**MeerKAT**
- Radio telescope array to be built in the Karoo (Northern Cape Province, South Africa).
- Will comprise 64 dishes, each 13.5m diameter.
- Frequency bands (receivers):
  - 0.58 – 1.015GHz
  - 1 – 1.75GHz
  - 8 – 14.5GHz
- On completion in 2016, will be the largest and most sensitive radio telescope in the Southern Hemisphere until the Square Kilometre Array (SKA, ~ 3000 dishes) is completed around 2024.
- A 7-dish precursor (KAT-7) is already producing science data at the site.
- MeerKAT Control and Monitoring (CAM) is growing out of experience gained with KAT-7.

Here are some highlights of the MeerKAT CAM architecture.

**KATcp protocol**
- TCP transport, text-based wire format applied throughout MeerKAT.
- Each hardware or software component has a KATcp server interface resulting in a unified view of system components, and a consistent set of base capabilities.
- Standardised component introspection.

**Proxy architecture**
- Inserts a KATcp layer in front of every component, device and monitoring point.
- KATcp servers have a common set of base capabilities, including logging, heartbeat/watchdog, etc.
- Proxies provide a protection layer for hardware – safeguards can be defined in software.
- We can build and test a fully simulated system (up to the KATcp boundaries).
- The KATConnection object provides a unified system abstraction by exposing all processes as software objects, each with a familiar KATcp interface.
- Component lifecycles are managed by a node manager component.
- Component health and comms are monitored by a katmonitor component (one per node).
- Each component (hardware or software) presents a standard interface built of sensors and commands, all introspectable:

**Sensors (software-defined monitoring points)**
Each sensor has:
- name
- description
- units
- absolute range
- warning/error ranges
Sensors support a reporting strategy per client:
- periodic
- event
- event + period
Aggregate sensors are easily defined

**Commands (Katcp requests)**
Standard requests include:
- help (introspection)
- build state
- specific commands and status

**Observation Framework**
An integrated framework supporting an “antenna subarray” concept allowing multiple independent concurrent observations:
- katscheduler – allows manual or automatic scheduling of observations
- katsyscontroller – manages server and component lifecycles, and emergency actions
- katexecutor – manages observations and observation logs
- katpool – system resource manager

**KATConfig**
- Text-based templated configuration, supports multiple instances of any component.
- Central configuration is published via XMLRpc.
- KATConfig implements the static system configuration:
  - Assignment of processes to nodes
  - Process startup sequence
  - Process command-lines
  - Logging handlers/levels/destinations
- KATcp introspection handles the dynamic aspects:
  - Components and monitoring points that come and go
  - Exploratory scripting with IPython