

The 40m Radiotelescope Control System

Pablo de Vicente

Observatorio de Yebes - CDT (IGN)



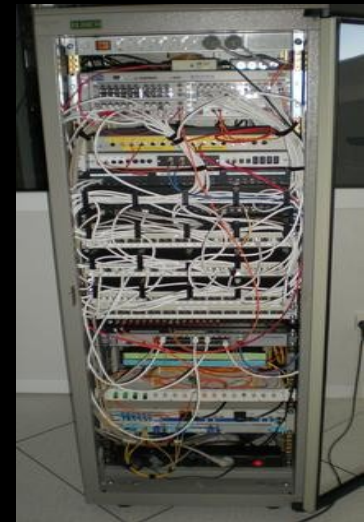
MINISTERIO
DE FOMENTO

DIRECCIÓN GENERAL
DEL INSTITUTO
GEOGRÁFICO
NACIONAL

OBSERVATORIO
ASTRONÓMICO
NACIONAL

Outline

- A software point of view of a RT: distributed system
- The infrastructure: ACS (ALMA Common Software)
- Programming new devices
- The ACU (Antenna Control Unit)
- The FS (Field System) connection
- Data acquisition
- The pipeline
- The pointing model implementation & analysis
- Lessons learnt



What is the control system of a telescope about?



The radiotelescope is a distributed system

- Operations and computations are distributed in different hosts



- Each physical device is associated to a software object living in a network (a "component").



For example:

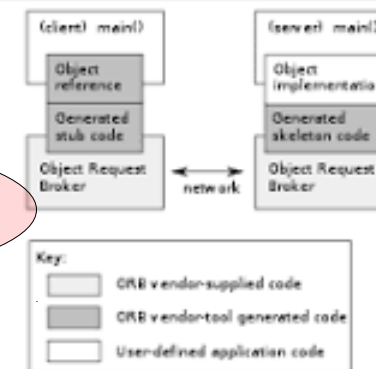
a component -> Spectral backend

- Components talk to other components via ethernet
- Need for an infrastructure that - eases communication &
- provides tools

CORBA via Alma Common Software

What's CORBA ???

The Common Object Request Broker Architecture (**CORBA**) is a standard defined by the Object Management Group (OMG) designed to facilitate the communication of systems that are deployed on diverse platforms.



[Common Object Request Broker Architecture - Wikipedia ...](https://en.wikipedia.org/wiki/Common_Object_Request_Broker_Architecture)

https://en.wikipedia.org/wiki/Common_Object_Request_Broker_Architecture

Feedback

CORBA
via Alma Common Software

Alma Common Software = ACS



The ALMA solution: three CORBA implementations + services

TAO/ACE, OmniORB and JacORB running in Linux

Why ACS?

- Decision taken in 2004
- CORBA is too complex. ACS hides its complexity to developers
- ACS provides useful tools, services & libraries
- Used by other telescopes
- ACS is free & supported by ESO
- Supports C++, Java, Python

ACS repositories

- Official release frequency ~ once per year
Redhat Enterprise / Scientific Linux
- Community fork synched with the official release
Other Linux distributions (Debian, ...)

The screenshot shows the GitHub interface for the repository ACS-Community / ACS. At the top, the GitHub logo is on the left, followed by a search bar containing 'This repository' and 'Search'. Navigation links for 'Explore', 'Features', 'Enterprise', and 'Pricing' are in the center, and 'Sign up' and 'Sign in' buttons are on the right. Below the navigation, the repository name 'ACS-Community / ACS' is displayed with a bookmark icon. To the right of the name are three buttons: 'Watch' (28), 'Star' (12), and 'Fork' (20). The main content area shows the repository is an 'Official ACS community fork'. A summary bar indicates 14,618 commits, 10 branches, 1 release, and 5 contributors. Below this, a green 'Refresh' button is next to 'Branch: master' and 'ACS / +'. A merge notification shows 'Merge branch 'master' into acscb-master' by user 'javarias' on 23 Jul, with the latest commit hash '9ce3f60683'. A sidebar on the right contains links for 'Code', 'Issues' (22), 'Pull requests' (0), and 'Wiki'.

How to program a component

- Each component exposes to the world:

Its methods

[Its characteristics]

- Do we want to generate a notification channel ?

- Automatic generation of templates

- Is it abstract or is it linked to a physical device?

- Physical device: ethernet/serial/GPIB connection?

- Do we need speed?

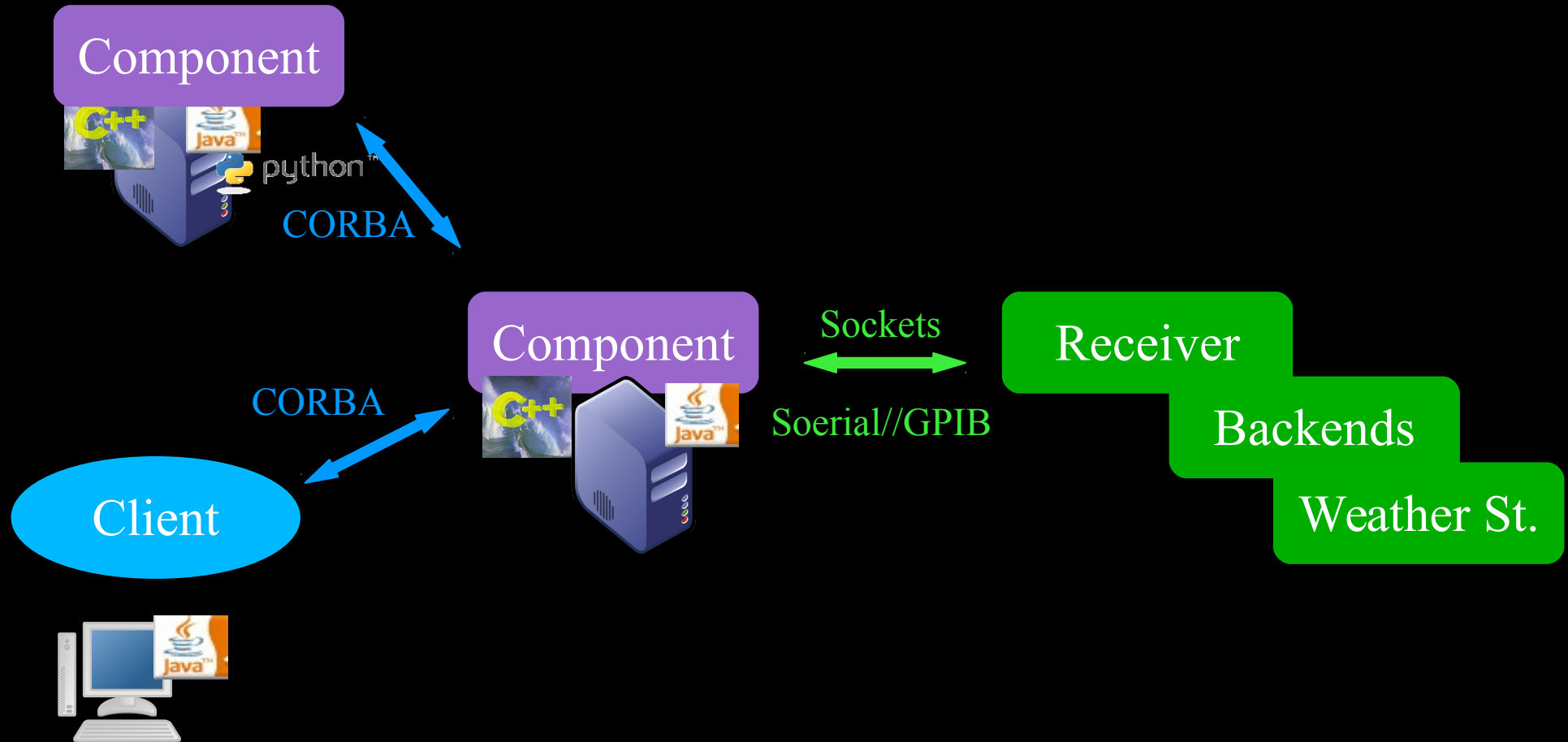
IDL file

C++

Python

Java

Clients & components relationship



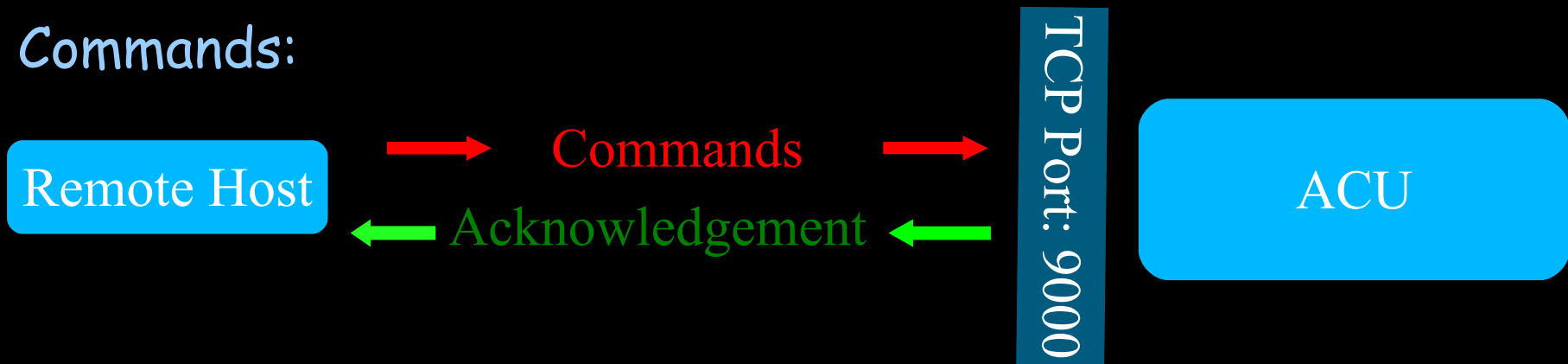
The ACU: Antenna Control Unit

- Computer provided by the telescope building company
- Runs a real time operative system or a real time extension
- Controls the main drives (Az & EL) and the subreflector
- It is usually a black box

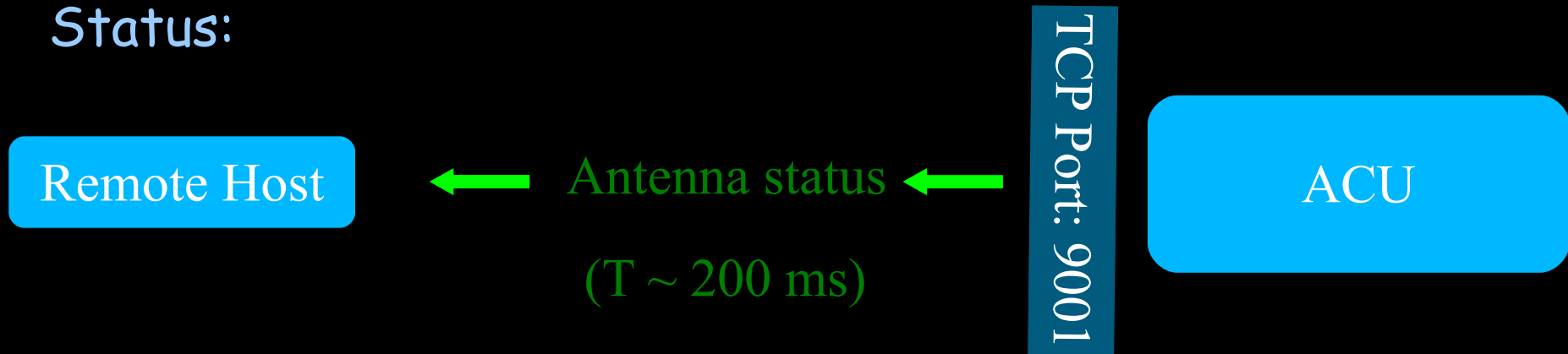


How to command and read the ACU remotely

Commands:



Status:



The ACU ICD: Interface Control Document

- Interface between the remote control and the ACU
- 2 Category of commands:

Mode commands

Positioning

Pointing corr.

Time source

Master setting

Trajectory & special commands

Az/EI Ra/Dec tracking tables

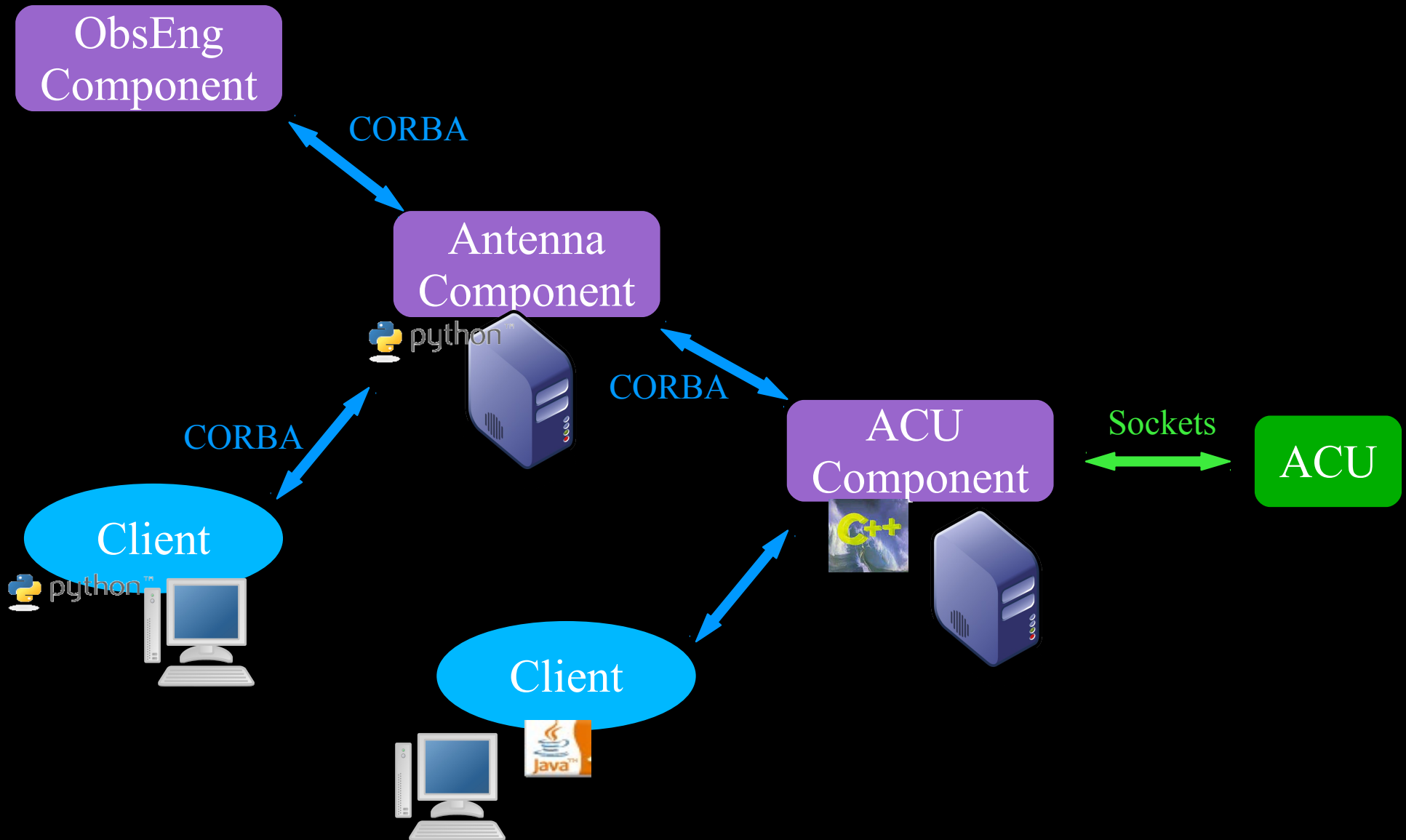
Az/EI Offset tracking tables

M2 Offset tracking tables

M2 Offset elevation dependency

- Description of command parameters
- Description of status words and fields

Software layers: abstraction and encapsulation



Graphical clients (ACS generic)

The screenshot displays the Acs Command Center interface, which is used for managing the deployment and operation of the ALMA software suite. The main window is titled "Acs Command Center" and includes a menu bar with "Project", "Tools", "Expert", and "Help".

Common Settings:

- Acs Instance: 0
- Cdb Root Dir: /home/almamgr/aries21/OANCDB
- Localhost (single-machine project) []
- Remote (distributed project) [X]
 - Use built-in ssh []
 - Use native ssh [X]
 - Use Acs Daemons []
- Host: ariesacu.oan.es
- User: almamgr

Acs Suite:

- Start []
- Stop []
- Kill []
- advanced []

Containers:

Name	Type	Remote Host
OCif100	cpp	ariesfrontends.oan.
OCif45	cpp	uioc-if45
OCpywriter	py	ariesfits.oan.es
OCpygildas	py	ariesfits.oan.es
OCatmosphere	java	ariesnew.oan.es

Deployment Info:

- Containers (34)
 - 'OCacu' [id 83034118]: 1 component
 - 'OCams' [id 78643207]: 1 component
 - 'OCantenna' [id 78839829]: 1 component
 - 'OCatmosphere' [id 71958545]: 1 component
 - 'OCbackends' [id 71434259]: 1 component
 - 'OCcryo' [id 70647833]: 0 components
 - 'OCdatawriter' [id 79757338]: 1 component
 - 'OCefem' [id 67698690]: 16 components
 - 'OCffts' [id 78643218]: 1 component
 - 'OCfrontends' [id 82182182]: 1 component
 - 'OCfs' [id 78184481]: 1 component
 - 'OCgps' [id 74842117]: 0 components
 - 'OCgpsmaser' [id 79167492]: 1 component
 - 'OCholoB' [id 73007133]: 0 components
 - 'OCholoR' [id 79822876]: 0 components
 - 'OCif100' [id 67436583]: 0 components
 - 'OCif45' [id 68878350]: 1 component
 - 'OCifc' [id 72744972]: 0 components
 - 'OCifsch' [id 71893003]: 0 components
 - 'OCifx' [id 67895306]: 0 components
 - 'OCjfrontends' [id 80085001]: 1 component
 - 'OCjholowriter' [id 71761950]: 0 component
 - 'OCobseng' [id 77791268]: 1 component
 - 'OCopticaltel' [id 82706454]: 0 component
 - 'OCpycal' [id 68550676]: 1 component

Object Explorer:

- File View BACI Engine
- By type | By device
- Search
- Tree view showing components like: telramHemt, ffts, fitsW, focuser, frontend, HET12P, HET2, HET22, HET45 (selected), HET5, HET6, HET8, HET87, HETCH, fsNet, FS40MCOMPY, gildasProcessor, gpsMaserComNet, holoal.

Object Details (Object: HET45):

- Operations: getAmplitude(), getBand(), getCalibration(), getHarmonic(), getIF0(), getIFatt(), getLOFreq(), getMode(), getNoiseDiodeAtt(), getObsFreq(), getPhaseCal(), getRate(), imageDistance(), imageGain(), instantaneousBandwidth(), locked(), maxLOFreq(), minLOFreq(), numberOfFeeds()
- Attributes: componentState, name

Log Window:

```

2015-10-02T10:30:01.984 [OCif100-GL - ] Switched state of component IF100: INITIALIZING -> INITIALIZED
2015-10-02T10:30:01.984 [OCif100-GL - ] Switched state of component IF100: INITIALIZED -> OPERATIONAL
2015-10-02T10:30:01.984 [OCif100-GL - maci::ContainerImpl:activate_component] Successfully initialized component. logName="LOG_Comp
2015-10-02T10:30:01.984 [OCif100 - maci::ContainerImpl:activate_component] Component 'IF100' activated.
2015-10-02T10:30:01.984 [Container-ActivationMethod - maci::ActivationMethod::call] Calling maci::CBCComponentInfo::done with descOut.id.t
2015-10-02T10:30:01.984 [Container-ActivationMethod - maci::ActivationMethod::call] Call to maci::CBCComponentInfo::done with descOut.id.t
2015-10-02T10:30:07.582 [OCif100 - maci::ContainerImpl:deactivate_component] Deactivating component with handle 16777290 (IF100 of t
2015-10-02T10:30:07.582 [OCif100 - maci::ContainerImpl:deactivate_component] Component 'IF100': calling cleanUp Lifecycle .
2015-10-02T10:30:07.583 [OCif100-GL - ] Switched state of component IF100: OPERATIONAL -> DESTROYING
2015-10-02T10:30:07.583 [OCif100-GL - ] Switched state of component IF100: DESTROYING -> DEFUNCT
2015-10-02T10:30:07.583 [OCif100 - maci::ContainerImpl:deactivate_component] Component 'IF100' reference set to nil.
2015-10-02T10:30:07.583 [OCif100 - maci::ContainerImpl:deactivate_component] Ethernalizing 'IF100'
2015-10-02T10:30:07.583 [OCif100 - maci::ContainerImpl:deactivate_component] Component 'IF100' etherealized.
2015-10-02T10:30:07.583 [maci::LibraryManager - maci::LibraryManager::unload] Unloading '/home/almamgr/introot/lib/libIF100.so'
2015-10-02T10:30:07.583 [maci::LibraryManager - maci::LibraryManager::unload] Executing DLLClose
2015-10-02T10:30:07.583 [maci::LibraryManager - maci::LibraryManager::unload] DLLClose executed
2015-10-02T10:30:07.583 [maci::LibraryManager - maci::LibraryManager::unload] Unloaded '/home/almamgr/introot/lib/libIF100.so'.
2015-10-02T10:30:07.583 [OCif100 - maci::ContainerImpl:deactivate_component] Component 'IF100' deactivated.
  
```

Taskbar:

- Acs
- OCweather
- OCefem
- OCscan
- OCgpsmaser
- OCgps
- OCacu
- OCams
- OCvobs
- OCjfrontends
- OCifx
- OCifsch
- OCifc
- OCif22
- OCif100
- OCif45
- OCpywriter
- OCpygildas
- OCatmosphere
- OCffts
- OCbackends
- OCpycal
- OCantenna
- OCopticaltel
- OCfrontends
- OCpyfrontends
- OCcryo
- OCdatawriter
- OCobseng
- OCholoR
- OCholoB
- OCjholowriter
- OCpycryo
- OCpyholo
- OCfs
- OCpythp
- Object Explorer

Graphical clients (specific)

Monitor Observation
_ □ ×

Oct 02, Fri, 2015
(275 / 365)

Latitude (°) 40,525
Longitude (°) -3,087
Height (m) 990,000

Antenna

Main Axis Status: tracking

Az (deg): 220,4848 dAz ("): 0,1

EI (deg): 56,8886 dEI ("): 0,1

Corrections

COL*A (") 0,00 Focus (mm)

NULE (") 20,00 X 0,00

Tilt X -42,83 Y -0,00

Tilt Y 28,83 Z 1,91

Weather

Temperature (C) 17,60

Humidity (%) 32,00


Pressure (hPa) 913,00

Wind Speed (m/s)

Inst. 1,90

Max. 10 min 7,20

Wind Direction (deg)



248,00

Scan

Project ID N-00.C-0046-2015

Number	Subscan	Total Time (s)	Type
49833	1 / 2	138	OnOff

	Start Time	Current Date/Time	End Time
UTC	10:42:41	10:43:13	10:44:59
LST	11:13:41		11:16:00

S/CH/C/X 22 / 45 / 87 GHz RX Cabin

Frontend:	Q_BAND
Frequency (MHz):	48747,548
LO:	LOCKED
Cleanup:	UNKNOWN
Calibration:	OFF

Source

IRC+10216

AR (h) 09:47:57,41

Offset 0,0000

DEC (deg) 13:16:43,56

Offset 0,0000

Epoch 2000,00

Frontend / Backend

HET45(R)-FFTS(3)

HET45(L)-FFTS(4)

IF Switch

Matrix Input IN_1_C

Fieldsystem Component

Active: ●

Frontend:	W_BAND
Frequency (MHz):	86250,000
LO:	LOCKED
Cleanup:	UNKNOWN
Calibration:	OFF

Alarms

CAUTION

s_c_ch bad position

x_ka bad position

k_w bad position

Commanding the telescope (example)

Ipython shell

```
Operator ID (NN) ? PV
Observer ID (NN) ? PV
RAEGE> status()
M1:   stopped. Current position: 225.00015, 89.99948
M2:   inactive. Current position:  0.00  0.00 -0.00  0.00  0.00 Errors: -0.00 -0.00 -0.00 -0.00 -0.00
      El Table offset position:  0.00  0.00  0.00  0.00  0.00

RAEGE> sourcecats(['/home/raegemgr/raege/Catalogs/user.cat'])

RAEGE> source('3C345')
[Errno 2] No such file or directory: 'user.cat'
Multiple matches:
Setting source velocity to 0.0.
Setting source 3C345 @ RA=16:42:58.810, Dec=+39:48:37.00 System: EQ 2000.0, Velocity: 0.00 LSR
Currently at Az=51.659 degs. El=16.890 degs.
For the time being offsets have to be in the same system as the source system. Command again the offset please.
(51.6586395371667, 16.889682957316857)

RAEGE> setTrajectoryParams(False, True)

RAEGE> on(60)
Beware: No frontend is being used. Data aquisition not available
Setting dataAquisition to 0
Setting up an "on" scan with 1 repetitions.
No data acquisition!

--- Scan: 1 (Type: Others) --- Start time: 08:32:39 End time: 08:33:42
(efem.dateTime(year=2013, month=10, day=15, hour=8, minute=32, second=39, millisecond=195),
 efem.dateTime(year=2013, month=10, day=15, hour=8, minute=33, second=42, millisecond=195))

RAEGE> point(3600,'arcsec',60,'otf',10,'no','x',0)
Data acquisition not available
No data acquisition!

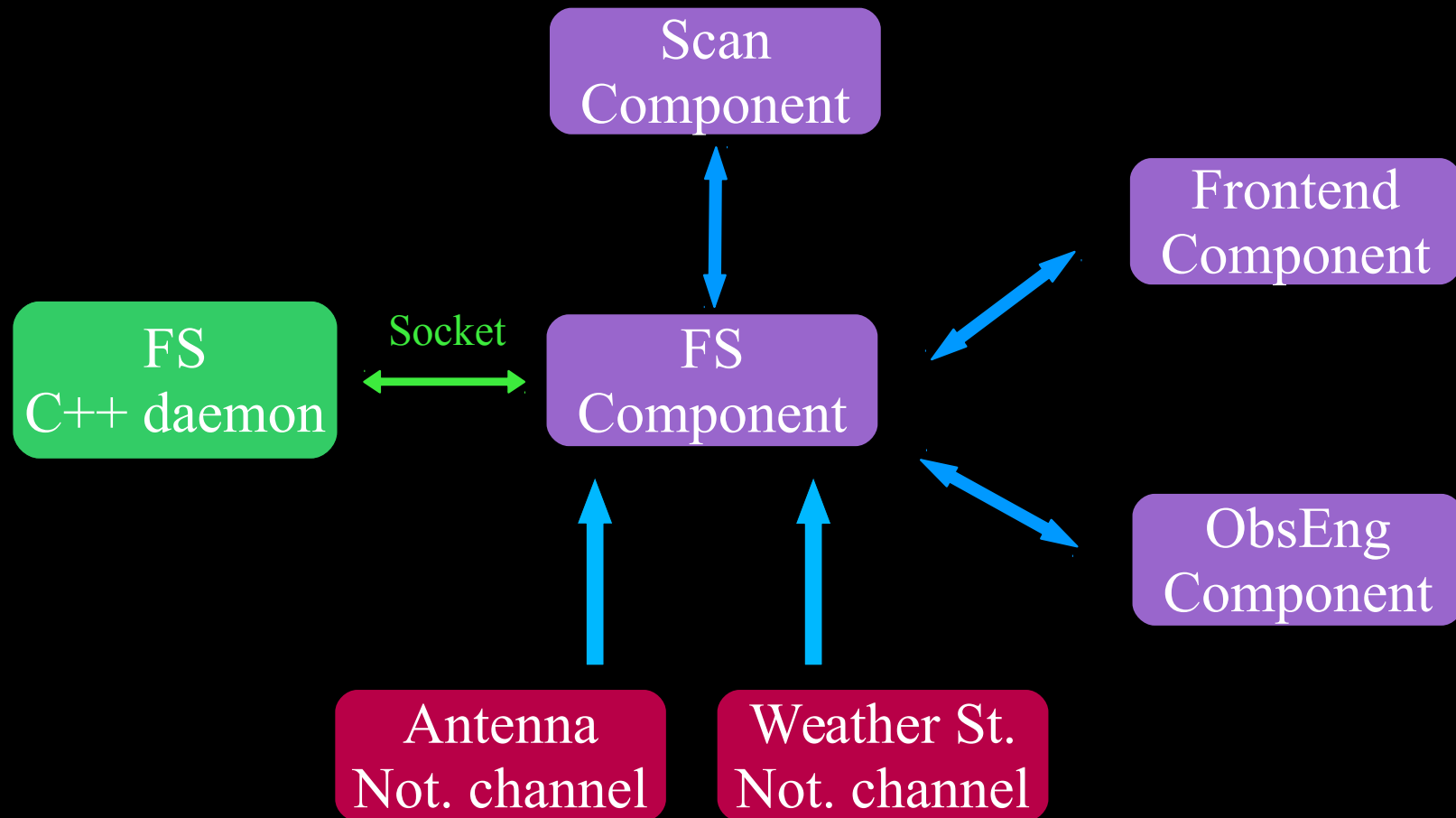
--- Scan: 2 (Type: Point) --- Start time: 08:34:56 End time: 08:37:33
(efem.dateTime(year=2013, month=10, day=15, hour=8, minute=34, second=56, millisecond=1),
 efem.dateTime(year=2013, month=10, day=15, hour=8, minute=37, second=33, millisecond=1))
```

Setting the source

Track the source 60 secs.

Pointing drift on the source

The FS component: connections in the Control System



Data Acquisition

- Data are captured and written in FITS files in real time

FITS



File Format Transport System

Every subscan

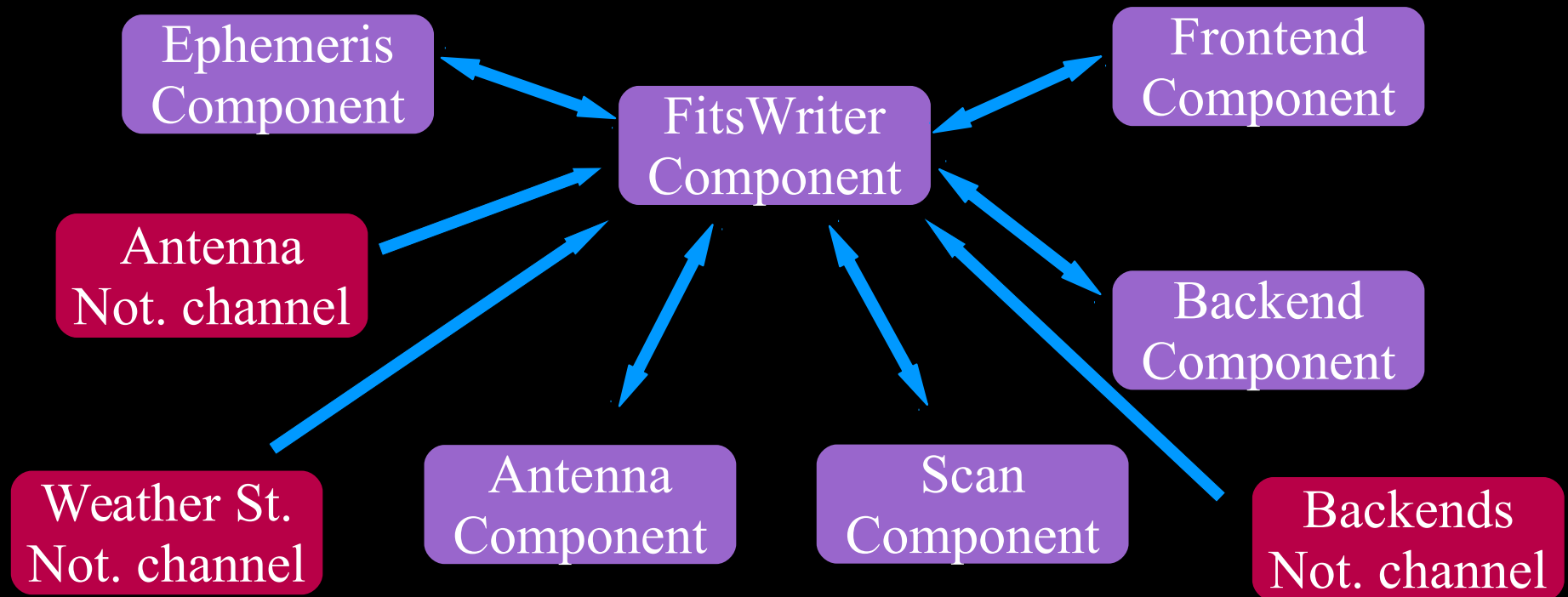


ONE file (MBFITS)

MBFITS: Multi Beam FITS

- Data organized in HDUs (Header Data Units)
- All scan information is stored in FITS headers
- Data from backends are in tables (1 HDU per Front-Back)
- Relative position information in tables
- Auxiliary information (weather, cal, ...) in tables

Data Acquisition



- Component coded in C++ (faster)
- Component uses cfitsio library
- Component uses a home made C++ cfitsio wrapper

Pipeline

- Coded in Python
- The pipeline generates GILDAS files: CLASS format



Institut de Radioastronomie Millimétrique

[Contact us](#) | [Useful links](#) | [Job offers](#) | [Search](#)
[Webmail](#) [FR / ES](#) | [Intranet access](#)

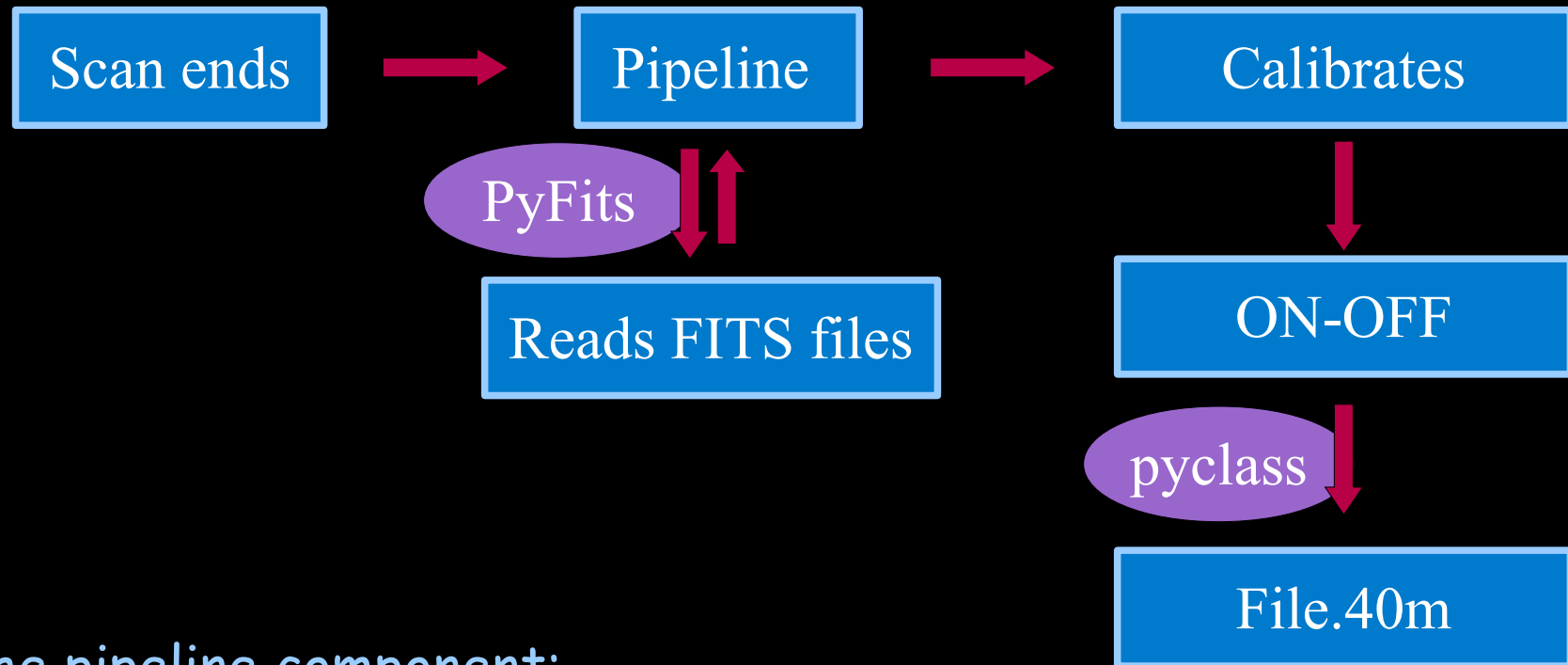
[About us](#) | [NOEMA](#) | [30m telescope](#) | [Science & technology](#) | [International cooperation](#) | [Public outreach](#) | [Science users](#)

[30m telescope](#)
[NOEMA interferometer](#)

[Science users](#)



Pipeline (single dish)



- The pipeline component:
 - Uses a python FITS module
 - Generates calibrated data (atmosphere, hot/load calcs)
 - Uses a pyclass filler

The pointing model: implementation

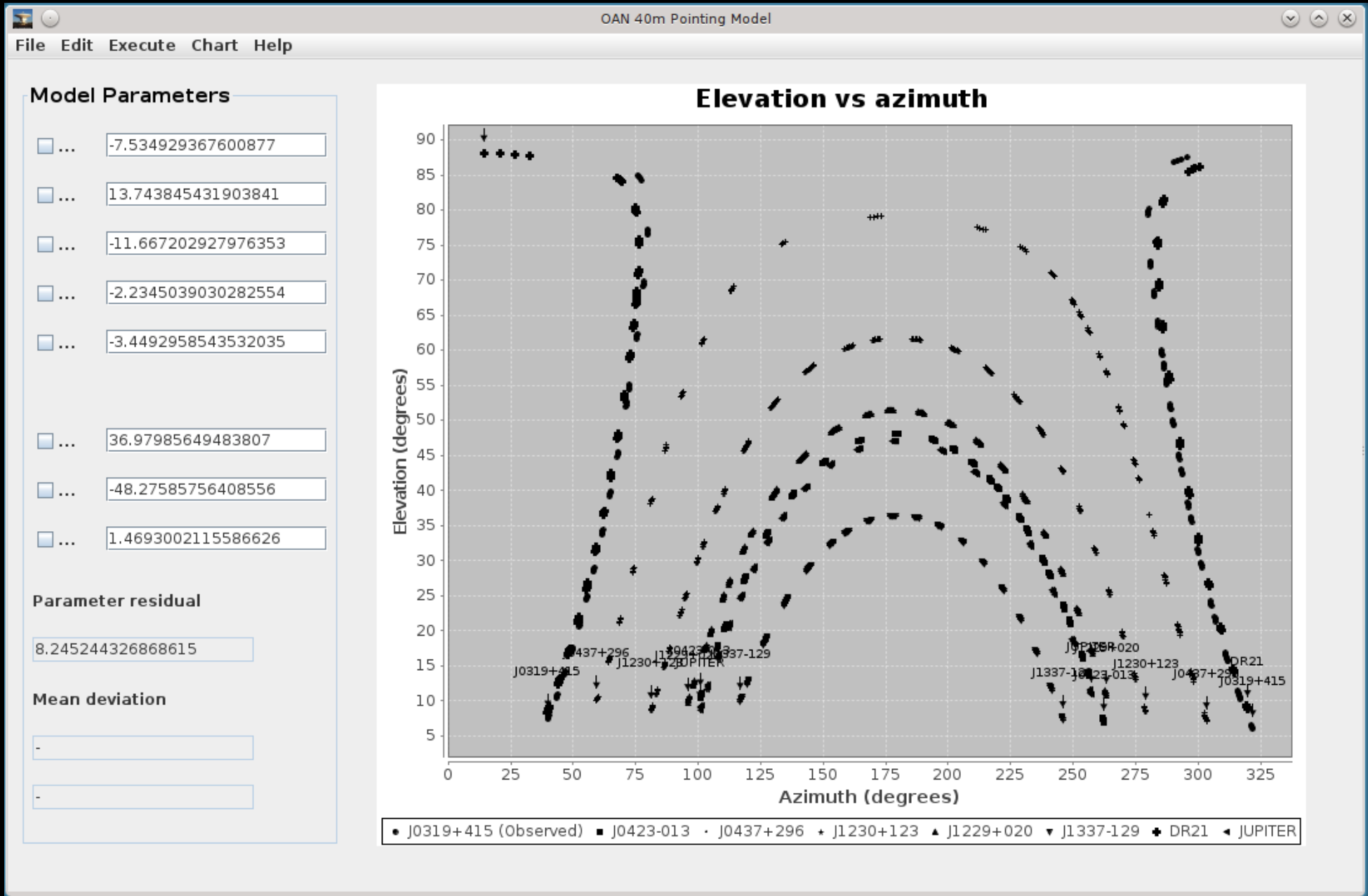
- The ACU implements a pointing model
 - composed of 9 parameters
 - which can be switched off

$$\delta E1 = P_1 + P_2 \sec(E1) + P_3 \tan(E1) - P_4 \cos(Az) \tan(E1) + P_5 \sin(Az) \tan(E1)$$

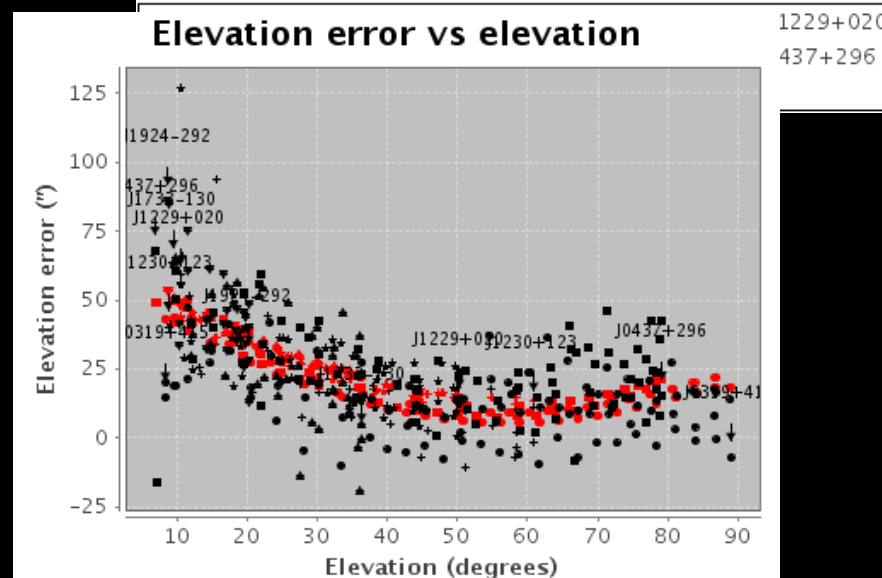
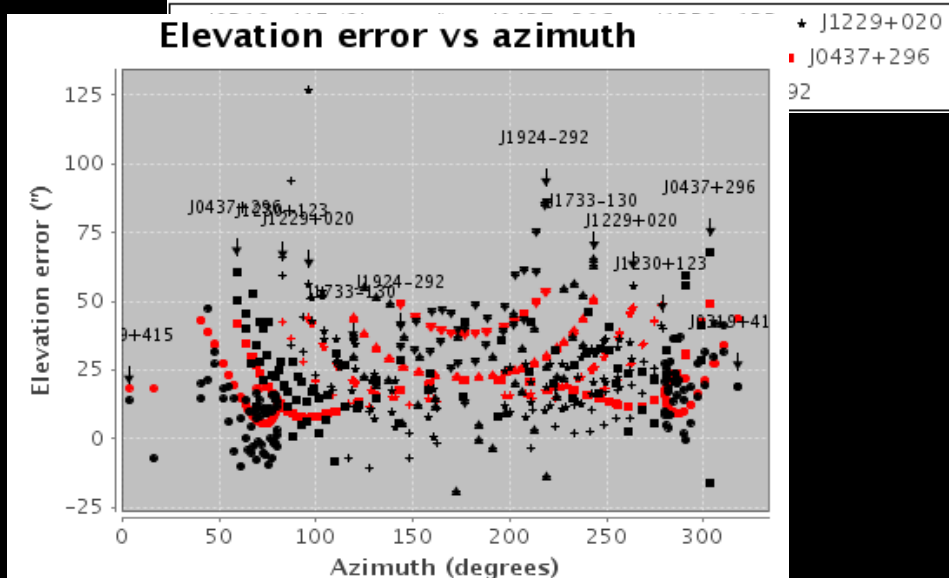
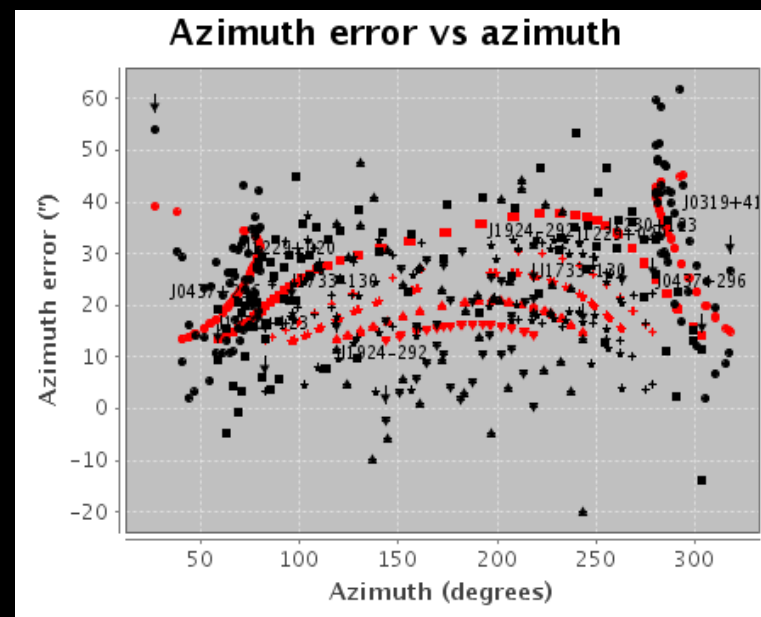
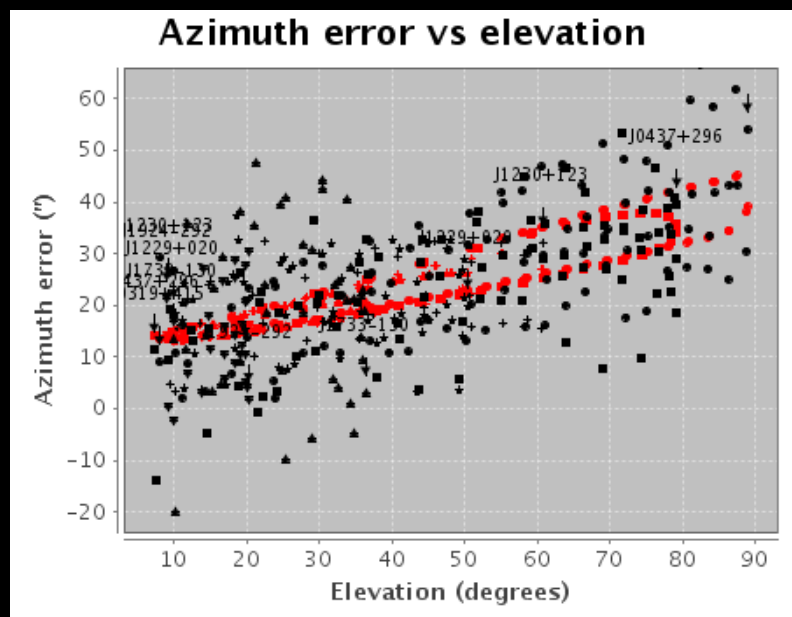
$$\delta Az = P_7 + P_4 \sin(Az) - P_5 \cos(Az) + P_8 \sin(E1) + P_9 \cos(E1)$$

- The antenna component uses a different model per receiver

The pointing model: analysis



The pointing model: analysis



• J0319+415 (Observed) ■ J0437+296 • J1230+123 + J1229+020
 ▲ J1733-130 ▼ J1924-292 ● J0319+415 (Predicted) ■ J0437+296
 • J1230+123 + J1229+020 ▲ J1733-130 ▼ J1924-292

• J0319+415 (Observed) ■ J0437+296 • J1230+123 + J1229+020
 ▲ J1733-130 ▼ J1924-292 ● J0319+415 (Predicted) ■ J0437+296
 • J1230+123 + J1229+020 ▲ J1733-130 ▼ J1924-292

Lessons learnt

- ACS has been a good choice: powerful and easy to use
 - The github repository facilitates the upgrades
 - The notification channel is an excellent service
 - We have not explored other services: alarms
-
- Tables in the ACU are a simple & flexible solution
-
- MBFITS is a good choice, but not fully exploited
 - FITS writer is fast but care is required when modifications
-
- Pyclass filler is a good tool for GILDAS.
 - CLASS is an excellent option for the processed data

Radiotelescope Component Relationship

or

What's going on when you command a scan?

