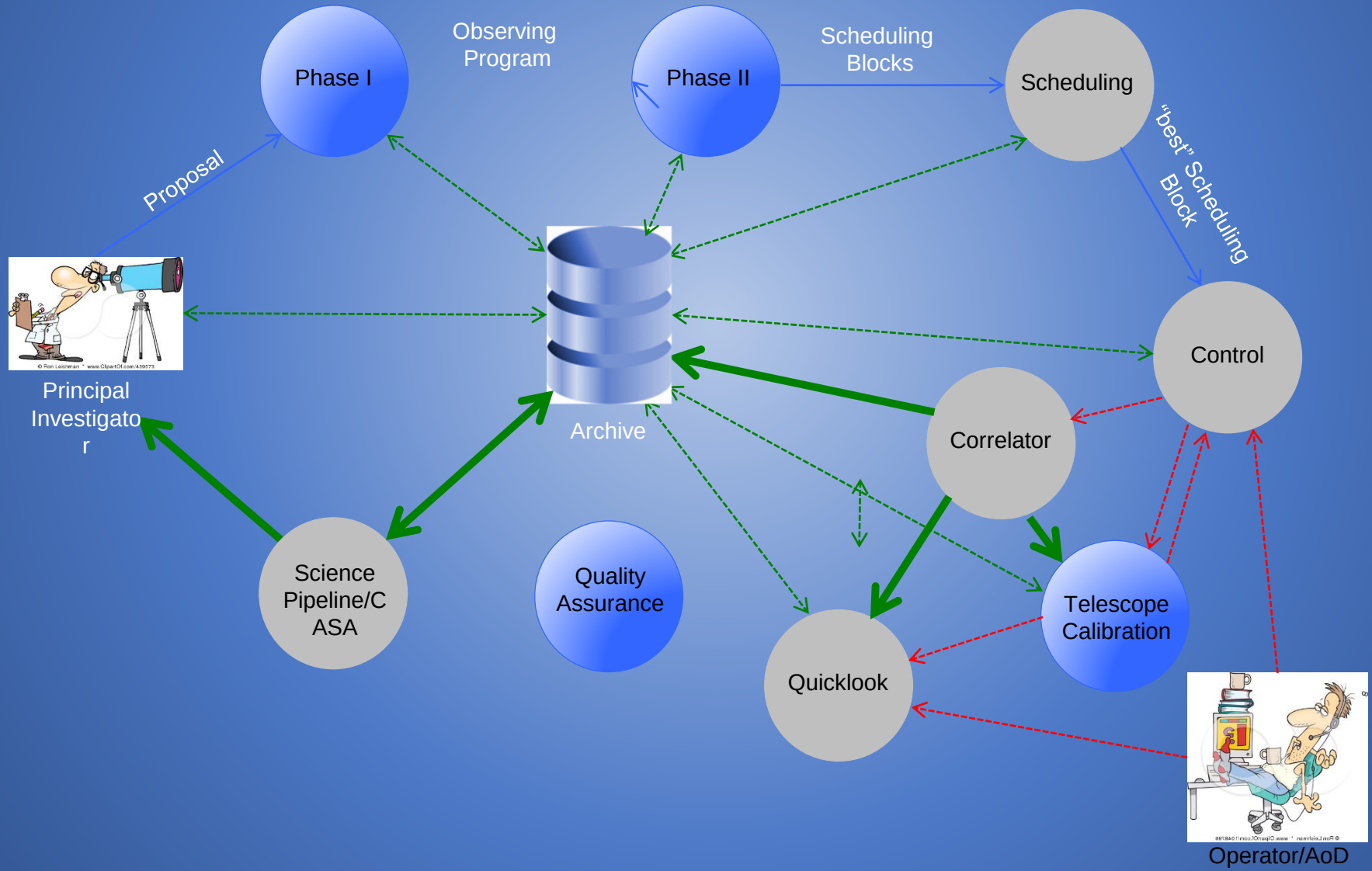
The ALMA Common Software

Alessandro Caproni

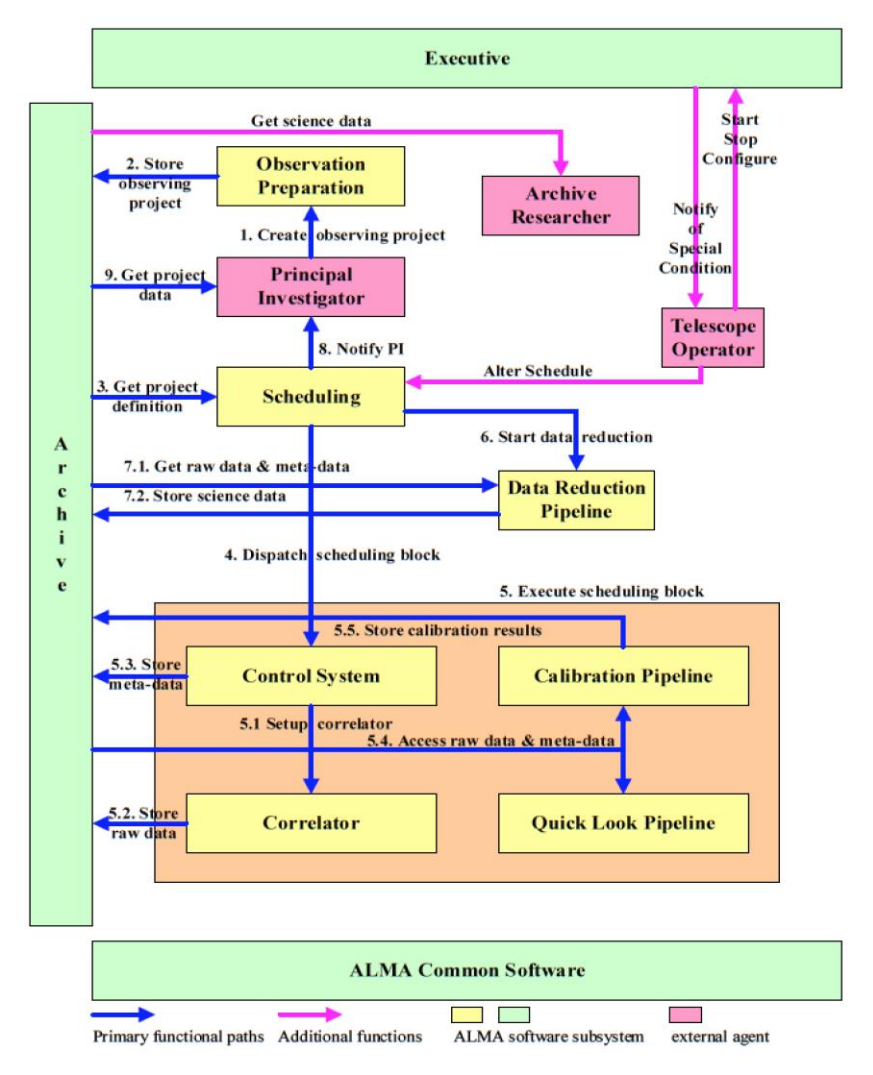
Summary

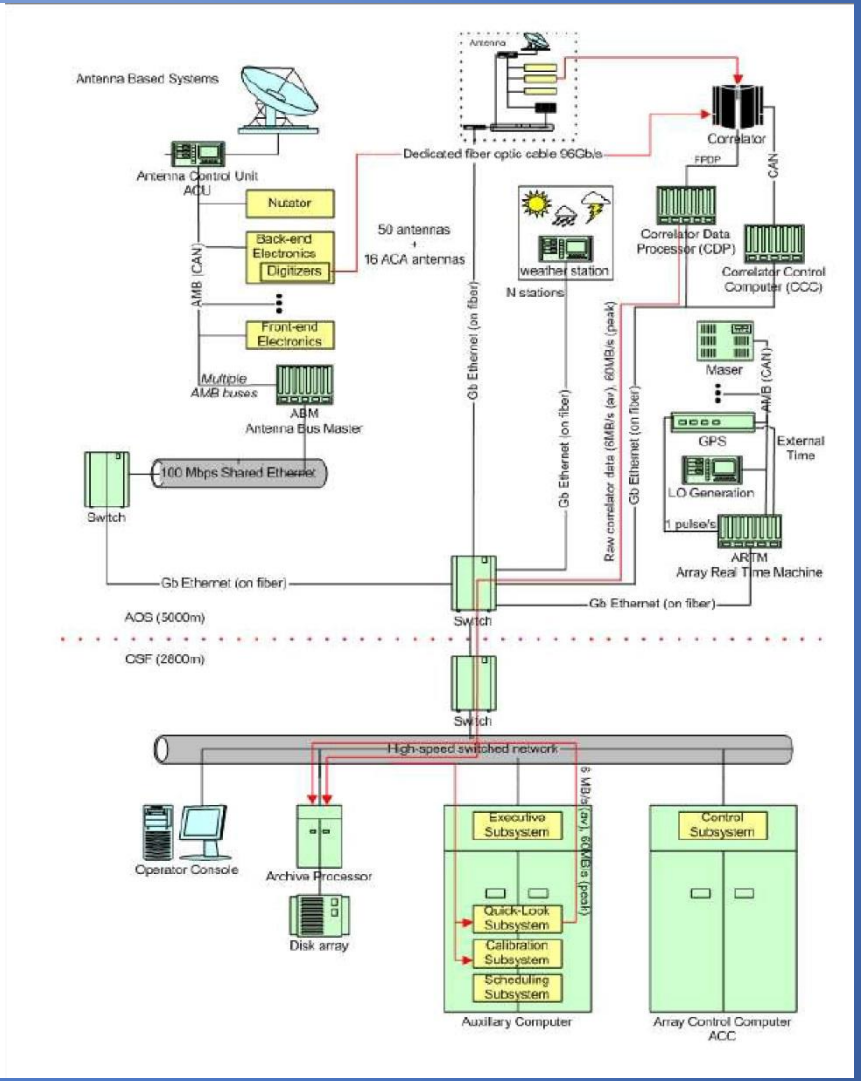
- What is ACS
- ACS services
- Component-Container paradigm
- Development
- Deployment
- Run-time

End-To-End data flow



ALMA – software and physical architecture





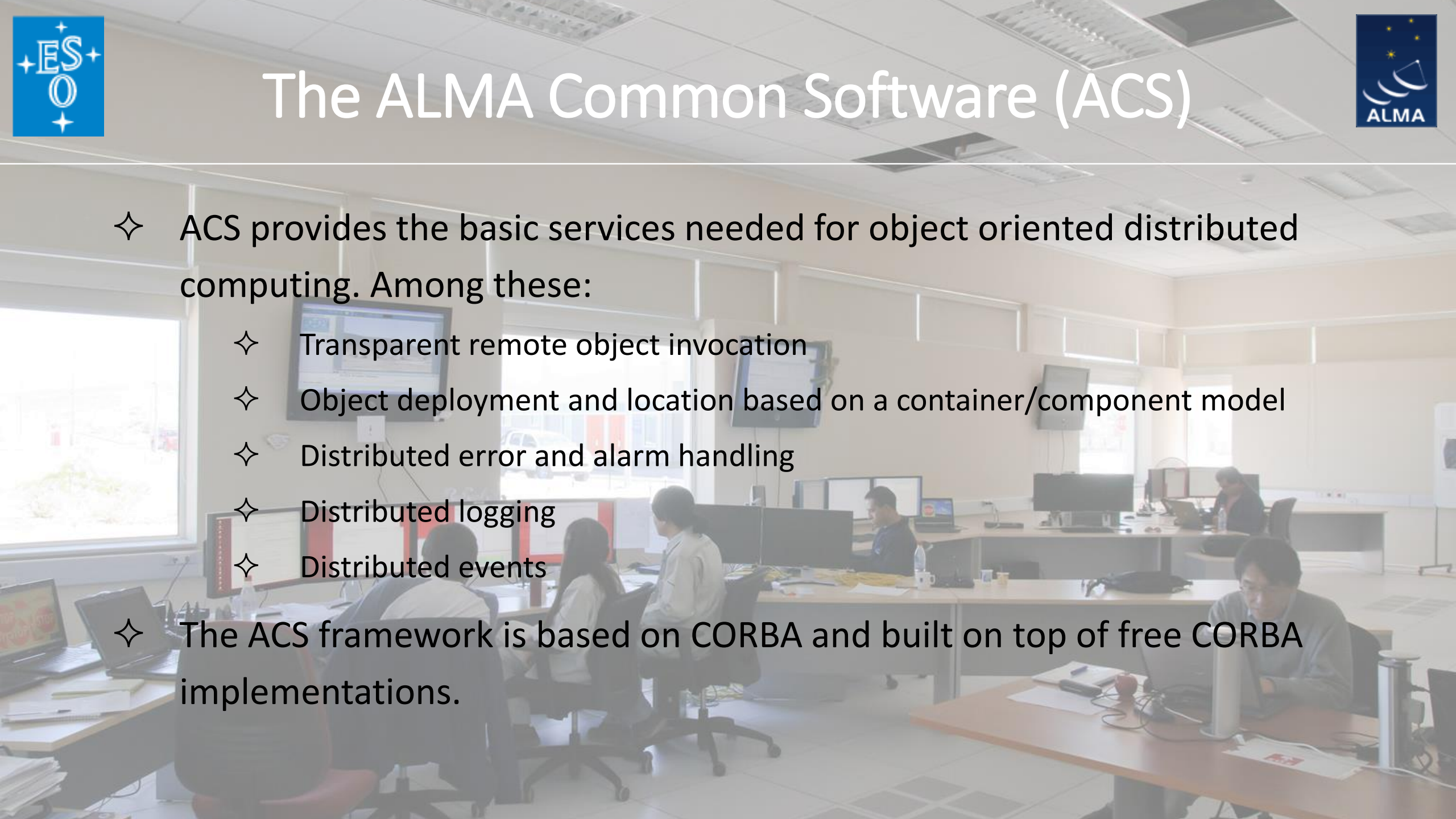
The observatory is a distributed system

- Servers and clients are distributed on different machines:
  - Possibly in different locations
  - With different purpose and functionality
  - With different requirements on performance and reliability
- Servers and clients may use different:
  - Hardware
  - System software
  - Programming languages

Requirements

- Developers of clients shall be unaware of the underlying server architecture & vice-versa
- It shall be possible to change the architecture of a server transparently to the client
- Client developers shall not even need to know whether a server is local or remote.

The ALMA Common Software (ACS)



- ACS provides the basic services needed for object oriented distributed computing. Among these:
  - Transparent remote object invocation
  - Object deployment and location based on a container/component model
  - Distributed error and alarm handling
  - Distributed logging
  - Distributed events
- The ACS framework is based on CORBA and built on top of free CORBA implementations.

ACS

- Operating system: RHEL 5.5 and 6.5 (32 and 64 bits)
  - CentOS/SL 5 and 6 binary compatible
  - Other linux versions supported by external projects
  - Windows added also by external initiatives
- Real-time: RTAI
  - VxWorks supported by and for APEX
- Languages: C++, Java, Python
- CORBA middleware: TAO (C++), JacORB (Java), Omniorb (Python), CORBA services.
- Embedded ACS Container: PC104, Debian, 300Mhz Geode, 256MB RAM, 256 MB flash (CosyLAB microIOC), …

LGPL and free software

- Use as much as possible open-source tools, instead of implementing things.
  - Do not reinvent the wheel
  - Reuse experience of other projects
  - Do not pay for licenses
  - Support from user community
- Wrap with convenience and unifying APIs
- ACS is distributed under LGPL license
- Open source projects may have drawbacks
  - Fast lifecycle and support only of the newest
  - Free/commercial support
  - Documentation not as good as commercial products

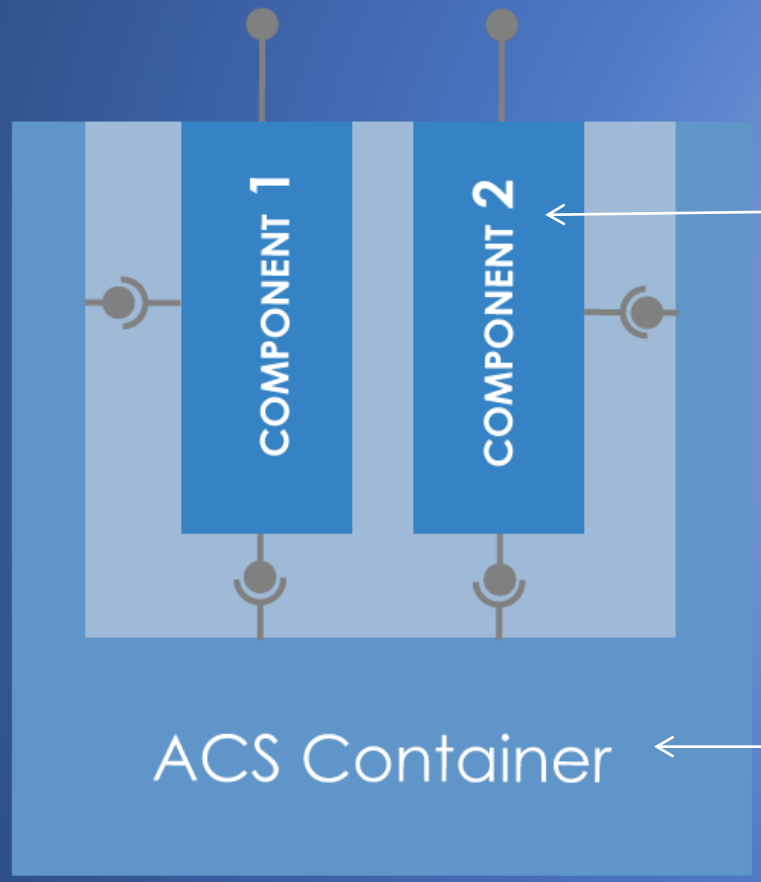
ACS services

- Naming service
- Interface Repository
- Notify Service*
- Logging Service
- Configuration database
- Alarm system
- Manager

ACS for developers

- Developers write components and graphical user interfaces clients in C++, Java, or Python.
- ACS provides an integrated build environment based on application code modules.
- Communication from an application to a component, and among components, uses ACS as middleware.
- No thinking about starting and stopping components, or on which machine they should run later.
- ACS keeps development, deployment and runtime separate

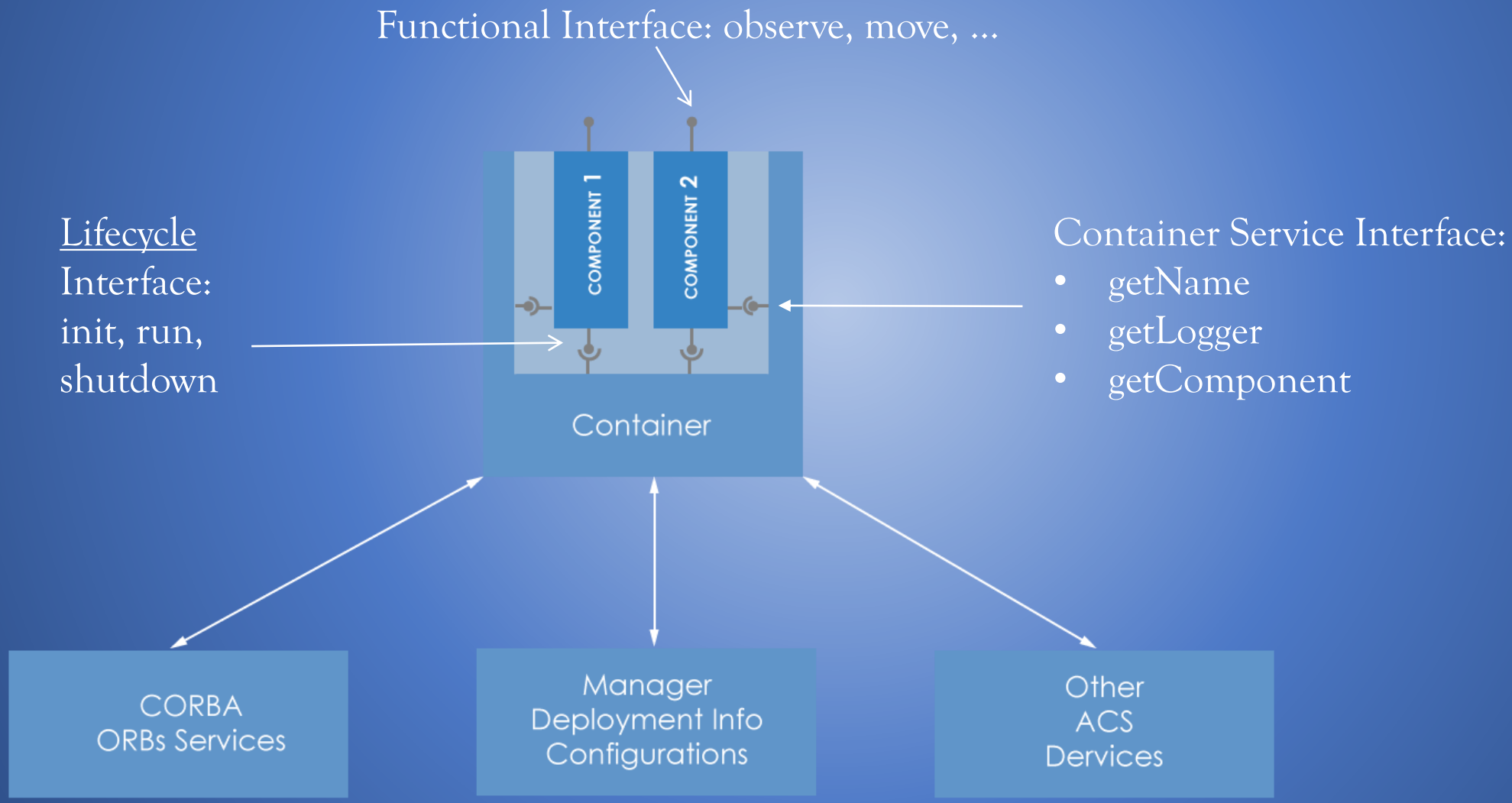
Container/Component



Components provide specific functionality to the system. They are started and stopped by the container, whom offers the component services

The container only cares about the lifecycle interface of the components deployed on it

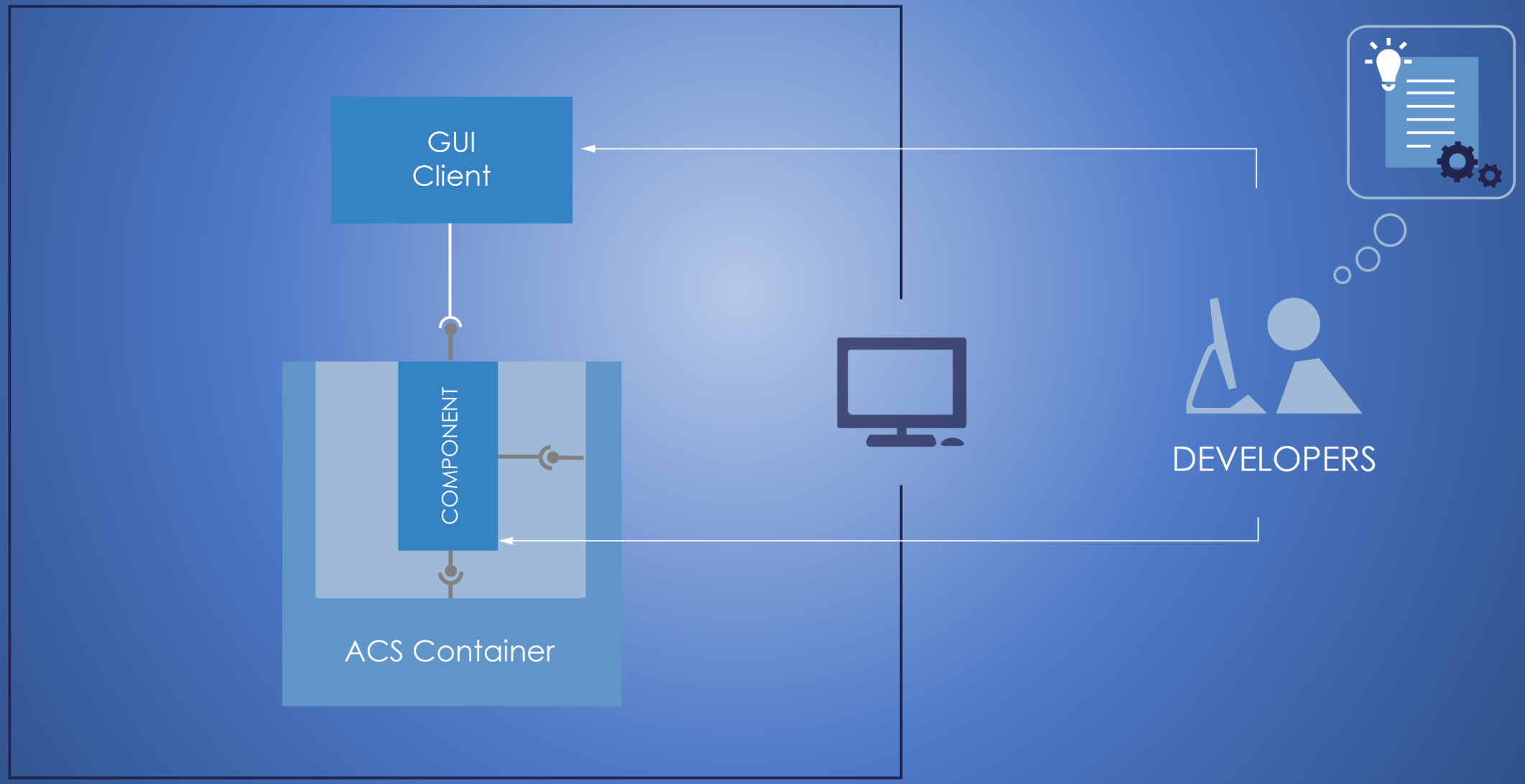
Container/Component interfaces



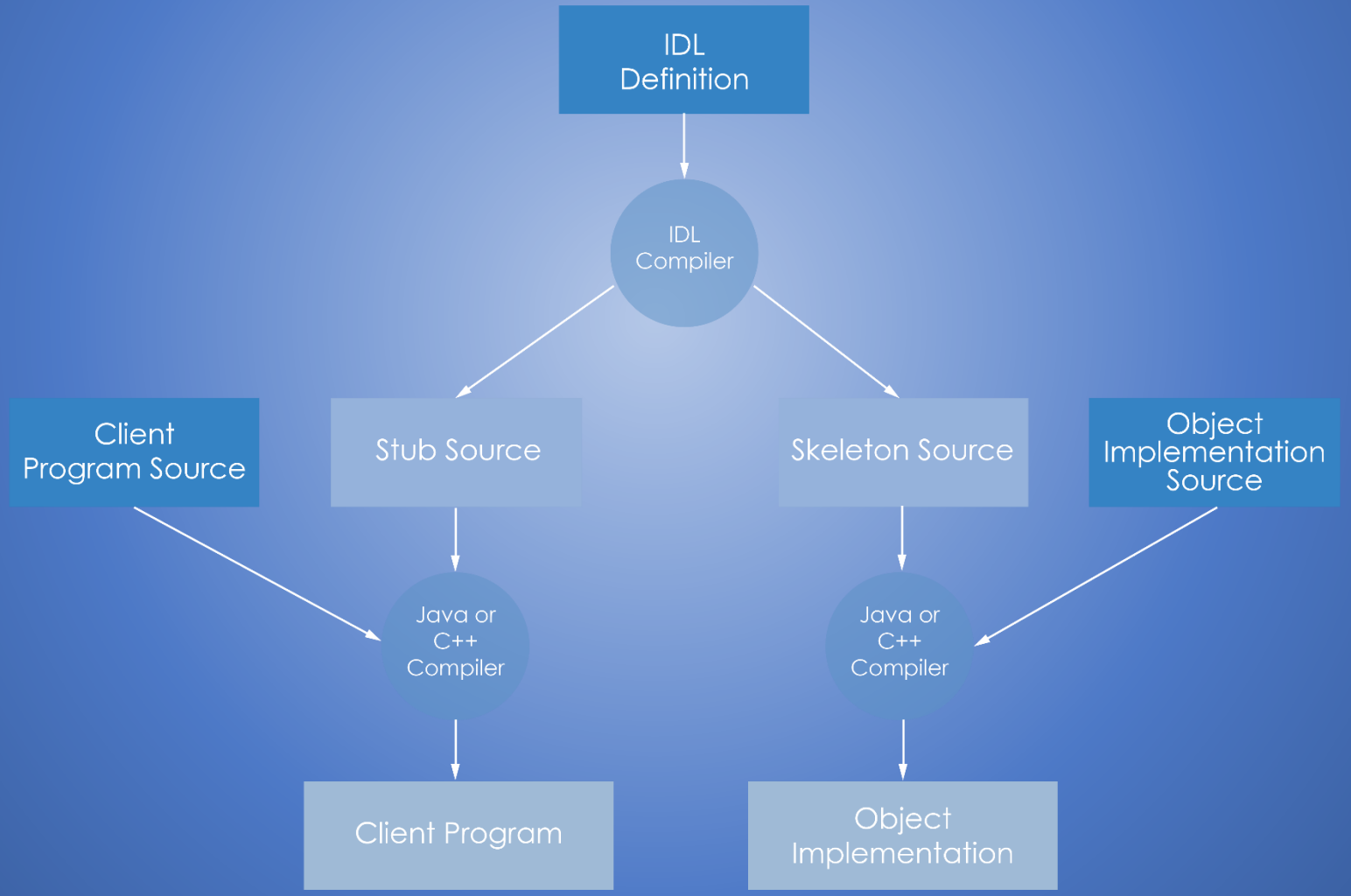
Development - 1

- First step: Identify objects
  - Mount
  - Camera
  - Telescope
- Second step: Define interfaces
  - Implementation comes later and is independent of interface
  - Deployment is also independent of interface definitions
  - Interfaces shall be kept as stable as possible, but it must be possible to have them evolve when needed.
  - A formal interface definition language is needed
  - Simulation

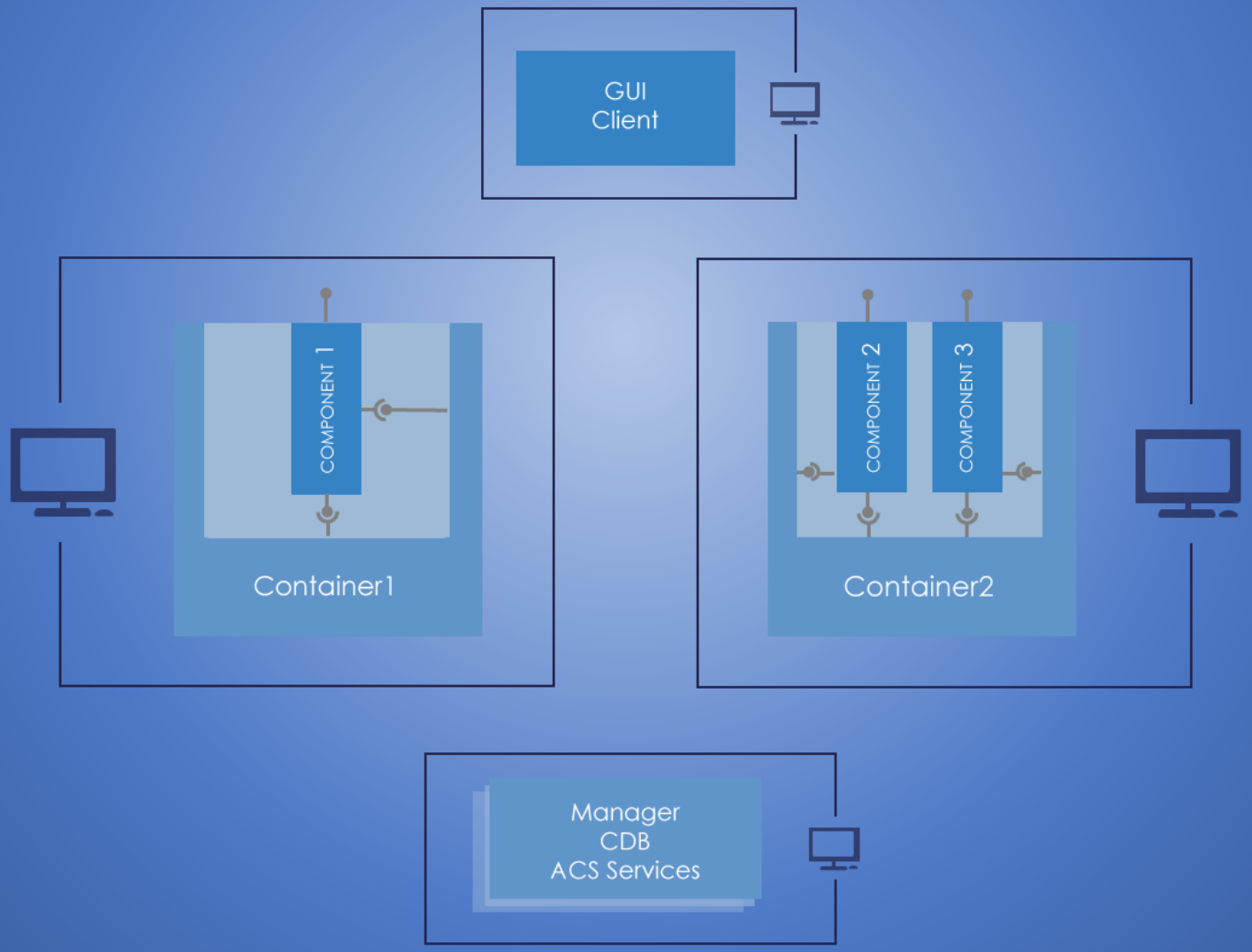
Development - 2



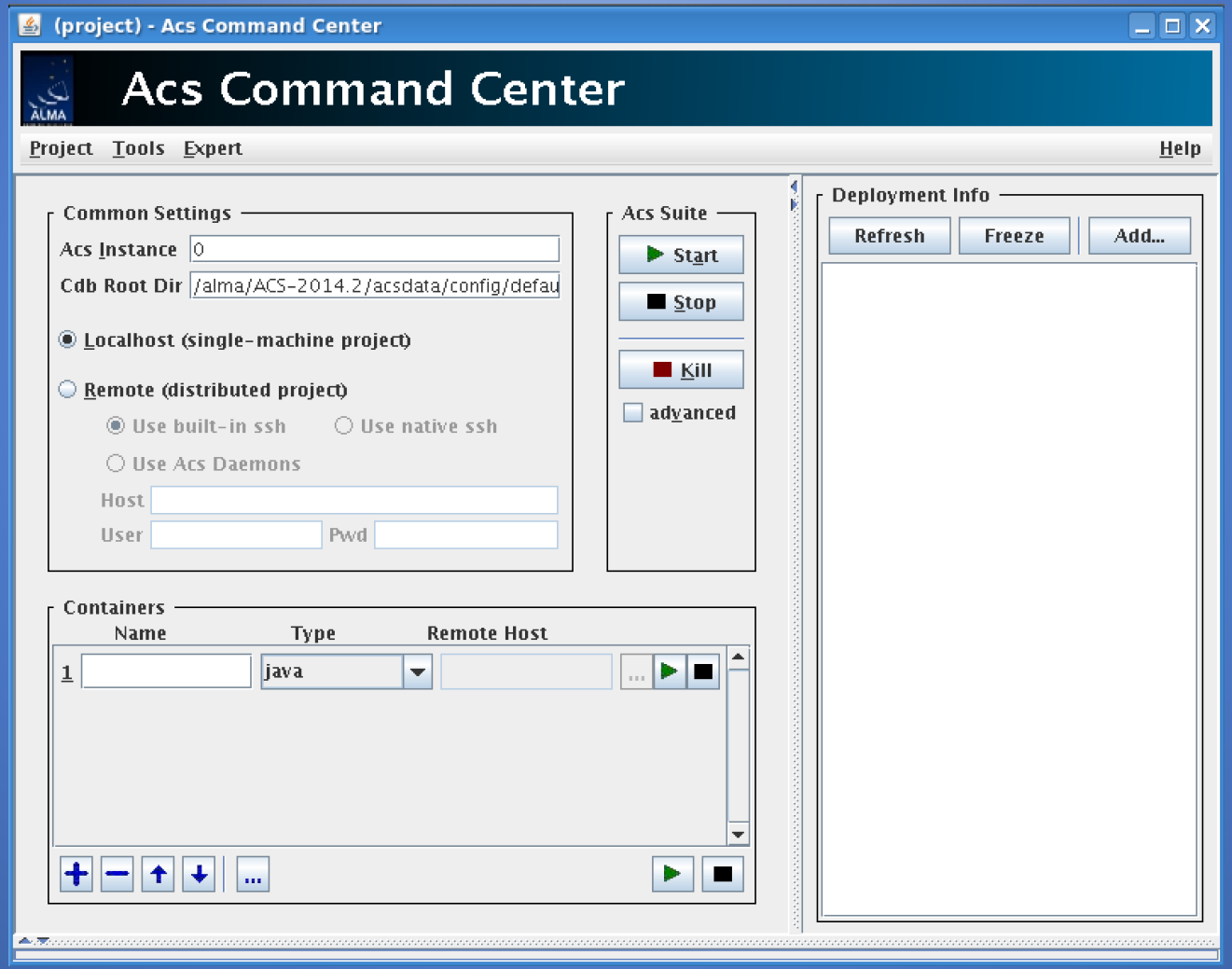
Development - 3



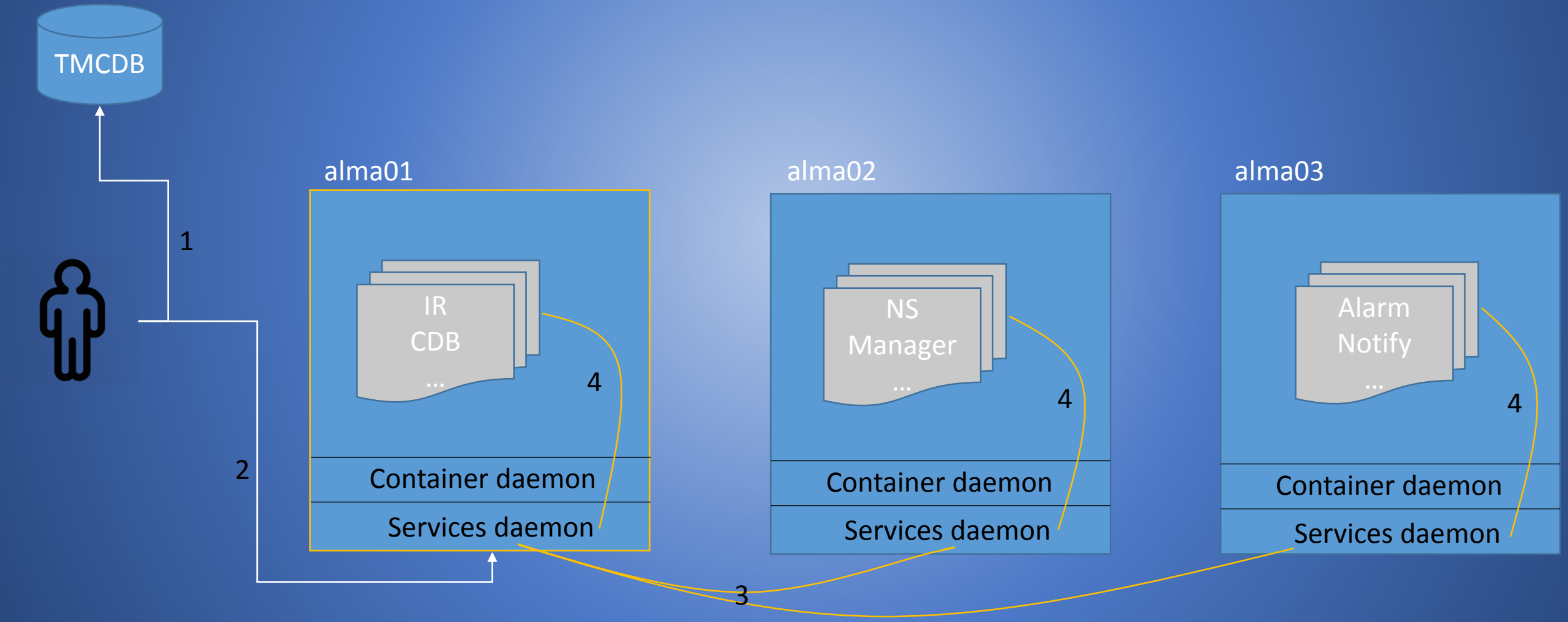
Deployment


GUI
Client
COMPONENT 1
Container1
COMPONENT 2
COMPONENT 3
Container2
Manager
CDB
ACS Services

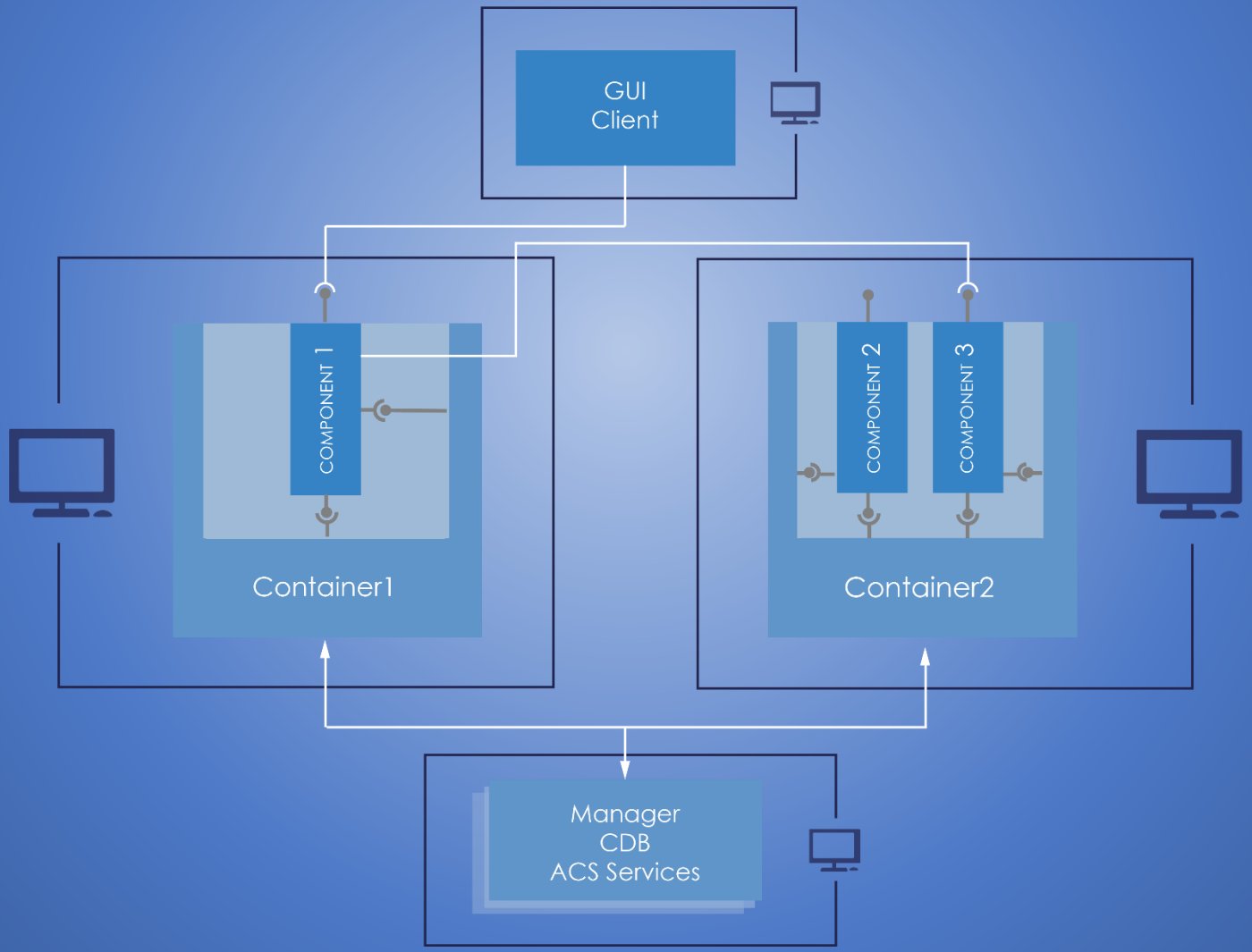
Acs command center



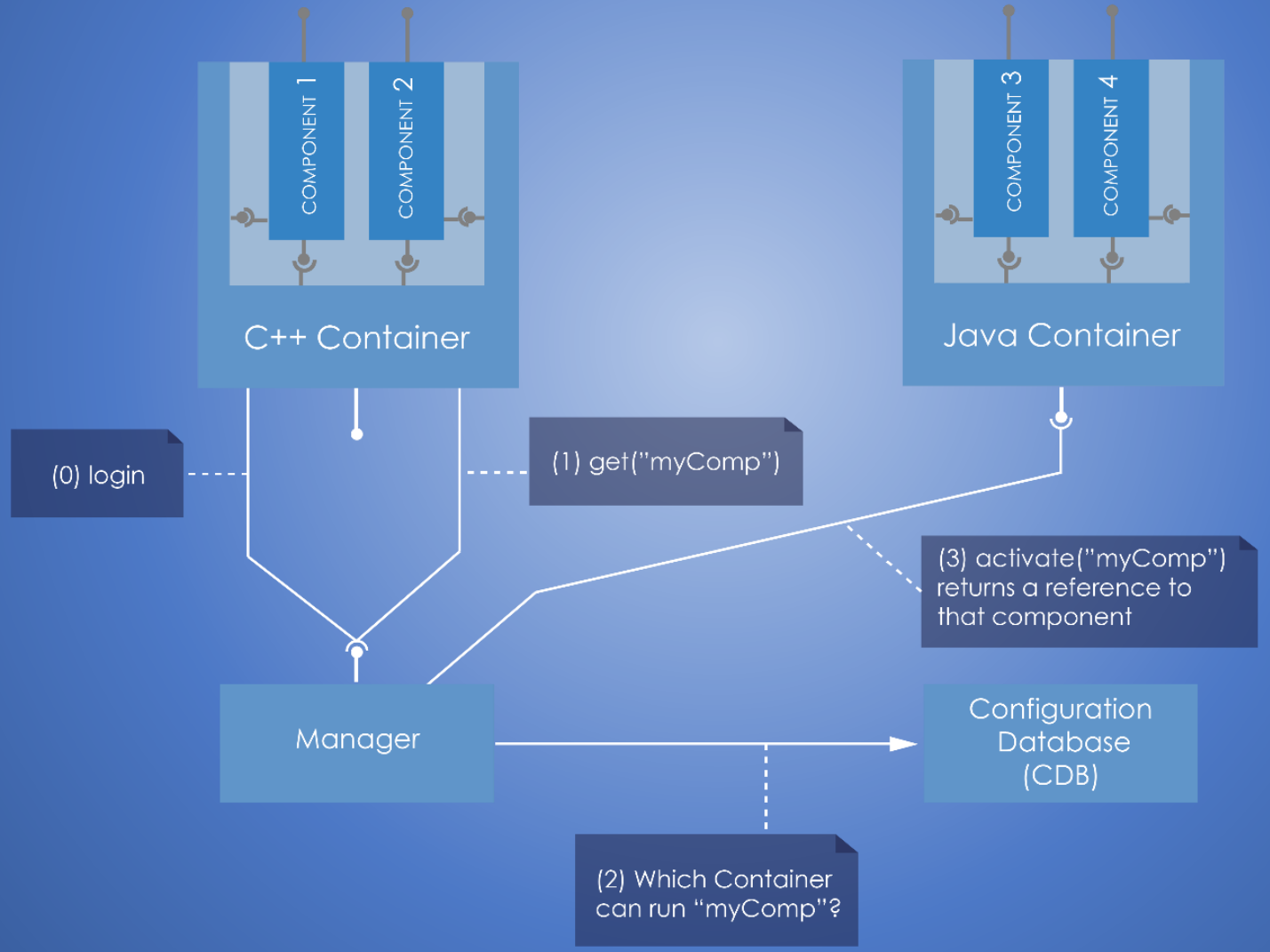
Daemons


TMCDB
1
2
3
4
alma01
IR
CDB
...
Container daemon
Services daemon
alma02
NS
Manager
...
Container daemon
Services daemon
alma03
Alarm
Notify
...
Container daemon
Services daemon

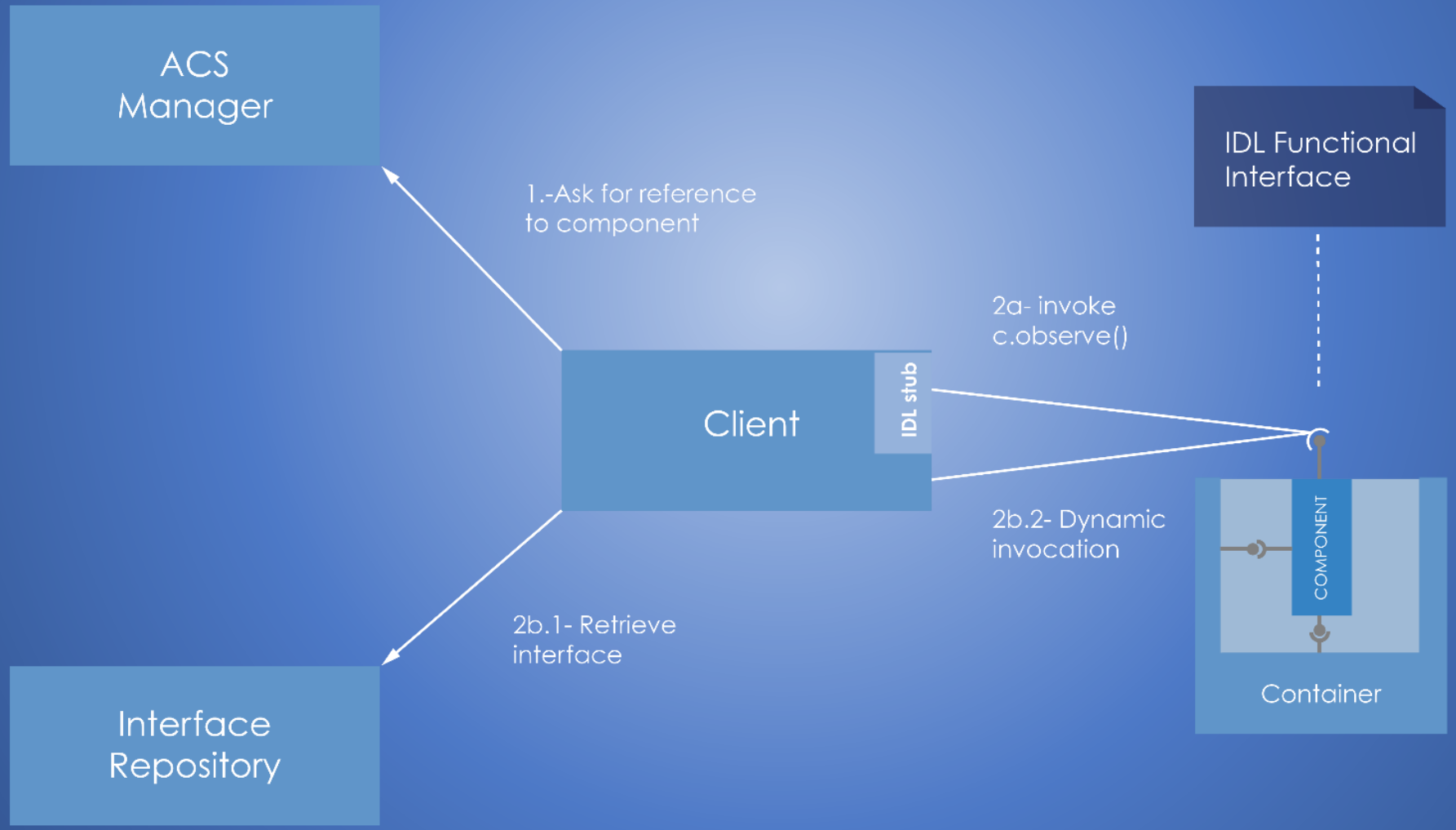
Runtime


GUI
Client
COMPONENT 1
COMPONENT 2
COMPONENT 3
Container1
Container2
Manager
CDB
ACS Services

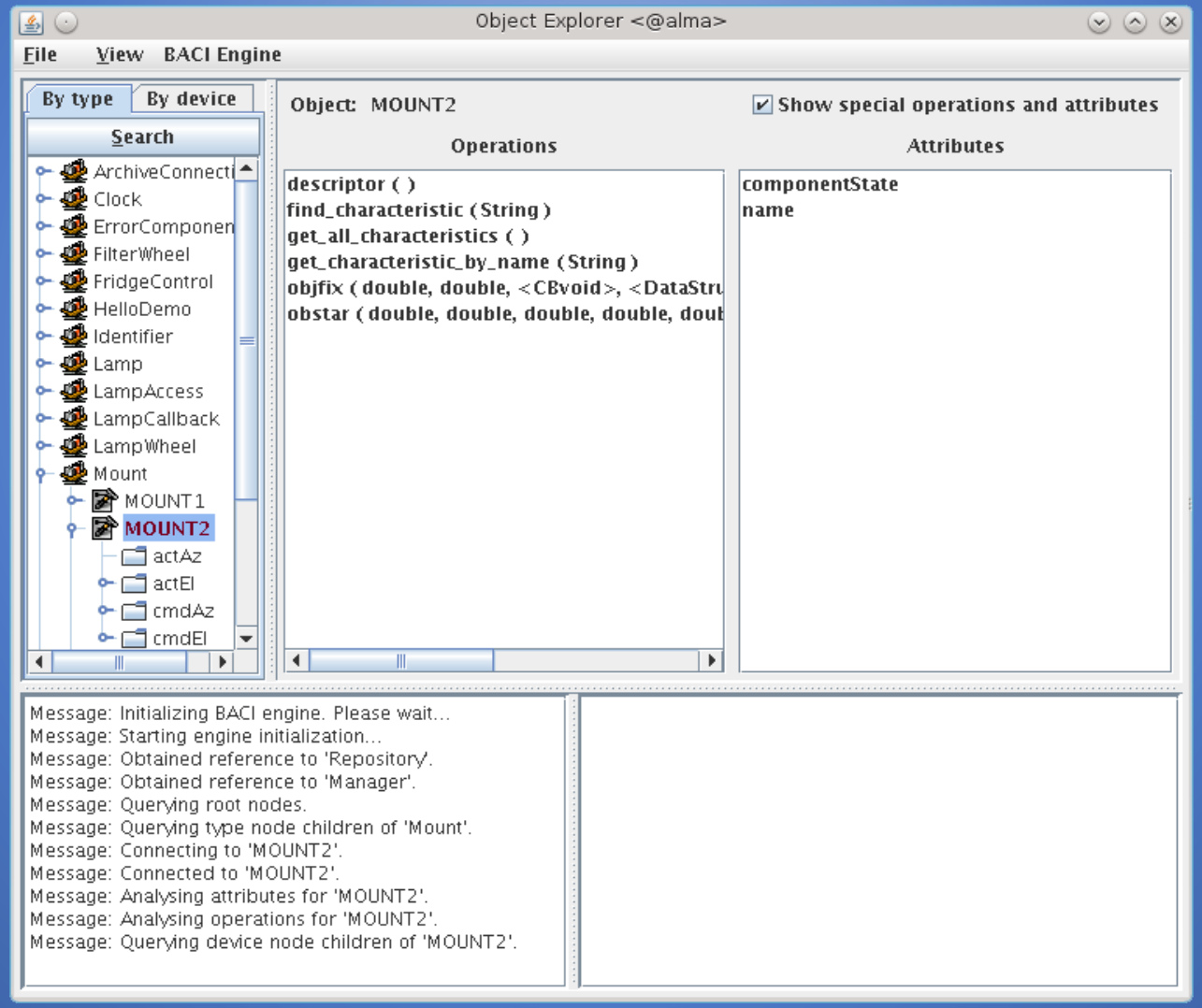
Component activation



Component client



Object explorer



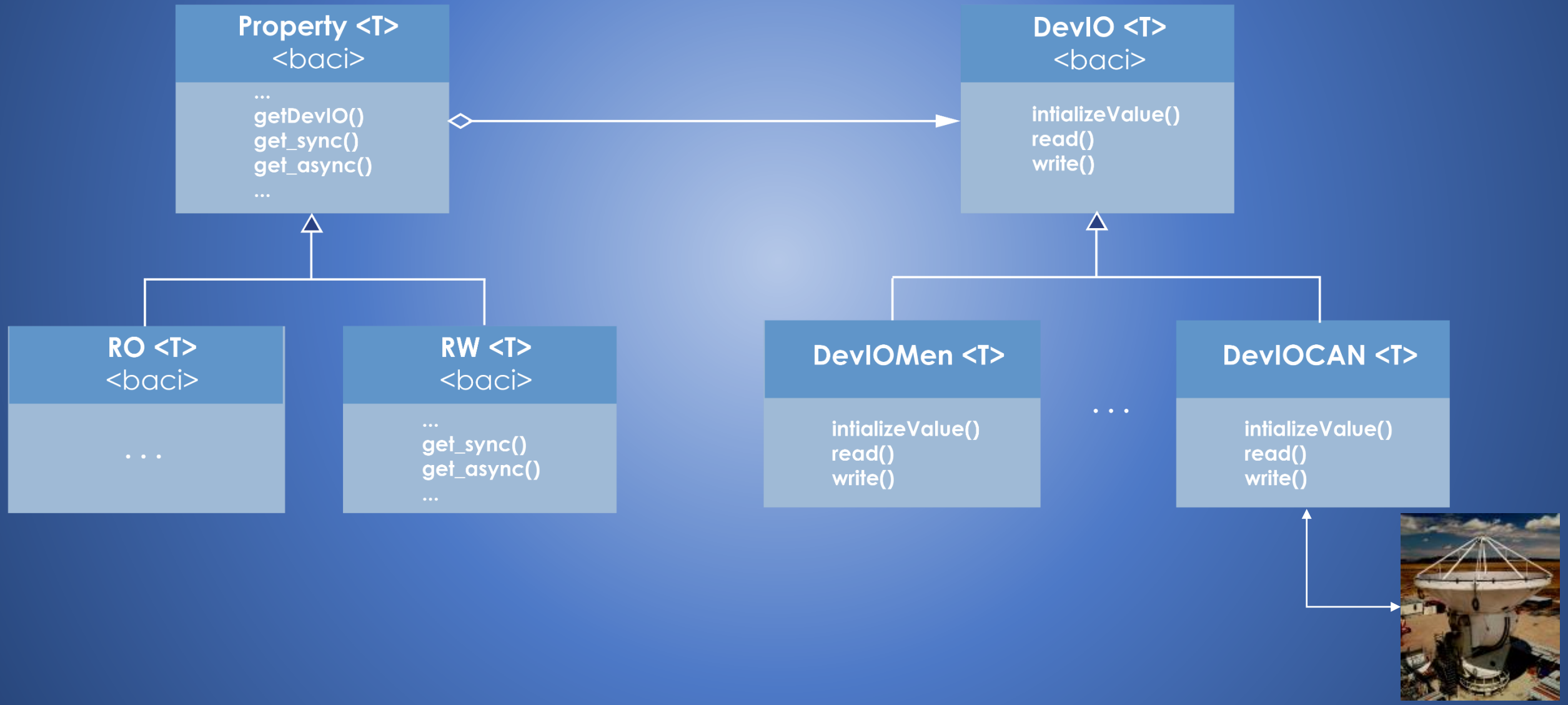
Characteristic component

- Executed within a container running on a given machine
  - Container spawns threads for component execution
- Follows a component lifecycle
- A Component is the natural base class for physical and logical "devices" (abstraction of hardware devices)
  - With properties (e.g. staus value, position – control/monitor points)
  - Characteristiscs i.e static data in the configuration database
    - units, default values, monitor*, alarm*, archive*

BACI property

- Statically defined item
- It has a typed value and attributes
  - Basi ctypes: double, long, string, pattern, enum, longSeq, …
- Read-only (RO) and read-write (RW) access
- Defines a interface, which is extended by developer
  - Developer implements functions read() and write() functions
- Combines value(s) with "attributes"
  - Description
  - Unit
  - Monitoring parameters
  - Alarms thresholds
- Value monitoring
  - Interval
  - On change
  - Keeps history (last 10 values)
- Value archiving
  - Same as for monitoring
- Alarms built-in

DevIO

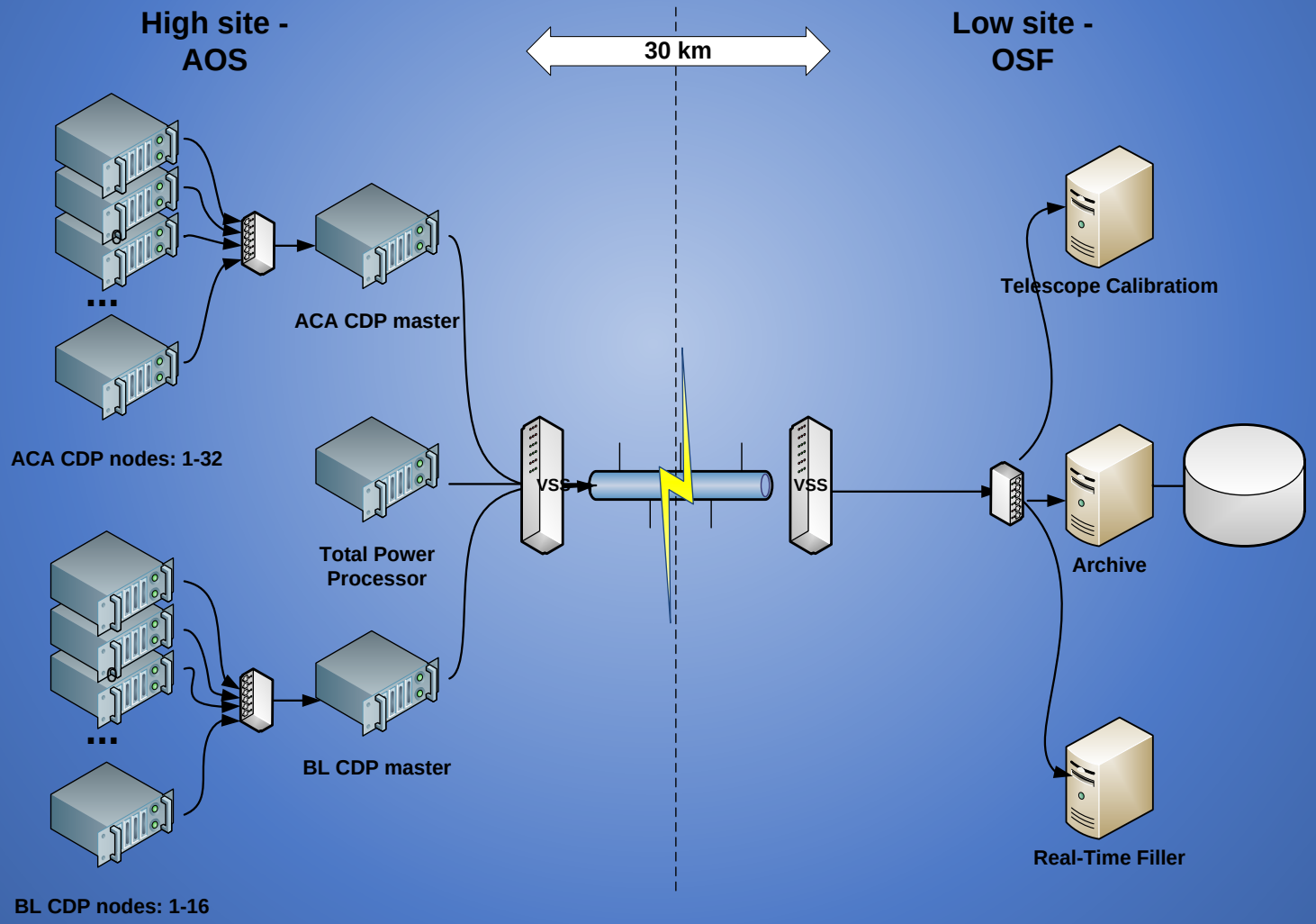

Property <T>
<baci>
...
getDevIO()
get_sync()
get_async()
...
DevIO <T>
<baci>
intializeValue()
read()
write()
RO <T>
<baci>
. . .
RW <T>
<baci>
...
get_sync()
get_async()
...
DevIOMen <T>
intializeValue()
read()
write()
. . .
DevIOCAN <T>
intializeValue()
read()
write()

Bulkdata

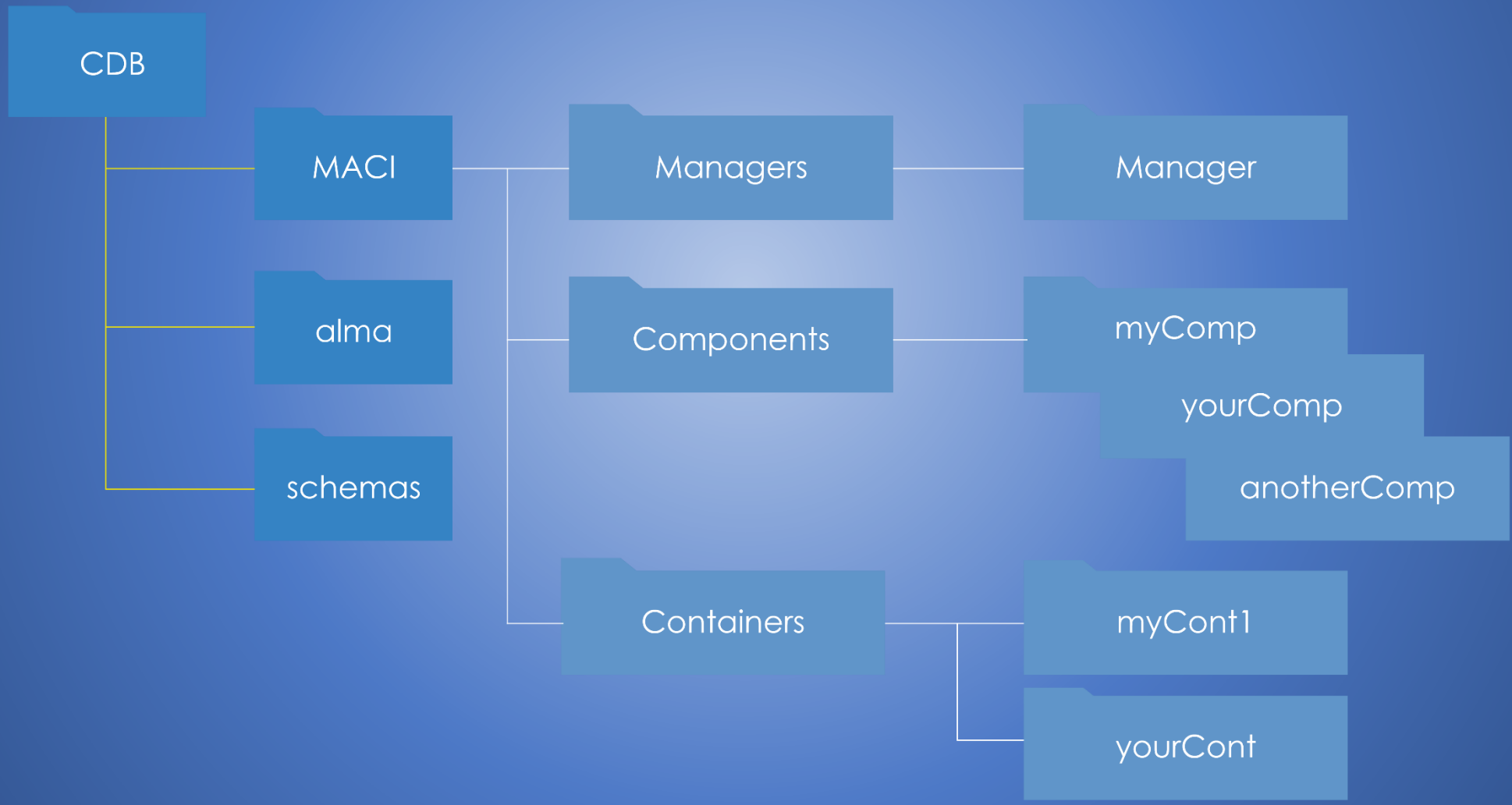
- ACS service for reliable and concurrent streaming of high volumes of (astronomical) data
- Used in two configurations:
  - Many senders to one receiver
  - One sender to many receivers – multicast

- Used by 6 ALMA SW sub-systems
- In operation since March 2013
- Total peak data rate: 64MBytes/sec

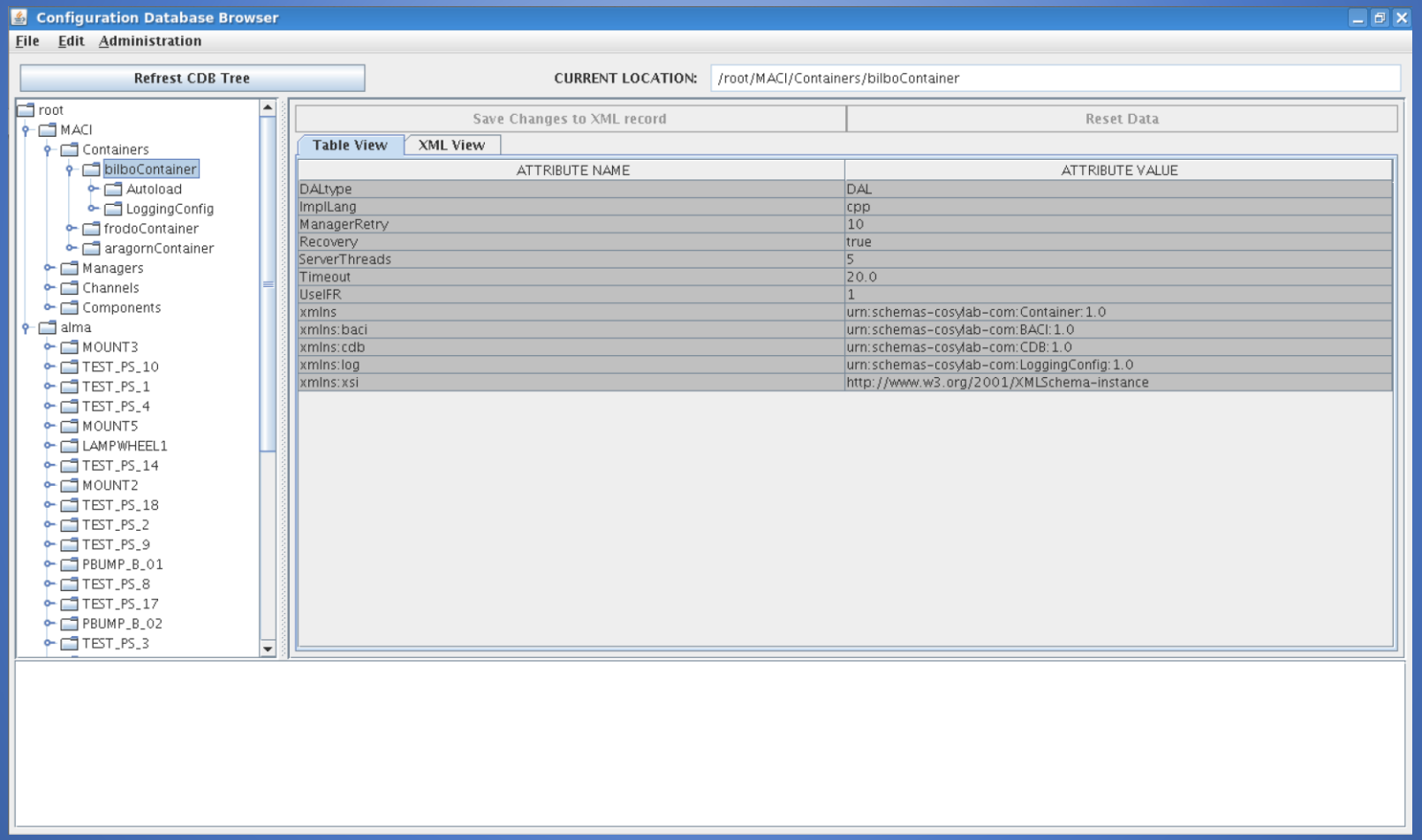
Bulkdata - 2


High site -
AOS
30 km
Low site -
OSF
ACA CDP master
ACA CDP nodes: 1-32
Total Power
Processor
VSS
VSS
Telescope Calibratiom
Archive
BL CDP master
BL CDP nodes: 1-16
Real-Time Filler

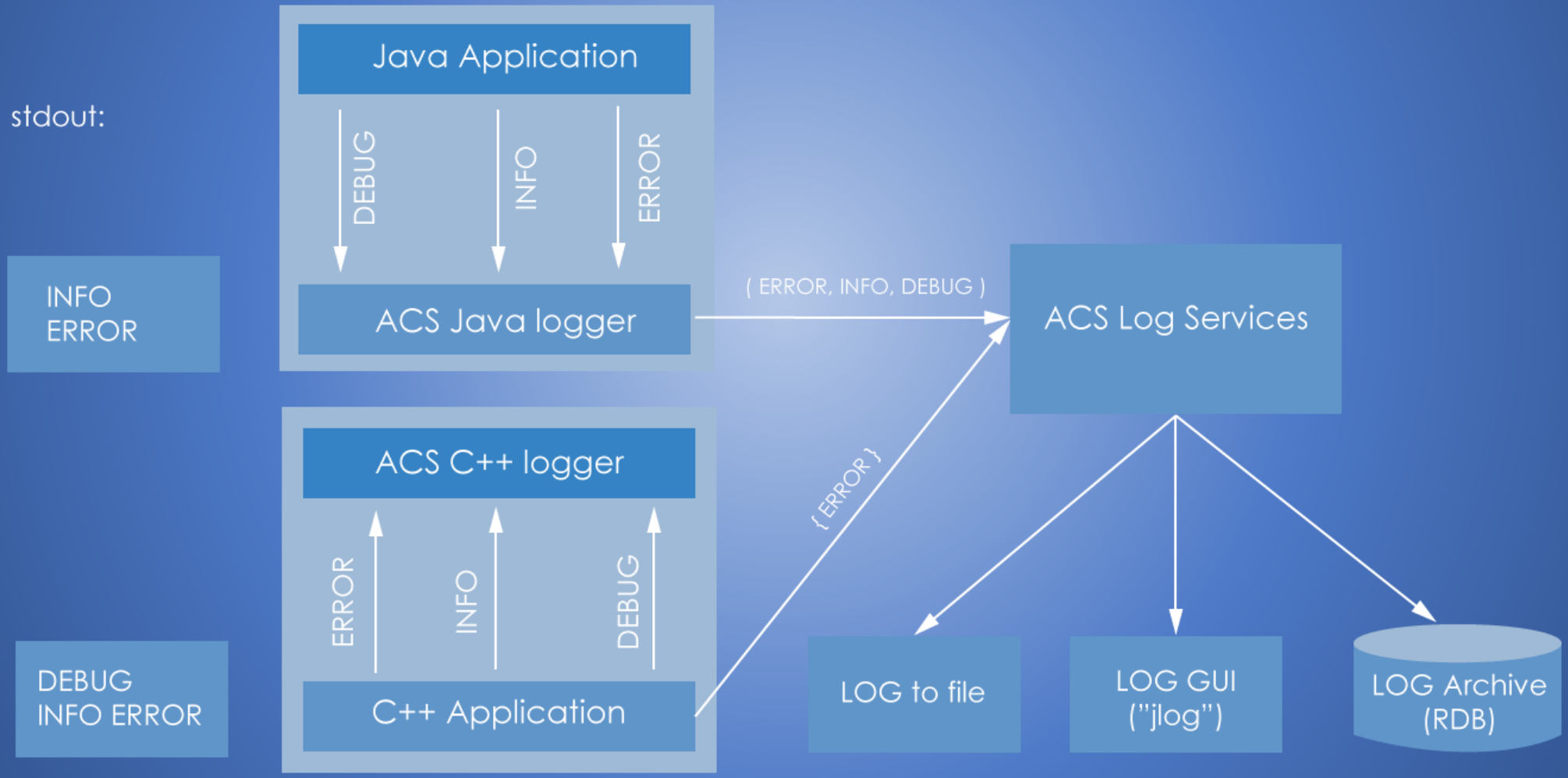
Configuration database (CDB)



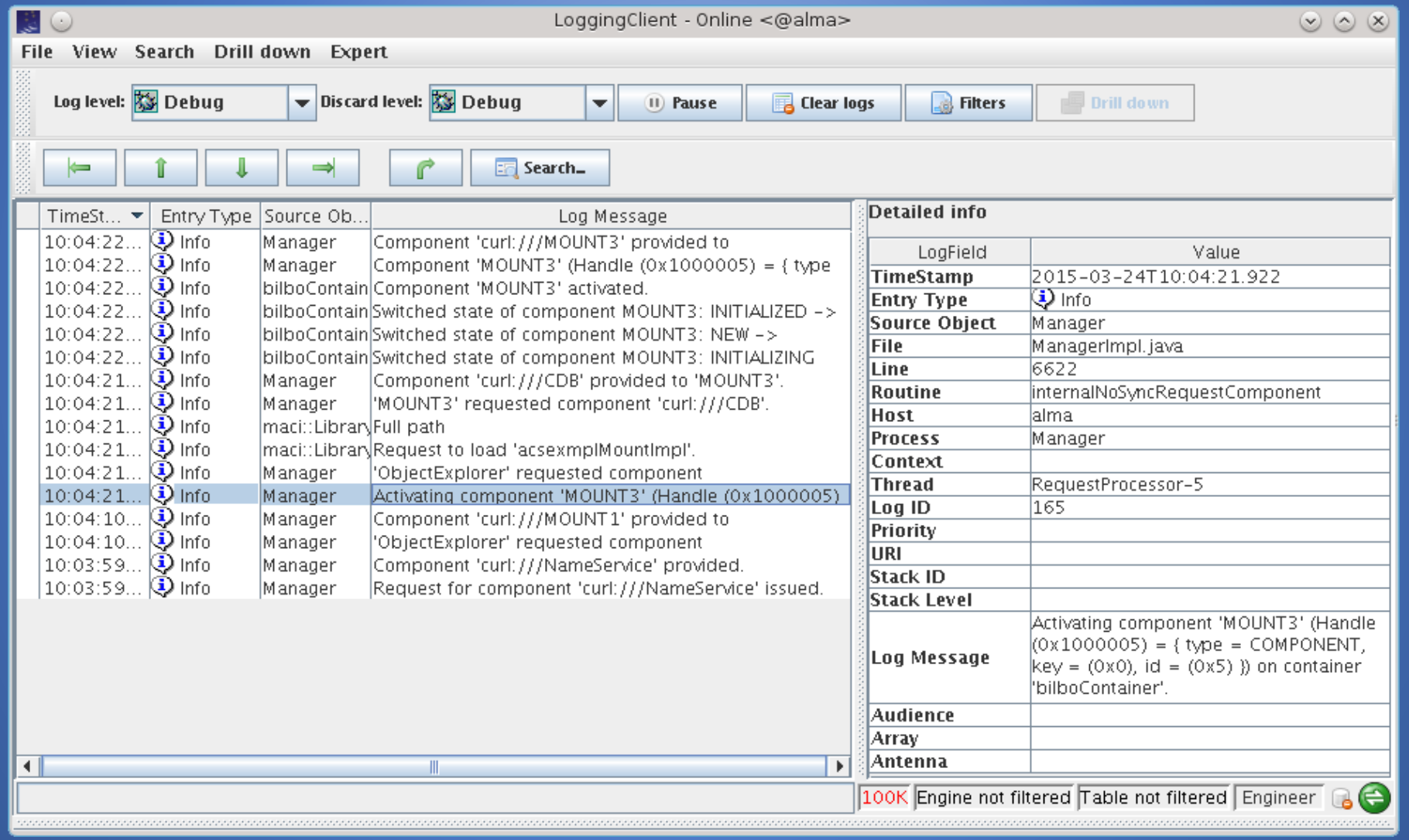
CDB browser


Configuration Database Browser
File
Edit
Administration
Refrest CDB Tree
CURRENT LOCATION:
/root/MACI/Containers/bilboContainer
root
MACI
Containers
bilboContainer
Autoload
LoggingConfig
frodoContainer
aragornContainer
Managers
Channels
Components
alma
MOUNT3
TEST_PS_10
TEST_PS_1
TEST_PS_4
MOUNT5
LAMPWHEEL1
TEST_PS_14
MOUNT2
TEST_PS_18
TEST_PS_2
TEST_PS_9
PBUMP_B_01
TEST_PS_8
TEST_PS_17
PBUMP_B_02
TEST_PS_3
Save Changes to XML record
Reset Data
Table View
XML View
ATTRIBUTE NAME
ATTRIBUTE VALUE
DALtype
DAL
ImplLang
cpp
ManagerRetry
10
Recovery
true
ServerThreads
5
Timeout
20.0
UseIFR
1
xmlns
urn:schemas-cosylab-com:Container:1.0
xmlns:baci
urn:schemas-cosylab-com:BACI:1.0
xmlns:cdb
urn:schemas-cosylab-com:CDB:1.0
xmlns:log
urn:schemas-cosylab-com:LoggingConfig:1.0
xmlns:xsi
http://www.w3.org/2001/XMLSchema-instance

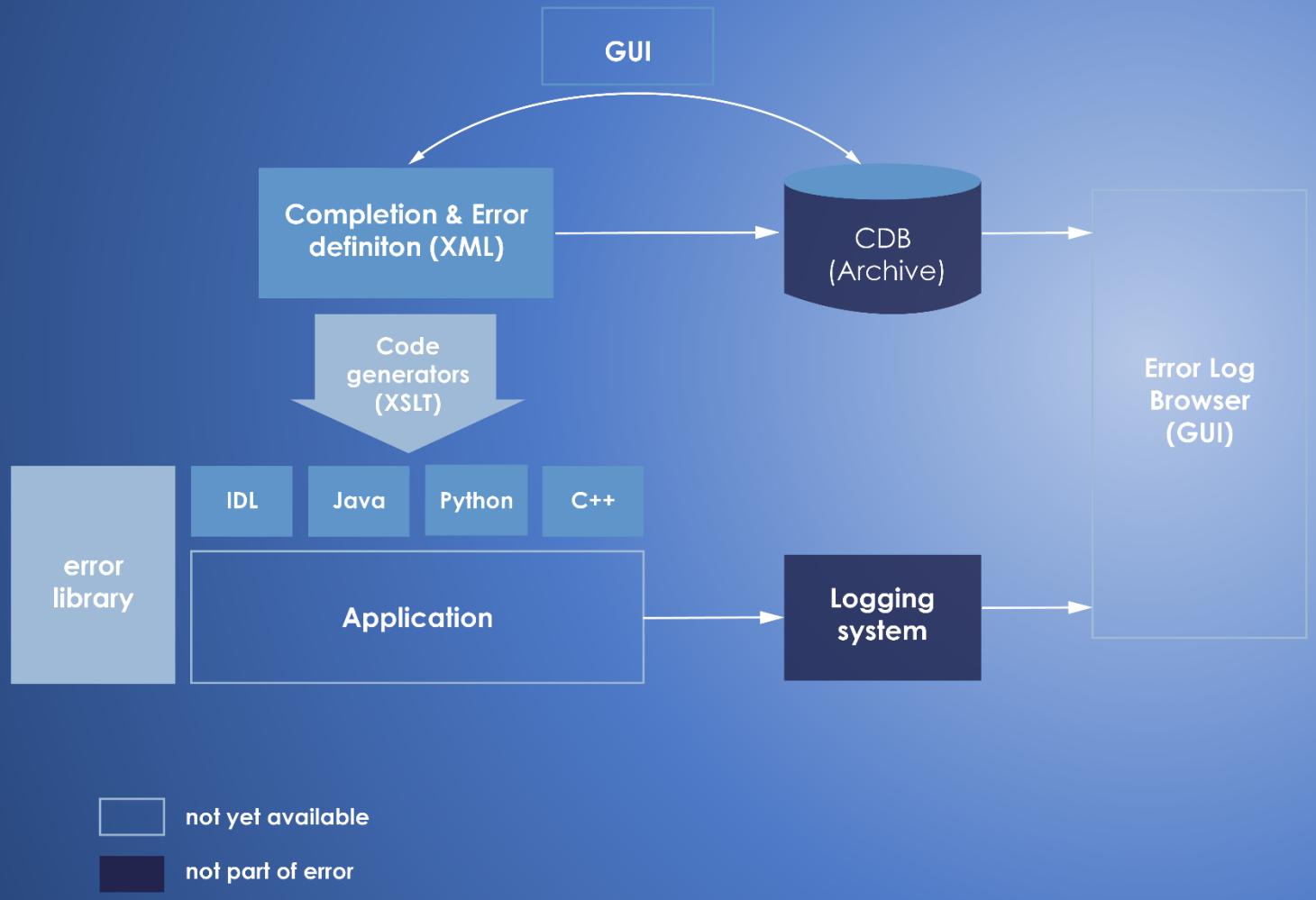
Logging system



Logging client (jlog)



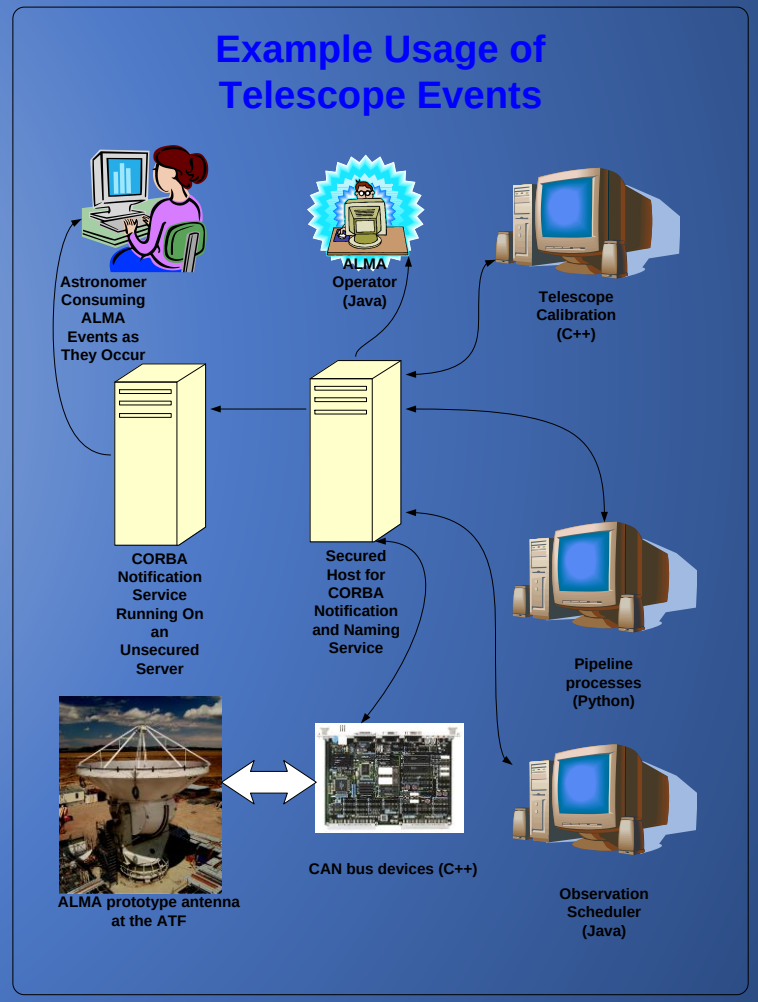
Error system



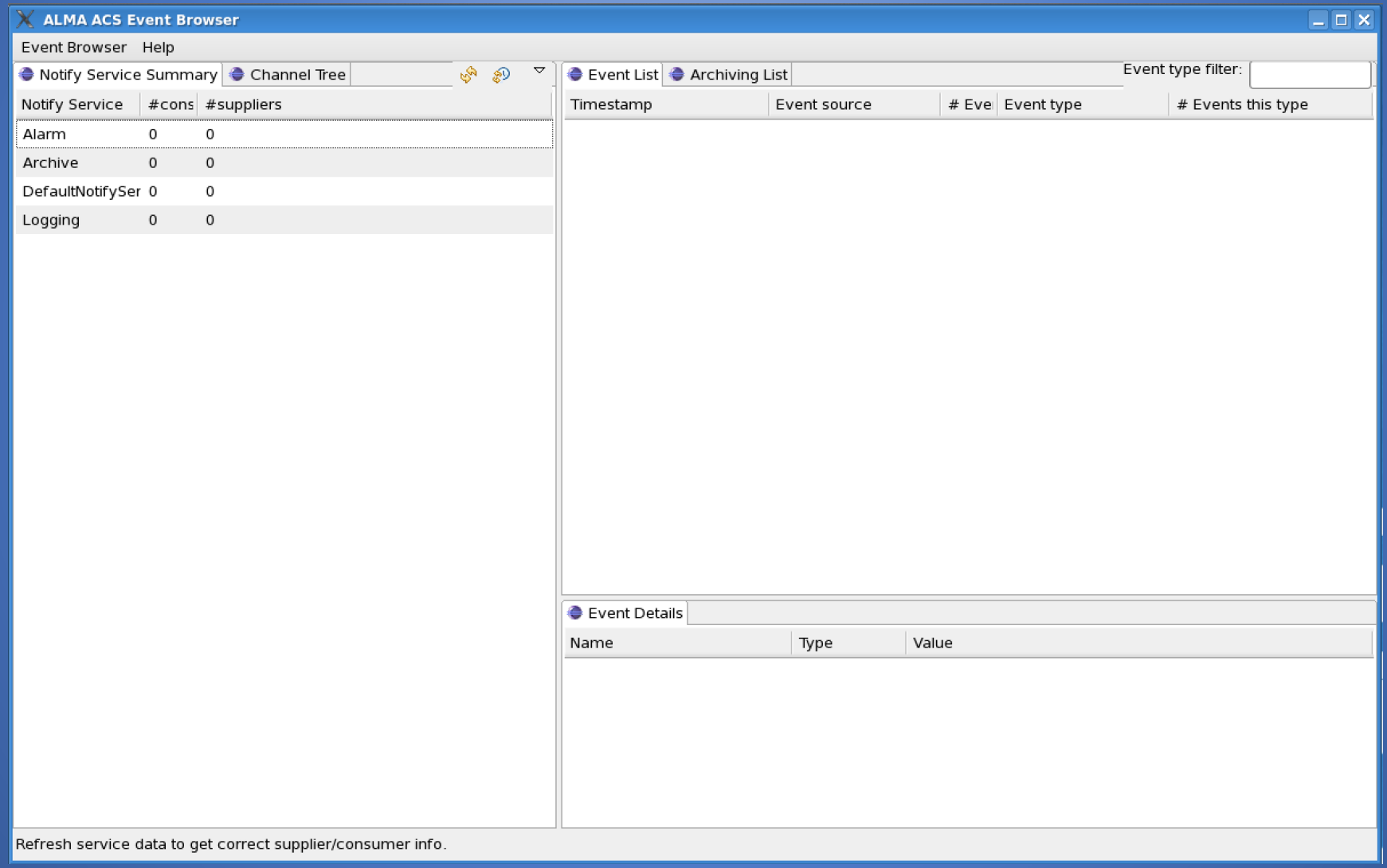

```
ErrorTrace (TimeStamp=Thu Oct 31 20:45:04 2013,
            FileDelayCal.py,
            Line=579,
            Routine=<module>,
            Host=gns,
            Process=14355,
            Thread=MainThread,
            Type=10,    Code=3,
            ShortDescrip=Unknown Error,
            Severity=Error,
            Data: )
     ErrorTrace (TimeStamp=Thu Oct 31 20:45:04 2013,
                 File=ArrayMountControllerImpl.java,
                 Line=1987,
                 Routine=throwIfIllegalParameterError,
                 Host=gas01,
                 Process=CONTROL/ACC/javaContainer,
                 Thread=RequestProcessor-177,
                 Type=10000, Codee=2,
                 ShortDescrip=Illegal Parameter Error,
                 Severity=Error,
                 Data: Name=DV02,
                 …
```

Events

- Events distributed by means of Notification Channels
- NCs are an alternative to direct “Request/Reply” calls.
- NCs decouple the communicating partners
- NCs can protect the sender from slow receivers
- Notification Channels runs inside CORBA Notify Services
- Publisher/Subscriber mechanism
  - ACS handle CORBA details of NCs
- Use of NCs makes debugging the system more difficult.
- Experimental NC over DDS



Event browser



Alarm System

The Alarm System is a messaging system that deals with abnormal situations by means of Fault States (FS):

- FS collection
- FS analysis and distribution (reduction rules)
- Alarm definition
- Alarm archiving

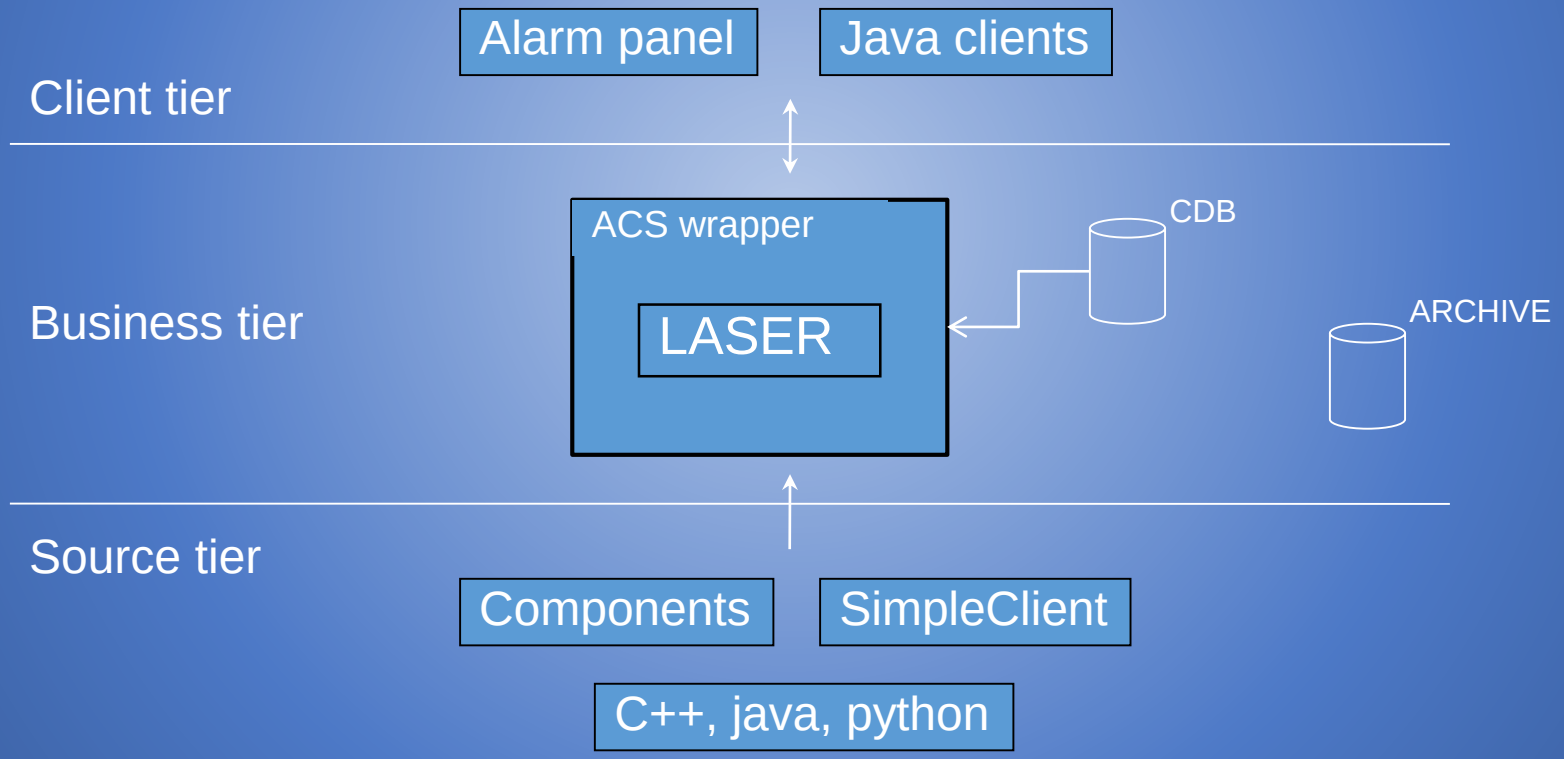
ACS comes with 2 implementations:

- **ACS** (default)
- **CERN** (explicitly set in the CDB)

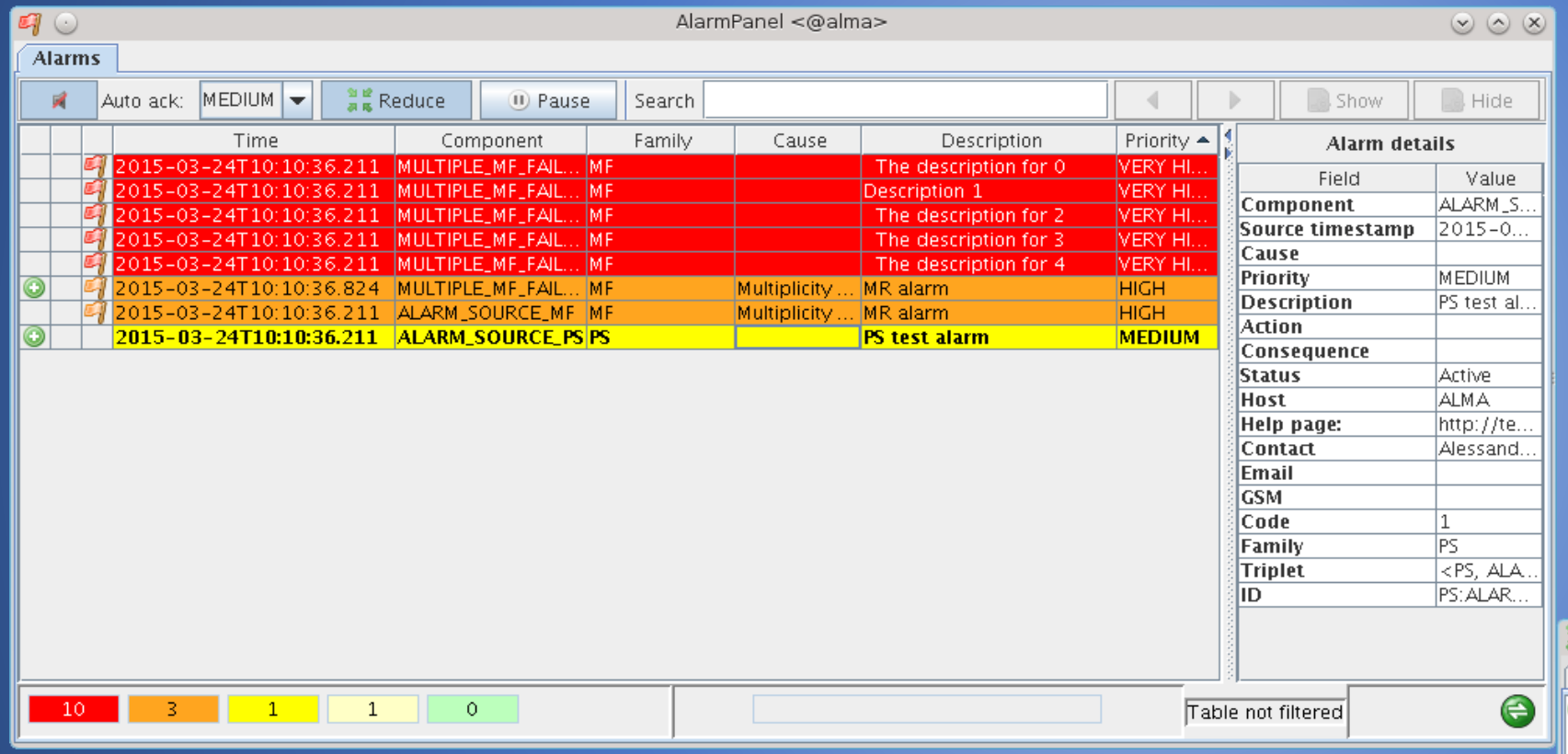
Alarms

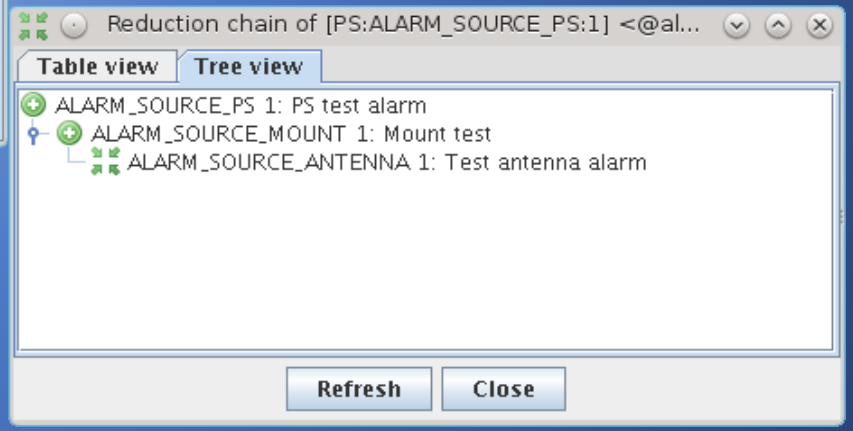
- 4 alarm levels (low, medium, high, critical)
- ACS generates alarms from BACI properties
- 2 type of reduction rules
  - NODE
  - MULTIPLICITY
- API is very easy, just one line of code

CERN Alarm System

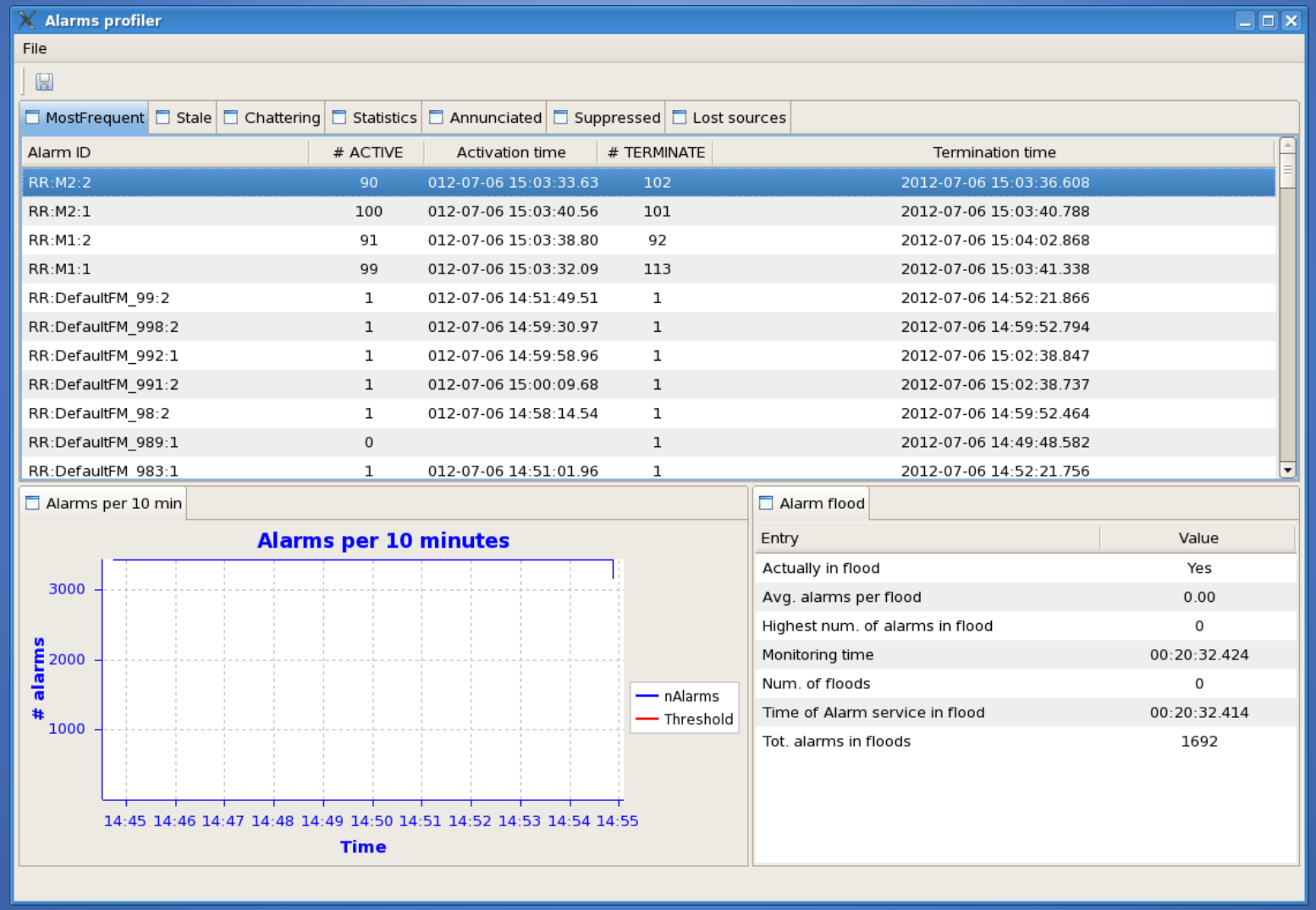


Alarm panel


AlarmPanel <@alma>
Alarms
Auto ack: MEDIUM
Reduce
Pause
Search
Show
Hide
Time
Component
Family
Cause
Description
Priority
2015-03-24T10:10:36.211 MULTIPLE_MF_FAIL... MF The description for 0 VERY HI...
2015-03-24T10:10:36.211 MULTIPLE_MF_FAIL... MF Description 1 VERY HI...
2015-03-24T10:10:36.211 MULTIPLE_MF_FAIL... MF The description for 2 VERY HI...
2015-03-24T10:10:36.211 MULTIPLE_MF_FAIL... MF The description for 3 VERY HI...
2015-03-24T10:10:36.211 MULTIPLE_MF_FAIL... MF The description for 4 VERY HI...
2015-03-24T10:10:36.824 MULTIPLE_MF_FAIL... MF Multiplicity ... MR alarm HIGH
2015-03-24T10:10:36.211 ALARM_SOURCE_MF MF Multiplicity ... MR alarm HIGH
2015-03-24T10:10:36.211 ALARM_SOURCE_PS PS PS test alarm MEDIUM
Alarm details
Field
Value
Component ALARM_S...
Source timestamp 2015-0...
Cause
Priority MEDIUM
Description PS test al...
Action
Consequence
Status Active
Host ALMA
Help page: http://te...
Contact Alessand...
Email
GSM
Code 1
Family PS
Triplet <PS, ALA...
ID PS:ALAR...
10
3
1
1
0
Table not filtered


Reduction chain of [PS:ALARM_SOURCE_PS:1] <@al...
Table view
Tree view
ALARM_SOURCE_PS 1: PS test alarm
ALARM_SOURCE_MOUNT 1: Mount test
ALARM_SOURCE_ANTENNA 1: Test antenna alarm
Refresh
Close

Alarm profiler

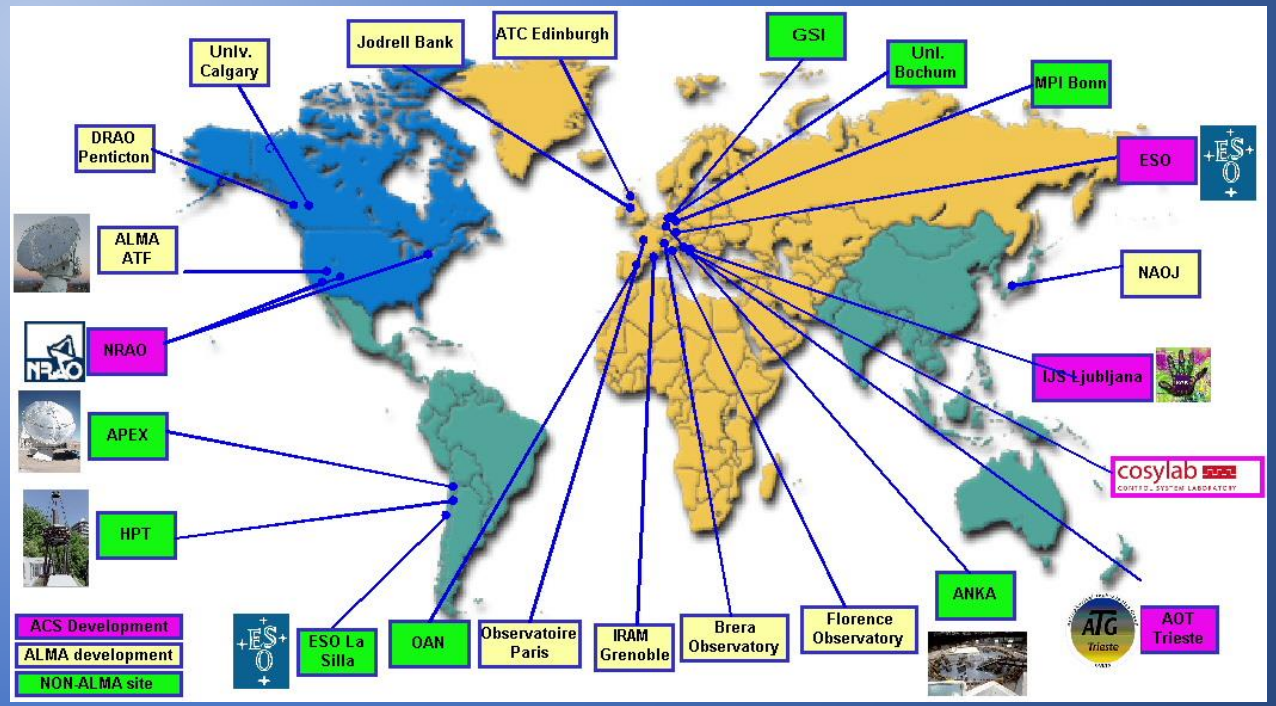


Releases

- Incremental releases (~4 releases/year)
- Feature complete
- Improving robustness
- Tools to help debugging
- Open to community after testing at the OSF

ACS outside of ALMA

- APEX
- Cherenkov Telescope Array (CTA)
- Large Latin America Millimiter Array (LLAMA)
- Radiotelescope IGN Yebes
- Sardinia Radio Telescope (SRT)
- Sparta@ESO



Conclusion

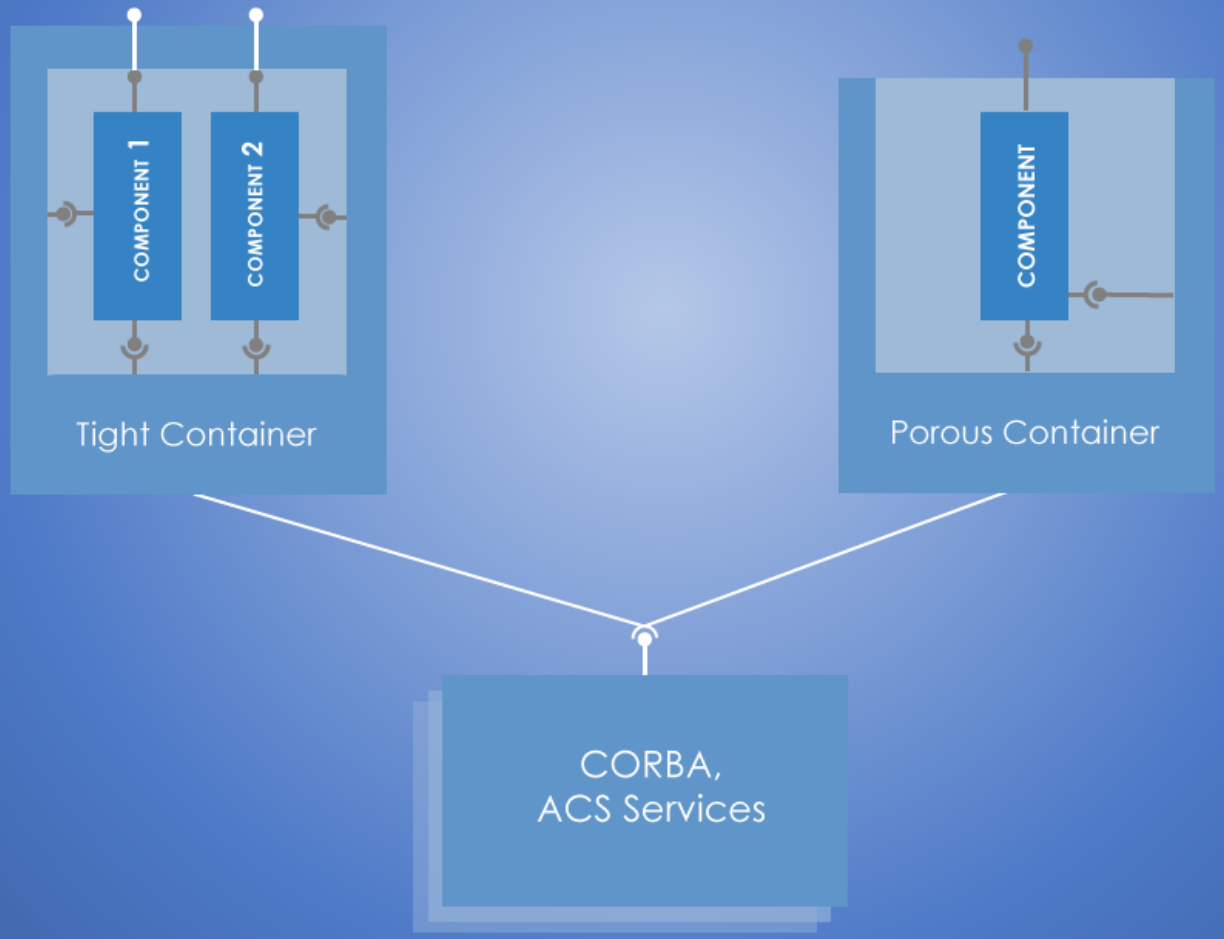
- Cons
  - Monolitic
  - Steep learning curve
  - Not yet complete
  - Slow evolving

- Pros
  - ACS is used in ALMA operations
  - Other telescopes uses ACS as well
  - Growing community (ACS@github)
  - C++, python and java (other languages possible)

Questions?

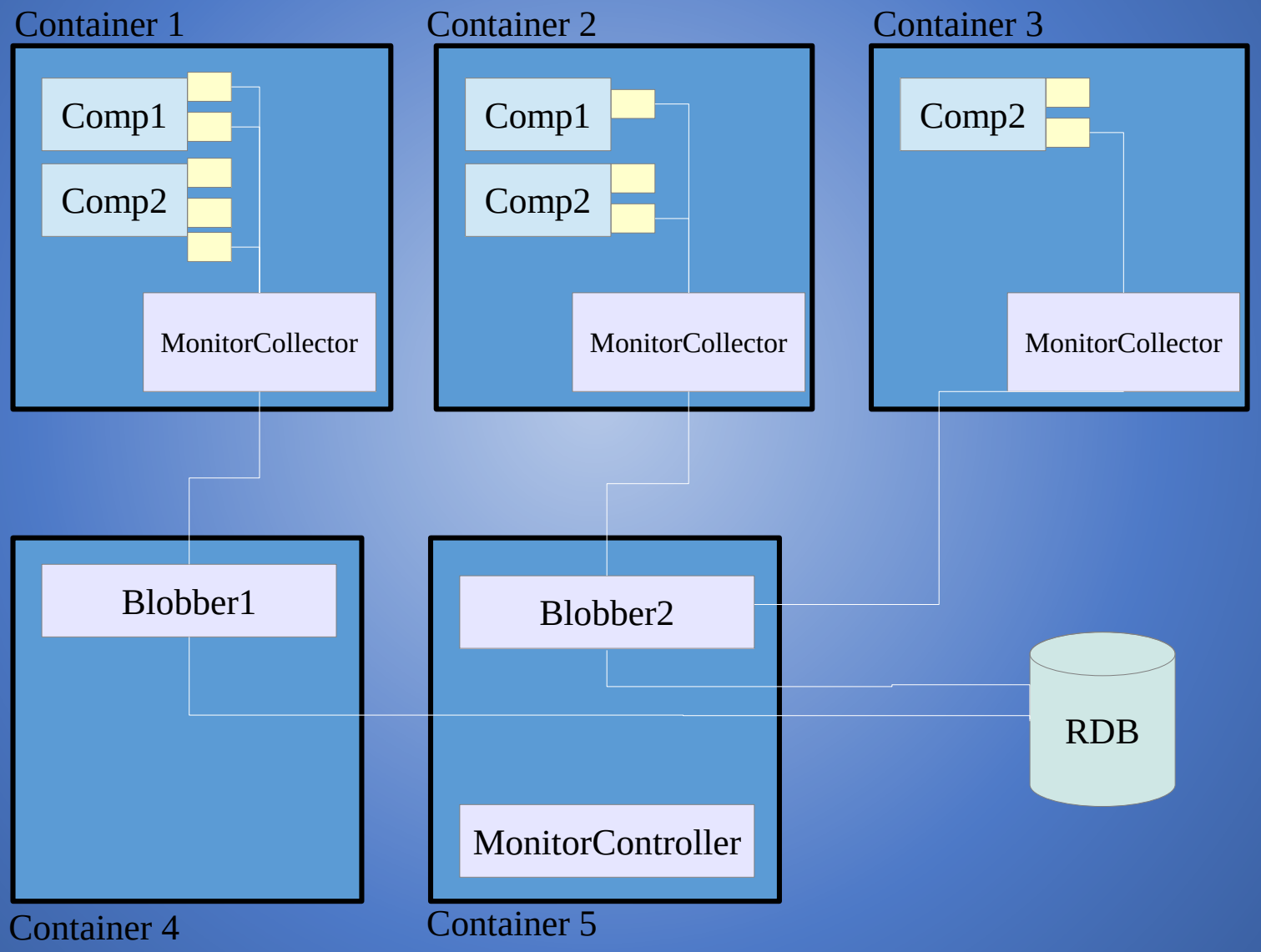
Tight and porous interfaces

Functional interface is intercepted by the container for logging and/or exception handling, security, …



Container manages lifecycle and offers services, but exposes the component's functional interface directly – less overhead

Monitoring - 1



Monitoring - 2