Computing Challenges for the Square Kilometre Array

Mathai Joseph & Harrick Vin
Tata Research Development & Design Centre
Pune, India

CHEP Mumbai 16 February 2006

Software for the SKA

There are different ways to view the complex SKA instrument and its software complement:

View 1

SKA hardware + control software + data management software + analysis software

View 2

A software 'telescope' that is implemented *by the SKA hardware*

Taking *View 2*

SKA software is like other complex software systems built today for many different applications.

SKA Software

Three dimensions of SKA software

- Instrument management
- Data indexing, archival, retrieval and dissemination
- Data and signal processing – astronomy calculations

The last dimension is of greatest interest to radio astronomers.

However, today they are handling all three dimensions.

SKA Software

Three dimensions of SKA software

- Instrument management
- Data indexing, archival, retrieval and dissemination
- Data and signal processing – astronomy calculations

Main problem: Scale and system complexity

- Large number of antennas (heterogeneous)
- Complex control and data communication networks
- Large volume of data storage
- Diverse and evolving usage models

Together, these pose a significant software challenge!

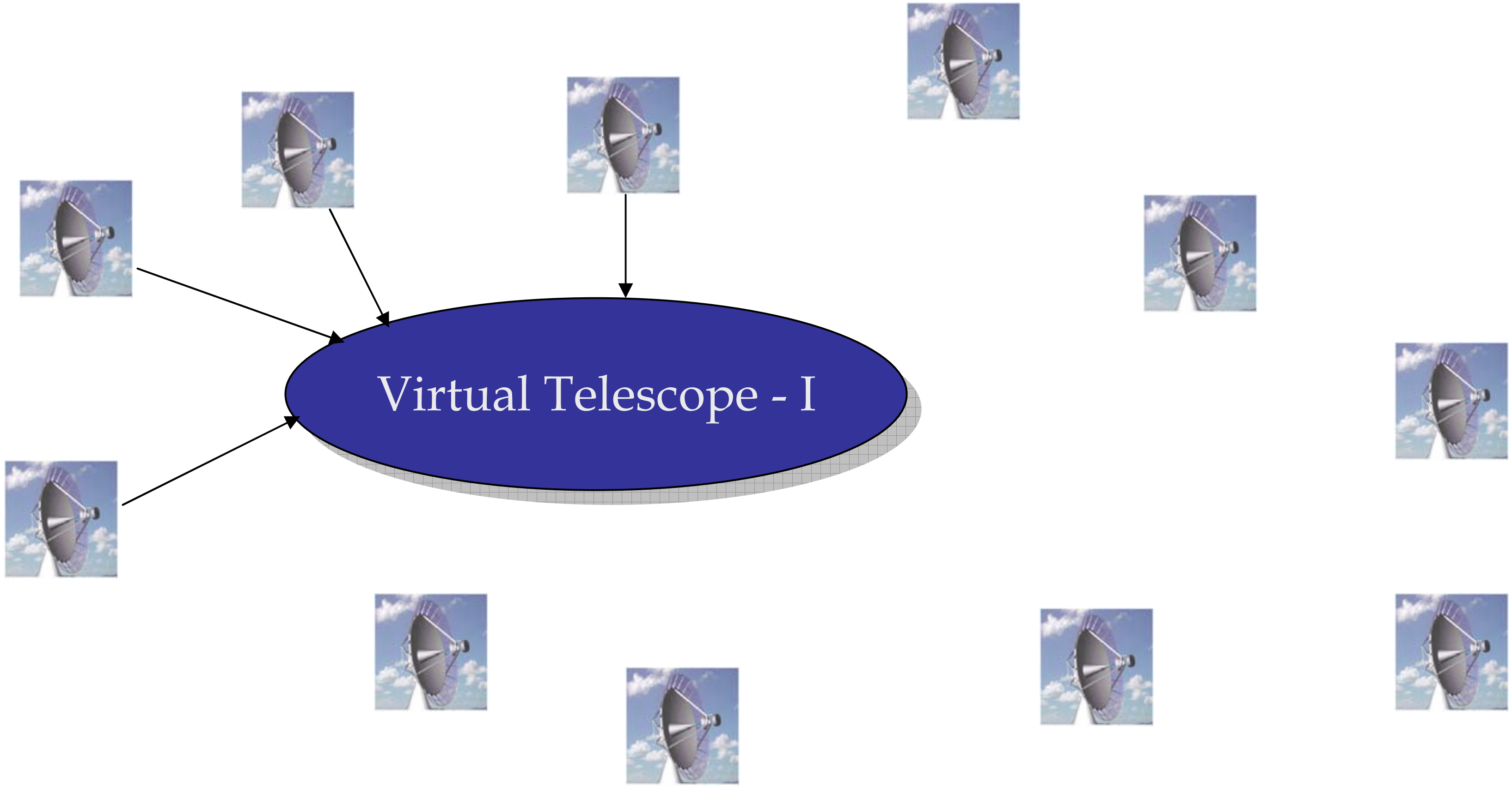
SKA Software & the Software Process

Like other complex software systems, the SKA software will evolve with time.

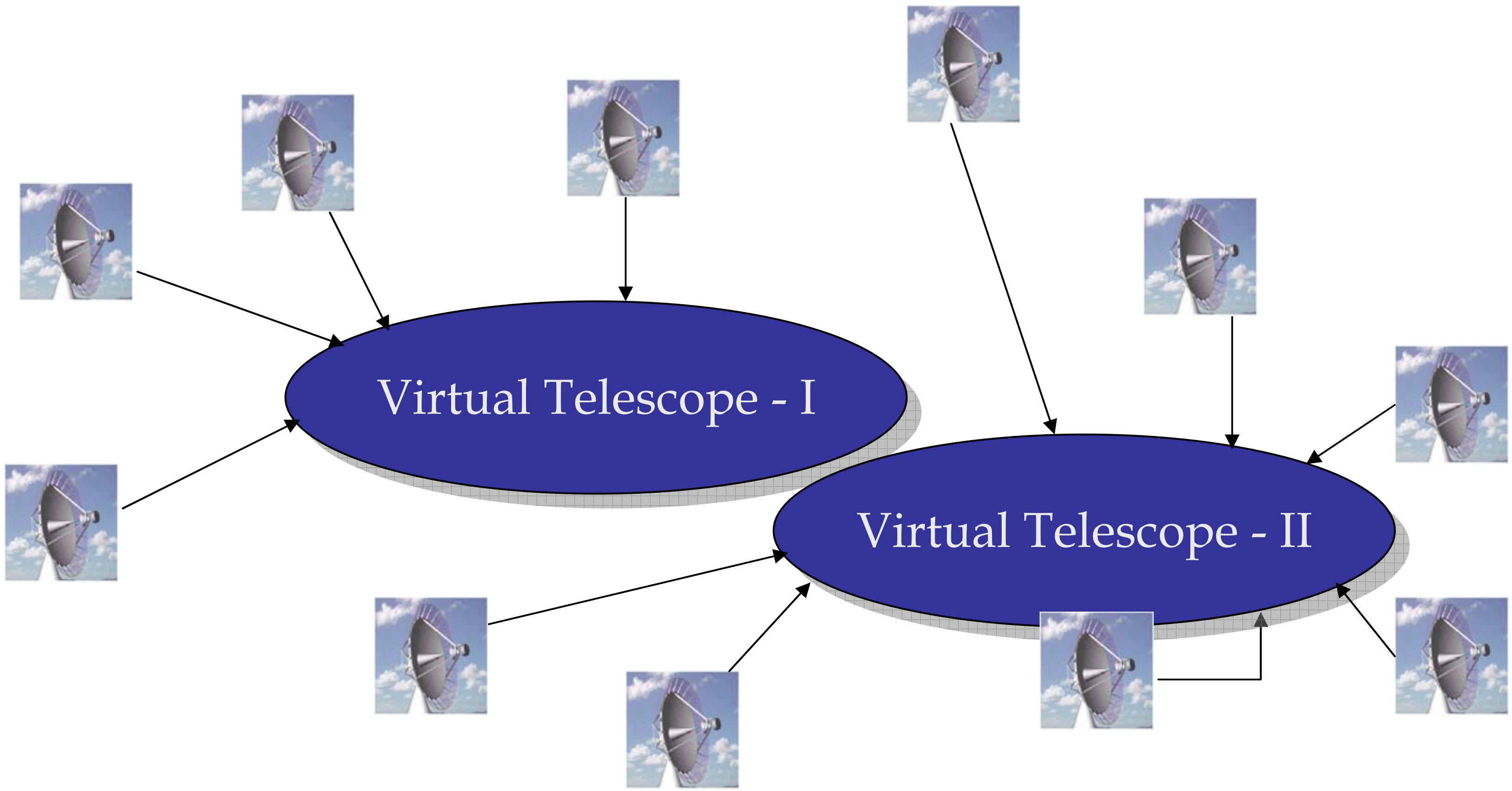
- Initial design and implementation,
- Growth in functionality and performance,
- Version management, &
- Backup, recovery and observational continuity.

Software maintenance can account for 50—75% of the total effort and cost of the software system.

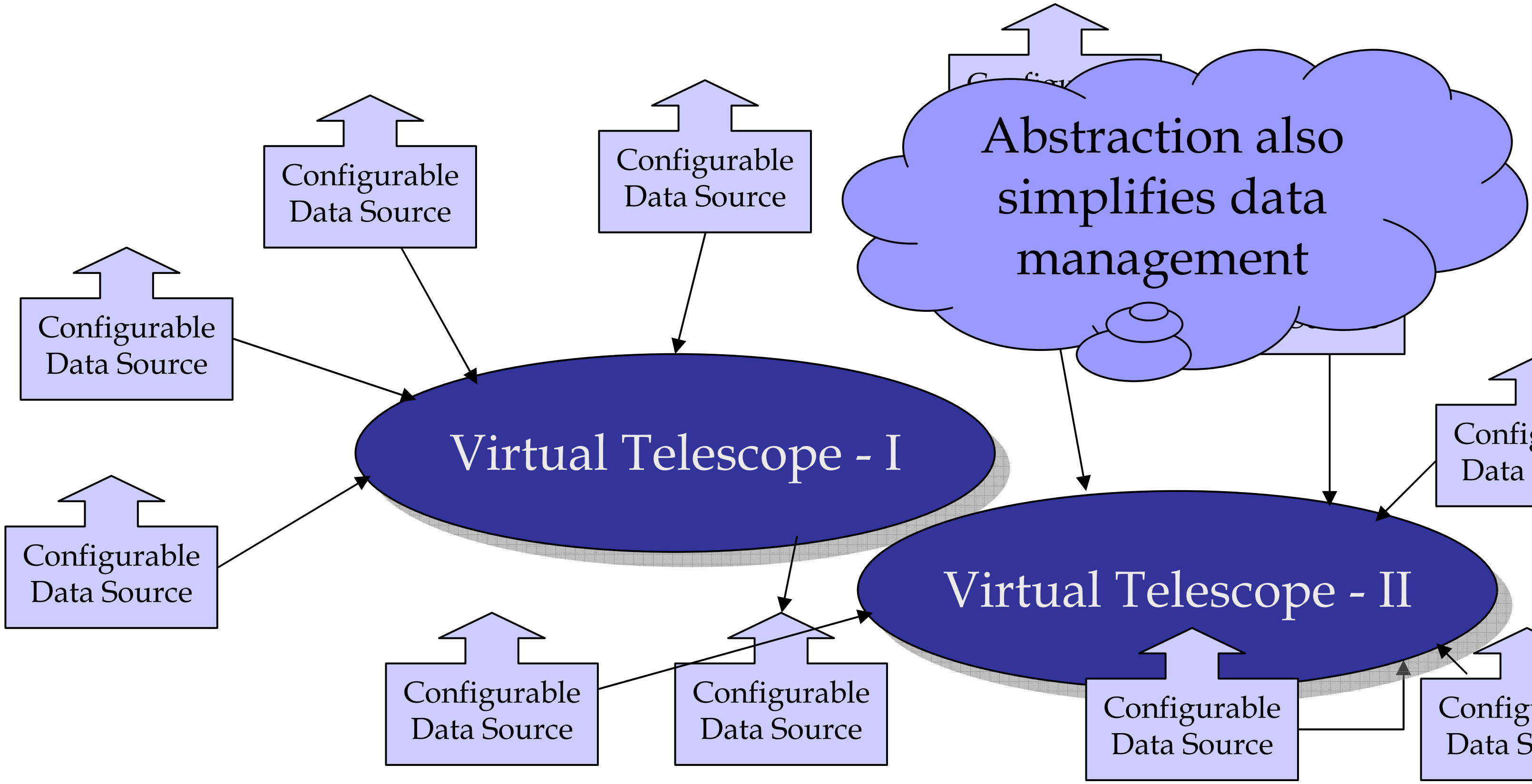
Manage Complexity Through Abstraction



Manage Complexity Through Abstraction



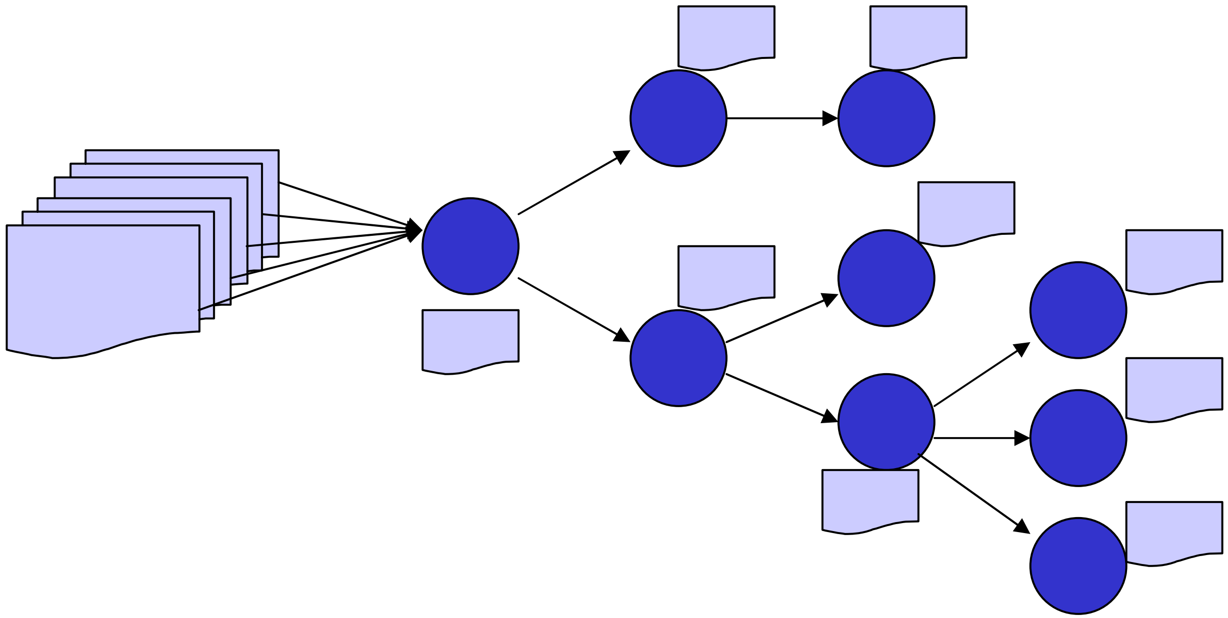
Manage Complexity Through Abstraction



Abstraction, virtualization, and hierarchical composition
➔ hide complexity, simplify management and upgrade

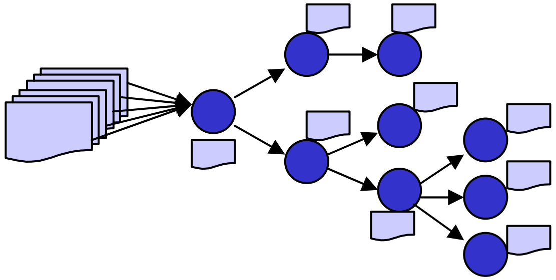
Virtual Telescope: A Usage Scenario (I)

- User sets up an experiment using a virtual telescope console:
  - Describe telescope configuration (e.g., sky region, time, …)
  - Describe data processing requirements
- Experiment scheduled and configured automatically
- Data streams collected, processed, stored
- Data quality monitored automatically; resources re-configured as needed without manual intervention
  - Control engineers informed of any abnormality
- Data post-processing done; data and results published

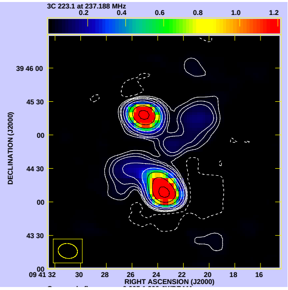


Virtual Telescope: Usage Scenario (III)

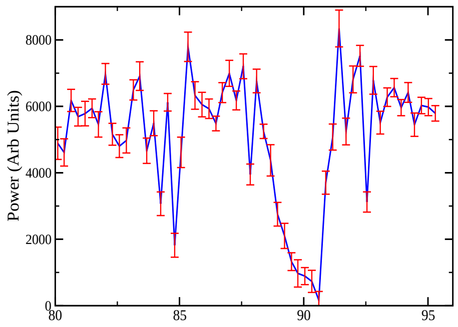
Support multiple consoles for different user communities



Astronomer's Console
Set-up experiment, monitor, visualize, process data; store/analyze results

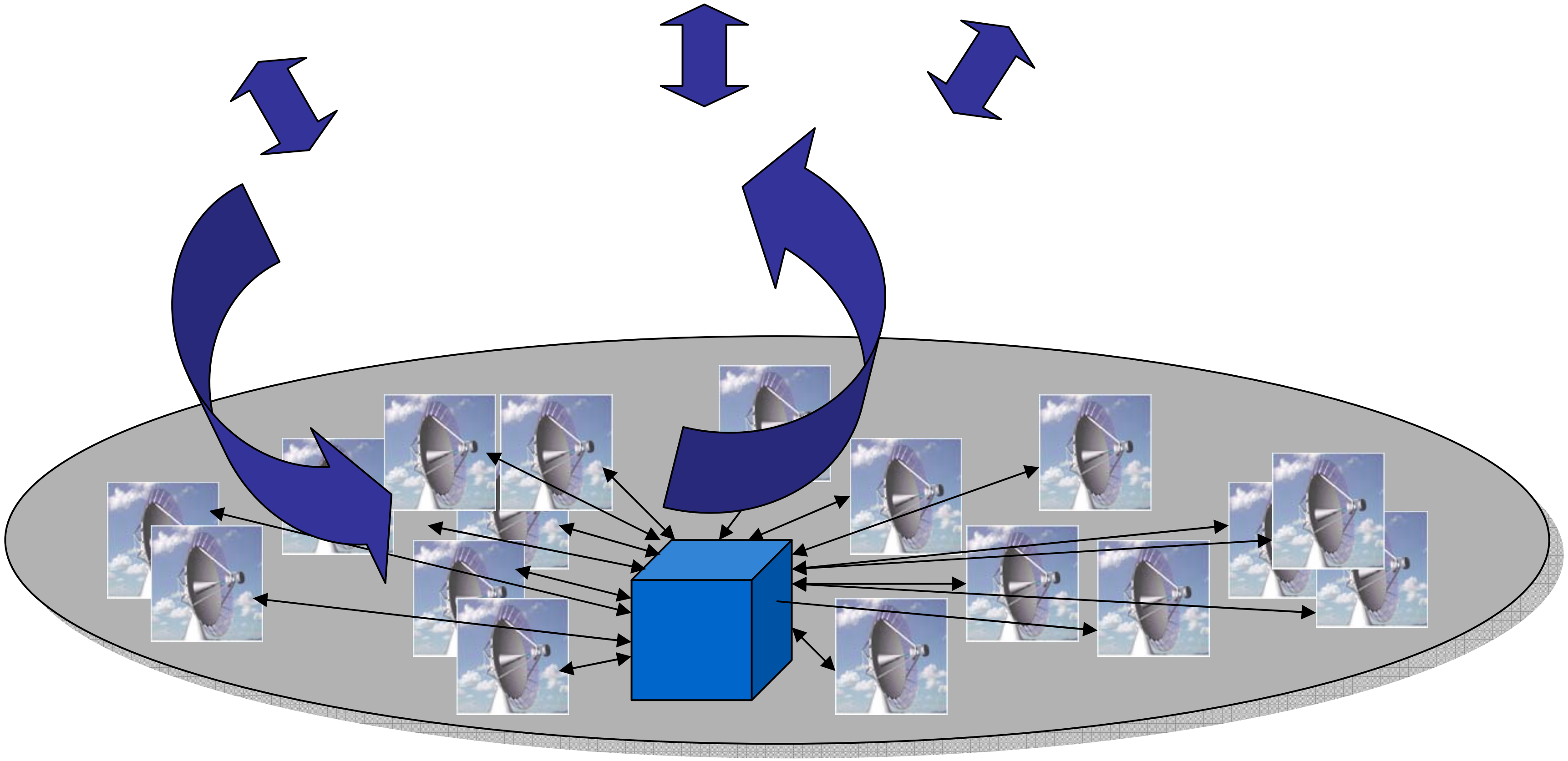


E-Museum Console
SKA tour, images/documents, history, search



Control Engineer's Console
configuration, monitoring, fault diagnosis, reliability

Data Administrator's Console
Archival, retrieval, disaster recovery



Virtual Telescope: Goals

- Usability and configurability:
  - Support for multiple simultaneous experiments
- Automation:
  - Scheduling of experiments and instrument configuration
- Availability:
  - Fault detection, isolation and recovery
  - Disruption-free maintenance and upgrade
- Data management:
  - Storage, retrieval and search of data
  - Multiple views and data models
- Extensibility:
  - Incorporate emerging technology and new usage models

Instrument Management: CS Challenges

- Analogy
  - Emulab: Emulation facility built from a cluster of servers
  - Usage model
    - Schedule and configure experiments
    - Fault detection, isolation, and recovery
  - Solution methodology
    - *Component virtualization:* server operating system
    - *System virtualization:* cluster operating system
- Design of instrument management system:
  - ‘Operating System for Radio Astronomy Telescopes’
  - Wide variety of instrument components
    - Antenna array, control systems, computational cluster, network & storage
  - Define a hierarchy of abstractions

The Data Management Problem

- Once published, data must remain available
  - Data may be used long after data collection
  - Allow others to reproduce work and/or do new science
- Goals:
  - Easy to search, access and process data
  - Ease of adding data and findings to corpus
- Challenge: Data avalanche
  - Vast volume of data: Multi-petabyte storage
    - Raw data, processed and filtered data, imagery, …
  - Lack of data models and descriptors for publication
  - Capturing data lineage is laborious

Data Management: CS Challenge

- Data models and descriptors
  - When, where, how was data captured?
    - Experimental configuration
  - Data lineage: how was the data processed/produced?
    - Association of raw, processed and image data
    - Description of data processing applications
  - Observations and results derived from datasets
  - …
- System: Federated data repository + search portal
  - Data visualization – capture queries and present results
  - Data analysis – data mining, machine learning, etc.
  - Distributed software systems – data and systems management

Is the SKA Software Unique?

The SKA ***application*** is challenging & unique. But many problems are similar to those seen elsewhere, e.g.:

- *Service-oriented transactional application*
- *Data storage integrity*
- *Redundancy and continuity*
- *Technology independent applications*

There are also many problems that need to be studied.

Success will need collaboration from radio astronomers and computer scientists.

Technology Independence

Large applications can be built to be independent of the hardware infrastructure:

- Platform independent
- Database, middleware independent
- Language independent

Similar abstractions can be used to allow for evolution of antennas.

Model-based software development standards are emerging -- can provide a framework for the SKA software development.

Conclusions

Constructing the SKA software will be a challenging task

- Many of the problems have solutions
- Others need investigation.

Important to use a mature software process

- To permit growth and evolution.

There should be some pilot development projects

- to evaluate performance, scalability.

Need to separate the software system architecture from the astronomy-specific analysis software.

Thank you.