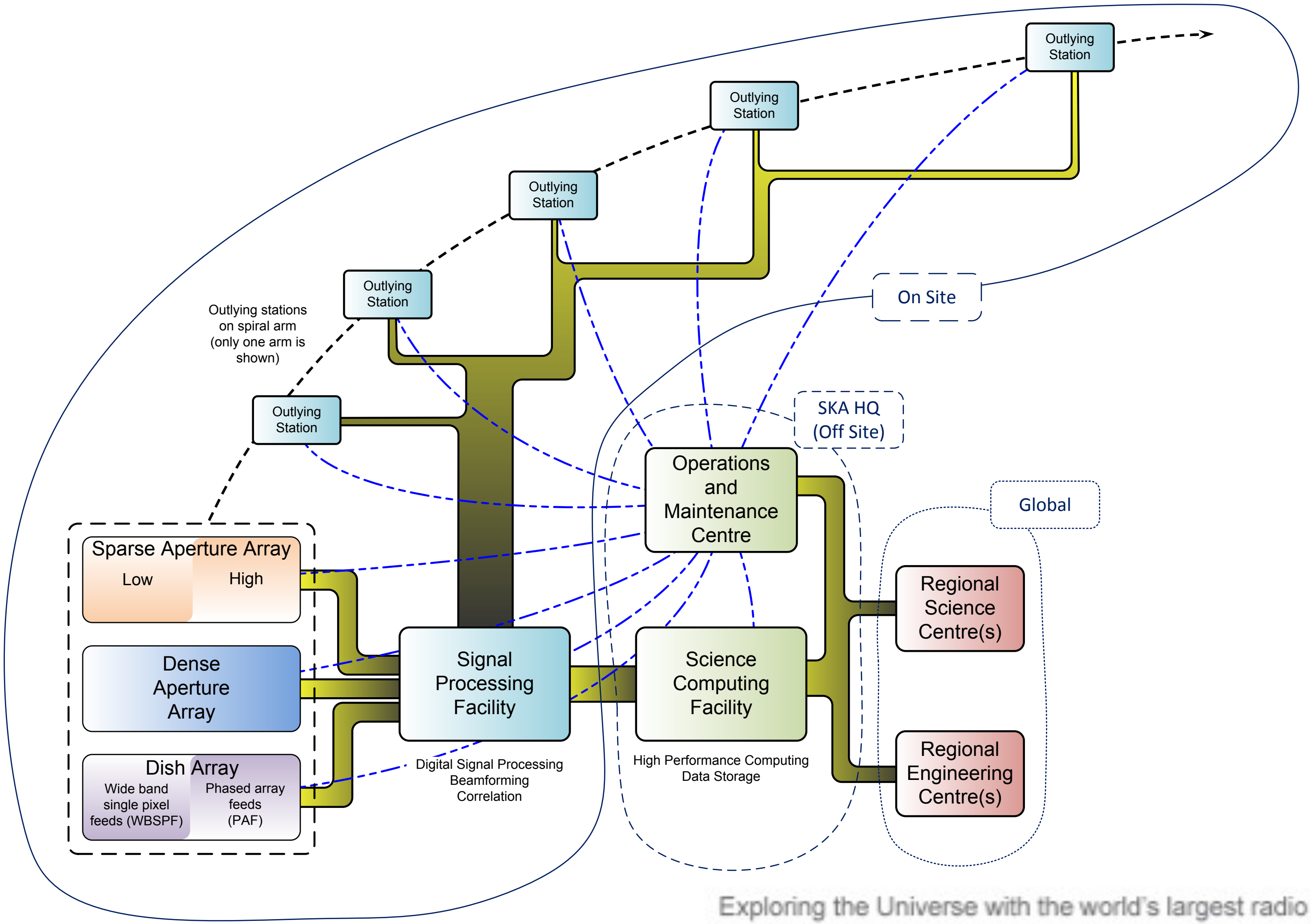
S&C System Overview

Outline

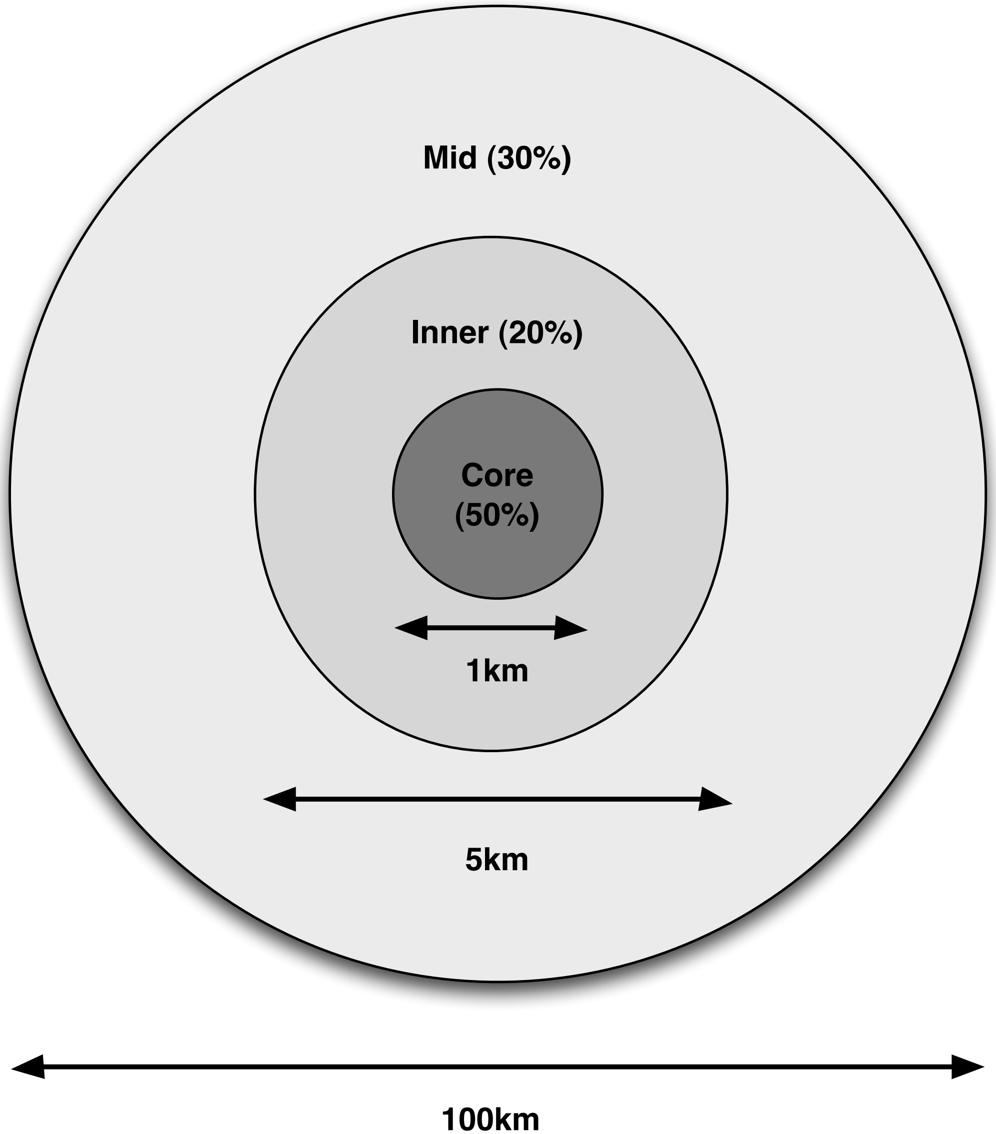
- Context
- System architecture

SKA high level schematic



Exploring the Universe with the world's largest radio telescope

Physical layout of SKA1 receptors



Science information flow

Survey Science Teams

- Science proposal
- Design studies
- Survey strategy
- Survey targets
- Survey validation
- Survey publication

→

SKA Science Office

- Science Case
- Scientific requirements
- SST interactions
- Science commissioning

→

SKA Telescope

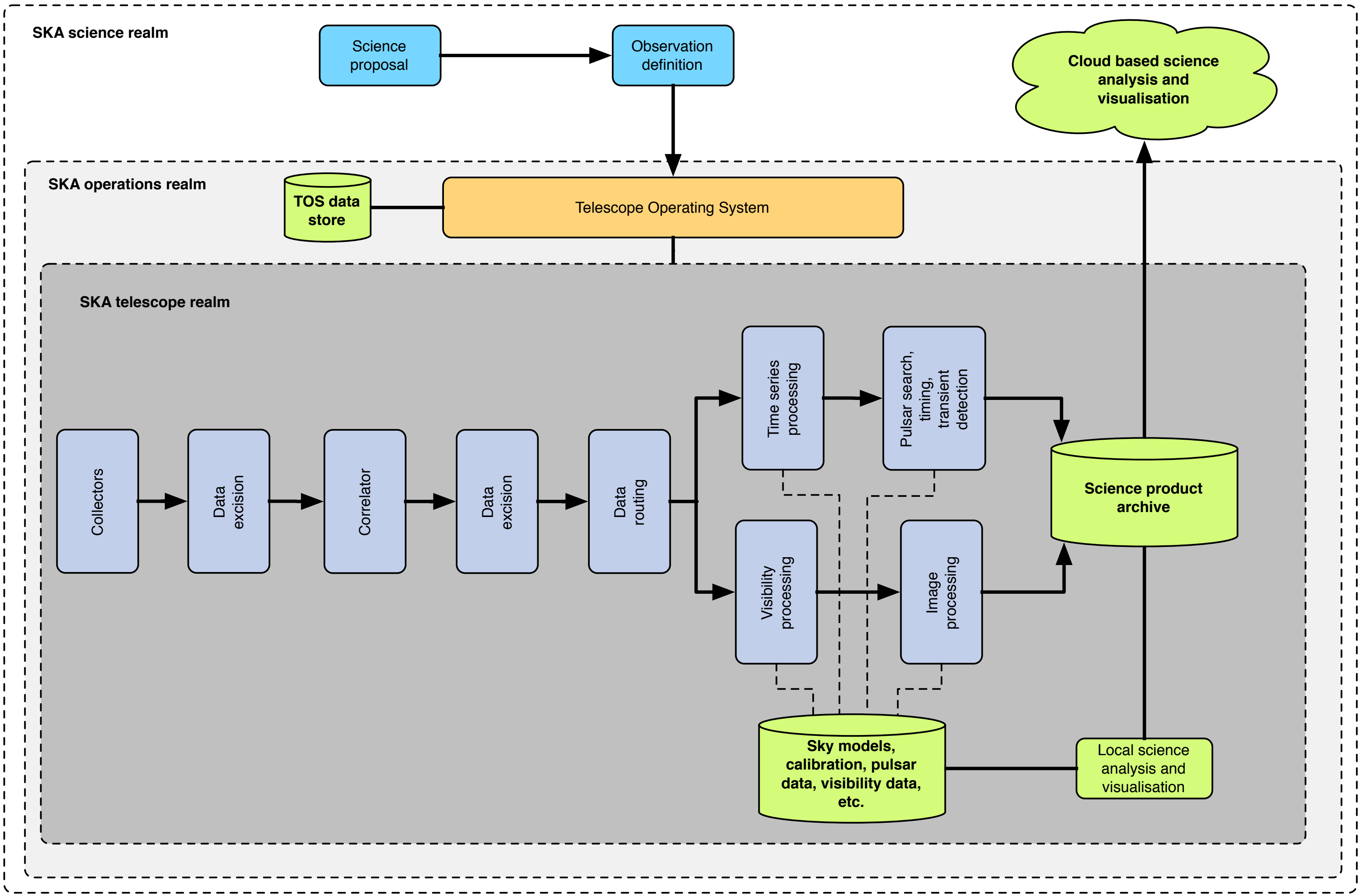
- Observing
- Science data processing and reprocessing
- Data validation

→

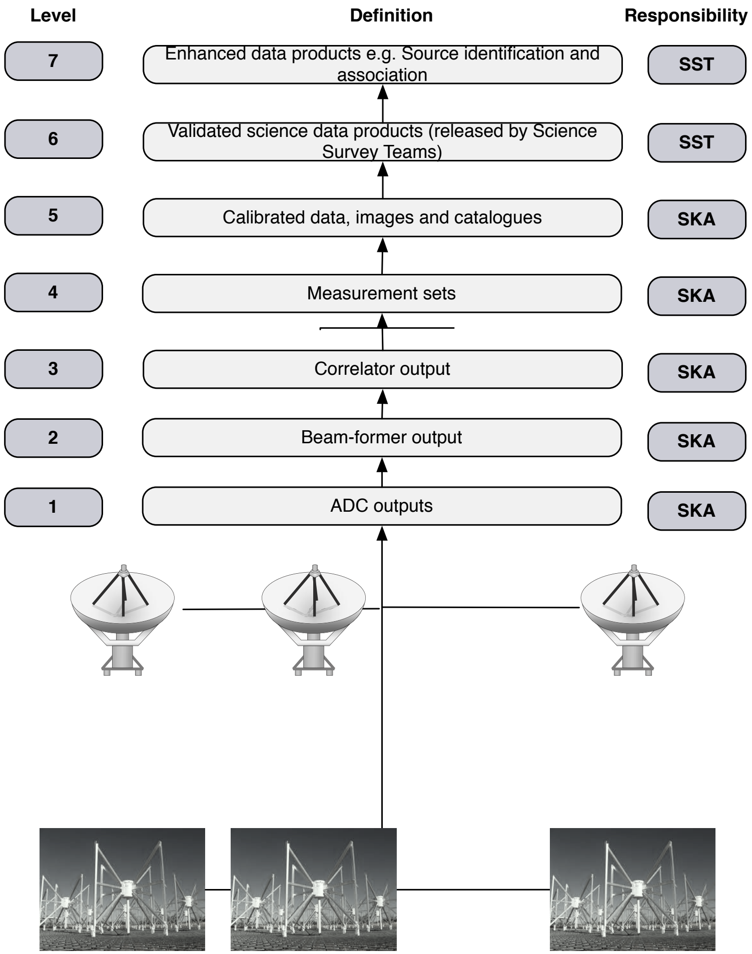
Science Archive

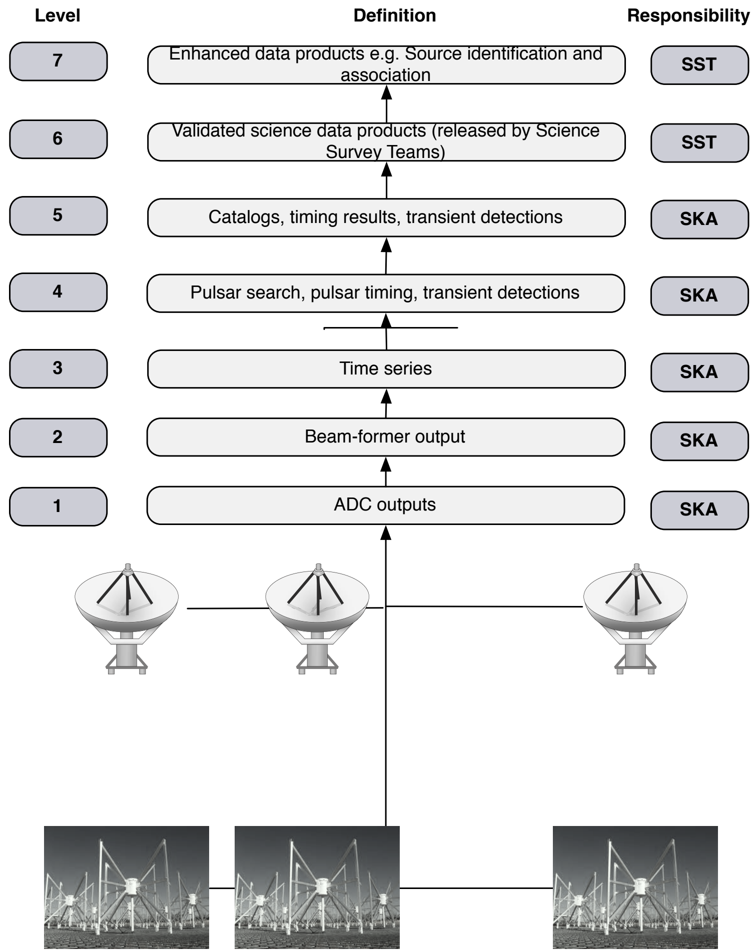
- Archive of observations
- Access to data products

Science data flow



Data level definitions





Outline

- Context
- System architecture

Overview of architecture

- An example of an architecture that could grow into SKA architecture
- Developed by ASKAP leads (Humphreys and Guzman) and so looks like ASKAP architecture
- Other possible architectures will be evaluated during Stage 1 of PEP

Key driving assumptions

- Automated, real-time data processing
- Service observing (mostly)
- All required information flows from M&C seamlessly
- Data rich environment
- HPC essential
- Multiple types of data products
- 2+ Tier data archive hierarchy

Comments on other architectures

- ASKAP and LOFAR
  - Similar in design to each other
  - Closest in scale and requirements to SKA Phase 1
- MeerKAT (KAT-7)
  - Architected for full system simulation

Comments on other architectures

- ALMA
  - Archive centric architecture
  - Common software layer
  - Multiple (regional) data access centres
- LSST
  - Processing and archive is a unified system, the Data Management System (DMS)
  - Impressive data/process model

Reference S&C architecture

- Straightforward SOA-like approach
  - Telescope Operating System
  - Common Processor
  - Science Data Archive
- Enterprise Service Bus
  - Except for visibility and time series data

Data model

- Each subsystem has own, distinct data persistence needs
  - Alternate (e.g. ALMA): one persistent store
- Multiple data formats
  - MeasurementSets, FITS, Catalogs
  - Translation to VO required
  - Alternate (e.g. LSST): data models translate easily or trivially to VO

Architectural goals

- System decomposition
- Loose coupling
- Reliability and Robustness
- Support technology evolution
- Promote software consistency

Loose coupling

- Hardware platform independence
- Language independence
- OS independence
- Implementation independence
- Location and server transparency
- Asynchronous communication

- Loose coupling not appropriate for large data

Reliability and robustness

- SKA will be remotely operated
  - Must be reliable and robust
- Have the ability to operate in degraded mode where appropriate
  - Identification of single points of failure. Relevant to location services, alarm management, logging and monitoring components.
- Support the deployment of redundant / high–availability services where possible.
  - Services must be stateless or idempotent

Support technology evolution

- Technologies very likely to change during 10 year development
- Must support evolution of hardware and software components

Promote software consistency

- Development of SKA software is likely to be distributed over multiple teams around the world
- Unless carefully managed, this may lead to fragmented software development processes, design and implementation.
- The architecture must aim to control this fragmentation and promote consistency.

System description

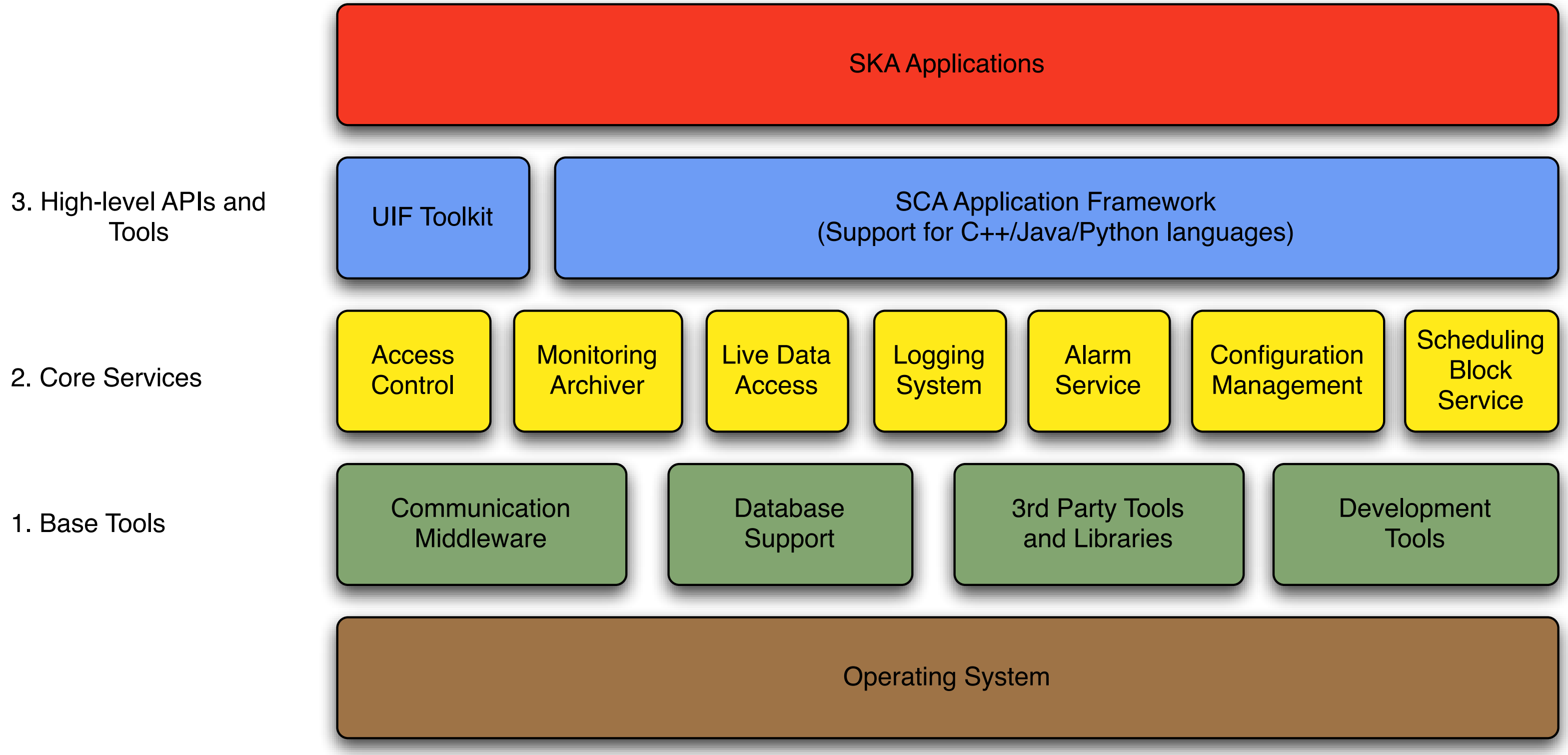
- Telescope Operating System
  - Observation preparation, scheduling and execution (data acquisition);
  - Operator displays, startup, shutdown and maintenance software.
  - Data acquisition coordination
    - antenna motors, beamformers, correlator, timing synchronisation, etc. levels 1 – 5 data products

System description

- Common Processor
  - on–line or near real–time data processing, producing level 5 data products
- Science Data Archive
  - archiving data products coming from the common processor (level 5 data products), pushing data products to regional centres and some user access for data analysis.

SKA Common Software

- Provides software infrastructure for all three components



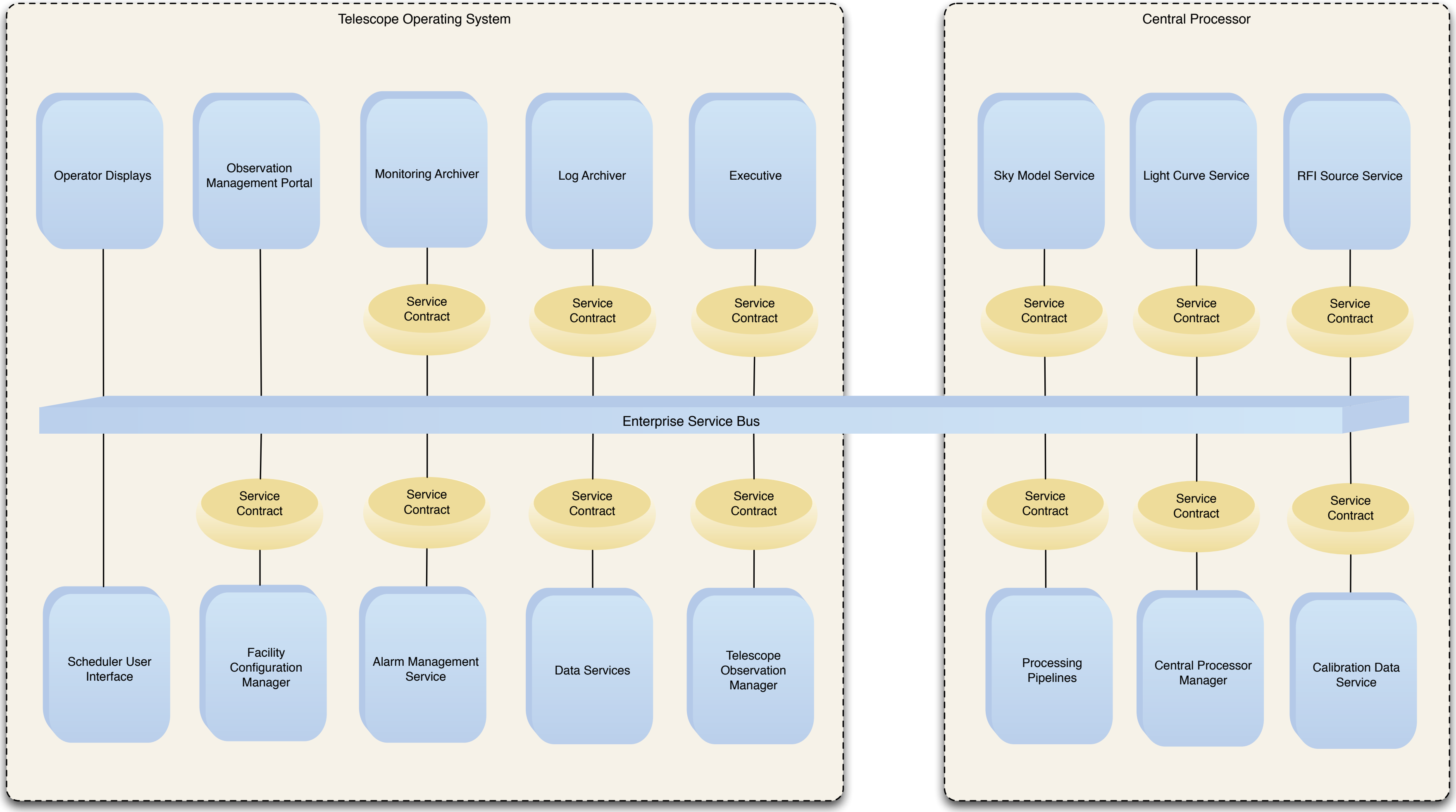
Benefits of Common Software

- Minimises the problem of integration hell by providing a common way to implement interface between components and common services
- Minimises the support burden required by the maintenance staff that would otherwise have to absorb development differences from the very distributed and heterogeneous SKA development teams.
- Minimises reinventing the wheel by letting application (business domain) developers focusing on domain−specific application code rather than technical infrastructure code, also referred sometimes as plumbing code.
- Allows an organised way of sharing and reusing design concepts and patterns.
- Promotes a cohesive set of development practices, standards and tools across different development teams

Capabilities of SCS

- Observation or experiment data model definition and management
- Logging and monitoring data collection and archiving
- Alarm Management
- Component communication infrastructure
- Component and common configuration. For example, station composition, station/element location, etc
- User Interface (UI) framework
- Authentication and access control
- Software development tools: build system, unit test frameworks, deployment system, etc.

TOS and CP subcomponents



Exploring the Universe with the world's largest radio telescope

Common Processor

- Responsible for transforming data from correlator or direct from beamformers into science products
- Specific functionality for imaging
  - Conditioning of the data to remove known errors such as RFI,
  - Calibration of observed visibilities;
  - Processing of calibrated visibilities into images/cubes;
  - Source finding/extraction within the produced images;
  - Detection of transient sources and production of light curves;
  - Quality evaluation of all data products;
  - Provenance tracking of all data products as they are produced and processed.
- Also time series processing, software correlator

Common Processor

- Both a hardware and software subsystem,
- High−performance computer and a suite of calibration, imaging and analysis software.
- Highly optimised and parallel software, designed specifically to support processing high−rate data streams in quasi real−time.
- The high data rate, and likely inability to buffer (store) the visibilities for more than perhaps 24 hours, requires processing occur at the same rate as data acquisition.

Visibility data buffer

- What to do with that huge data flow?
  - 3.3 TB/s input
- Two options
  - Buffer
  - Stream

Buffered

- Store data to non−volatile storage medium for part or the whole of an observation
- Benefits
  - Enables multi−pass algorithms, which iterate over the dataset multiple times;
  - Simplifies failure handling, allowing handling of most common processor hardware failures without data loss;
  - Enables optimisation of the ordering of processing
  - Allows potentially smaller memory (RAM) footprint given there is no need for all spectral channels to be processed simultaneously.

Feasibility of buffering

- For single write and single read, required a storage system capable of sustaining simultaneous 3.3TB/s write and 3.3TB/s read would be required.
- Current record is from Jaguar supercomputer at Oak Ridge National Laboratory
  - Demonstrated bandwidth of 240GB/s (3.5% SKA1 requirement).
- SSD's? Phase change memory

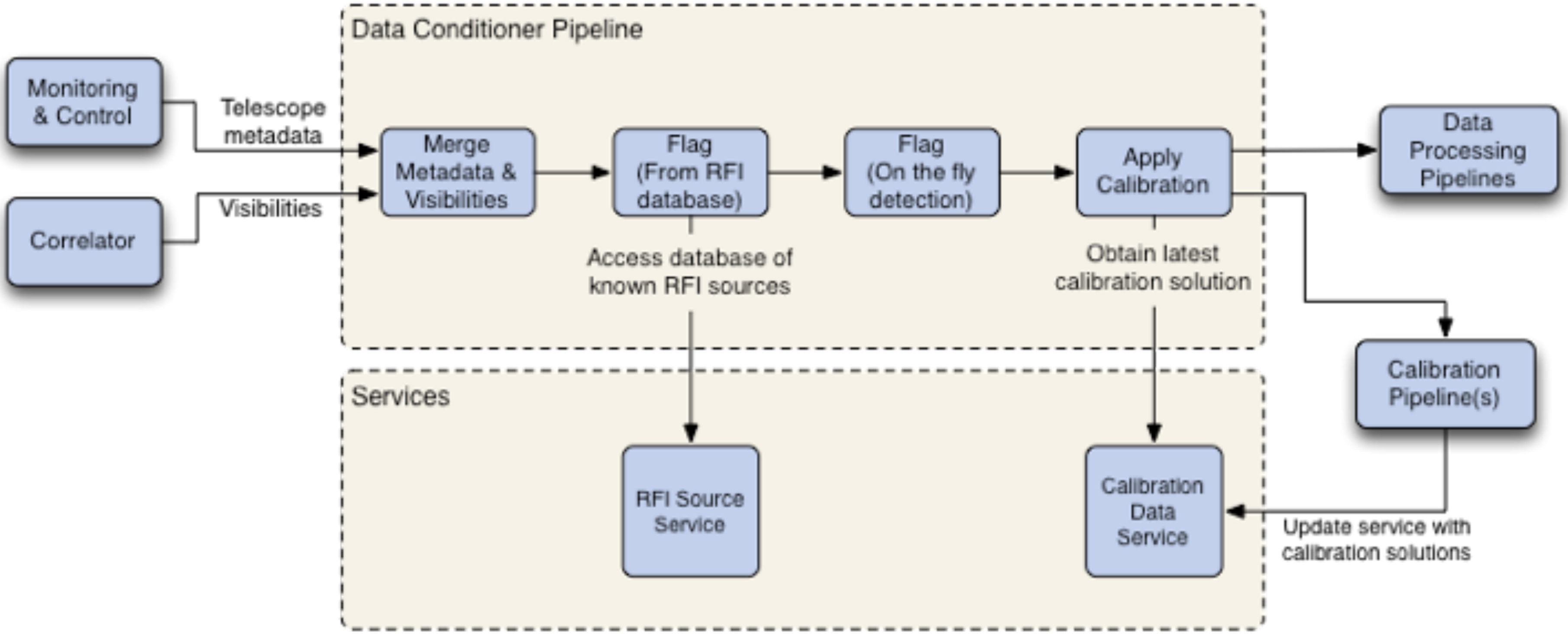
Streamed

- Correlator integrations stream through pipelines
- Eliminates costly buffer
- Drawbacks
  - No support for algorithms which require iteration over the dataset;
  - Node failures will usually result in the loss of data;
  - Few data ordering optimisations possible;
  - Jitter (variation) in processing time may result in data loss;
  - Potentially larger memory (RAM) footprint.

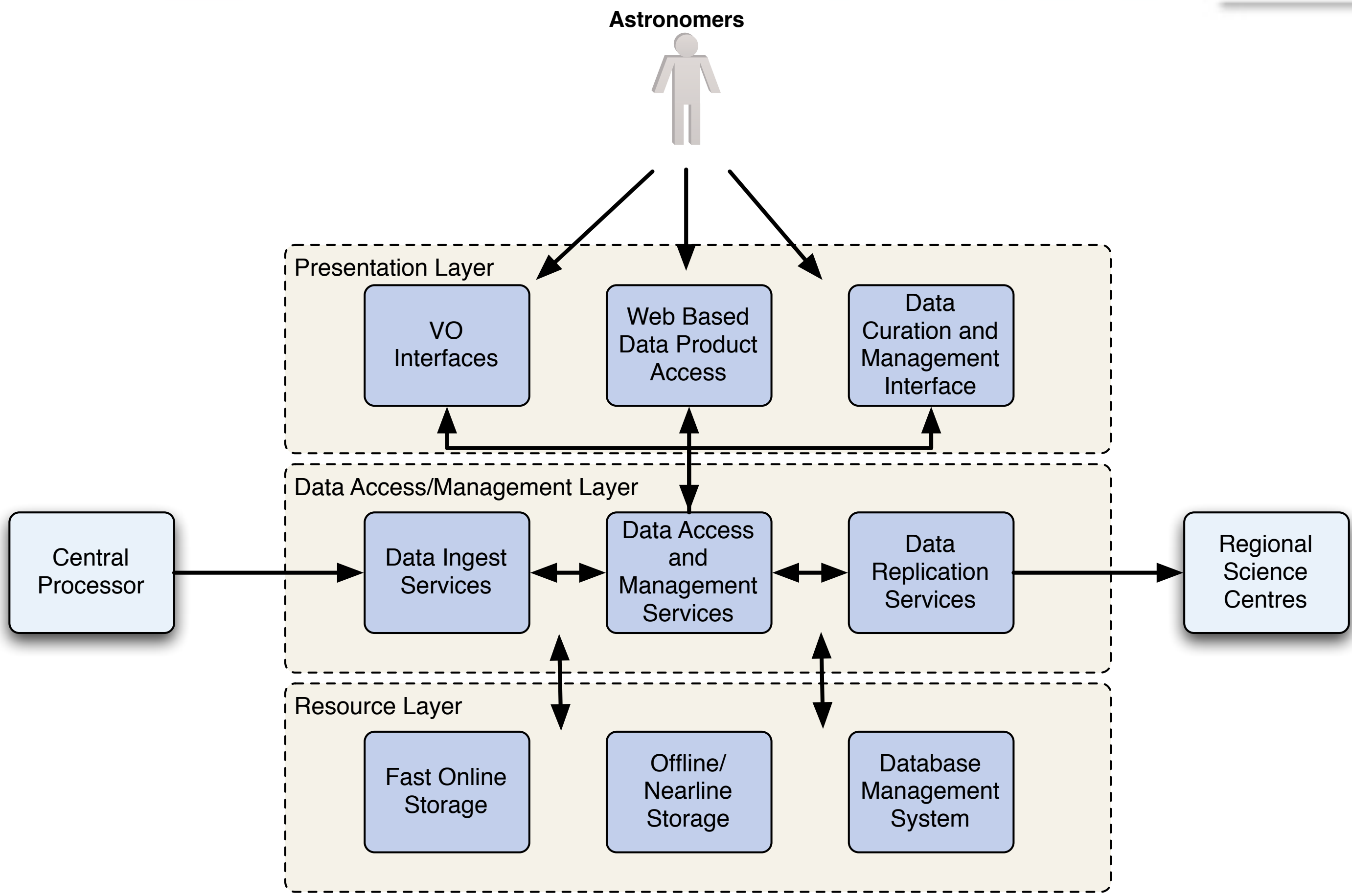
Pipelines

- Streaming: acyclic (one−pass) processing
- Buffering: cyclic (multi− pass) processing
- Each pipeline consists of
  - one or more data sources
  - one or more processing stages
  - one or more data sinks.
- System pipelines
  - data conditioner
  - calibration pipelines.
- One or more science oriented processing pipelines
  - imaging
  - source finding and fitting

Data conditioner pipeline



Science Data Archive



Summary

- Reference architecture exists
- Based on ASKAP architecture
- Conservative SOA-like approach
  - Data do not flow across ESB
- Many questions remain
  - e.g. scaling to large number of antennas
- Will explore other options as part of PEP Stage 1