ALMA Common Software

Basic Track

A walk through ACS functionality

I. Oya, CTAO gGmbH

Topics covered

- ACS services
- ACS generic GUIs and clients

ACS Services

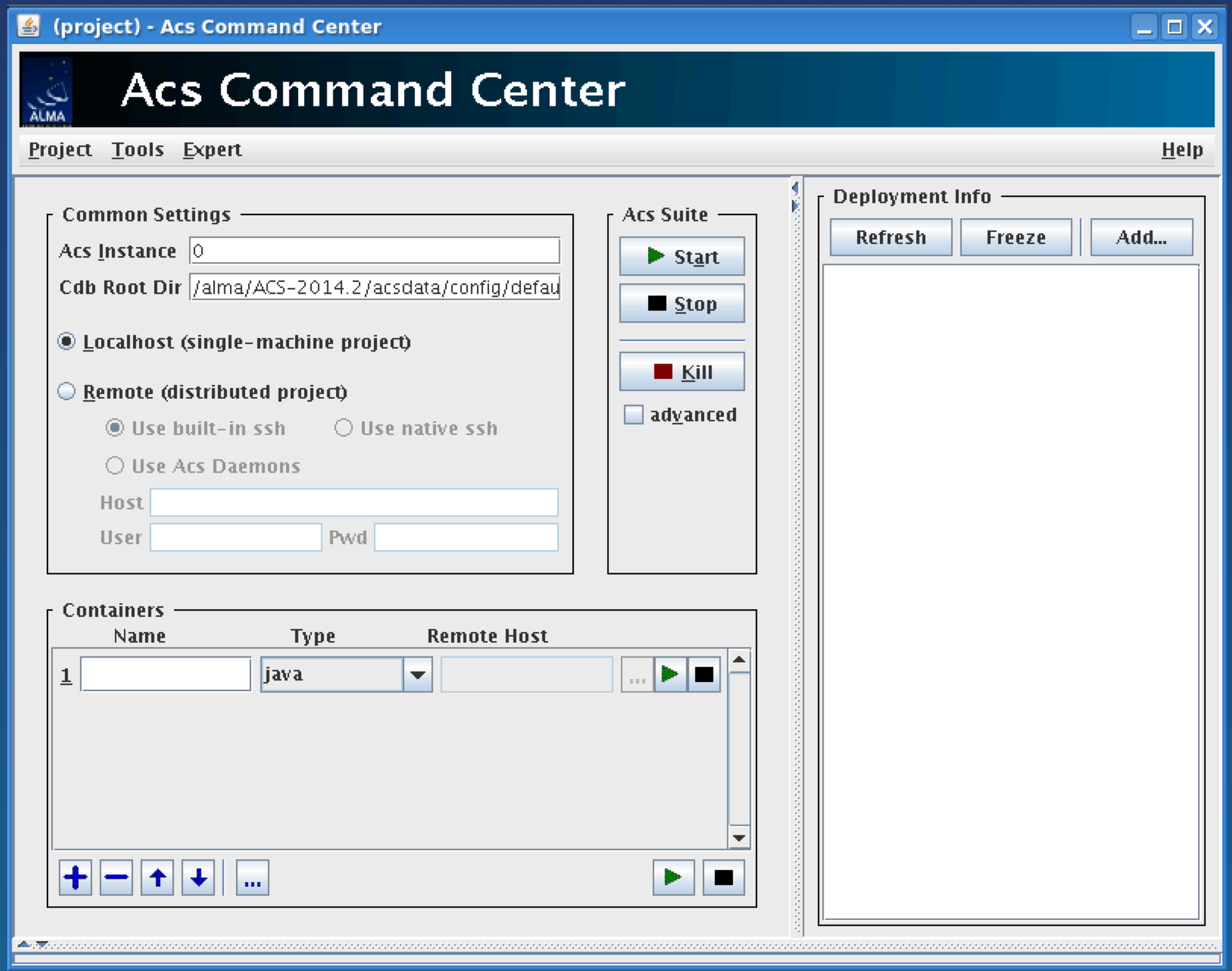
- Every application needs a set of **core** integration infrastructure services, like for instance:
  - messaging (request/response and publish/subscribe)
  - logging
  - error management
  - alarms
  - configuration data
- In ACS these services have been identified as essential for the application domain
- These have been implemented mostly on top of standard CORBA Services
  - DDS/zeroMQ/other, implementations or studies to replace at some point CORBA for some of them
- The ACS work consists in wrapping the implementation to simplify their usage by the application developers

Messaging

- Enables the communication between system components
- Supports both request-response and publish-subscribe message patterns
  - anonymous publish/subscribe data transfer is seen more and more as a key need for the messaging system

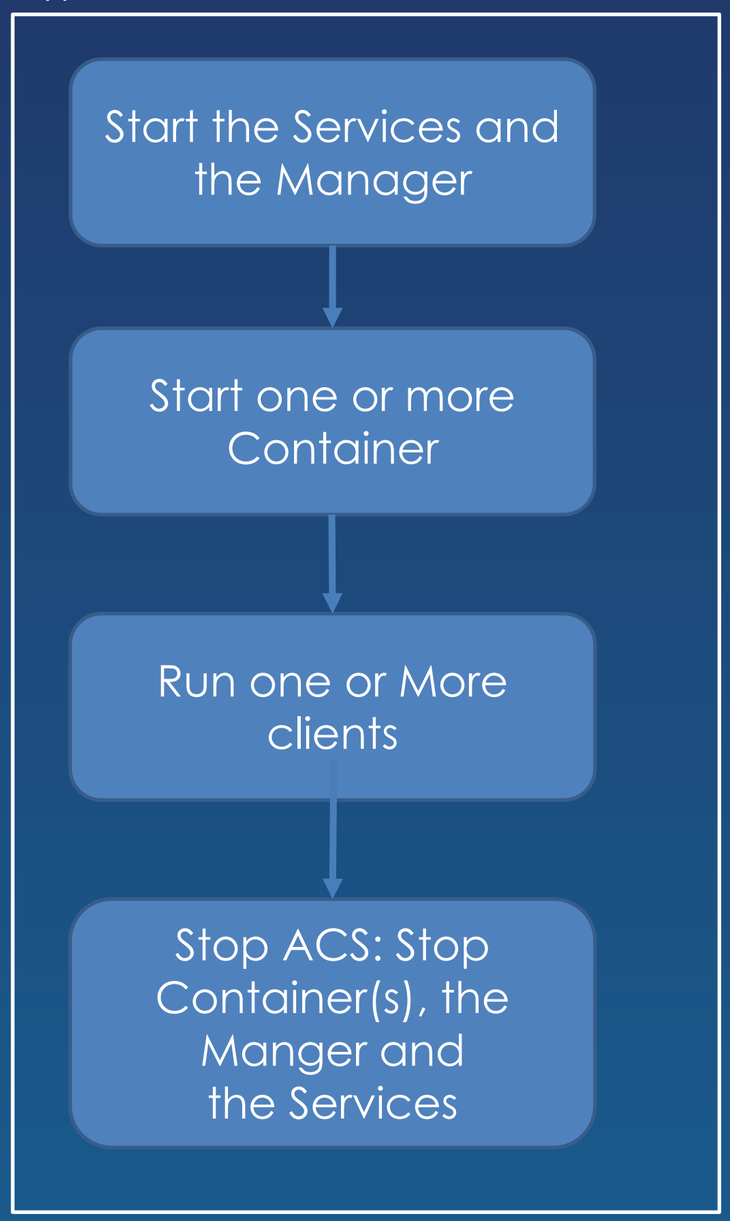
- In ACS:
  - CORBA messaging provides request-response
  - CORBA Notify Service provides publish-subscribe
    - New technologies may replace the Notify Service, offering some advantages
    - DDS implementation exists, zeroMQ/other. studies

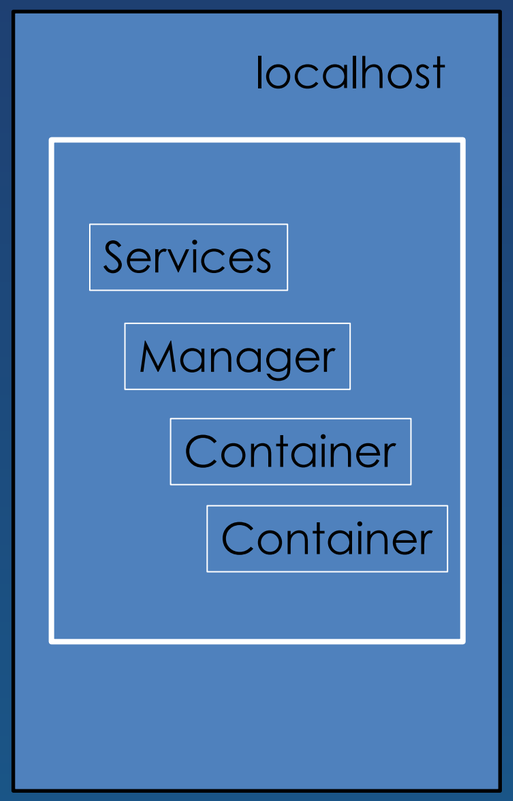
ACS Command Center: starting/stopping ACS



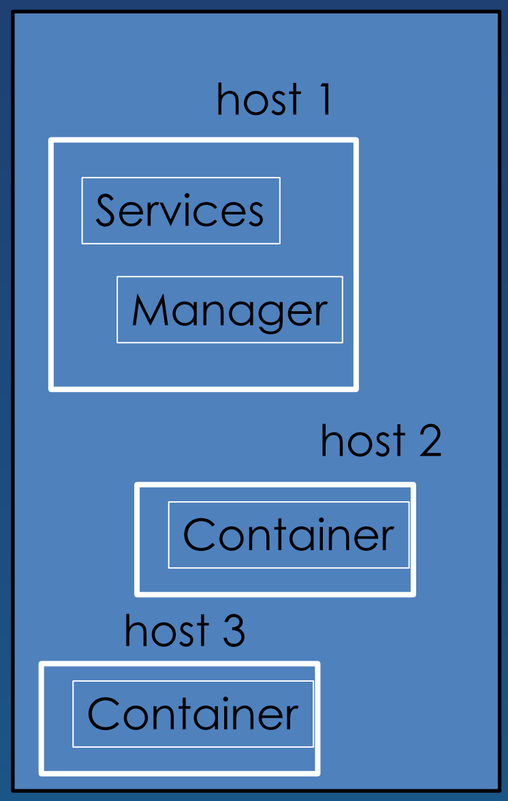
A typical ACS Session

Typical ACS session



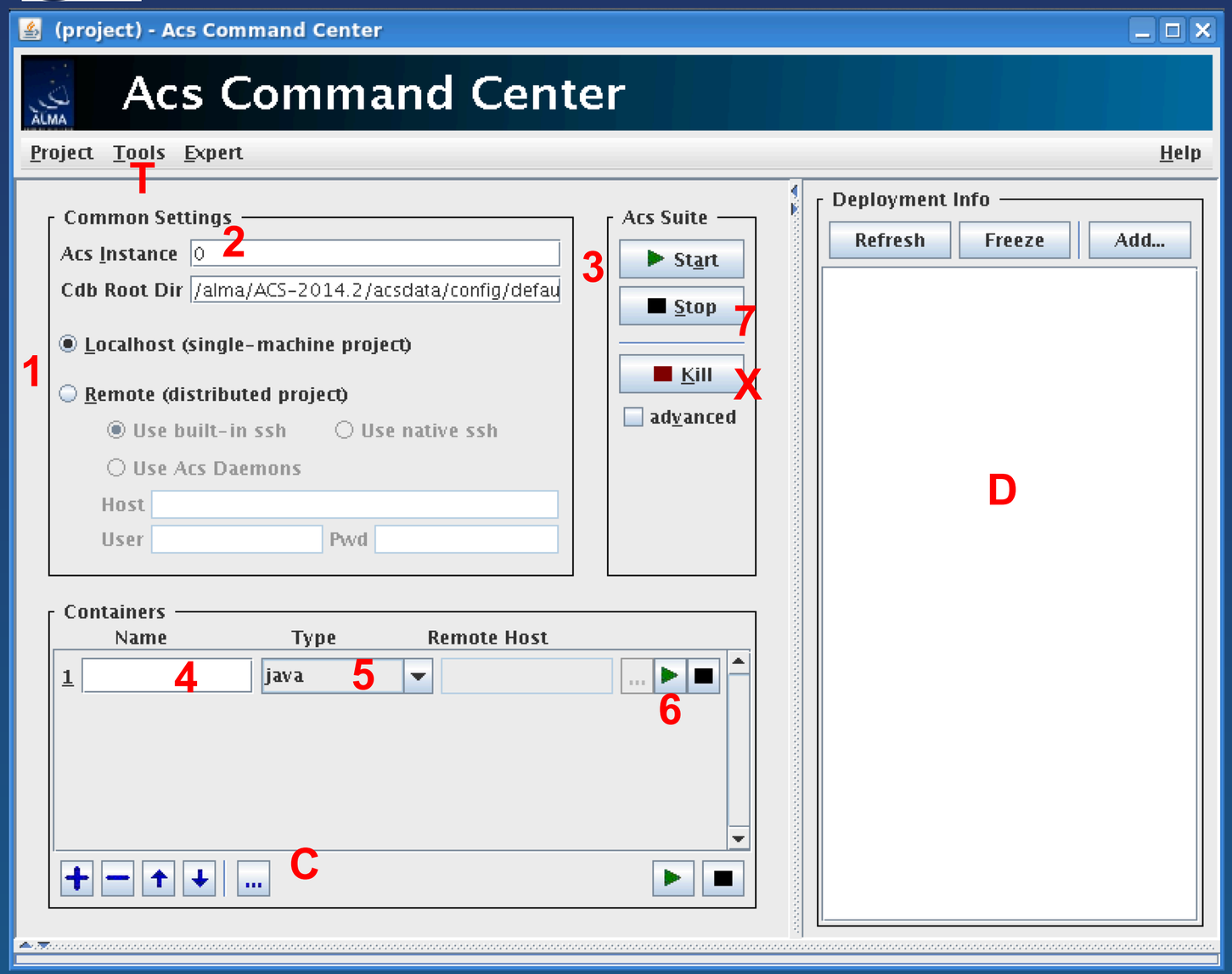


Simple deployment scenario



Complex deployment scenario

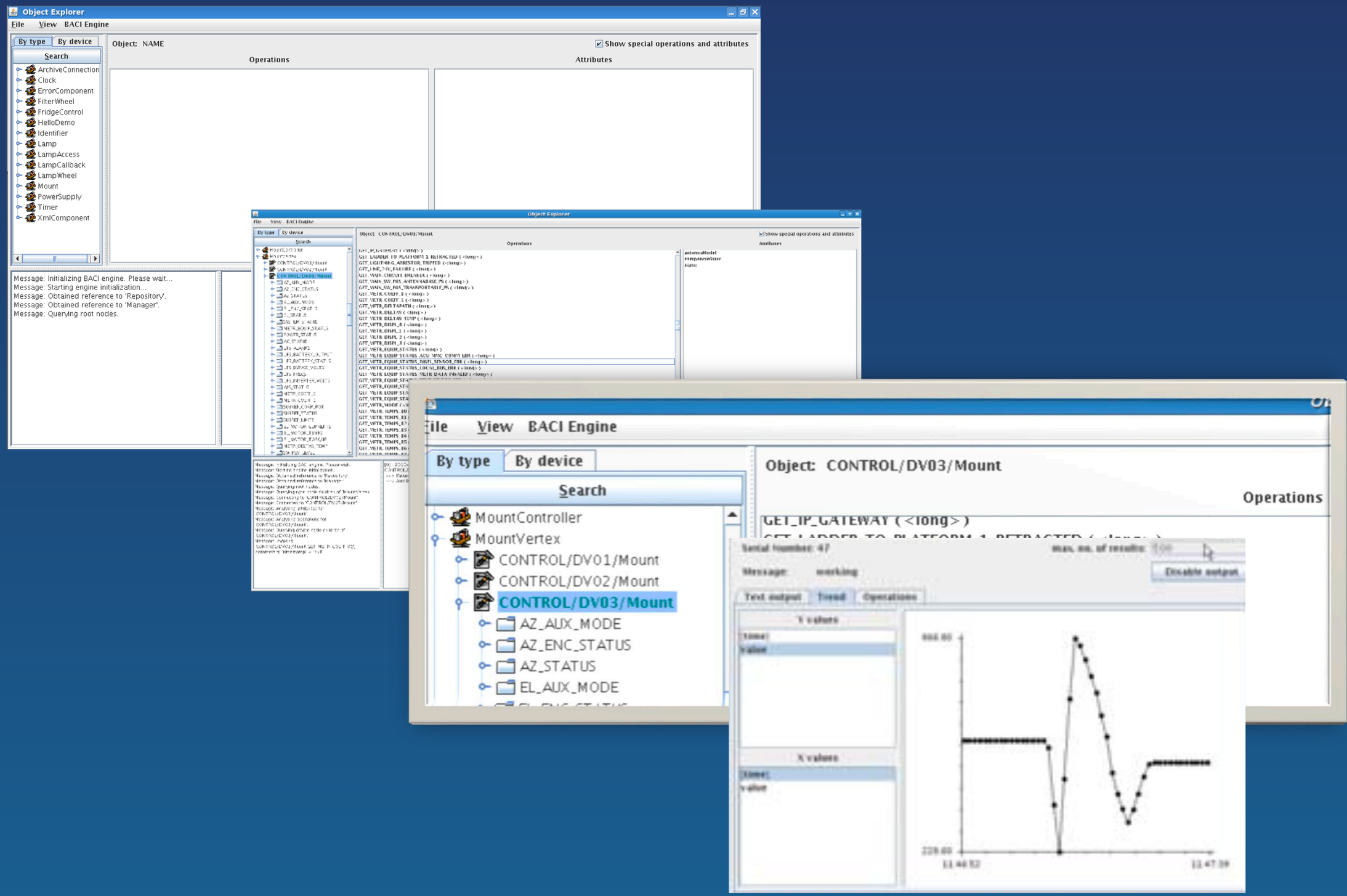
✶Simple deployment scenario



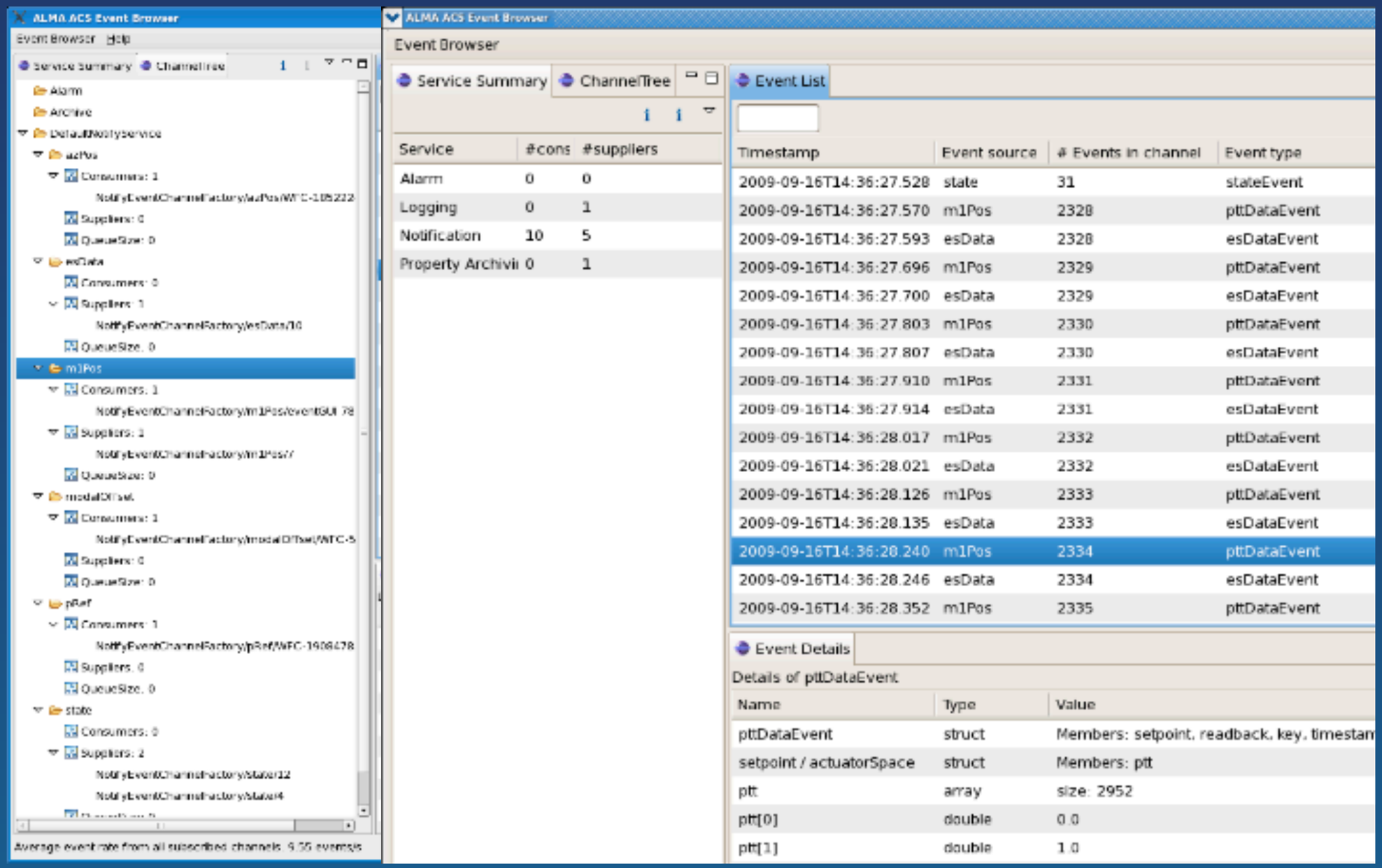
a. Select localhost (1)
b. Specify ACS instance (2)
c. Play button starts Services and manager (3)
    a. Output logs will show in Log Area (L)
    b. freshly started Manager now appears in the deployment info view (D)
d. Specify Container Name (4)
e. Specify Container Type (5)
f. Start Container (6)
    a. More containers can be added via (C)
g. Run your clients.
    a. Predetermined clients can be started from (T)
h. Stop Containers, Manager and Services (7)
    a. All output goes to the logging area (L)
i. If you encounter problems, and find the Acs session in an inoperable state, you can attempt to terminate it (X)

Based on the description Marcus Schilling, that also includes details for the complex case: Link

Generic client: object explorer



Generic client: event browser



[Further details here]

Logging system

- Logging is fundamental for the operation of distributed systems, in order to understand and keep track of what happens between concurrent components
- Logging is used to publish any kind of status and diagnostic information for interested clients and for archival
- The current implementation of ACS is based on the Notification Service
  - Replacement to use newer technologies may come (DDS, zeroMQ, …)

Logging tools: jlog logging client

LoggingClient - Online

File View Search Drill down Expert

Log level: Info Discard level: Debug Pause Clear logs Filters Drill down

Search...

TimeSt...	Entry Type	Source Ob...	Log Message
22:04:09...	Info	Manager	Request for component 'curl:///NameService' issued.
22:04:09...	Info	Manager	Component 'curl:///NameService' provided.
22:04:09...	Info	Manager	ORB status: connectionThreadsUsed=0%, lost calls=0,

Detailed info

LogField	Value
TimeStamp	
Entry Type	
Source Object	
File	
Line	
Routine	
Host	

LoggingClient - Online

File View Search Drill down Expert

Log level: Info Discard level: Debug Pause Clear logs Filters Drill down

Search...

TimeStamp	Entry Type	Source Object	Log Message
19:08:14.412	Info	CONTROL/DV04/FrontEnd	FrontEnd band# 3 locked at 9.208718e+10
19:08:14.320	Info	CONTROL/DV06/FrontEnd	FrontEnd band# 3 locked at 9.208718e+10
19:08:14.011	Info	CONTROL/DV06/FrontEnd	Optimized EDFA value 1.686493, for a target voltage of 2.5V
19:08:12.750	Info	CONTROL/DV04/FrontEnd	Optimized EDFA value 1.486290, for a target voltage of 2.5V
19:08:10.456	Info	CONTROL/DV05/FrontEnd	FrontEnd band# 3 locked at 9.208718e+10
19:08:10.169	Info	CONTROL/DV08/FrontEnd	FrontEnd band# 3 locked at 9.208718e+10
19:08:09.522	Info	CONTROL/DV05/FrontEnd	Optimized EDFA value 2.451853, for a target voltage of 2.5V
19:08:09.517	Info	CONTROL/DV05/FrontEnd/WCA3	WCA PLL adjust to near zero
19:08:09.457	Info	CONTROL/DV02/FrontEnd	FrontEnd band# 3 locked at 9.208718e+10
19:08:09.455	Info	CONTROL/DV08/FrontEnd	Optimized EDFA value 1.668773, for a target voltage of 2.5V
19:08:09.341	Info	CONTROL/DV05/FrontEnd/WCA3	WCA Lock Acquired
19:08:09.168	Info	CONTROL/DV05/FrontEnd/WCA3	WCA Frequency set to: 9.20872e+10
19:08:09.149	Info	CONTROL/PM02/FrontEnd	FrontEnd band# 3 locked at 9.208718e+10
19:08:08.966	Info	CONTROL/DV05/FrontEnd	Attempting lock with EDFA value 2.451853, and target PM current 0.000541
19:08:08.854	Info	CONTROL/DV03/FrontEnd	FrontEnd band# 3 locked at 9.208718e+10
19:08:08.685	Info	CONTROL/PM02/FrontEnd	Optimized EDFA value 1.704569, for a target voltage of 2.5V
19:08:08.680	Info	CONTROL/PM02/FrontEnd/WCA3	WCA PLL adjust to near zero
19:08:08.583	Info	CONTROL/DV02/FrontEnd	Optimized EDFA value 1.612308, for a target voltage of 2.5V
19:08:08.577	Info	CONTROL/DV02/FrontEnd/WCA3	WCA PLL adjust to near zero
19:08:08.527	Info	CONTROL/DV03/FrontEnd	Optimized EDFA value 1.612405, for a target voltage of 2.5V
19:08:08.503	Info	CONTROL/PM02/FrontEnd/WCA3	WCA Lock Acquired
19:08:08.401	Info	CONTROL/DV02/FrontEnd/WCA3	WCA Lock Acquired
19:08:08.351	Info	CONTROL/DV03/FrontEnd/WCA3	WCA PLL adjust to near zero

Detailed info

LogField	Value
TimeStamp	2010-11-15T19:07:52.217
Entry Type	Info
Source Object	CONTROL/Array001
File	alma.Control.TotalPowerProcessor.TotalPowerProcessorImpl
Line	232
Routine	addDigitizer
Host	gas01
Process	CONTROL/ACC/javaContainer
Context	
Thread	RequestProcessor-29
Log ID	4550
Priority	
URI	
Stack ID	
Stack Level	
Log Message	Adding new Processor Client for Antenna: DV08 Polarization: 1
Audience	
Array	
Antenna	

100K Engine not filtered Table not filtered Engineer

Error System

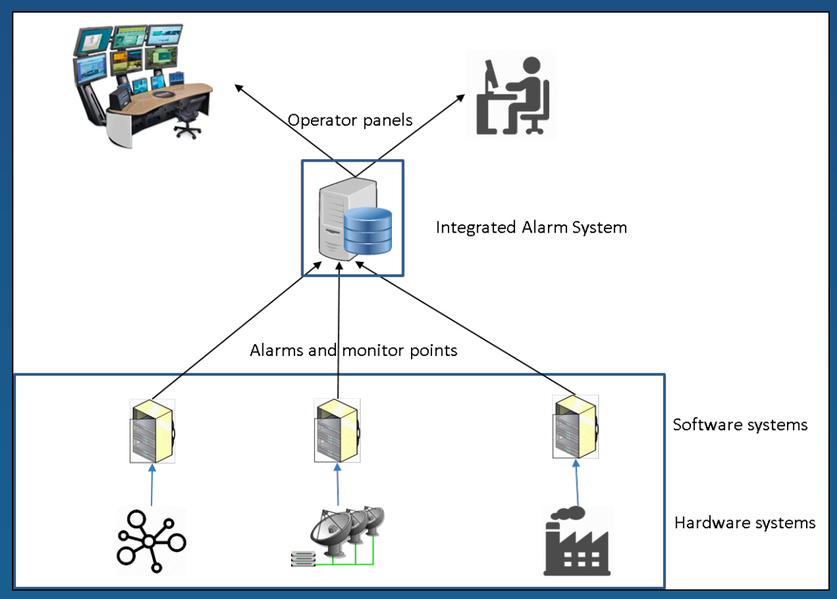
- Provides a unified way of dealing with errors/exceptions through the system
- CORBA supports “distributed” exceptions
- The ACS Error System provides additionally the following features:
  - Error format standardisation
  - Error handling design patterns
  - Error trace
  - Error logging
  - Synchronous and asynchronous error handling
  - Error browsing and definition tools

Alarm System

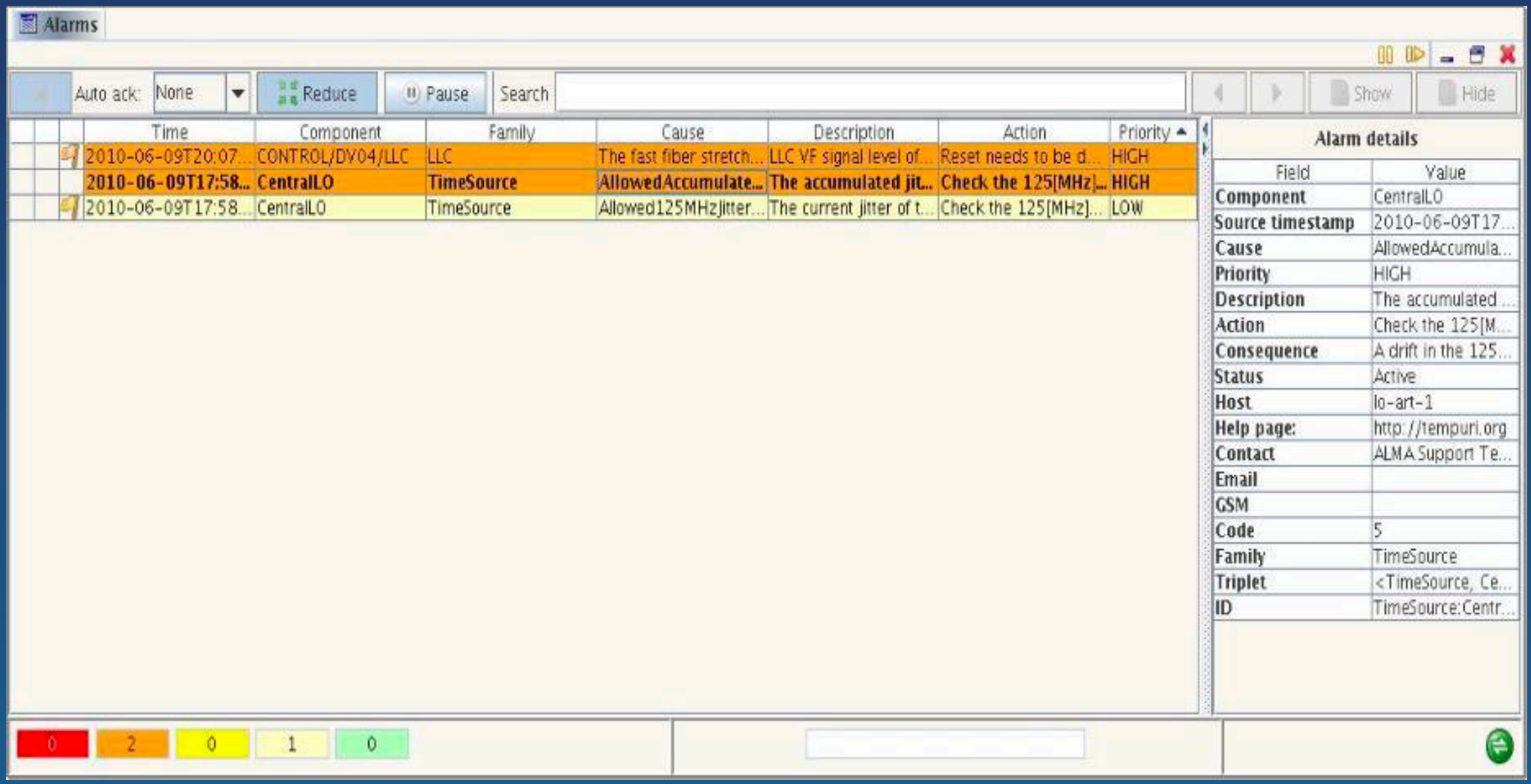
- Deals with **abnormal** situations
  - Fault states (FS)
  - Range from severe alarms to warning states
- Provides
  - FS collection, analysis and distribution, definition and archiving
  - FS reduction
  - Dedicated alarm consoles
- The ACS alarm system is a porting of the CERN LASER system out of the box

- Alma using the “Integrated Alarm System (A. Caproni)to provide further capabilities – CTA intends to use it as well.

https://integratedalarmsystem-group.github.io/



✳ Alarm System Console

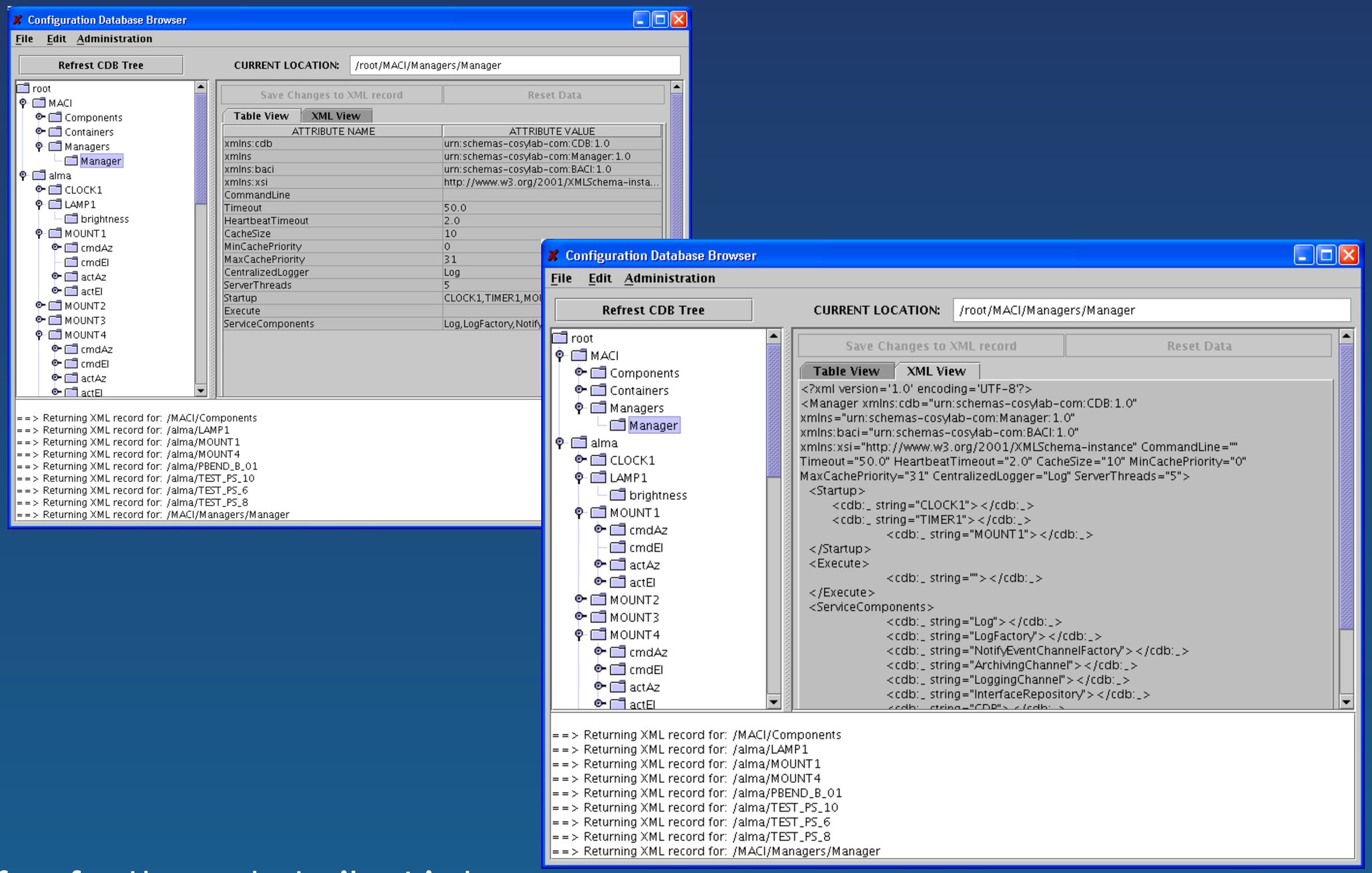


Configuration Database

- The ACS Configuration Database (CDB) addresses: defining, accessing and maintaining the configuration of a system
- For each component in the system, there might be a set of static (or quasi-static) configuration parameters that have to be configured in a persistent store and read when the component is started up or re-initialized.
- This includes the “deployment structure” of the system, i.e., which statically deployed Components are part of the system and their inter-relationships
- This information is used by the component/container infrastructure in runtime

- See presentation of Tomás: Software deployment: Configuration Database and GIT

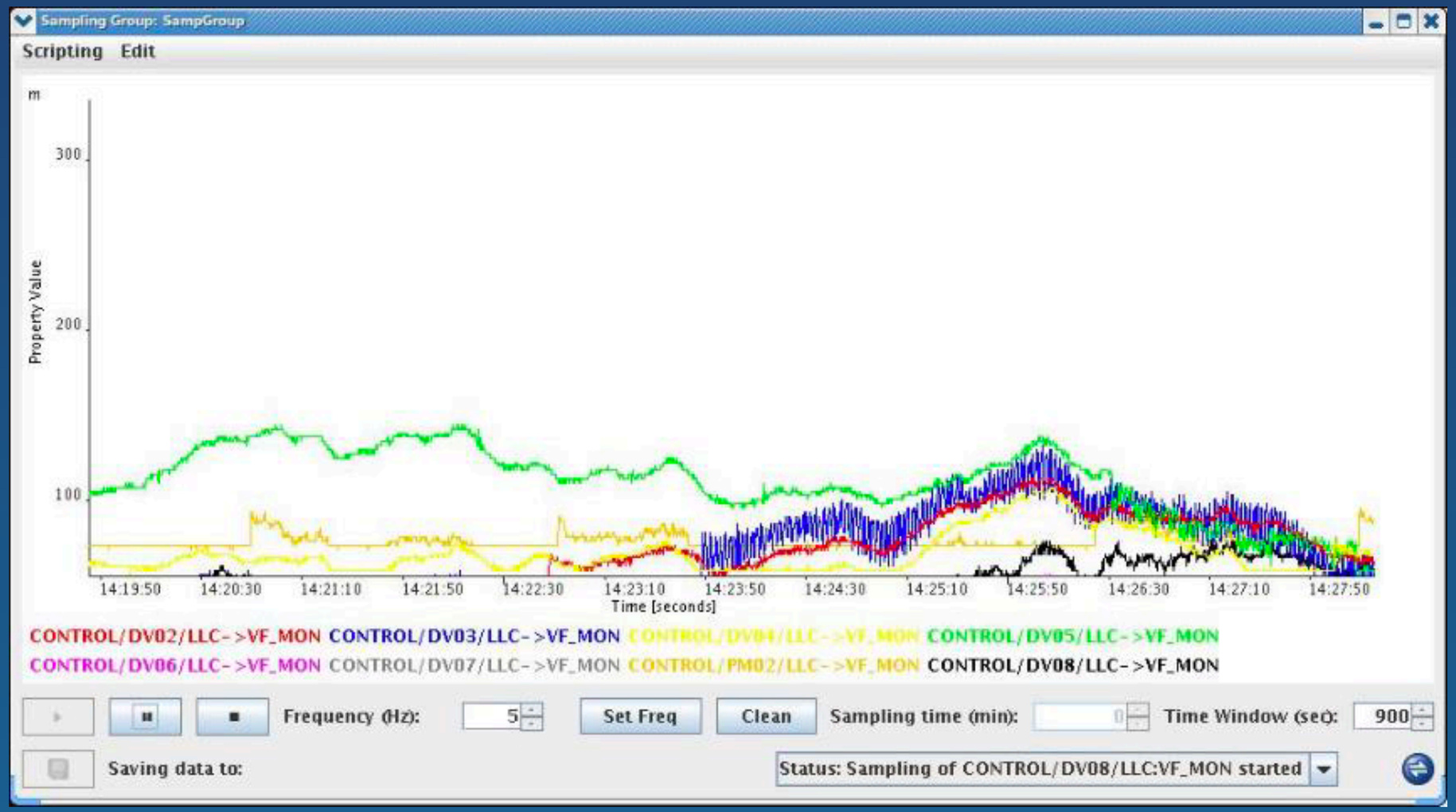
Configuration Database Browser



See here for further details: Link

Sampling System

- ✧ Sampling of any Property
- ✧ High sustained frequency
- ✧ Optimized data transport
- ✧ Simultaneous sampling
- ✧ Plotting GUI



More details here: Link

Component simulation

- ✧ The component simulator is based on the CDB y Python
- ✧ Useful in ALMA early stages but nor much used anymore
- ✧ Further details: Link

Questions?

Acknowledgements
ACS presentations were originally developed by the ALMA Common Software development team and has been used in many instances of training courses since 2004. Main contributors are (listed in alphabetical order): Jorge Avarias, Alessandro Caproni, Gianluca Chiozzi, Jorge Ibsen, Thomas Jürgens, Matias Mora, Joseph Schwarz, Heiko Sommer.

The Atacama Large Millimeter/submillimeter Array (ALMA), an international astronomy facility, is a partnership of Europe, North America and East Asia in cooperation with the Republic of Chile. ALMA is funded in Europe by the European Organization for Astronomical Research in the Southern Hemisphere (ESO), in North America by the U.S. National Science Foundation (NSF) in cooperation with the National Research Council of Canada (NRC) and the National Science Council of Taiwan (NSC) and in East Asia by the National Institutes of Natural Sciences (NINS) of Japan in cooperation with the Academia Sinica (AS) in Taiwan. ALMA construction and operations are led on behalf of Europe by ESO, on behalf of North America by the National Radio Astronomy Observatory (NRAO), which is managed by Associated Universities, Inc. (AUI) and on behalf of East Asia by the National Astronomical Observatory of Japan (NAOJ). The Joint ALMA Observatory (JAO) provides the unified leadership and management of the construction, commissioning and operation of ALMA.