



# TMCDB

## ALMA Telescope Monitor and Configuration DB

ISS013E54329

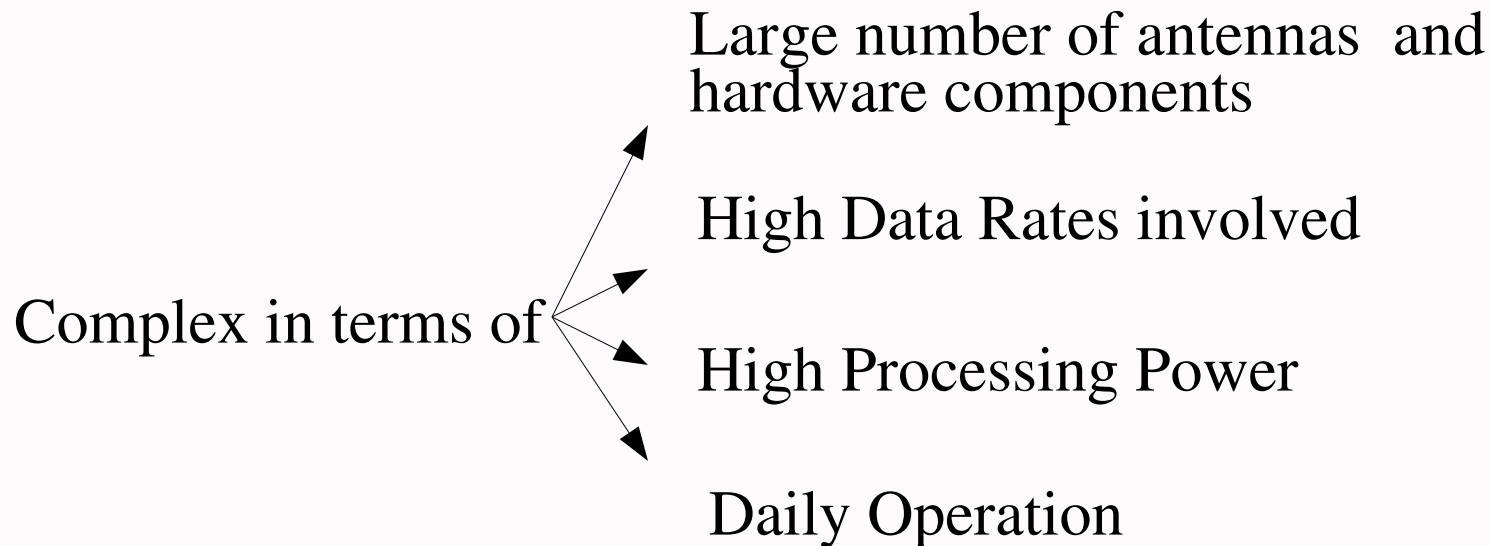


Pablo Burgos  
ALMA

# Motivation

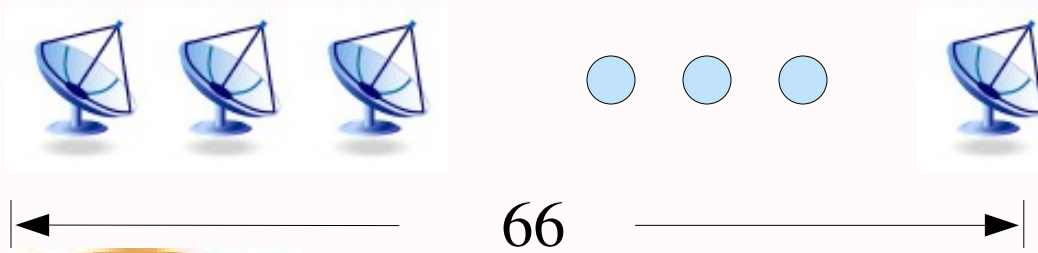


ALMA will be the most complex telescope that has ever been built by human kind



... for naming a few ...

# In terms of software operation ALMA is complex too...



Static Components: FLOOG, LORR, LO2 x4, PSA, PSD etc = 40 components per antenna

2640 static components whole Array

Components from other subsystems ARCHIVE, Correlator, TELCAL etc

Dynamic components

**+** More than 100 containers

---

More than 3000 entities that needs Configuration Monitoring, Startup and Deployment info



# TMCDB has been design to be the answer to this problem



Main TMDCB Design Goal: Simplify ALMA Software Operation

By means of

Providing interfaces to persist/store/retrieve data needed for configuring, initializing, monitoring and deploying of software modules for telescope operation, tracking this information over lifetime of ALMA project





... a maintenance database

Btw, maintenance databases are meant to:

Keep Maintenance schedules and its  
history

Part and suppliers

Vendors information

# Objetives



To understand the motivation for TMCDB design and the driving forces shaping its development

To understand the TMCDB role, and how its design fits on the overall ALMA system

Have a grasp on the current architecture design

To realize about current development status

To know how to activate and play at home with it :)



## Schedule

TMCDB: It's Layered Architecture

Activating  
TMCDB

Ongoing TMCDB  
Development

Conclusion

S



## Schedule

### TMCDB: It's Layered Architecture

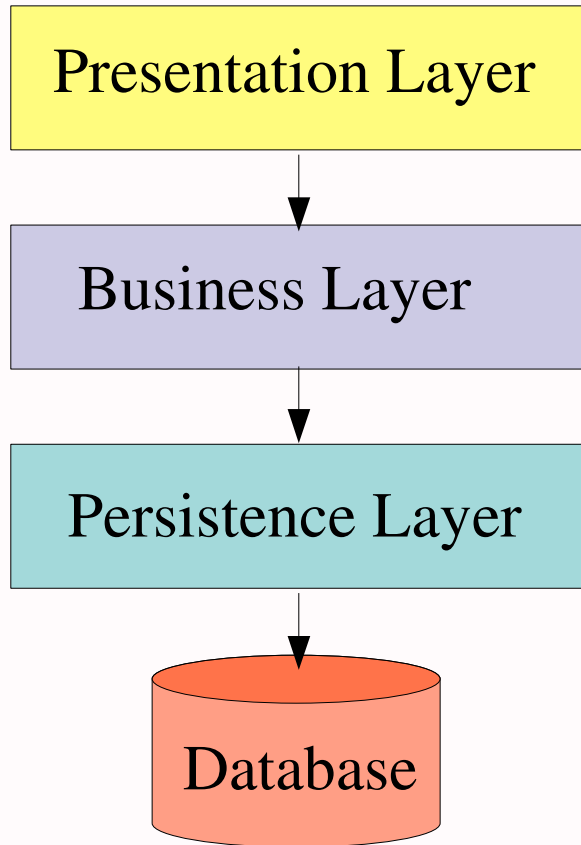
Activating  
TMCDB

Ongoing TMCDB  
Development

Conclusion

S

# TMCDB Layered Architecture



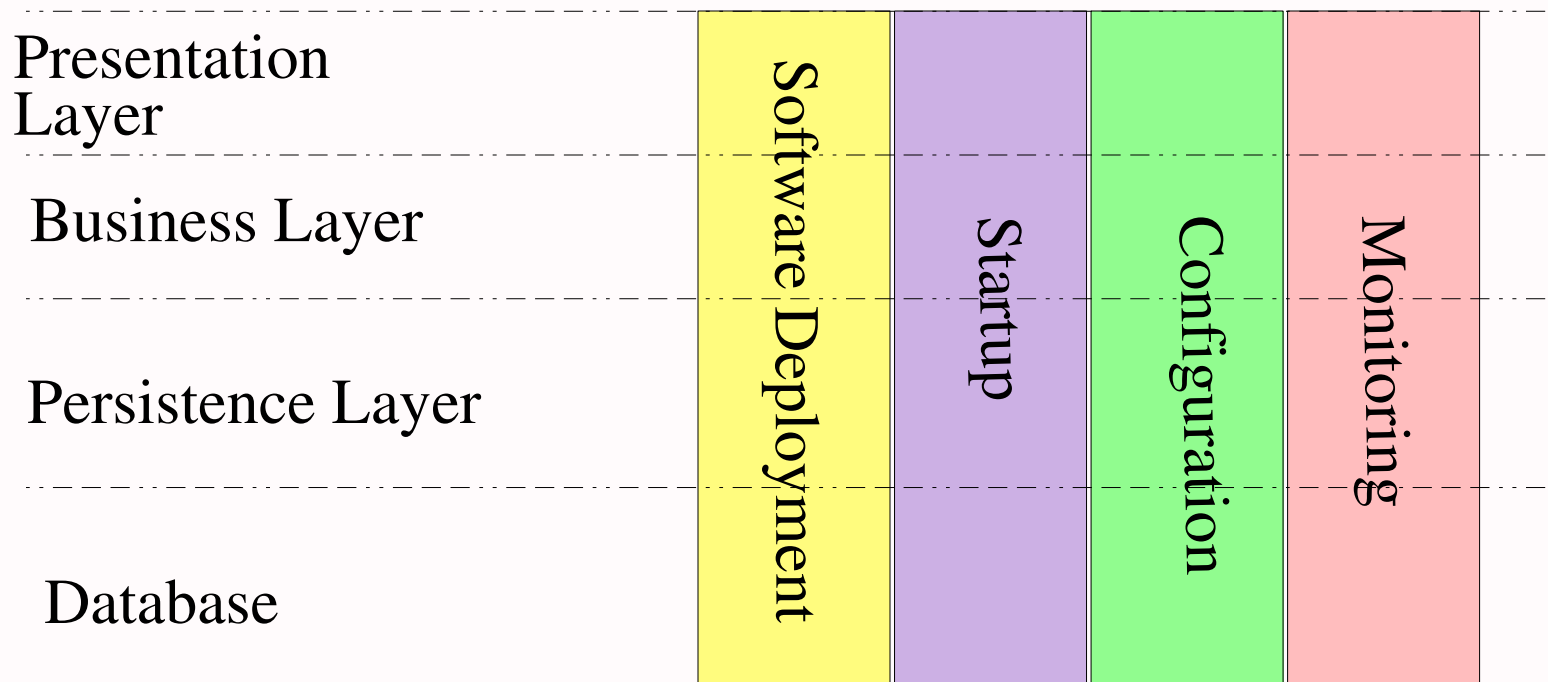
User Interface Logic. Collected data is only as useful as the analysis that can be done on it

Domain Model Logic

Responsible for storing/retrieving data from data source

Persistent representation of the system state

# TMCDB Sections





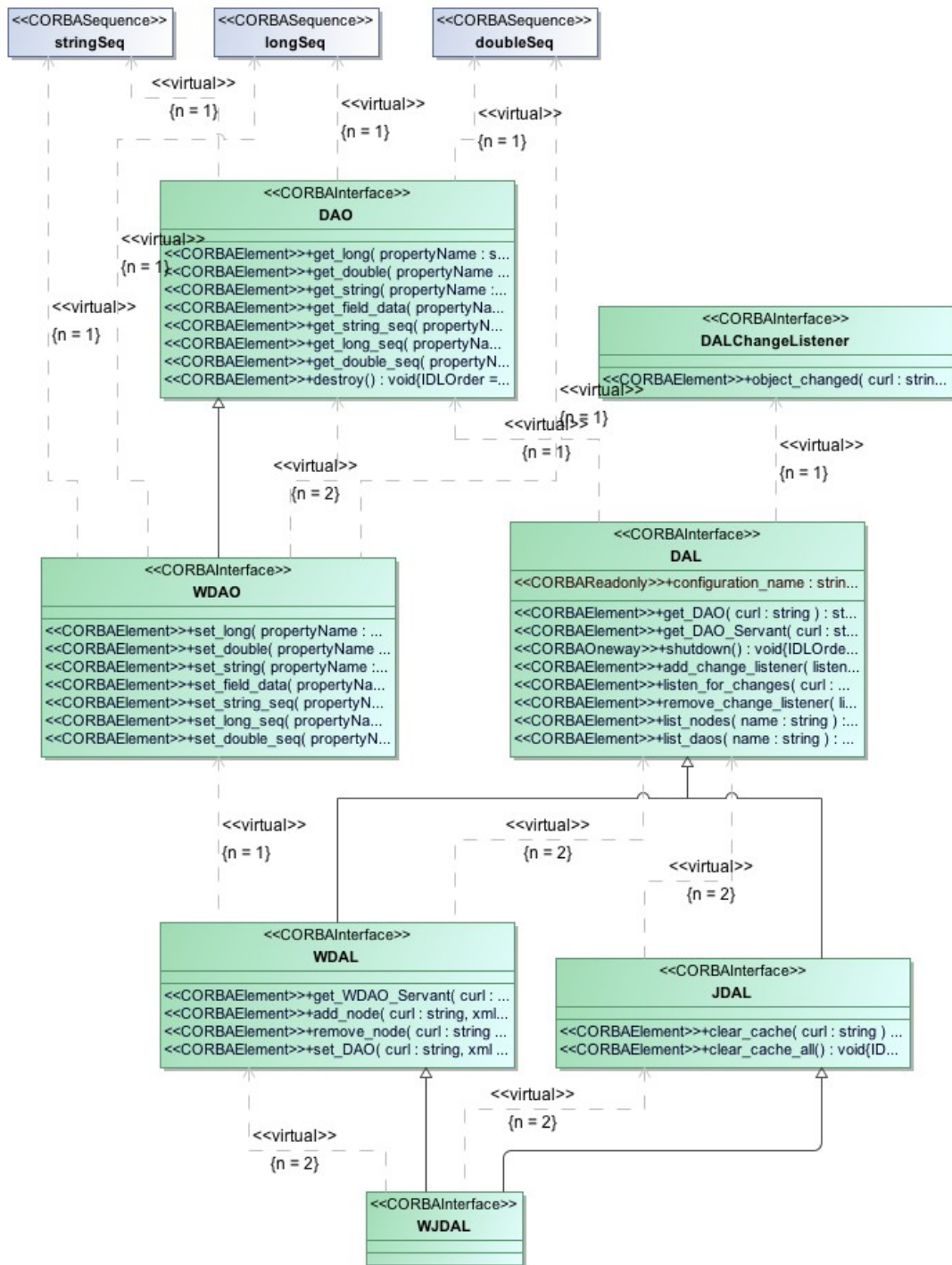
Software Deployment Section: information about properties of Computers, Containers that run inside Computers and associated Components

# SW Deployment: How it works? (1)

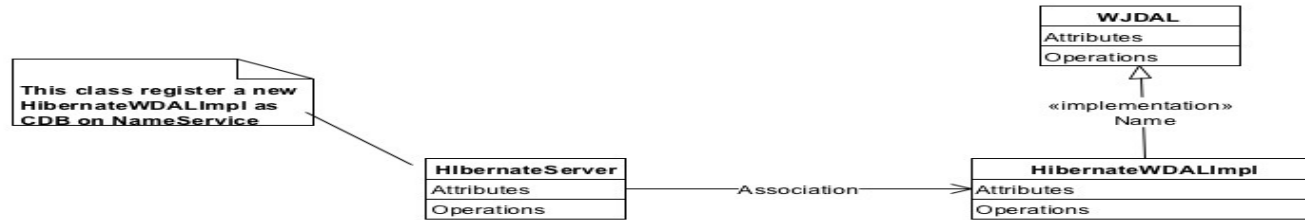


HibernateCDBJDaI is the new Servant that incarnates WJDAL

ALMA system never knows what is the CDB behind the scenes (classicCDB or HibernateCDB :) )



# SW Deployment: How it works? (3)



# SW Deployment: How It works?(4)



Name	Kind	Type	Host	Port
ArchiveNotifyE...		IDL:NotifyExt/EventChannelFactory...	192...	3006
LoggingChannel		IDL:omg.org/CosNotifyChannelAdmi...	192...	3005
Log		IDL:omg.org/DsLogAdmin:BasicLog:...	192...	3003
LogFactory		IDL:omg.org/DsLogAdmin:BasicLog:...	192...	3003
Manager		IDL:ijs.si/maci/Manager: 1.0	192...	3000
ArchivingChannel		IDL:omg.org/CosNotifyChannelAdmi...	192...	3006
NotifyEventCha...		IDL:NotifyExt/EventChannelFactory...	192...	3002
InterfaceRepos...		IDL:omg.org/CORBA/Repository.1.0	192...	3004
AlarmNotifyEv...		IDL:NotifyExt/EventChannelFactory...	192...	3007
ACSLogSvc		IDL:alma/ACSLog/LogSvc: 1.0	192...	3011
LoggingNotifyE...		IDL:sandia.gov/NotifyMonitoringExt...	192...	3005
CDB		IDL:cosylab.com/CDB/WJDAL: 1.0	192...	3012
TAO_MonitorA...		IDL:sandia.gov/CosNotification/Noti...	192...	353...

# Configuration Section



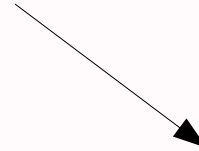
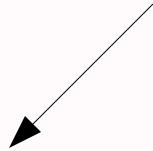
Handles information about Base Elements and which of them are online (BaseElementOnline)

A BaseElement is one set of high level structures that makes up the ALMA Telescope. They may contain assemblies, arranged in groups

Base Elements examples: Antenna, Pad, FrontEnd, Weather Station, Correlator, Central Rack, Holography Tower and Array



Startup Section: Houses all data needed to start the entire ALMA Software System



Software Deployment  
Startup List

Computers expected to be operational and online  
ACS version to be run  
Containers to start and computers on which to run them  
Components that are started automatically and containers in which to run them  
Dependencies

BaseElement Startup List

Base Elements to be started  
Example: Antennas and for each antenna:

The pad it is on  
Which FrontEnd the antenna has installed  
Antenna pointing model

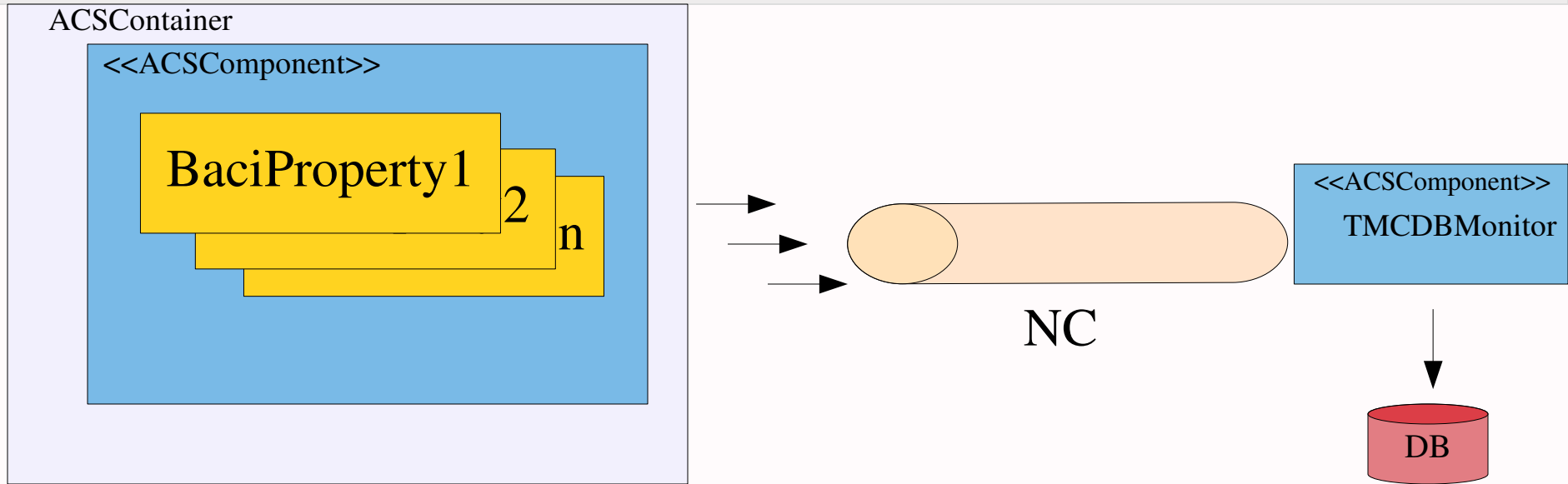
# Configuration and Startup: How It works? (1)



# Configuration and Startup: How it works? (2)



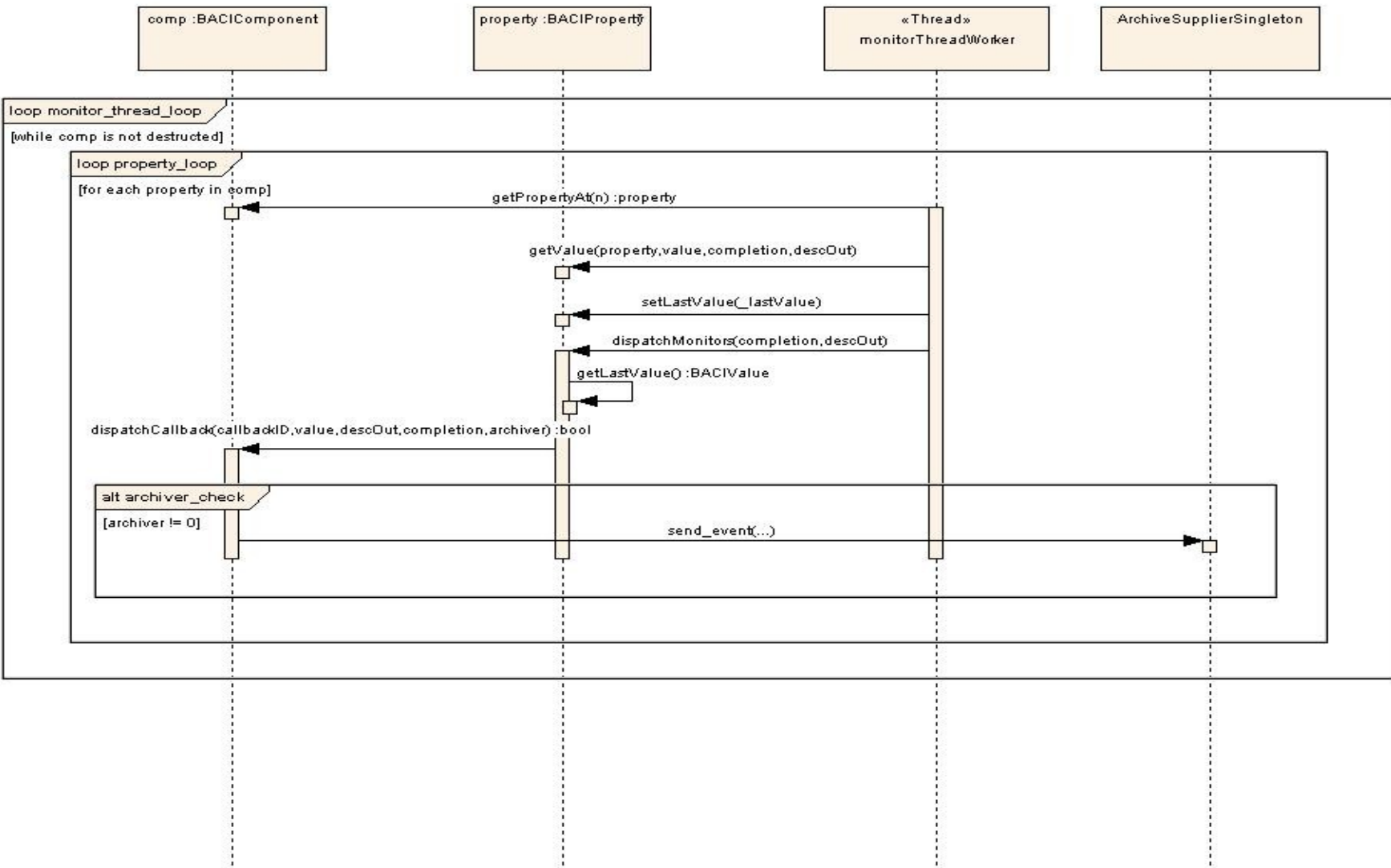
# Monitoring



Supplier

Consumer

# Monitoring. How it works?



# Monitoring: Presentation Layer



Monitor Database Query

Location	Device	Property
DA41	DGCK	AC_STATUS_OK
DM02	DRXBBpr0	ALARM_STATUS_REG_B
PM02	DRXBBpr1	ALARM_STATUS_REG_B_ALMINT
	DRXBBpr2	ALARM_STATUS_REG_B_LOCKERR
	DRXBBpr3	ALARM_STATUS_REG_B_POWALM
	DTXBBpr0	ALARM_STATUS_REG_B_SIGALM
	DTXBBpr1	ALARM_STATUS_REG_C
	DTXBBpr2	ALARM_STATUS_REG_C_ALMINT
	DTXBBpr3	ALARM_STATUS_REG_C_LOCKERR
	FLOOG	ALARM_STATUS_REG_C_POWALM
	FrontEnd/ColdCart3	ALARM_STATUS_REG_C_SIGALM
	FrontEnd/ColdCart6	ALARM_STATUS_REG_D
	FrontEnd/ColdCart7	ALARM_STATUS_REG_D_ALMINT
	FrontEnd/ColdCart9	ALARM_STATUS_REG_D_LOCKERR
	FrontEnd/Cryostat	ALARM_STATUS_REG_D_POWALM
	FrontEnd/IFSwitch	ALARM_STATUS_REG_D_SIGALM
	FrontEnd/LPR	ALIVE_COUNTER
	FrontEnd/PowerDist3	ALMA_TIME
	FrontEnd/PowerDist6	AMBIENT_TEMPERATURE
	FrontEnd/PowerDist7	AMB_CMD_COUNTER
	FrontEnd/PowerDist9	AMB_INVALID_CMD
	FrontEnd/WCA3	ANALOG_5_GOOD
	FrontEnd/WCA6	ANALOG_5_VOLTAGES
	FrontEnd/WCA7	APP_FULL

Properties to Query

Add Query  
Remove Query

Start: yyyy-mm-dd hh:mm:ss  
2008-11-19 20:08:59

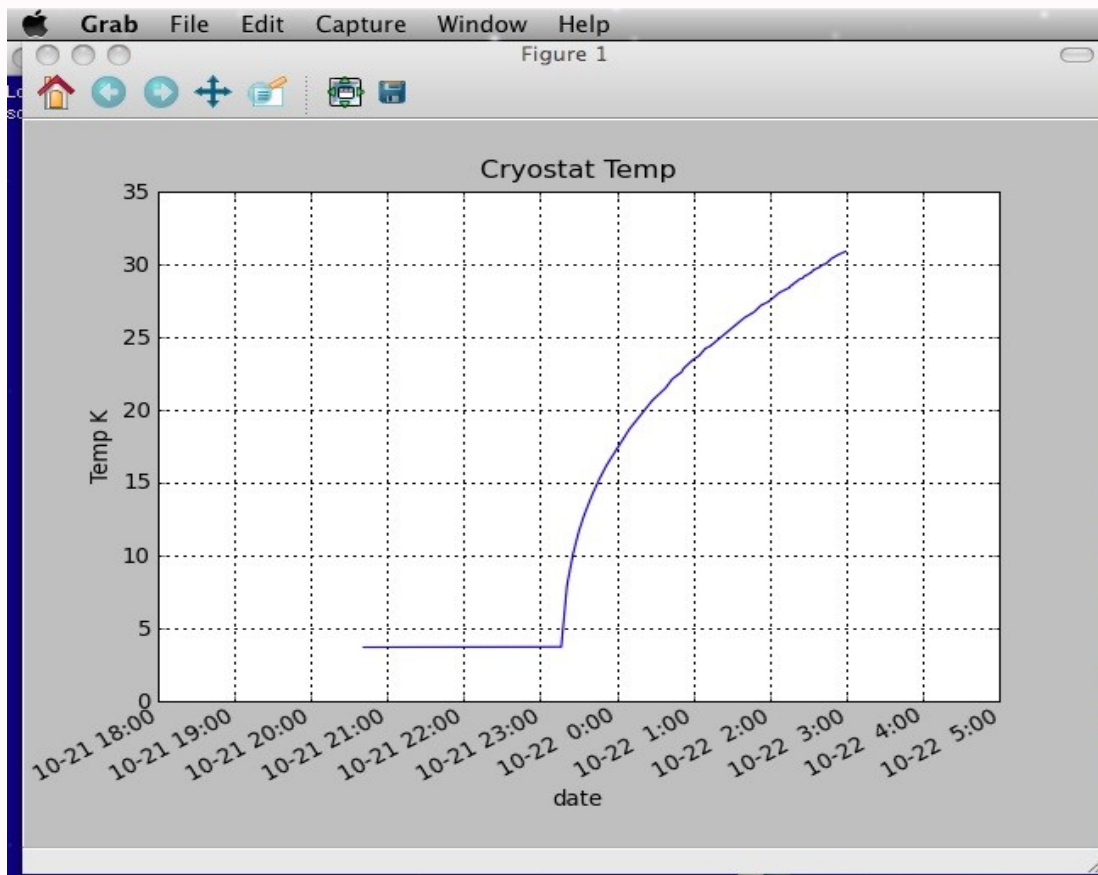
End: yyyy-mm-dd hh:mm:ss  
2008-11-19 21:08:59

Averaging: seconds  
10

Get Current Query Raw Query Statistics

Status: ok

# Monitoring: Presentation Layer





Finally, this is the reason to keep all different TMCDB sections as a whole...

We can create powerful queries against the DB

<http://almasw.hq.eso.org/almasw/bin/viewfile/Archive/TMCDBRepository?rev=1;filename=TMCDB-v1.6.3.png>

# Big Picture of Current Level...



Section Layer	Monitoring	Startup	Configuration	Software Deployment
Presentation	<p><i>TMCDBMonitorGUI</i></p> <p><i>TMCDBMonitorReport</i></p>			
Business				
Persistence				
Database				





## Schedule

TMCDB: It's Layered Architecture

Activating  
TMCDB

Ongoing TMCDB  
Development

Conclusion

S

# Activating TMCDB



Export some environment variables:

```
ENABLE_TMCDB=1
```

```
TMCDB_CONFIGURATION_NAME='<dbname>'
```

For an initial bulk loading data from CDB

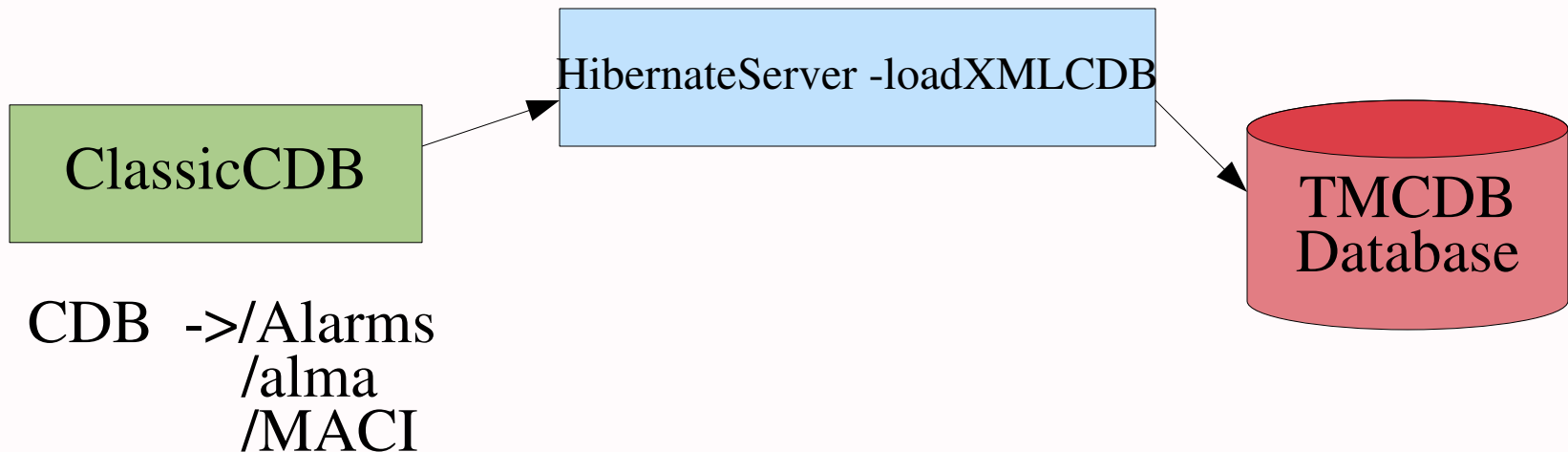
```
LOAD_FROM_XML=1
```

```
ACS_STARTUP_TIMEOUT_MULTIPLIER=100
```

# Initial bulk data loading



**TMCDDB can be populated with initial Software Deployment information from an existing classic CDB**



# Activating TMCDB



Update dbConfig.properties file located on \$ACSDATA/config with this data:

```
# for HsqlDB:  
alma.tmcdb.backend=hsqldb  
archive.tmcdb.backend=hsqldb  
archive.tmcdb.user=sa  
archive.tmcdb.location=jdbc:hsqldb:hsql://localhost:8090
```

```
tmcdb.db.backend=hsqldb  
tmcdb.confname=test  
tmcdb.hsqldb.user=sa  
tmcdb.hsqldb.passwd=  
tmcdb.hsqldb.url=jdbc:hsqldb:hsql://localhost:8090
```

# Activating TMCDB



Start HSQLDB database, and load TMCDB schema executing

```
CreateHsqldbTables.sql
```

...then let's start acs and the containers

```
acsStart -noloadifr
```

```
acsStartContainer -cpp CONTROL/DA41/cppContainer
```

```
acsStartContainer -java CONTROL/ACC/javaContainer
```

```
acsstartupLoadIFR MonitorInterface.idl PSABECompSimBase.idl
```

# ..retrieving SW Deployment info



```
Type "help", "copyright", "credits" or "license" for more information.
>>> import Acspy.Util.ACSCorba
>>> dal = Acspy.Util.ACSCorba.cdb()
>>> dao = dal.get_DAO_Servant("alma/CONTROL/DA41/PSABE")
>>> att = dao.get_double("MID_1_VOLTAGE/alarm_timer_trig")
>>> print att
0.0
>>> att = dao.get_double("MID_1_VOLTAGE/default_timer_trig")
>>> print att
10.0
>>>
```



...time to play with it...



## Schedule

TMCDB: It's Layered Architecture

Activating  
TMCDB

Ongoing TMCDB  
Development

Conclusion

S

# Presentation Layer



**This layer is where most development time will be needed in the future**


**ObOps is on charge of this.**

**Currently they are working on the TMCDB Configuration Tool**


**It can look like the next slide....**

# Presentation Layer





## TMCDB Tools





Logs... User Settings...

SW Deploymt Browse Config

Monitoring Configuration User Mgmt

- Antenna
- Pointing Model
- Front End
- Weather Station
- Correlator
- Pad Admin
- CentralRack
- Master Clock
- Array
- Mount


TMCDB Version 2.4 User: pburgos UTC: 20081004T12:43:00.02 ConfigurationID=343 StartupConfigid:342

 **Antenna** 

AntennaType

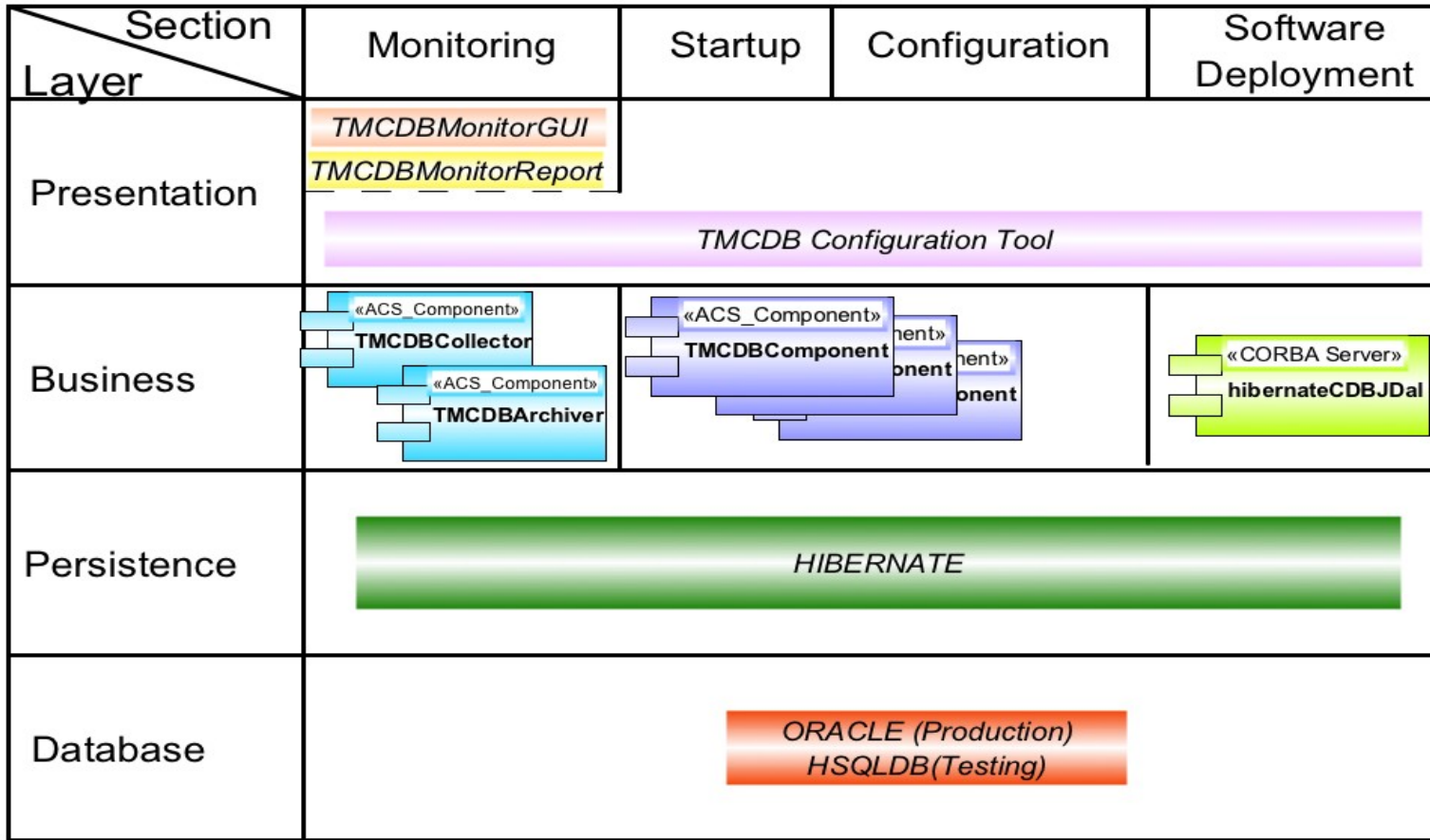
AntennaId	AntennaName	AntennaType	Dish Diameter	Commision Date	xpos	ypos	zpos
1	DV01	VA	11.99	20120303	0	0	0
2	DV02	VA	11.98	20080801	44	43	34
3	DV03	VA	12.05	20081010	54	43	343
4	DV04	VA	12.1	20081201	32	466	434
5	DV05	VA	12.05	20110123	123	45	44
6	DV06	VA	12.03	20110303	2321	43	13
7	DV07	VA	12.02	20110201	23	245	332
8	DV08	VA	12.02	20120112	34	3	214
9	DV09	VA	12.04	20091223	3452	34	2442
10	DV10	VA	12.02	20100229	232	35	243

DV04 Selected--> Retrieving Components ....

Synchronize to DB 



# In the middle term....





## Schedule

TMCDB: It's Layered Architecture

Activating  
TMCDB

Ongoing TMCDB  
Development

Conclusion

S

# Conclusions



All 4 TMCDB sections, Configuration, Monitoring, Startup and SW Deployment are the response to the different use cases expected at ALMA project

TMCDB is the answer to ease the operation of the ALMA Telescope



**¡Muchas Gracias!**

**Thank you!**

