

SKA-Low: growing pains

What does SKA-Low look like on the ground?

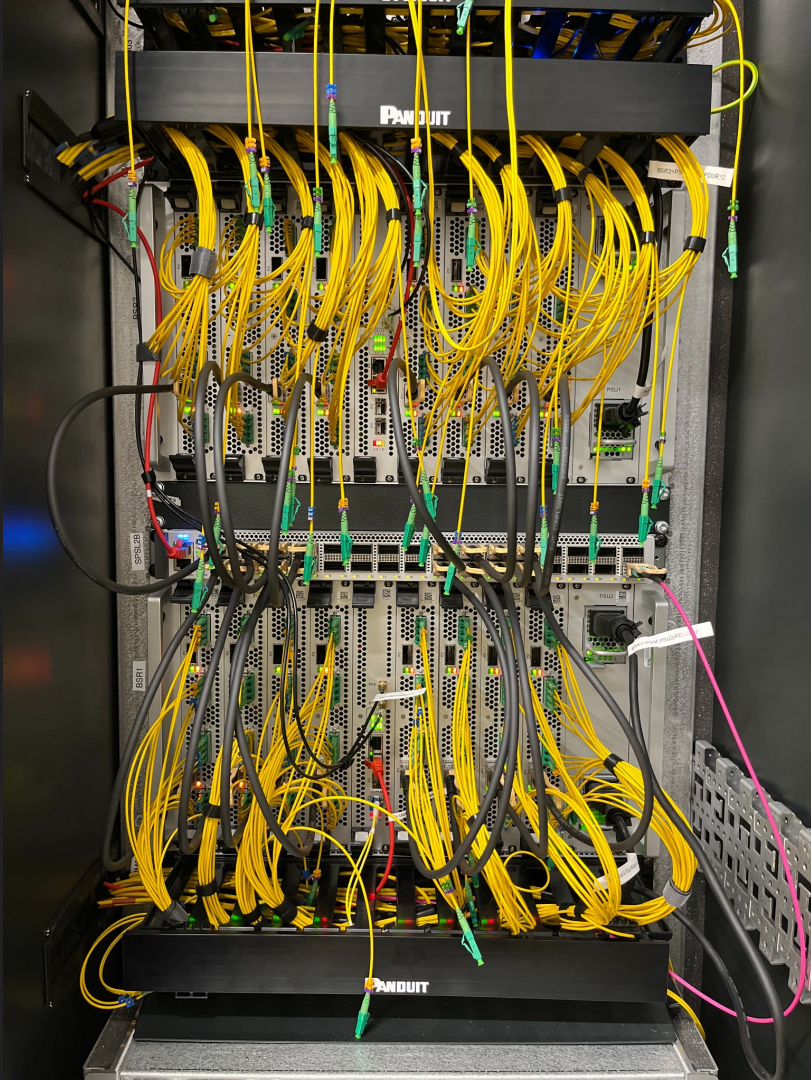




# SKA-Low station - field node

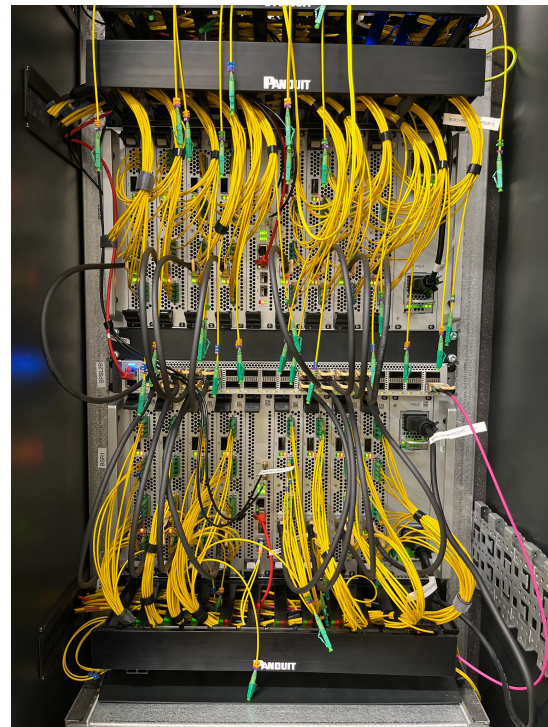
- 256 antennas
- 512 LNAs
- 256 FEMs
- 24 smartboxes
- 1 FNDH
- ~1km of 12-core fibre
- >3km of coax
- More than 1,000 physical cable connections
- **28 Tango devices**
- **More than 1,500 Tango attributes**
- Multiple hardware variants for each component
- ??? firmware variants





# SKA-Low station - signal processing

- 256 fibre inputs
- 512 pre-ADUs and ADCs
- 16 TPMs
- 2 subracks
- 1 100Gbps switch
- **21 Tango devices**
- **More than 3,000 Tango attributes**
- Multiple hardware variants for each component
- Multiple firmware and BIOS variants



# One SKA-Low station



~50 Tango devices, ~5000 attributes

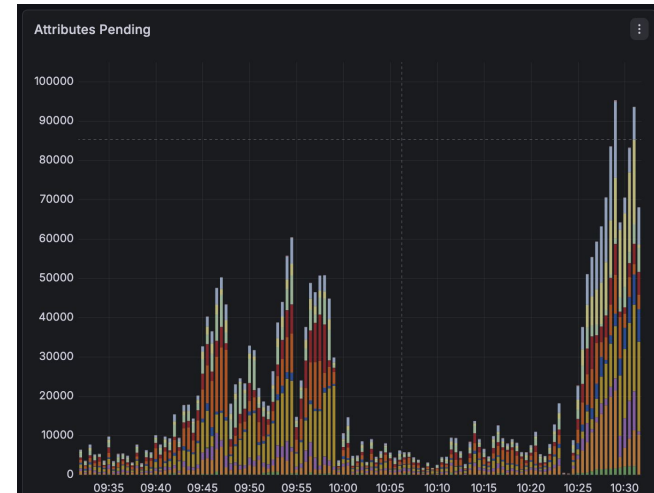


# Deployment strategy

- Self-managed Kubernetes - one cluster in Perth and one on site
- Packaged using Helm charts from many repositories
- Traceability and auditability is considered essential - held in git
- Custom tooling transforms telescope inventory into deployment configuration in the form of Helm values
- Deployment is designed to be immutable and idempotent
- json2tango and yaml2archiving run against the entire system every deployment to keep TangoDB in sync

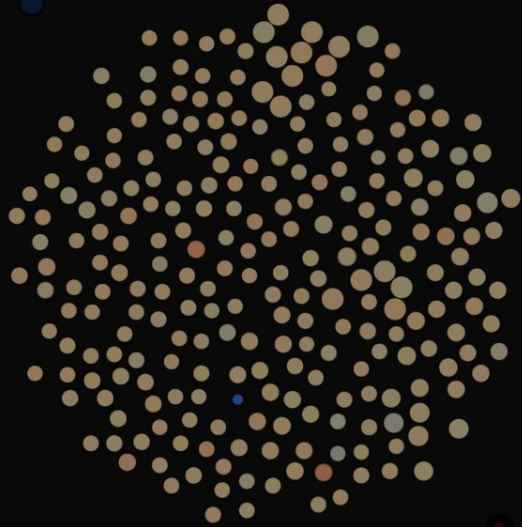
# Archiving and monitoring

- Historically - HDB++, TimescaleDB, Grafana in quasi-real-time.
- Octopus - in-house dashboard tool with realtime and control
- Most devices push, but we also poll for periodic data points
- Watching the watchmen:
  - attributes pending
  - attributes in error
  - per-attribute ingest rates

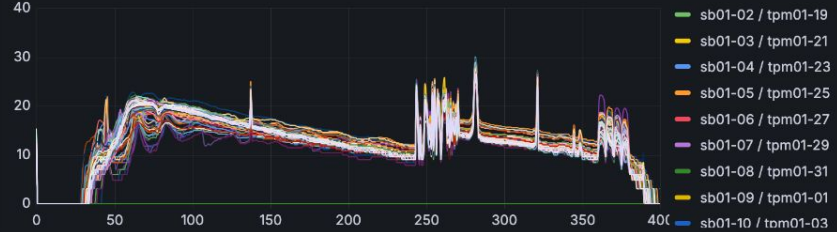


# Archiving and monitoring

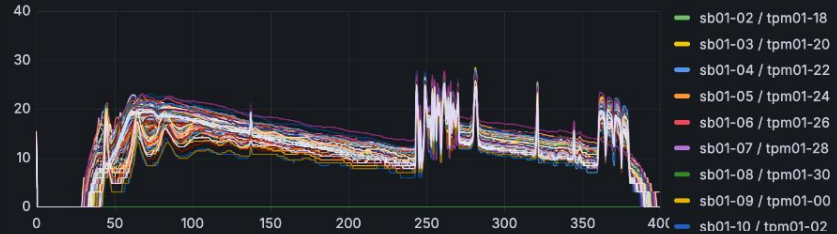
colour = ADC power, size = preADU atten, polys overlaid ⓘ



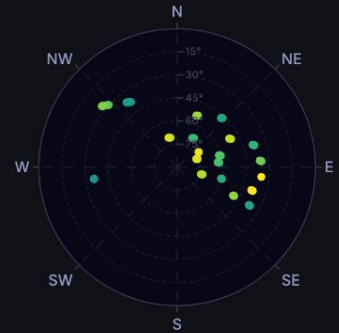
Integrated power (dB), Y pol



Integrated power (dB), X pol



s10-2 pointing over 4h



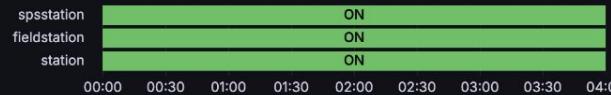
s10-2 elevation ⓘ

No data

AdminMode  
**ONLINE**

TileProgrammingState  
**Synchronized**

States

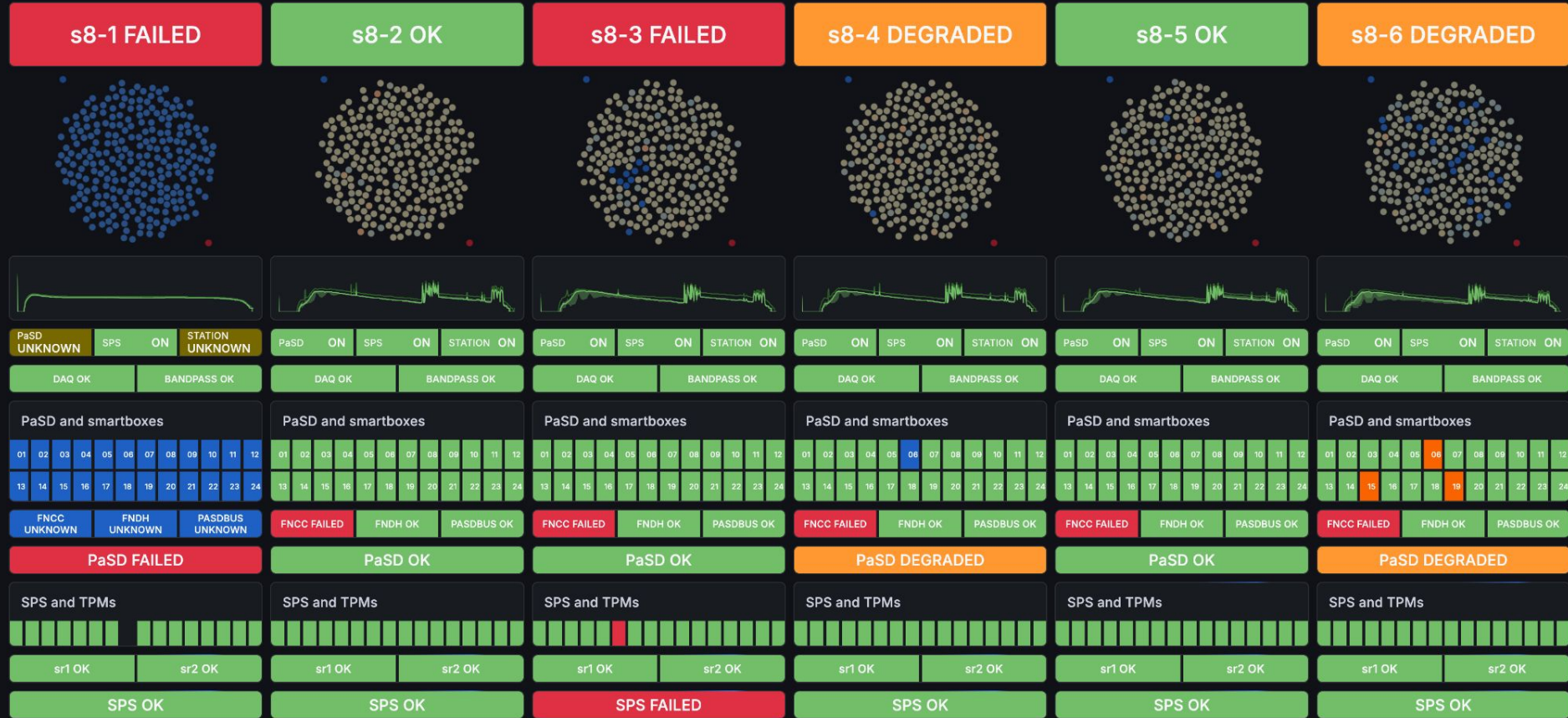


HealthStates



# Archiving and monitoring

Cluster s8



# Growing pains

- archiving
  - Volume - figuring out archiving rates and TimescaleDB quirks
  - Steady-state is OK - most problems at startup, modifying configuration, and handling interruptions and errors
  - Per-attribute operations + non-negligible round-trip time hurts
- json2tango
  - some queries slow enough to cause timeouts on Database DS
  - class properties in PyTango must be duplicated per-device

# Debugging tools - core dump catcher

- A service runs in the clusters to catch core dump files
- Makes them available for download with a web interface
- Scripts to simplify the process of installing and running gdb and debug symbols in the OCI image that produced the core dump
- A handful of bugs found, reported and fixed thanks to this tooling!
- Also gives us empirical evidence to encourage teams to upgrade their Tango versions

**Are we using Tango well at SKA?**



**Thank you!**