

# A joint web archive for the SAAO and SALT

Lucian Botha

Christian Hettlage

*South African Astronomical Observatory*  
*Southern African Large Telescope*



# A data archive for the SAAO and SALT

The South African Astronomical Observatory (SAAO) operates various telescopes in Sutherland (South Africa), including the Southern African Large Telescope, one of the biggest optical telescopes in the world.

Currently the data taken at Sutherland is not made available to the public, and access for data owners is not ideal either: Clearly there is a need for a data archive.

Resources are limited, and for that reason the first phase of the archive is limited to providing a website for searching and retrieving data.

# SAAO/SALT hardware and data

Virtualization - Proxmox VE with a mix of KVM and LXC containers across both sites.

EqualLogic Dell Storage - Running on both sites with 20Gbps storage throughput.

Overall data storage ~ 300TB

Current Science Data Storage - 40 TB SALT data  
- 12 TB SAAO data

# Data flow and pipelines

Data collected on the plateau could potentially reach 1 TB per night.

Not all data is stored on our NAS. Hosted facilities copies data back to home institutes.

SALT data is grabbed from all instruments during a night's observation.

Data is sync'ed to Cape Town via bash scripts and cron jobs.

Pipelines run in the morning to reduce data and users are notified of the data product.

# General data archive workflow

Search

Login ➔

Register 👤

 CART

Target name

Resolver

Simba ▼

resolve

Right ascension

Declination

Search radius

Radius units

Arcseconds ▼

Target type

☐ Galaxy ☐ ISM ☐ Solar System Body ☐ Star

Proposal code

Principal investigator

Proposal title

Observation night

Telescope

Any ▼

Include:

☐ Arcs ☐ Biases ☐ Flats ☐ Standards

search

☒ Observation

☒ File

☒ Data Category

☒ Night

☒ Target

☒ Proposal

☒ Telescope

☒ Instrument

☒ Release Date

Previous page

401

...

500

of

21324

Next page

# General data archive workflow

Users can either register, or login with their SALT accounts (others to follow).

They can search public data and data owned by them.

Proprietary observations aren't included in search results, unless the user owns them.

Data files can be added to a cart and requested in bulk.  
Alternatively, individual files can be viewed and saved in JS9.

# System architecture

## *Database server:*

MySQL 8

(observation and admin database; the former is populated from FITS headers by a Python script)

## *Backend:*

nginx as proxy server

pm2 for running the server

GraphQL-Yoga server (on top of Express)

## *Frontend:*

nginx server serving static files (which use React and Apollo)

# API considerations

## Observation queries

- SQL where condition described as a JSON object

- Requested columns as an array of strings

- MySQL statement generated from the where condition, the requested tables and the database model

## GraphQL

- Not strictly necessary

- Input validation „for free“

- Documentation much simpler than for RESTful API, thanks to tools like GraphQL-Playground

- No more worrying about POST vs PATCH vs PUT...

Classic HTTP endpoints for authentication and file downloads



# Software used

TypeScript

graphql-yoga

PassportJS (with local strategy, cookie-based session)

React

Apollo

react-virtualized

JS9

Bulma, Font Awesome, styled components

target-position (own npm package for accessing Simbad etc.)

General tools: Reviewable, TS Lint, Prettier, Husky, Travis, Jest

# What hasn't worked that well?

Using Prisma instead of „just“ MySQL was not effective.

Apollo is a great tool (and we'll keep it), but its automated caching comes with a caveat or two.

Travis (to some extent).

Daily meetings were a great help. But there should have been a stakeholder on these meetings...

# Future developments

Cloud Computing and Storage - Commercial or Research

Data Retentions and Data Management policies

Archive future - get all instruments and telescopes on board

To become interoperable and allow users to use VO tools and applications on the archive data