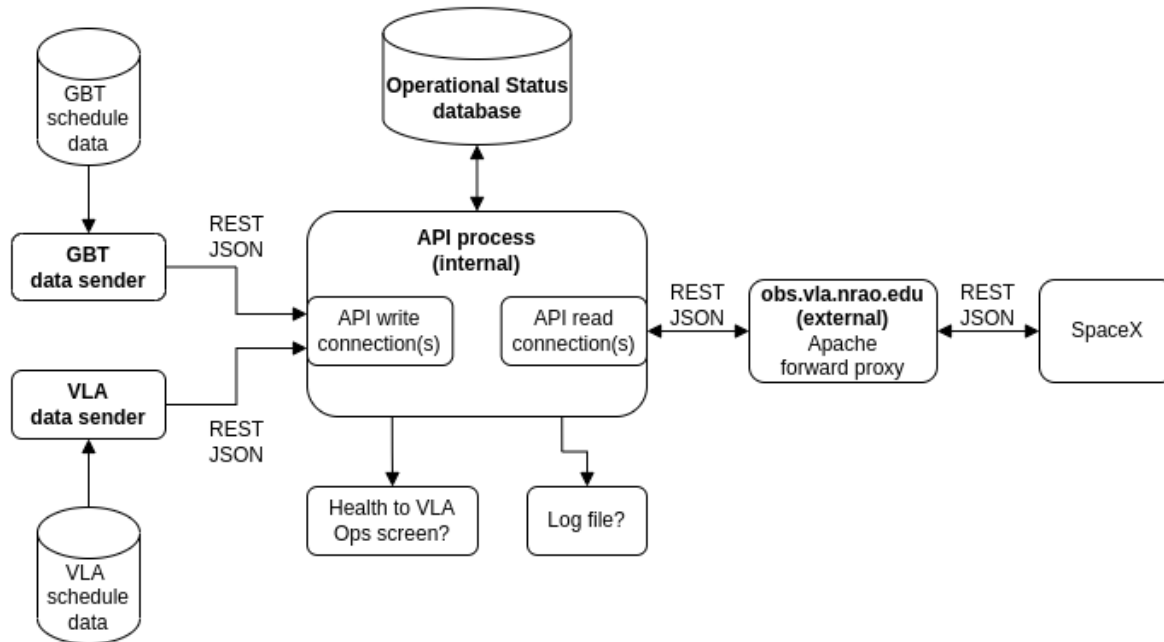


Hello World Approach

This page will outline efforts to develop a "Hello World" version of ~~Obs. Status Sharing~~ Operational Data Sharing (ODS) by "end of summer" 2023. "end of summer" ~ Autumnal Equinox = September 23, 2023 😊

High level design diagram



Component Descriptions

GBT data sender - will retrieve GBT schedule data and send to Obs Status DB as schedule data becomes available. Schedule data is usually available well in advance of 20 minutes prior to observation, but schedule can change.

VLA data sender - will retrieve VLA schedule data and send to Obs Status DB as schedule data becomes available. VLA schedule is not currently fixed 20 minutes prior to observation. Will require coordination with Science Ops, but if 20 minute commitment can be agreed schedule data is available to send to Obs Status DB.

Obs Status DB - will aggregate GBT and VLA schedule data for query by API process. Will be hosted on VLA database server. Growth is not anticipated to be that large even over a few years. Schema needs finalization, but proposed columns are:

- site - GBT or VLA
- effective datetime UTC - used to determine effective schedule when previous schedule has been updated.
- obs start datetime UTC
- obs end datetime UTC
- center RA of the sources
- center Dec of the sources
- radius over which the sources are distributed (for GBT map observations)
- tracking rate of instrument (for non-sidereal observations)
- frequency range of observation (in Hz or MHz)
- text field (for notes)

The JSON keys are described in each Observatory data sender.

API process - will expose REST interface to SpaceX. API process will receive request from SpaceX, query Obs Status DB for GBT and VLA observation status data and format and send response to SpaceX. API format to be finalized. Interface security needs to be finalized. API process will be deployed on a VLA web server.

Tasks

1) Finalize DB schema and deploy DB.

Schema described here: [OSSDB schema json v0.1.pdf](#). Test version deployed.

2) Finalize API process security requirements and document here.

Ticket entered with helpdesk. Awaiting response.

3) Finalize API format and document here.

Described here: [ODS Database API Operation Description](#)

4) Develop GBT data sender.

In progress.

5) Develop VLA data sender.

It's described [here](#).

6) Develop and deploy API process.

Demo version deployed internally: http://turing.aoc.nrao.edu:8191/ods_data

Open Questions

1) How will the exposed interface be secured?

answer: We can use forward proxy from obs.vla.nrao.edu. obs.vla.nrao.edu rewrites all http port 80 requests to https port 443. Apache forward proxy can forward external URL (example: obs.vla.nrao.edu/obsstatussharing) to any internal host and port. We do this on obs.vla.nrao.edu for the CST running on elbow:8000 and know forward proxy passes muster with InfoSec.

2) Which VLA web server will host the API?

answer: See above. We can host anywhere internally and use forward proxy from obs.vla.nrao.edu to expose the API publicly.

3) Will VLA observational data be fixed 20 minutes in advance?

VLA and GBT will update observation data as it is scheduled. API will provide all scheduled observations current and future to the request time.

4) What VLA will send to DB and expose in the API during Maintenance and Test periods (or when OST is not used)?

answer: If test and maintenance observations need protection from RFI they schedule the observations in a way that the "data senders" can forward the schedule to Obs Status DB. Otherwise, and maybe for "Hello World", test and maintenance observations do not need to be forwarded to the Obs Status DB.

5) Will requests from SpaceX be limited to GET requests or would they be able to modify the database?

answer: API process, and by extension SpaceX or other API user, will have read-only access to the DB. However, there may be other information in the API request that warrants POST. Example: security token.

General Notes

1. "tracking rate of instrument": to distinguish between sidereal vs non-sidereal tracking (TBD)
2. The API component units need to be defined (ISO 8601 for UTC date-time, RA+Dec in degrees, etc), and API component units need to be easily translatable from DB column data types.
3. At this initial phase, the Operators are not interacting with this service (i.e., it need to be fully automated)

VLA Notes

1. Even if VLA Operations agree to set targets 20 minutes in advance, there is no guarantee that observations will end on time (for example, due to a sudden change in weather conditions or hardware failure).
2. Manual observation scripts can be read, but may not meet the 20-minute advance notice.
3. The VLA executes multi-band observations. For example, Super Novae explosions can be characterized by **all** VLA bands, with just a few minutes of separation between each band. The "frequency range of observation" needs to be compatible with what the API consumer are expecting.
4. The VLA data sender needs to be **per scan** instead of per SB. Flux and polarization calibrators are usually far from Science target field, and some SBs have Science Targets in different fields.
5. For the VLA, the **initial 10 minutes of observation are critical**. It is when the Observers are instructed to set the Attenuation gains. If the flux of a satellite changes the this setting, all the Observation block can be compromised (TBD).
6. The array configuration is not relevant (TBD).

GBT Notes

1. We anticipate pulling data from the DSS, which should give us the scheduled observations sufficient for the 20 minute window. However, this may not account for projects that are not Open Skies, such as BTL.
2. We'll be reporting scheduled targets with the understanding that observers are technically free to do whatever they want in their sessions i.e. there's no enforcement that they have to abide by what's in their proposal.
3. We're also deliberately ignoring pointing to calibration sources, raster mapping, daisy scans, etc. with the assumption that these should all be close to the source and/or aren't in scope for a "hello world" demo.

