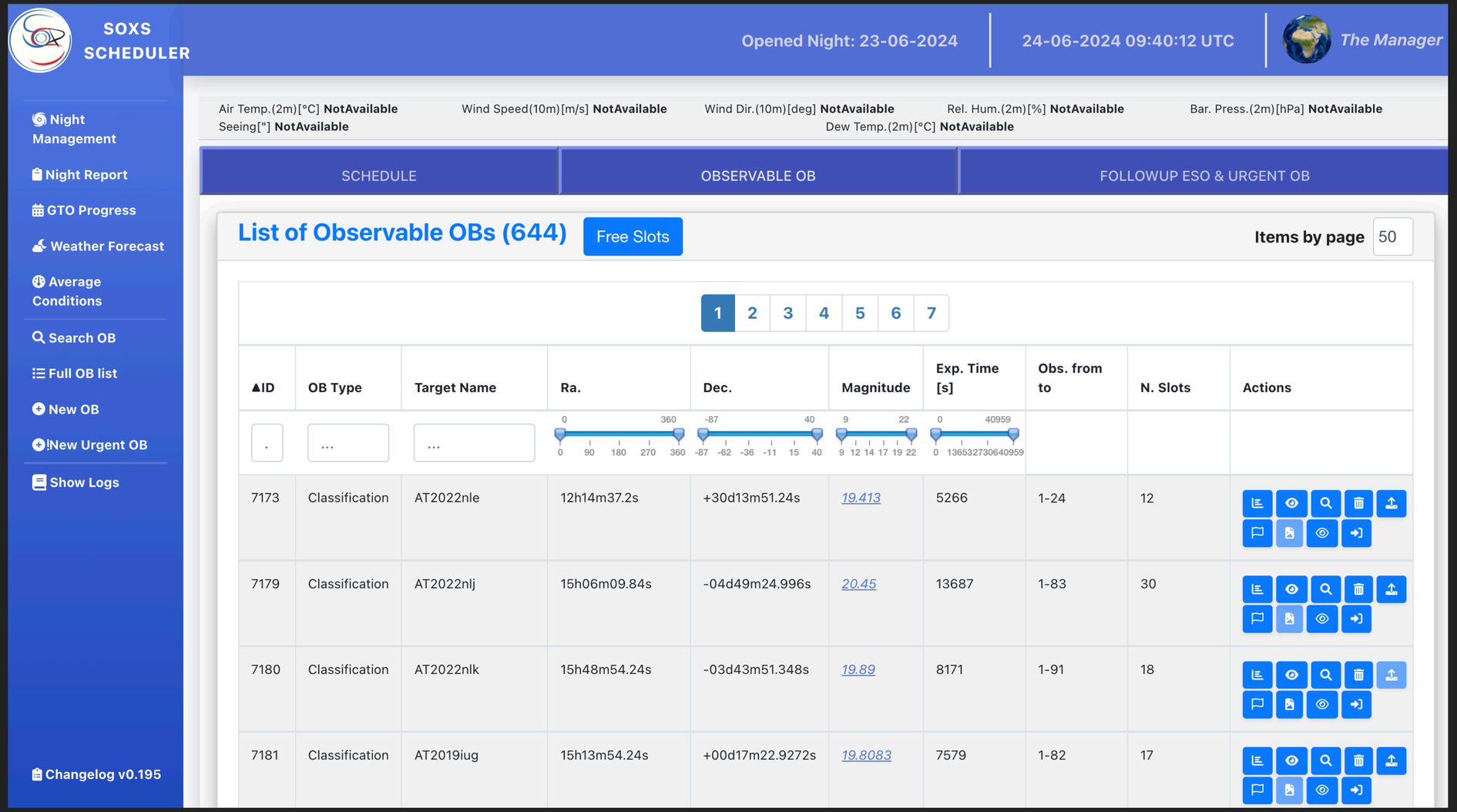
SCIENCE TEAM INTERFACE



SCIENCE TEAM INTERFACE



SCIENCE TEAM INTERFACE



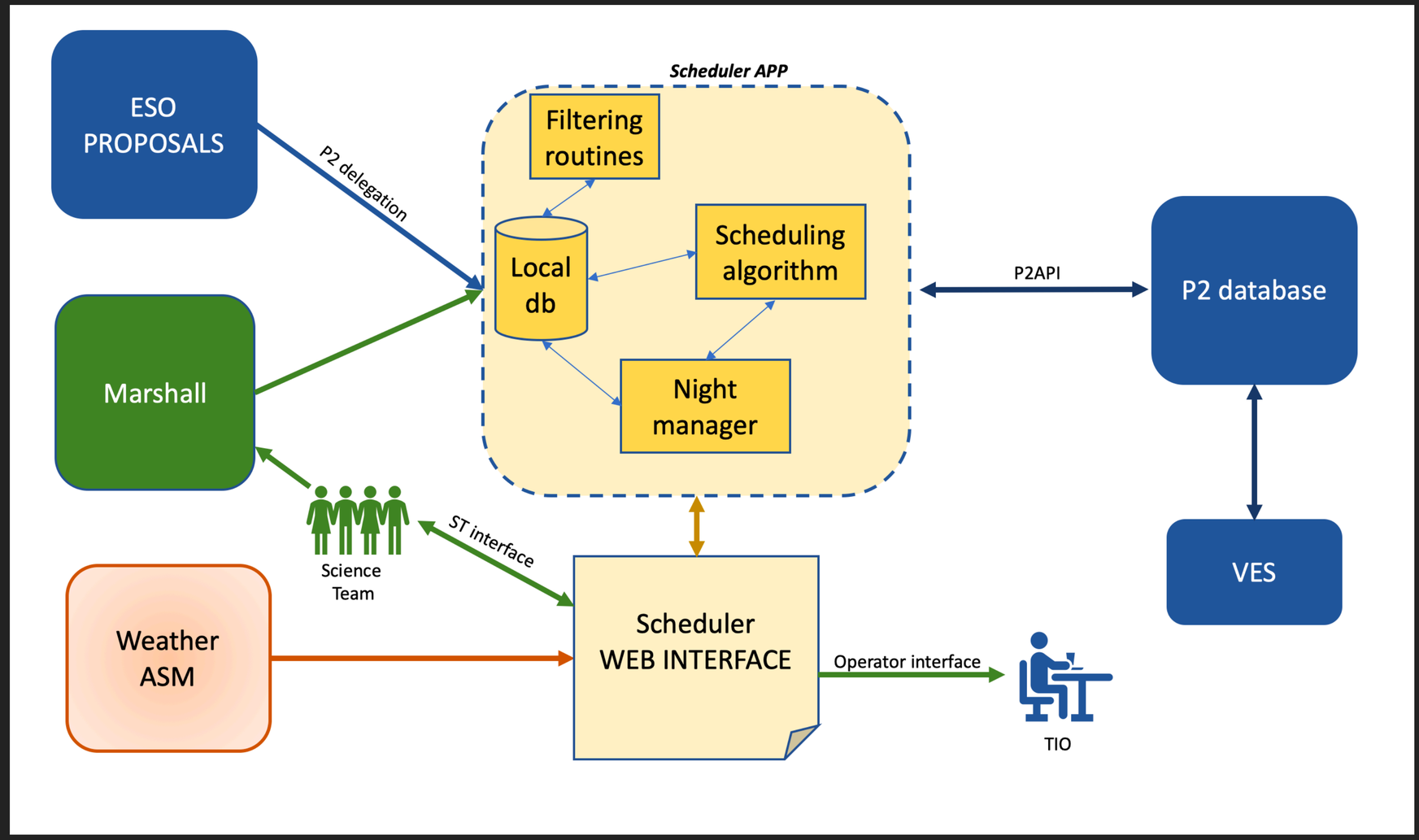
MANY USER-FRIENDLY TOOLS

The scheduler interface has many tool to aid the science team during their operations, mainly to:

- Modify the proposed schedule
- View the OB specifics
- View the Guaranteed Time of Observation status
- View weather forecasts

COMPONENTS

- Fed by Marshall and ESO p2DB
- Several routines distribute the workload
- Synched to P2 with automatically

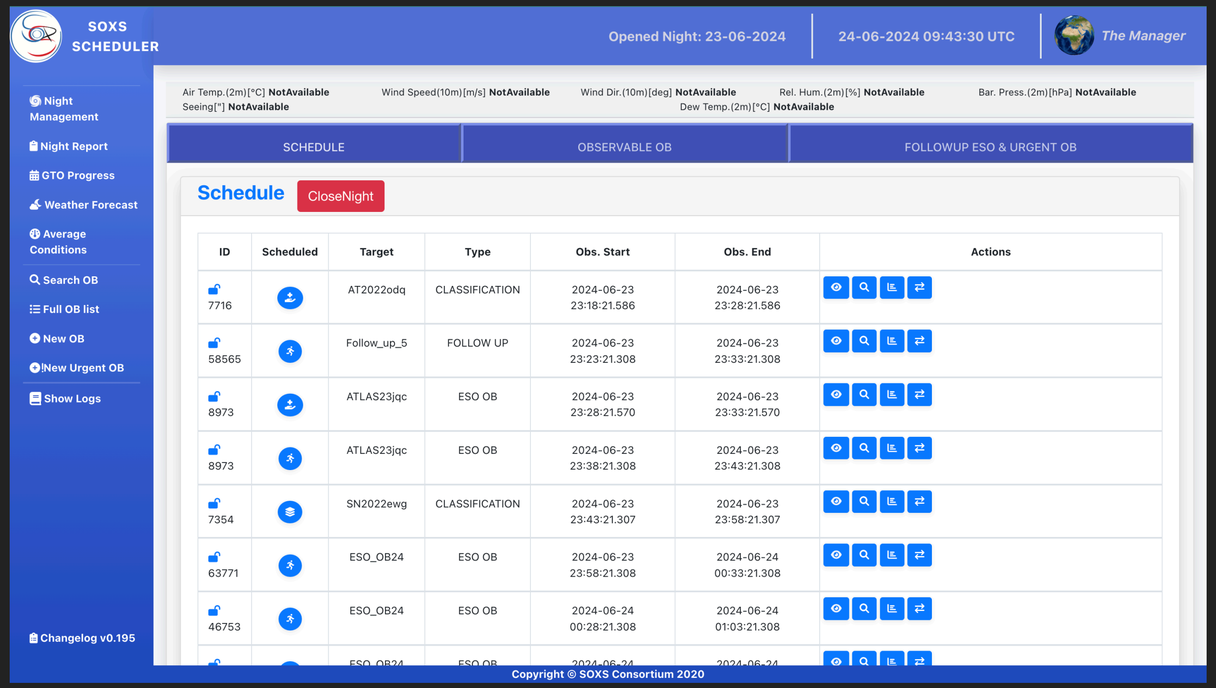


FILTERING

- Using Astroplan
- In order to identify the time period during the night when the target respects all given constraints (airmass, moon sep)
- Every OB is associated with an “observability grade” based on target coordinates, observatory location
- Scripts run every 300 s, the OBs are always updated

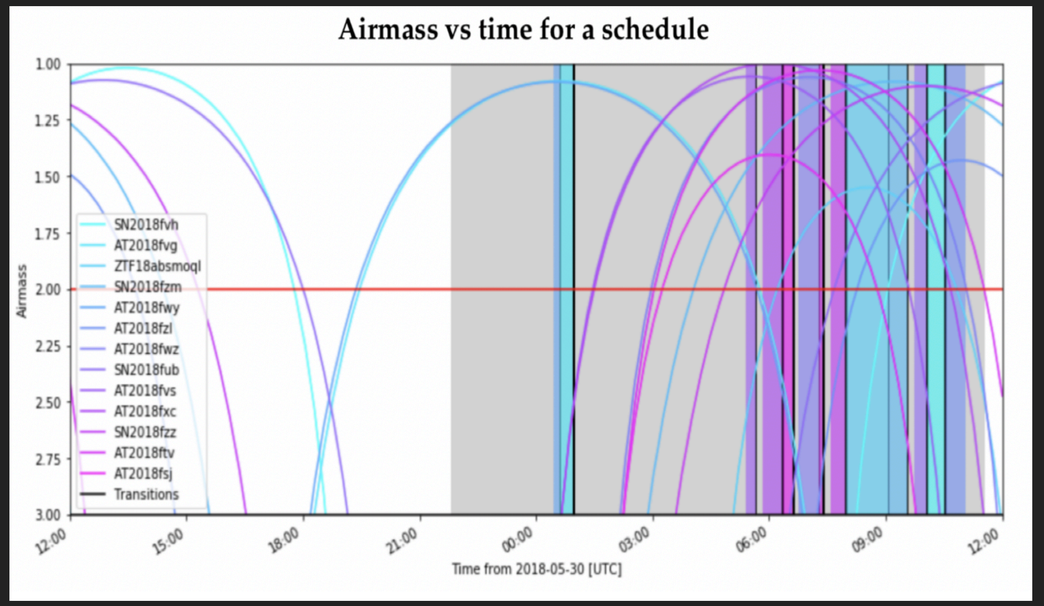
SCHEDULING

- “Static” schedule
- 2 algorithms
- Follow up OBs with Priority Scheduler from Astroplan
- Classification targets with GGF
- Still TBD, we may open up other possibilities



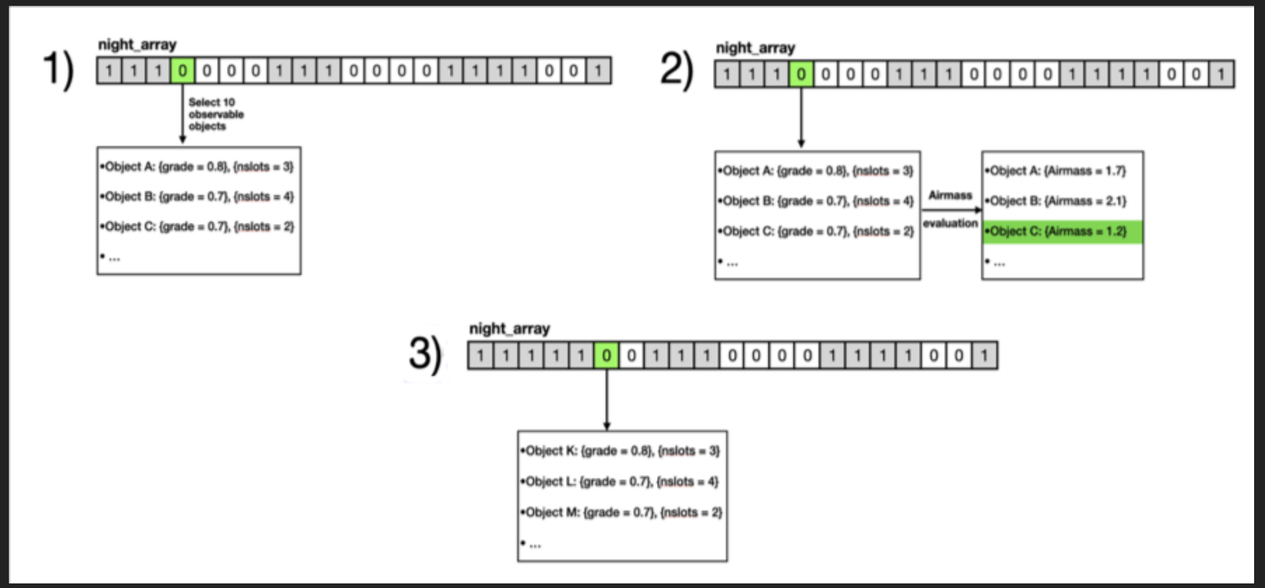
PRIORITY SCHEDULER

- Sorts and schedules by priority (duh…)
- Optimizes constraints with 1 minute time-grid



GRADED GAP FILLER

- During filtering, every OB is associated with a “grade”
- The algorithm is greedy, finds a hole in the schedule and wants to fill it
- Chooses 10 targets with highest grade, checks their constraints for that hole
- Winner gets scheduled



NIGHT MANAGING

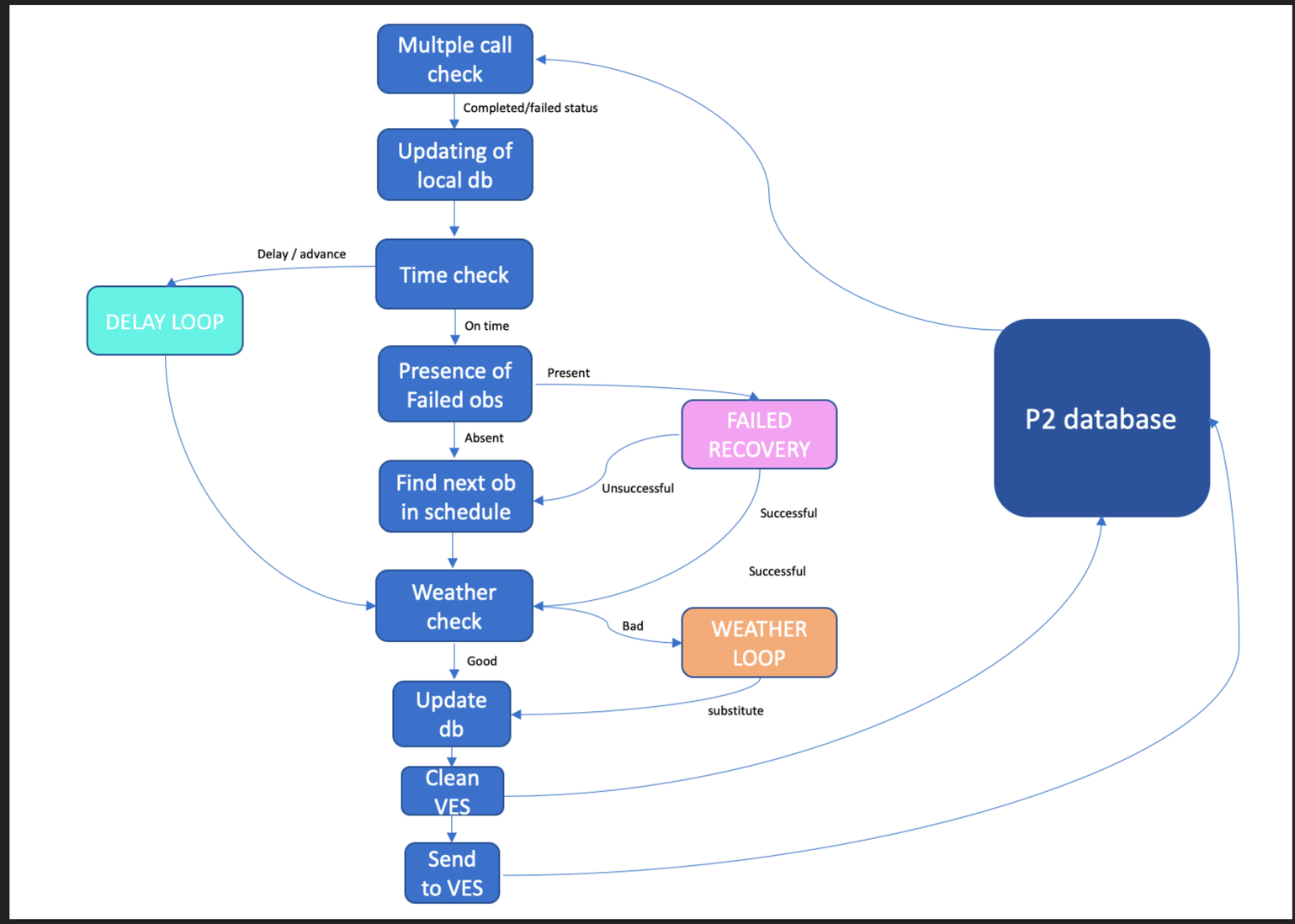
- “Static” part of the night is optimized
- A dynamic algorithm reacts in real time
- Actual automation, TIO will not need to intervene unless something goes wrong
- The algorithm is summarized in the “Next OB” function

NEXT OB FUNCTION

We identified three main sources of problems:

- Delays/advances in schedule
- Presence of failed OBs
- Weather conditions change

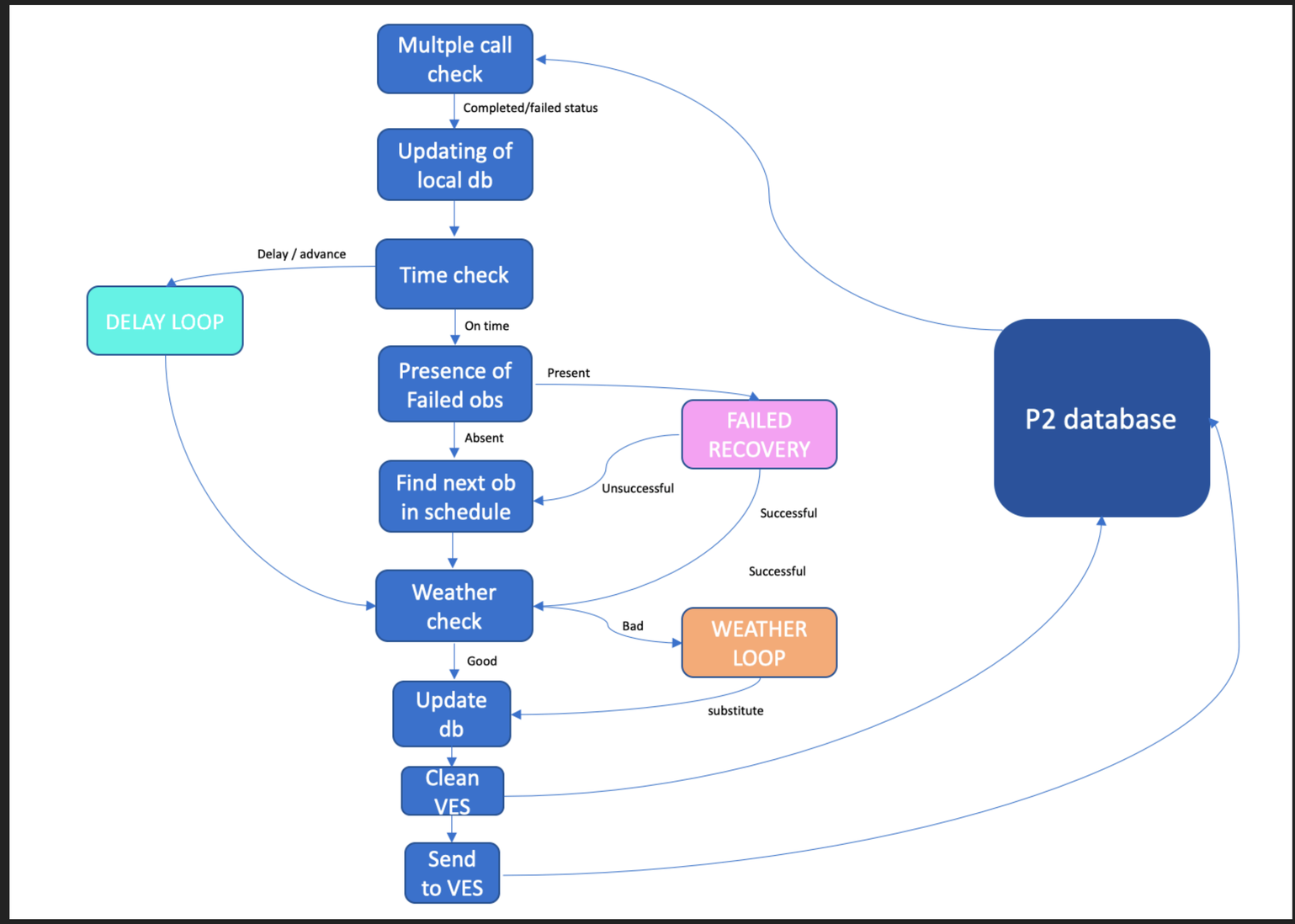
Each of them was coded in a “recovery loop”



NEXT OB FUNCTION

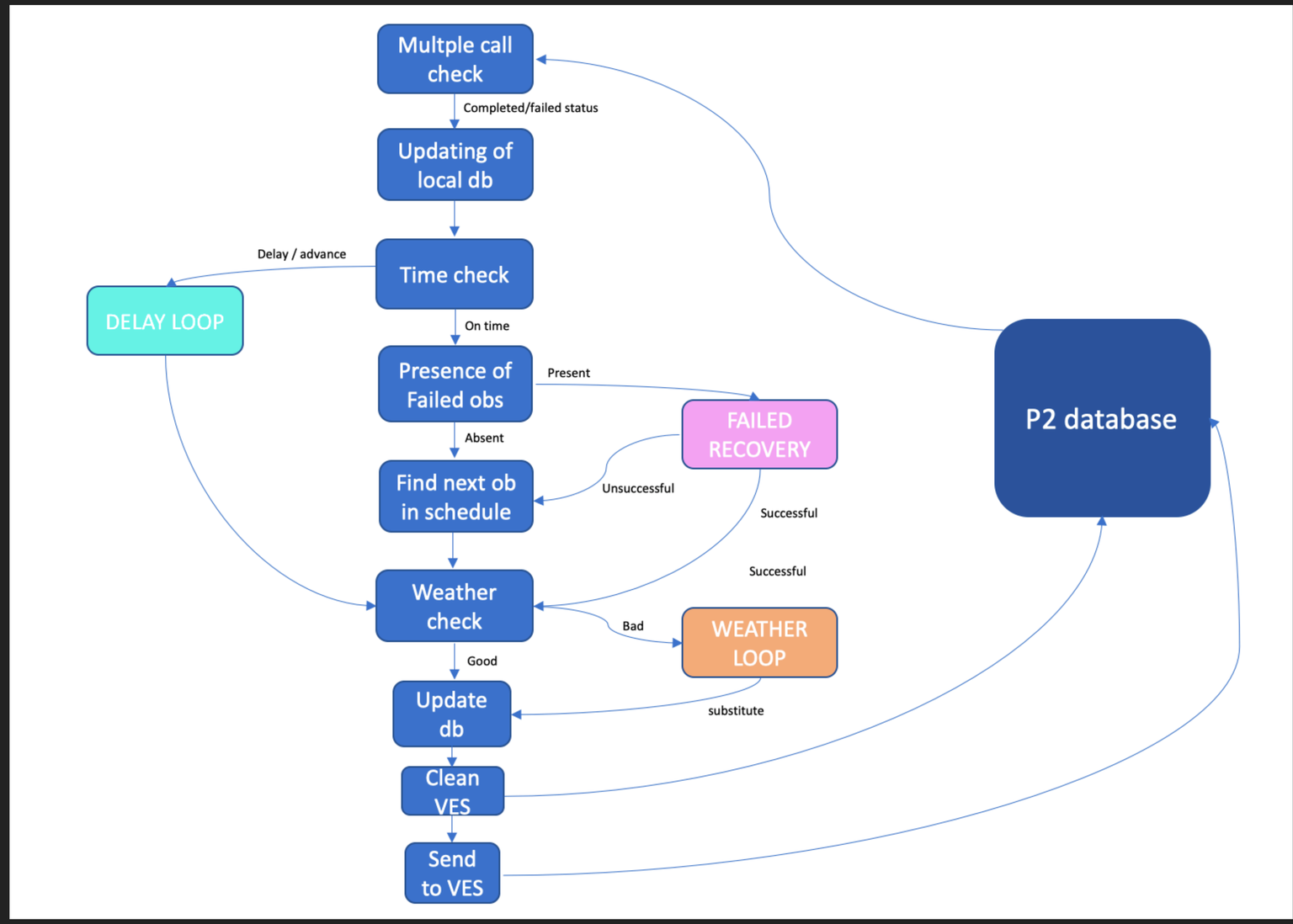
They all share common ideas:

- Performing the related check
- Preserving and re-optimizing prioritized targets
- Re-arranging remaining time



NEXT OB FUNCTION

- Relieves the Operator from any decision-making process
- Every change is promptly synchronized with P2 and local db
- The algorithm is automatically triggered



WARNINGS

- Talking to a machine: your word is its law
- Overbooking is bad
- Scientific priority will have the last word
- We will all have to deal with our internal gto: shoot your shots carefully!

THE IMMEDIATE FUTURE

- Real life testing in September (la Silla)
- A few housekeeping/troubleshooting features to be added
- We do listen to user feedback!

THE IMMEDIATE FUTURE

- Real life testing in September (la Silla)
- A few housekeeping/troubleshooting features to be added
- We do listen to user feedback!

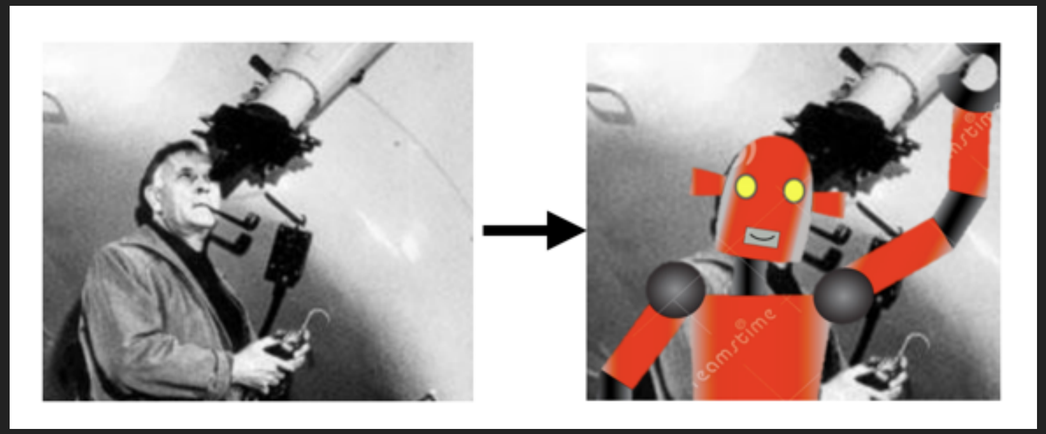
THANK YOU!

THE SCHEDULER

THE SCHEDULER

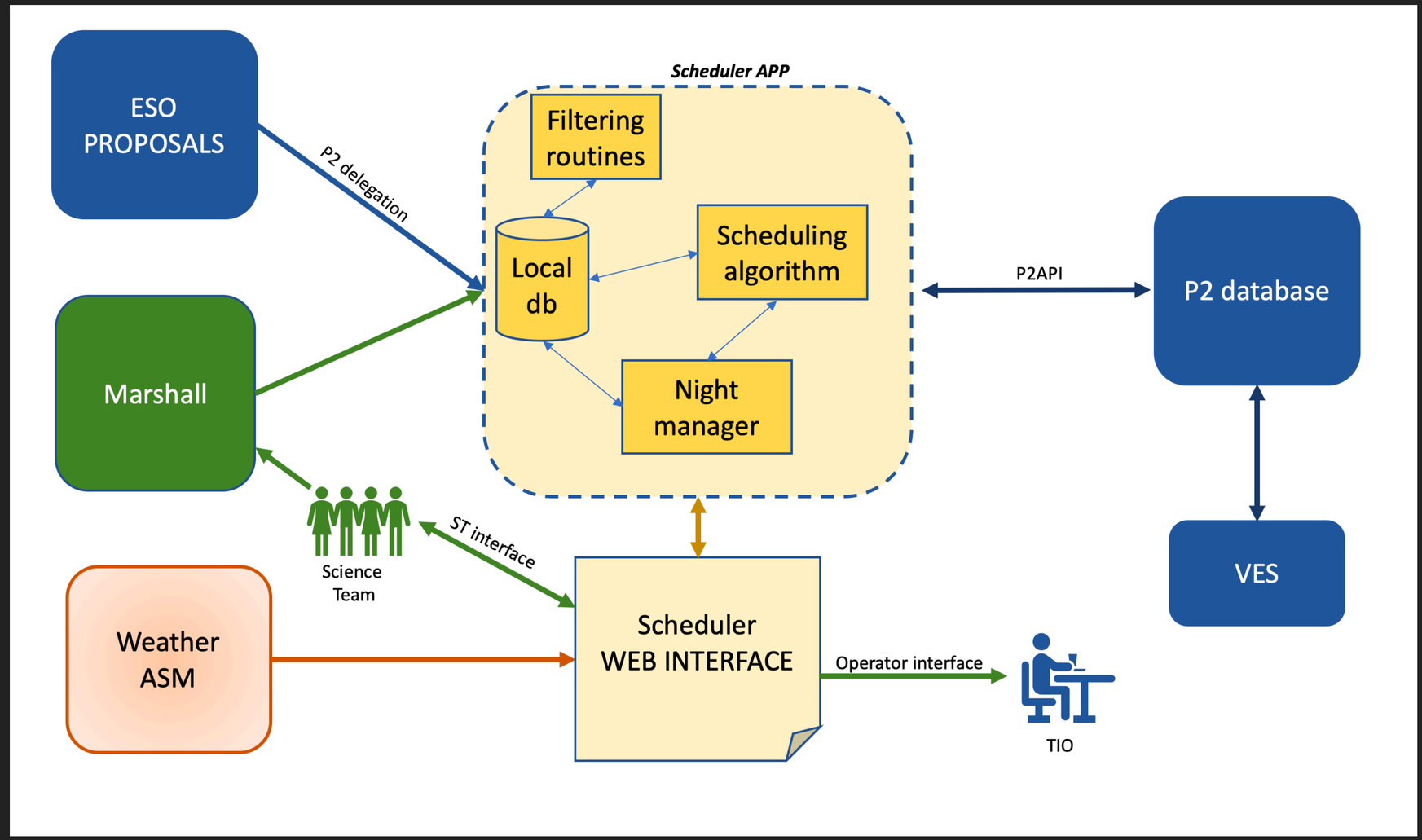
GOALS AND REQUIREMENTS

- Full automation of the scheduling process
- Providing a schedule of observable targets
- Optimization of the observation quality and time at the telescope
- Reliable automated decisions for management of unforeseen events
- Operation tracking
- Flexibility



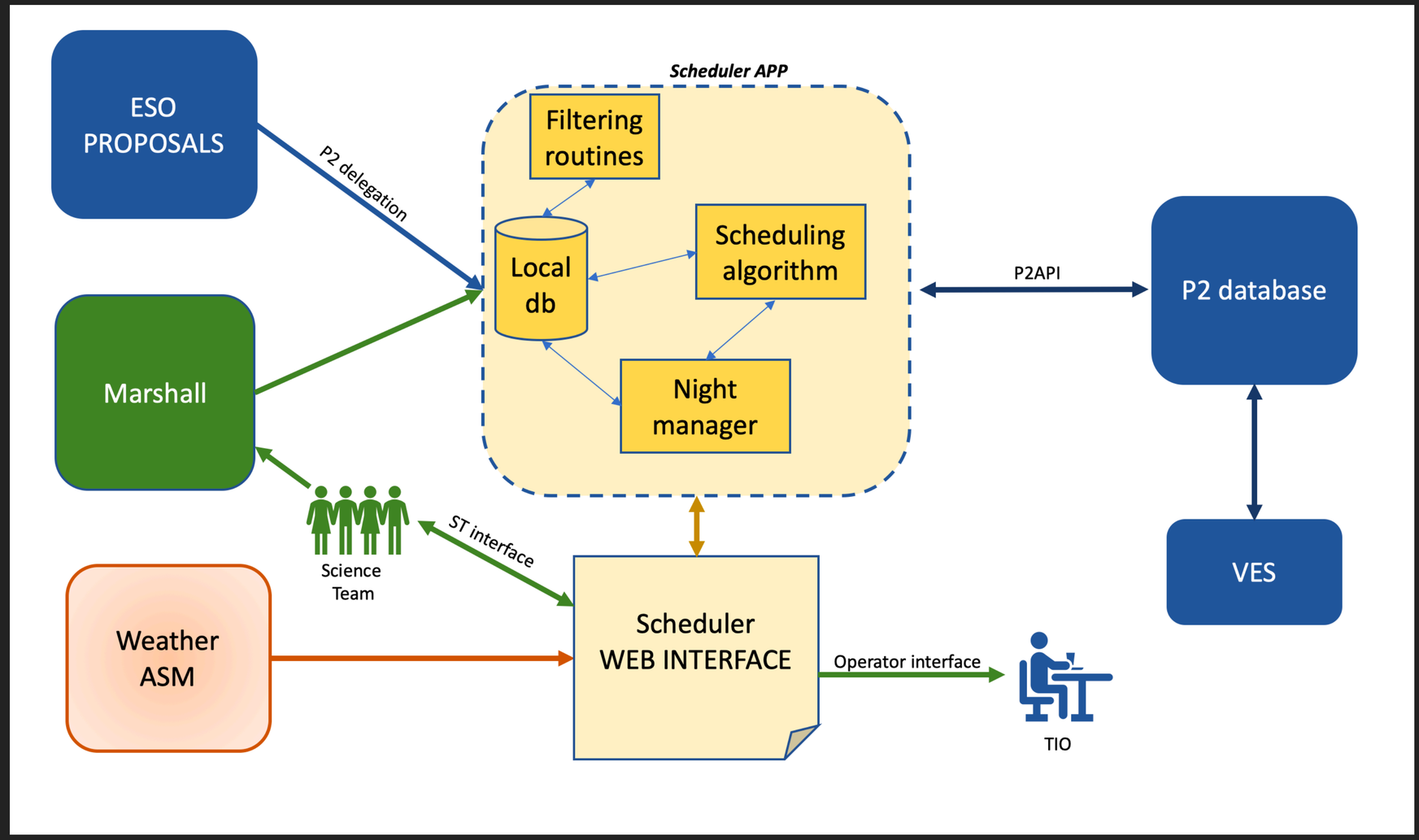
THE SCHEDULER

- RESTful API
- Python, Flask micro-framework
- MySQL database
- Containerized using Docker
- Applicable to any ground-based facility



THE SCHEDULER

- Fed by various databases
- Various operating routines deal with different tasks
- Accessible by different users with their customized interface



KEY CONCEPT:

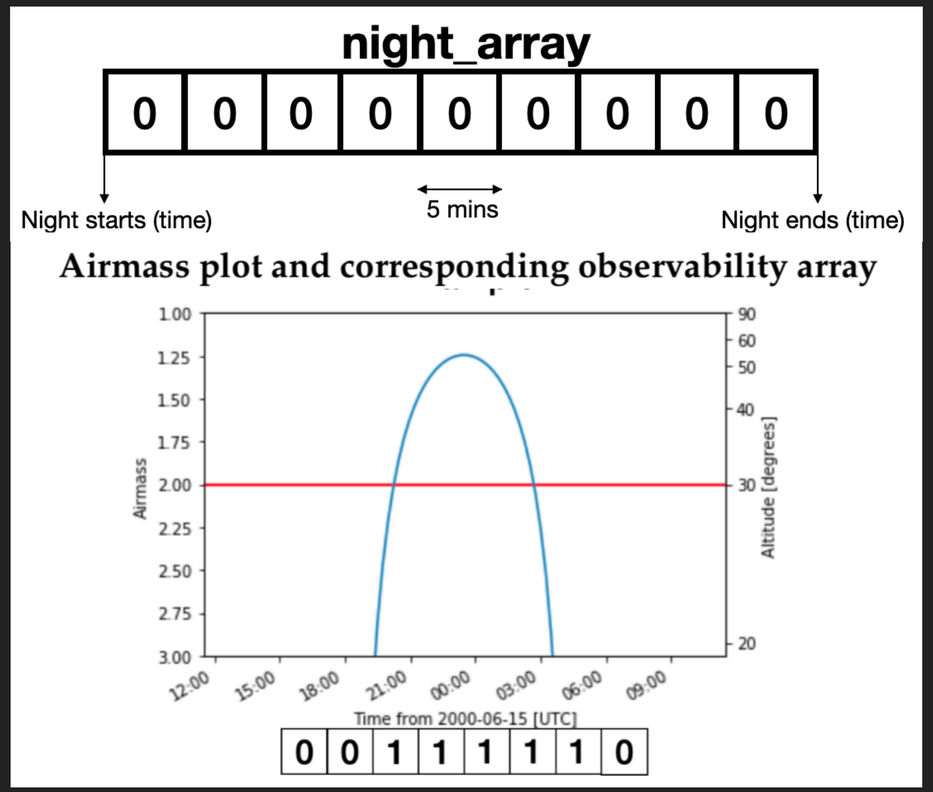
Observing Blocks (OBs) are the smallest unit for astronomical observations, containing all the information needed to carry it out (coordinates, exposure time, instrument setup…)

The scheduler will work with four main types of OBs:

- Follow Up OBs (High priority - SOXS)
- Classification OBs (Low priority - SOXS)
- ESO OBs (High priority)
- Urgent OBs (Highest priority - SOXS)

FILTERING

- Using Astroplan library
- Parametrization of the night with array
- Observability arrays are created for each OB
- Processing and Updating scripts run every 15 minutes



GRADING SYSTEM

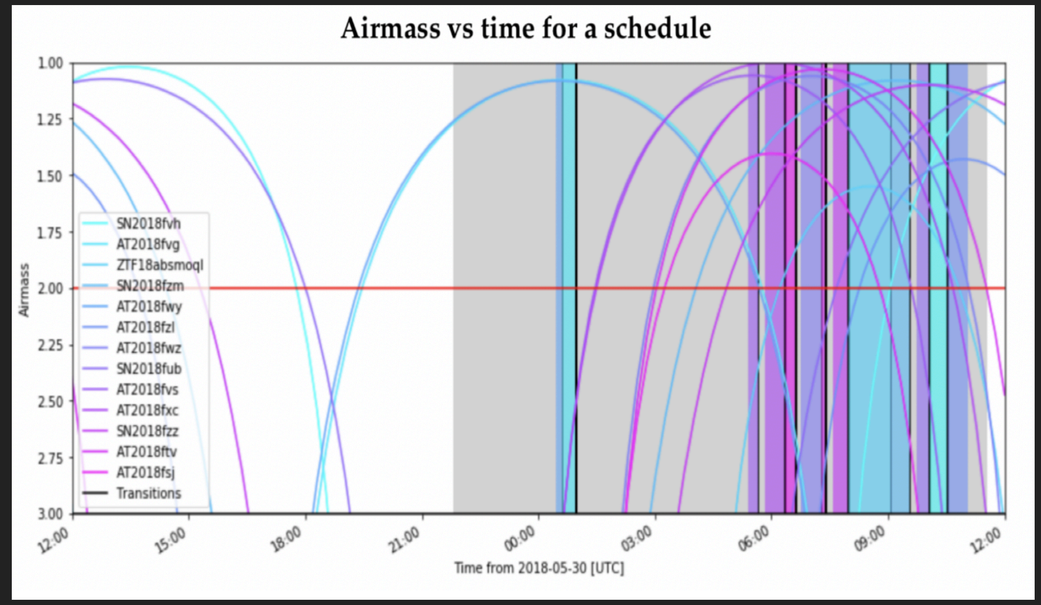
- Adaptation and extension of what is currently used for observation planning
- every target is associated with a grade P, i.e. the convolution of all the probabilities P_i the target has to satisfy the i-eth constraint
- e.g. for airmass:

$$P_z = \frac{h_z}{h_z(z_{requested} = 1.7, \delta = lat_{observatory})}$$

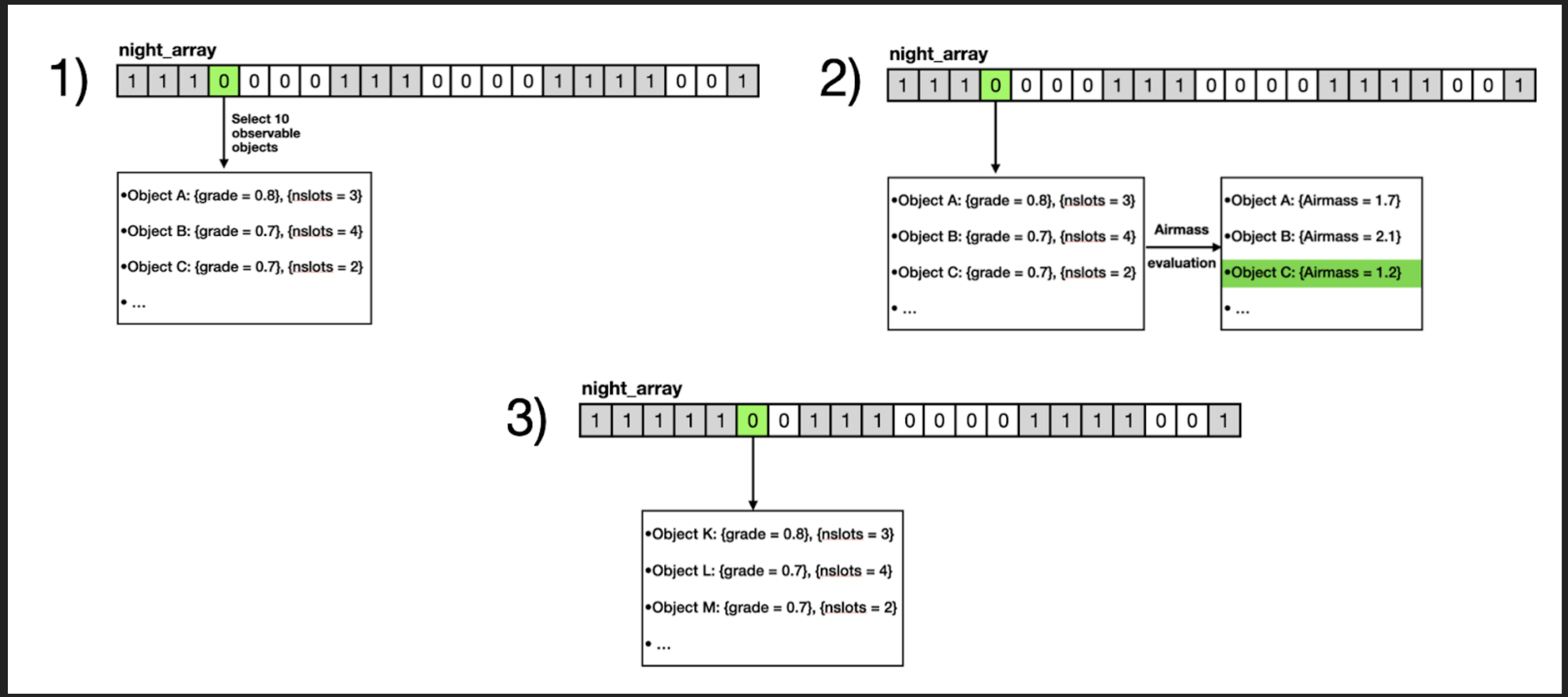
$$h_z = 24/360 * a\cos\left(\frac{\sin(alt) - \sin(lat_{observatory})\sin(\delta)}{\cos(lat_{observatory})\cos(\delta)}\right)$$

SCHEDULING ALGORITHM – PRIORITY SCHEDULER

- Priority scheduler, built-in scheduler in Astroplan
- Sorts targets by priority, finds for each one the best time of observation
- Lacks the objective of making complete use of the night
- Computationally demanding
- Used for Follow Up targets and ESO proposals



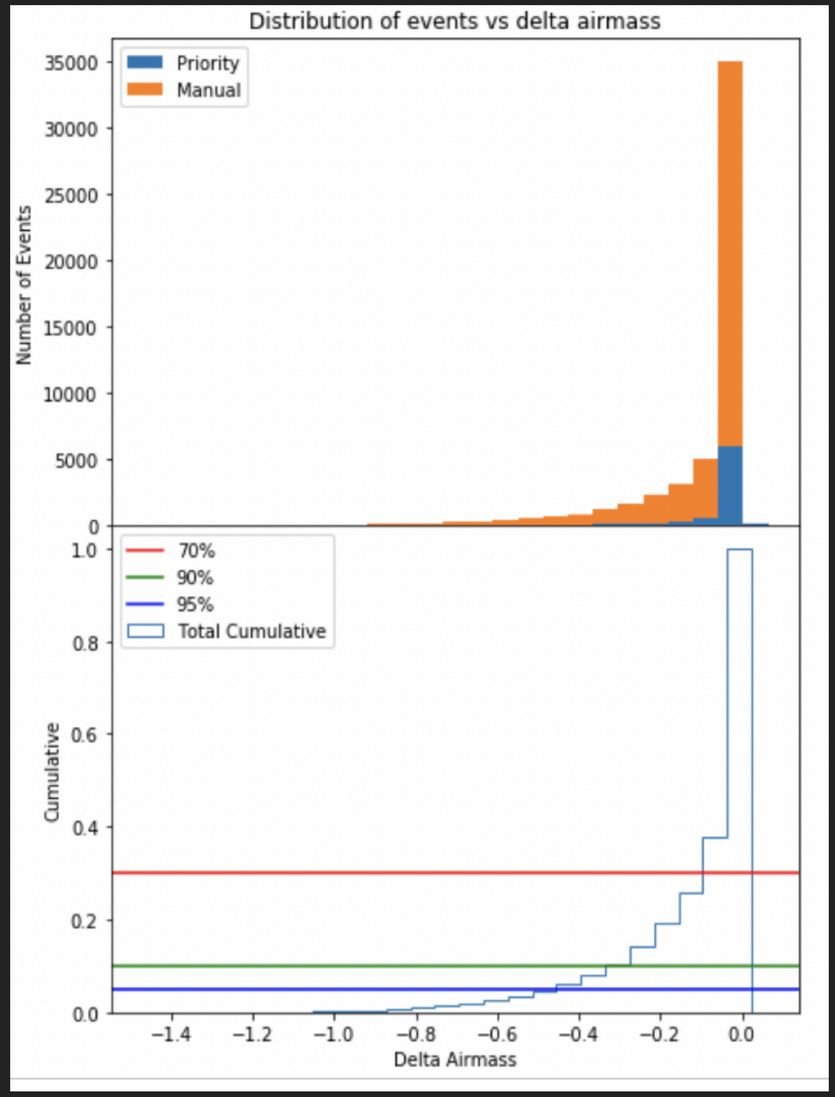
SCHEDULING ALGORITHM – GRADED GAP FILLER (GGF)



PRIORITY AND GGF COMPARED

Can we trust this algorithm?

- Our first Monte Carlo simulations, with the 2018 ePESSTO data
- 40 different nights simulated, with 100 schedules per night
- 90% of events are scheduled at airmass less than 0.3 from their best -> deg

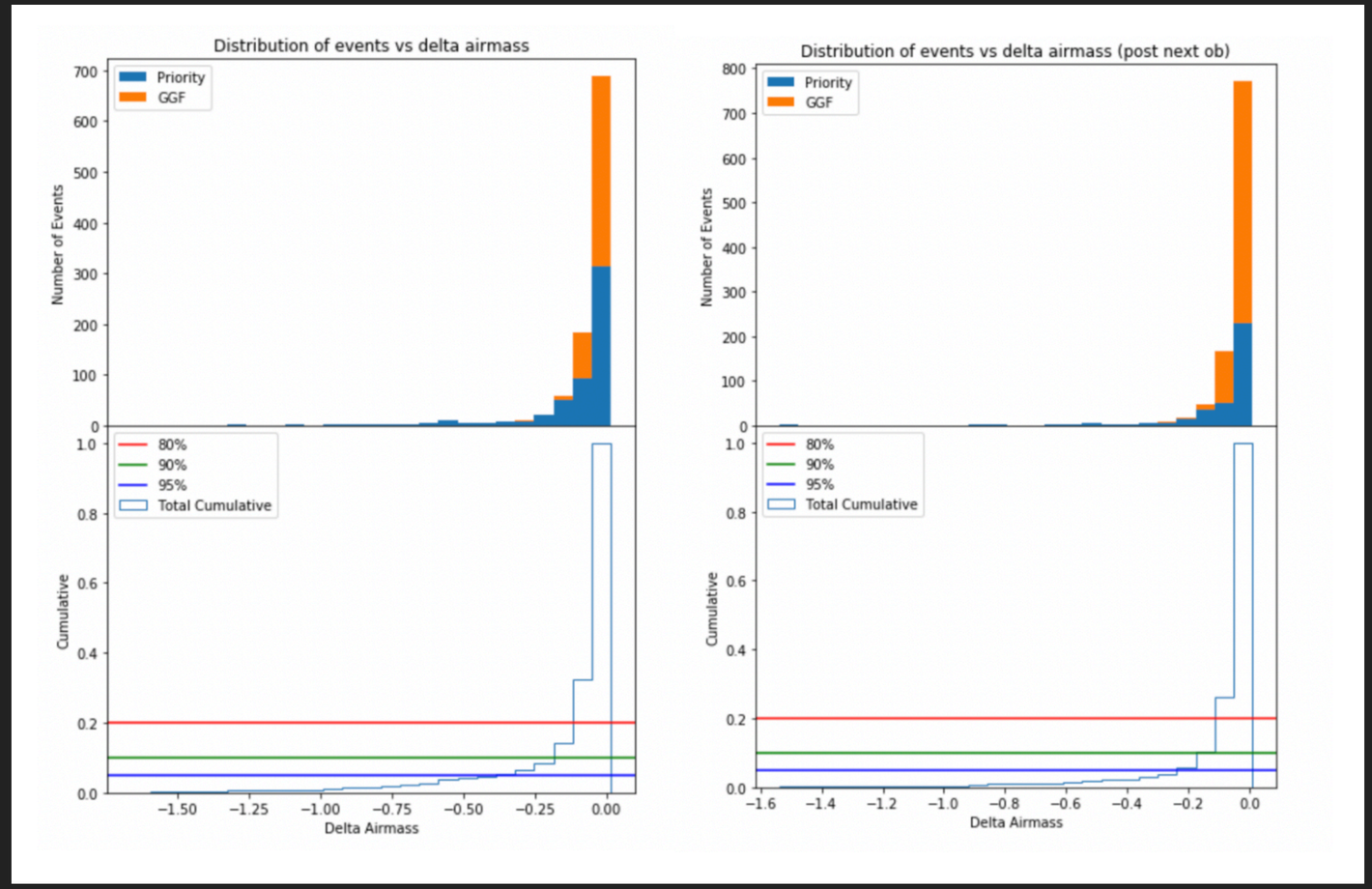


STRESS TESTING THE ALGORITHM

- Summer nights
- Gross overbooking (an average of 11/day High Priority targets with exposure of about 40 minute)
- Randomized weather conditions (based on published ESO data)
- Further “delays” every 2-3 days

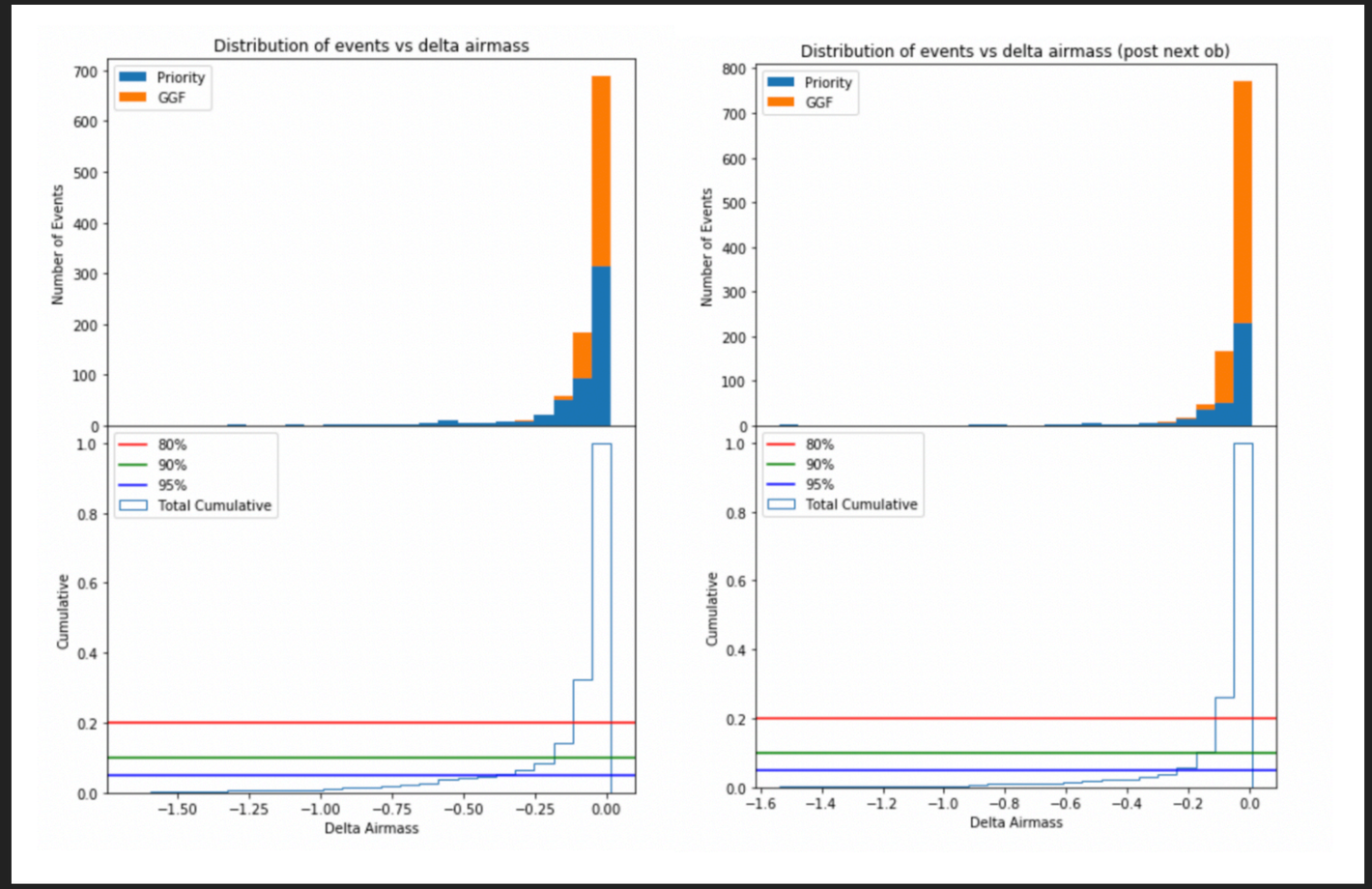
STRESS TESTING THE ALGORITHM

- 54.4% of high priority targets observed within a day from creation
- Further 18.2% recovered within 30 days.
- Highlights the important of the work from the Science Team.

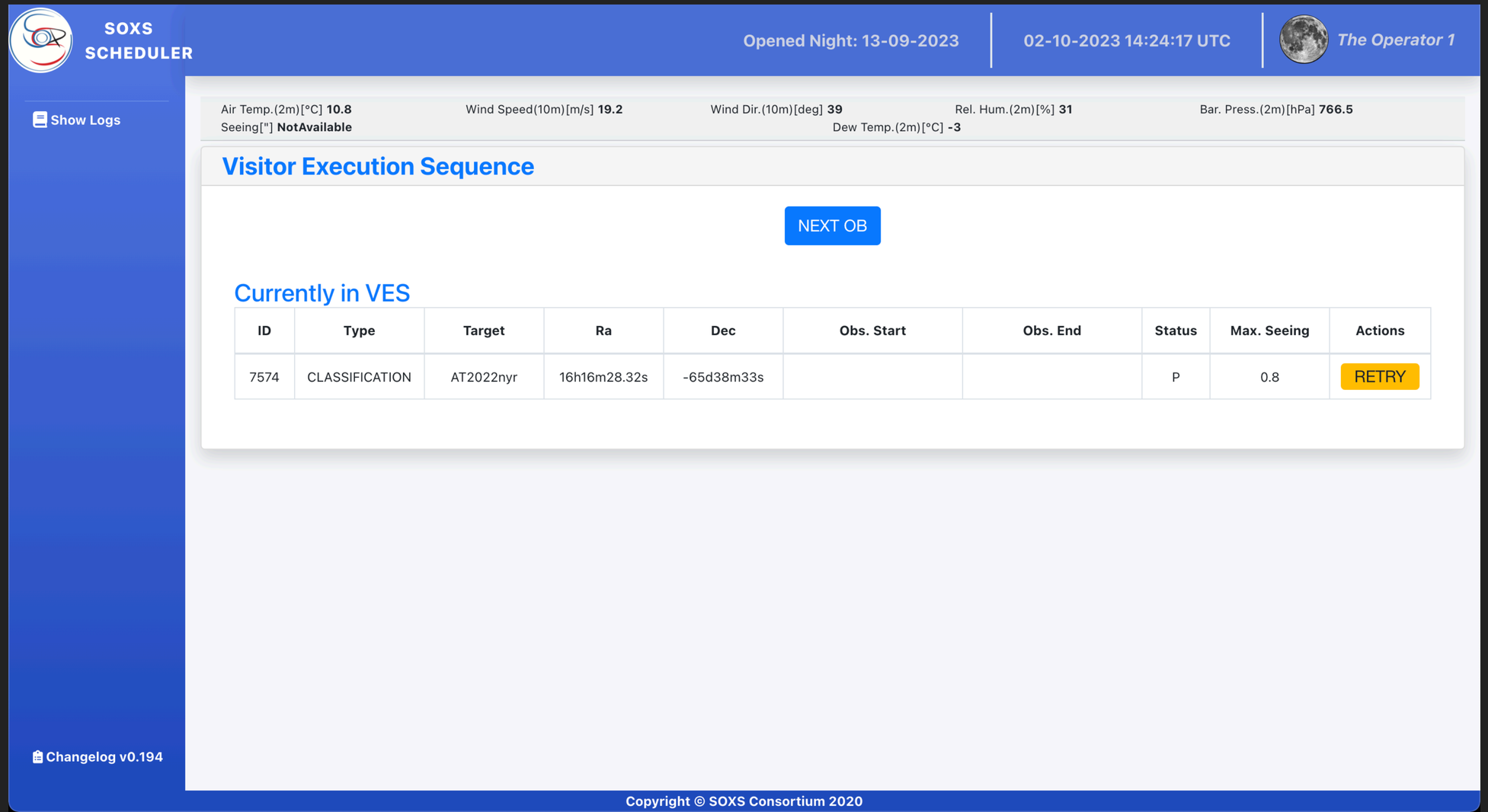


STRESS TESTING THE ALGORITHM

- 95% of the resulting obs were observed within 0.2 airmass from their best
- For a 45° altitude, it corresponds to 3° vertical shift



THE OPERATOR INTERFACE



THE DAEMON

- Even more automated
- Exploits the characteristics of the Broker of Observing Blocks
- Actions by the TIO only in case of problems