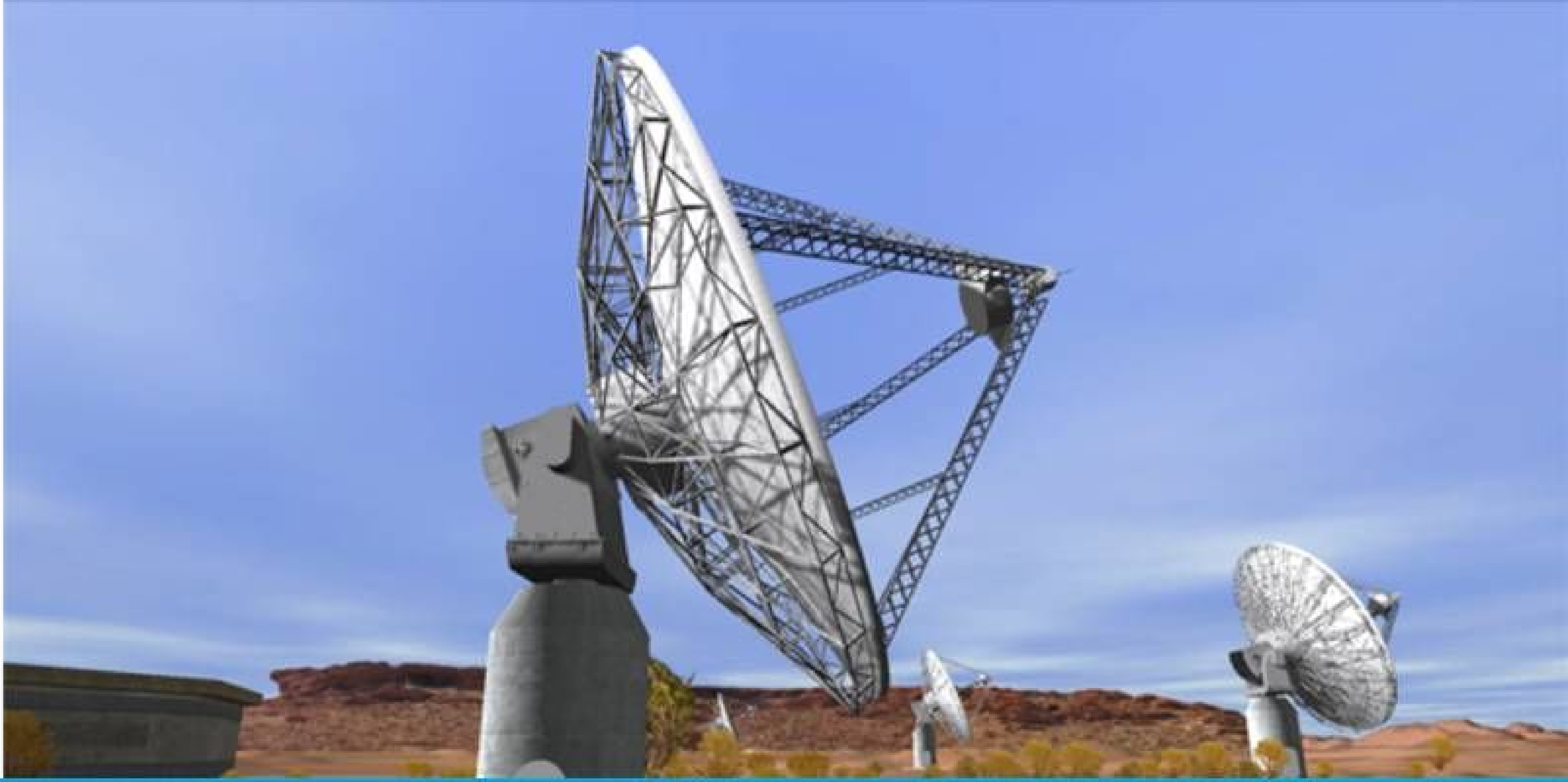


Preliminary Design of the Australian SKA Pathfinder (ASKAP) Telescope Control System

Juan Carlos Guzman

ASKAP Computing IPT – Software Engineer

ASKAP Monitoring and Control Task Force Lead

ICALEPCS09 – Kobe, Japan – 13th October 2009

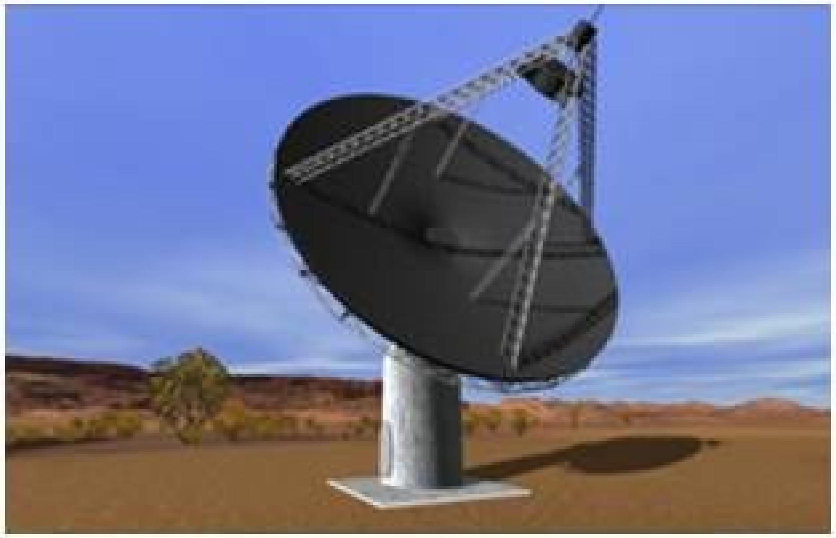
CSIRO

Outline

* Overview of the ASKAP Project
* ASKAP Software Architecture: EPICS and ICE
* Current and Future Developments

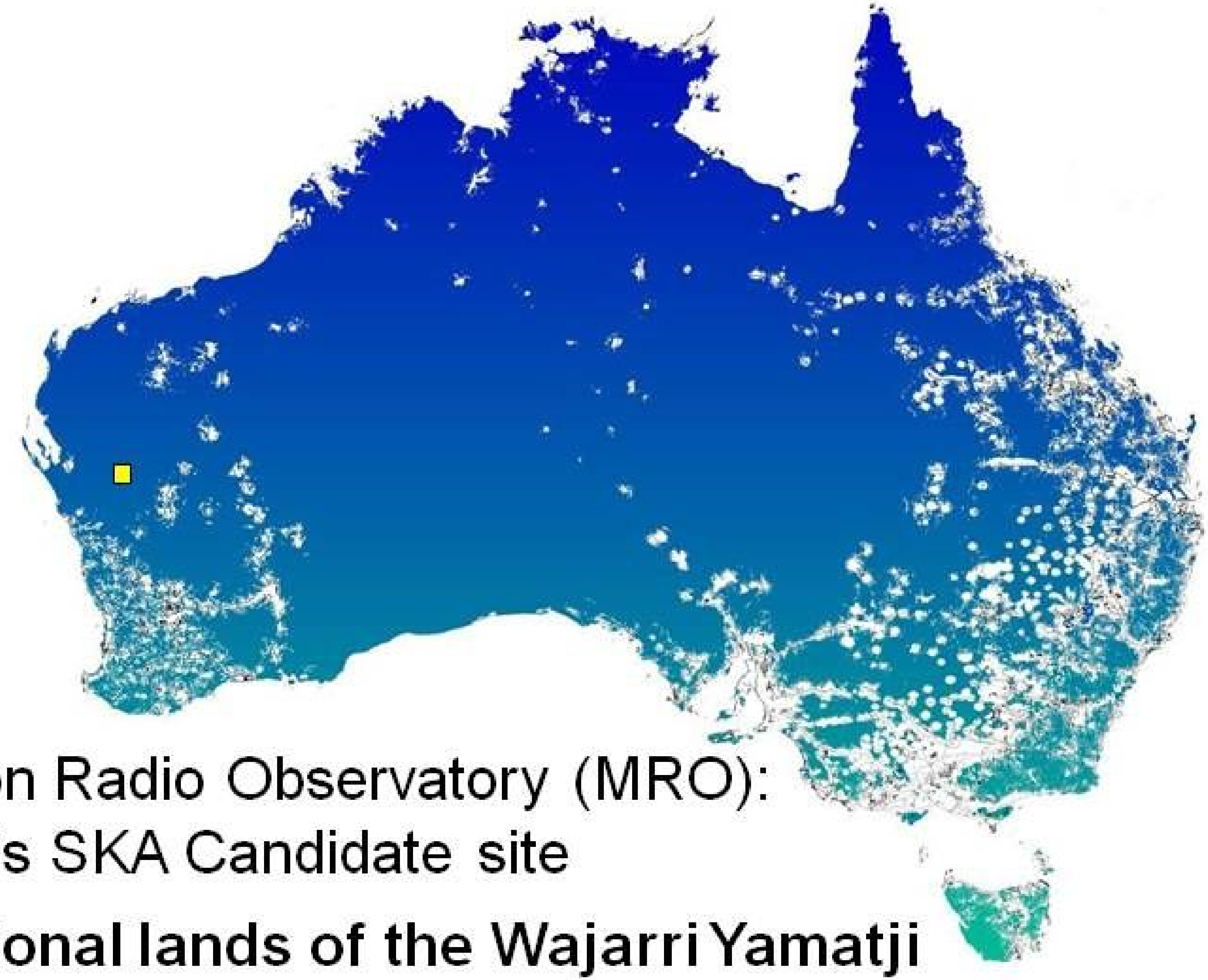
Australian SKA Pathfinder Project = 1% SKA

- Wide field of view telescope (30 sq degrees)
  - Sited at Boolardy, Western Australia
  - Observes between 0.7 and 1.8 GHz
  - 36 antennas, 12m diameter, 3-axis sky-mount
  - 30 beam phased array feed on each antenna
  - Mainly a survey telescope, but it has to support also targeted observations
- Scientific capabilities
  - Survey HI emission from 1.7 million galaxies up z ~ 0.3
  - Deep continuum survey of entire sky
  - Polarimetry over entire sky
- Technical pathfinder
  - Demonstration of WA as SKA site
  - Phased Array Feeds (PAF)
  - Computing



CSIRO

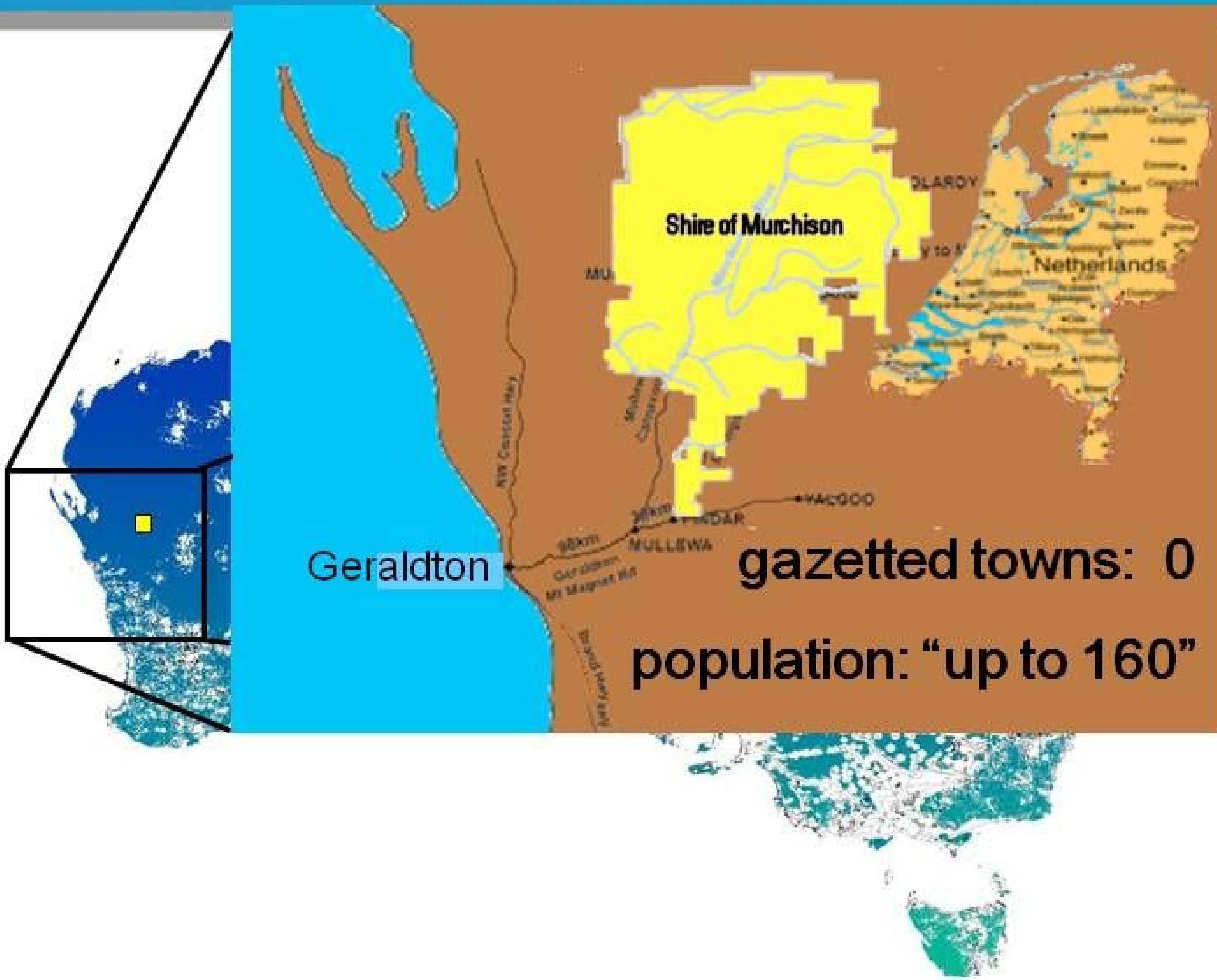
ASKAP Site



Murchison Radio Observatory (MRO):
Australia's SKA Candidate site

Traditional lands of the Wajarri Yamatji

ASKAP Site



Shire of Murchison
Netherlands
Geraldton
YALGOO
MULLEWA
gazetted towns: 0
population: “up to 160”

ASKAP Site (current office)

ASKAP Site (current staff)

ASKAP Computing Challenges

- Remote operations of telescope
  - Technical operations/maintenance from Geraldton
  - Science operations from Sydney
  - Science archive in Perth
- Novel active element in data flow - Phased Array Feeds (PAF)
  - Parallel mosaicing!
  - Performance of PAFs unknown
- Very large data flow
  - ~ 30 VLAs running in parallel (~ 10TB/hr of visibilities)
  - Cannot archive all observed data
- Automated processing vital
  - Parallel and distributed computing necessary
- Science analysis on the fly
  - Optimal algorithms running automatically
- Limited computing staff
  - ~ 60 FTE-years planned for mid 2006 to mid 2013

CSIRO

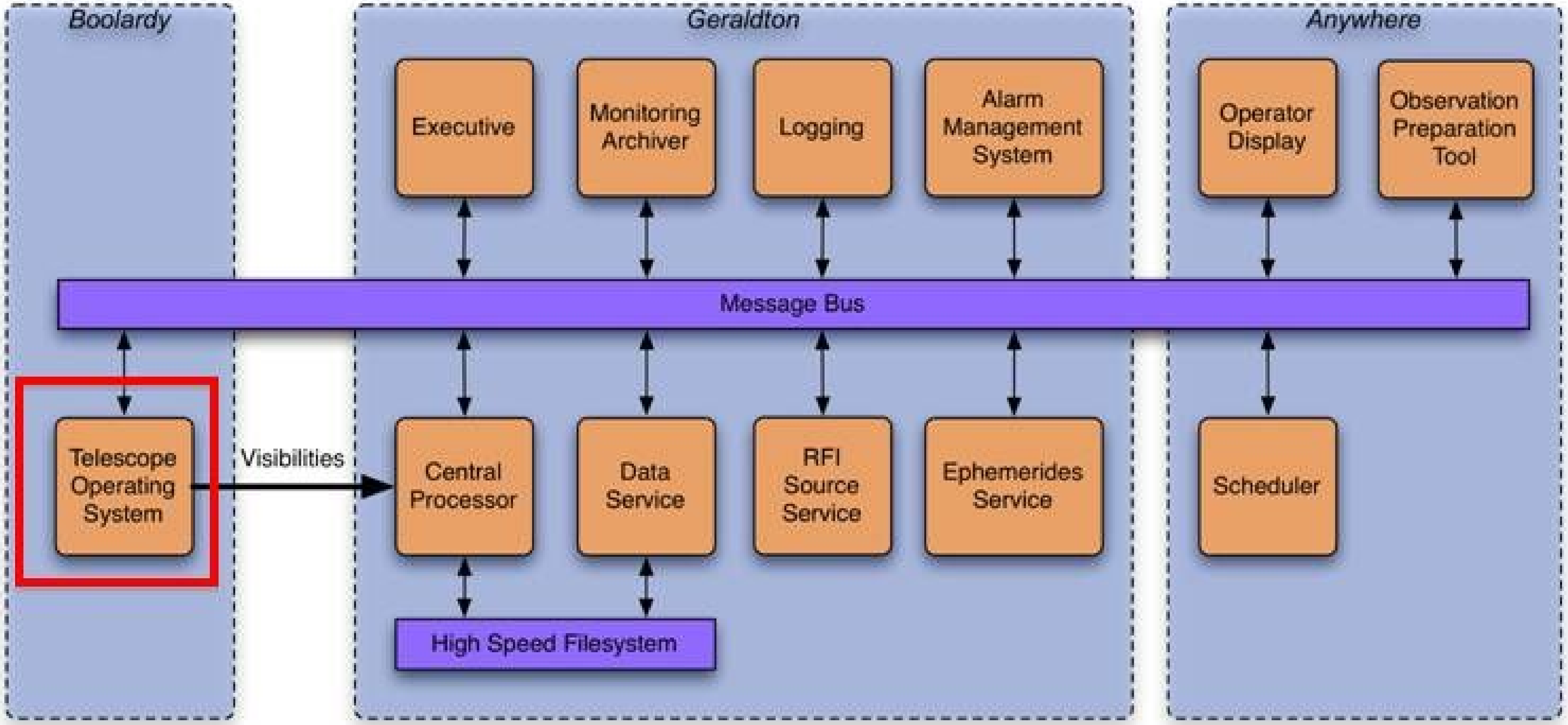
ASKAP Timeline

- ASKAP Antenna contract awarded to 54th Research Institute of China Electronics Technology Group Corporation (known as CETC54) late 2008
  - Installation of Antenna #1 at MRO starts in late Dec 2009
  - Antenna #36 to be delivered in Dec 2011
- Digital, Analogue and Computing PDRs completed in 2009
- CDRs to be completed in early 2010
- Boolardy Engineering Test Array (BETA)
  - 6 antenna interferometer
  - PAF, Digital and Computing back-end to be installed in the first antenna mid-2010
  - Commissioning starts in 2011
- Full ASKAP commissioning to begin in 2012
- Full ASKAP operational in 2013

Outline

- Overview of the ASKAP Project
- ASKAP Software Architecture: EPICS and ICE
- Current and Future Developments

ASKAP Top-Level Software Architecture



Telescope Operating System (TOS) Responsibilities

- Provides narrow interface to top-level control during execution of observations (Scheduling Blocks)
- Coordinate operations of many widely distributed devices built both in-house and COTS (heterogeneous distributed system)
- Accurate time synchronisation between hardware subsystems
- Deliver visibilities and its meta-data to Central Processor for subsequent processing
- Publishes monitoring data to Monitoring Data Archiver component
  - Number of monitoring points: ~150,000
  - Monitoring data should be archived permanently: ~10TB/year (modest compared to science data)
  - Monitoring data archiver to provide reporting tools
- Handles safety operations with the instrument such as automatic wind stowing, network failures, etc.
- Provides interfaces to hardware/firmware engineers and maintenance staff
- Will be deployed in Boolardy

TOS Technologies: Evaluation of control software frameworks

- Four alternatives selected
  - Being used in other major research projects or facilities
  - Supported on Linux
- One commercial (SCADA)
  - PVSS-II
- Three non-commercial
  - Experimental Physics and Industrial Control System (EPICS)
  - ALMA Common Software (ACS)
  - TANGO

TOS Technologies: Evaluation of control software frameworks – cont.

	PVSS-II	EPICS	ACS	TANGO
Developed by	ETM (Siemens)	ANL, LANL	ESO	ESRF, SOLEIL, ELETTRA, ALBA
Released	1995	1990	2000	2000
Number of projects	~ 100	~ 100	7	6
Platforms	Linux, Windows, Solaris	Windows, Linux, MacOS, Solaris, vxWorks, RTEMS	Linux, vxWorks, RTAI	Windows, Linux, Solaris
Programming Languages	C/C++, CTRL (scripts)	C/C++ (IOC); Java, Python and other (clients)	C++, Java, Python	C++, Java, Python
Middleware	N/A	Channel Access	CORBA	CORBA
Supported I/O devices	~ Hundreds	300	10 (?)	350
Database	Yes (proprietary)	Yes	Yes (XML)	Yes (MySQL)
GUI Toolkit	Yes	Yes	No	Yes
Logging support	Yes	Yes	Yes	Yes
Archiving Tools	Yes	Yes	No	Yes
Alarm Handler	Yes	Yes	Yes	?
Bulk Data Support	No	No	Yes	No
Security	Yes	Yes	No	No (proposed)
Redundancy Support	Yes	No	No	No

CSIRO

TOS Technologies: Evaluation of control software frameworks – cont.

- Developing all technical software infrastructure in-house is costly and risky
- There are several alternatives both commercial and non-commercial that can "do the job"
- PVSS relatively expensive for ASKAP scale and we will tie ourselves to a commercial entity
- ACS (CORBA-based), despite being used in a radio interferometer, is relatively new and too complex for our needs
- TANGO (CORBA-based) is also relatively new and has a small user base mainly within the European synchrotron community
- For ASKAP we have chosen EPICS as the software framework for the monitoring and control system
- Evaluation report (ASKAP-SW-0002) version 1.0 released in late December 2008

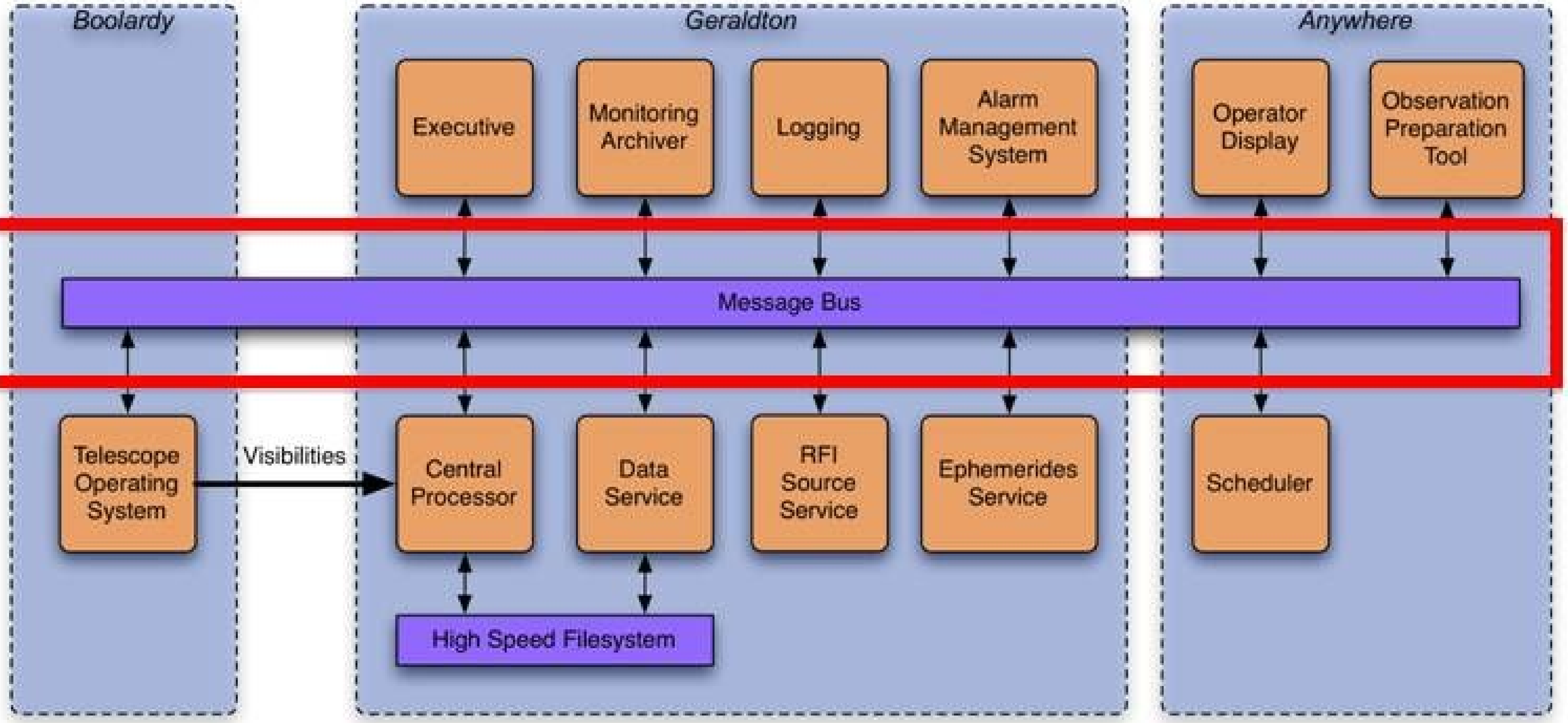
TOS Technologies: Why EPICS?

- It is free, open source and with a very active community
- Both clients and servers can run in many platforms; not only VxWorks!
- Proven technology
- Proven scalability
- All the client needs to know is the PV name. No messing around with fixed addresses
- Lots of software tools available on the web
- Real-time database design appeals also to non-programmers
- Presents a unified interface to high-level control (easier integration)
- Common software for hardware subsystems developers

TOS Technologies: Limitations of EPICS

- Close loop control via Channel Access up to 20 Hz
  - Fast control (hard real-time) will be implemented at lower level
- Channel Access supports limited data types (e.g. no XML)
  - Complex data types are used only at the top-level (ICE middleware)
  - Alternatives exist to extend data types: arrays, aSub record
- Channel Access is optimised for 16k packets (no support for bulk data transfer)
  - The maximum size can be increased, but there is a penalty on performance
  - Bulk data transfer will use a different mechanism to avoid overloading of the monitoring and control network, e.g. point to point UDP (visibilities) or files (beamformer weights)
- No built-in request/response-type of communication
  - Workarounds are available, e.g. CAD/CAR/SIR records or aSub record
  - Real-time control is usually asynchronous (non-blocking)
- Has not being used recently in astronomical projects
  - Misconceptions and "bad press" rather than technical limitations

Technologies for Top-Level Message Bus



Technologies for Top-Level Message Bus: Evaluation of Middlewares

- Requirements
  - Support Linux
  - Language bindings for Java, C++ and Python
  - Support for Request / Response style communication
    - Synchronous
    - Asynchronous
  - Support for Publish / Subscribe style communication
  - Promote loose coupling
  - Support for fault tolerance (replication, etc.)
  - (Desirable) Free license
  - (Desirable) Mature
- Three alternatives evaluated
  - Apache Tuscany
  - ActiveMQ (JMS)
  - Internet Communication Engine (ICE)

Technologies for Top-Level Message Bus: Evaluation of Middlewares

- Apache Tuscany was found to be too immature for our needs
- Both ActiveMQ and ICE were found to meet our needs, but we have selected ICE over ActiveMQ/JMS primarily because of its interface definition language
  - Avoids us having to define our own interface definition language
  - Avoids us having to build our own bindings between the language and the interface definition
  - ICE's interface definition language is reasonably strict, so eliminates ambiguity which is present in other methods of defining such interfaces
- Also many of our interfaces appear to be best suited to an object oriented model

Outline

- Overview of the ASKAP Project
- ASKAP Software Architecture: EPICS and ICE
- Current and Future Developments

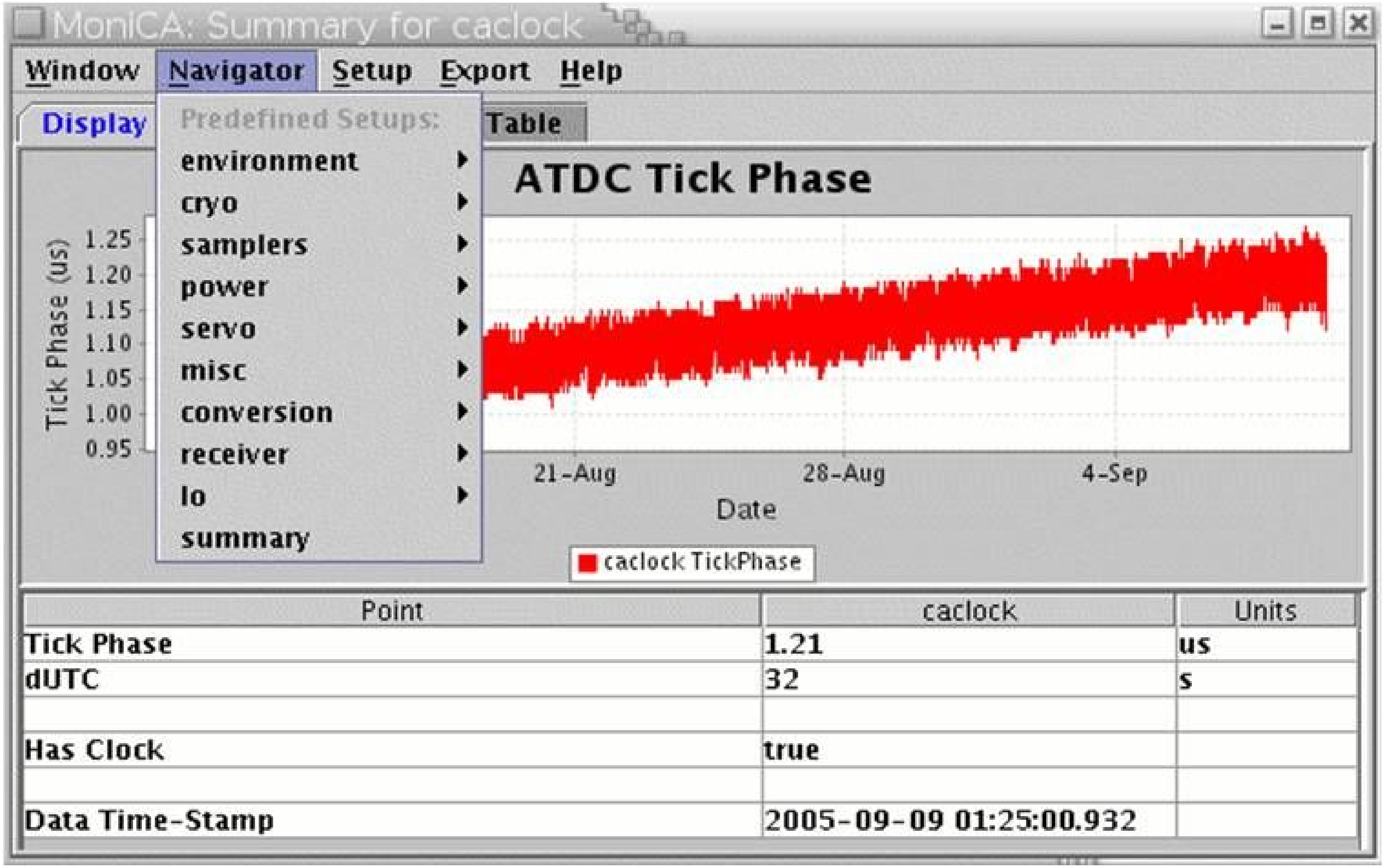
Current Developments

- Parkes Testbed Prototype (12-m single dish with 20-element dual polarization PAF)
  - Operator Display (EDM)
  - Observation scripts in Python (cothread and EpicsCA)
  - 5 Linux IOCs (deployed in one PC)
    - Antenna drives (TCP socket)
    - Digital Back-End (TCP socket)
    - Local Oscillators (vxi11)
    - PAF monitoring (usb2spi, developed in-house)
    - 64m interface (TCP socket)
- MoniCA, our EPICS Channel Access (CA) Archiver
- EPICS device support for SNMP devices (based on devSNMP)
- EPICS device support for in-house USB-to-SPI module
  - Used to monitor PAF (temp, power levels, humidity) and control power switches
  - Uses FTDI USB DLP-2232M module
  - Driver support based on ASYN driver framework
- Top-level
  - ICE common interface for all components
  - Logging Service

MoniCA

- Monitoring system developed in-house over recent years, primarily for Australia Telescope Compact Array (ATCA)
- Used at all existing ATNF observatories
  - Familiar to Operations staff
- Many "data sources" exist and new ones can be added
  - EPICS Data source already implemented and in use at Parkes
- Server and client (GUI) implemented in Java
- Extensive use of inheritance
  - Relatively easy to extend/customise
- Supports aggregate/virtual points derived by combining values from other points
- Archive to different types of databases: single file or relational database (MySQL)
- There is an open-source version available in google code, but not much documentation yet (see http://code.google.com/p/open-monica/)

MoniCA Client



Other Technologies in our Software Infrastructure

- Subversion
- Trac
  - Software project management (tickets, milestones,bug report)
  - Wiki
  - Linked to subversion
- Recursive build (developed in-house)
  - Support for Python, C++ and Java packages as well as EPICS applications
  - Dependency analysis
  - Creates deployable packages (tarball and debian packages)
  - Perform automatic unit and function tests
- Hudson Continuous Integration (http://hudson.dev.java.net)
  - Open source
  - Support for Subversion projects
  - Easy to use and very useful
  - There is even an iPhone app available for free

Future Developments

- ASKAP Computing CDR planned for March 2009
- MoniCA enhancements
  - Support for monitoring ICE components
  - Scalability beyond 20,000 points
  - More documentation
- Develop hardware subsystems interfaces (EPICS IOCs) in conjunction with other teams
- BETA integration starts mid-2010
- Non-critical for BETA -> development deferred after BETA:
  - Facility Configuration System
    - Goal is to centralise "static" configuration of all devices deployed in Boolardy
  - Alarm Management System
    - EPICS Alarm Handler for BETA
    - Evaluate alternatives for ASKAP
- This list only presents developments in Monitoring and Control. There are more developments in area of Data Reduction (HPC, data analysis, data quality, algorithms, etc.)

Australia Telescope National Facility
Juan Carlos Guzman
ASKAP Computing IPT - Software Engineer

Phone: 02 9372 4457
Email: Juan.Guzman@csiro.au
Web: http://www.atnf.csiro.au/projects/askap/

www.csiro.au

Thank you