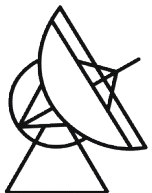


# APEX Control System (APECS)

ERATec 2015

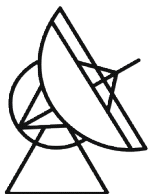


Oct. 2015

APECS, ERATec 2015

1





Oct. 2015

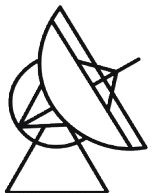
APECS, ERATec 2015

2



# Background

- APEX 12m telescope at 5100m on Chajnantor (Partners: MPIfR, ESO, OSO)
- ALMA prototype antenna
- Inherited hardware interface to telescope computer
- Re-used ALMA communication (ALMA Common Software, ACS) and telescope control software (Test Interferometer Control Software, TICS)
- Developed observer level software (APECS) at MPIfR



Oct. 2015

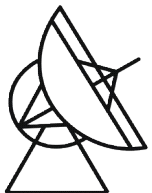
APECS, ERATec 2015

3



# ACS

- Distributed system based on CORBA (Common Object Request Broker Architecture)
- Supports C++, Python and Java code
- Container / Component design
- Automatic property monitoring and alarms
- Centralized logging
- Handles all distributed communication
- Community Edition



Oct. 2015

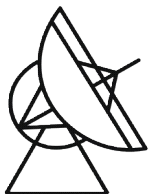
APECS, ERATec 2015

4



# TICS

- Test Interferometer Control System developed for ALMA Test Facility (ATF, VLA site)
- CAN bus components to communicate with Antenna Control Unit (ACU)
- Real-time coordinate transformations (equatorial / horizontal)
- Basic observing patterns (tracking, linear and arc OTF strokes)



Oct. 2015

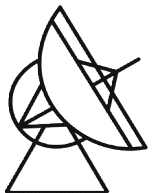
APECS, ERATec 2015

5



# APECS

- Generic, re-usable, high-level interfaces to accommodate expected installation of many PI instruments
- New raw data format for multi-beam receivers (MBFITS)
- Modular Python interface to set up and script observations (apecs)
- Observing Engine to configure and coordinate devices, data acquisition and calibration
- MBFITS Writer to capture backend and telescope data (→ see also talk by R. Schaaf)
- Calibrator to perform atmospheric calibration and spectral line reduction
- Continuum data reduction package (BoA)
- Automatic observation logger with optional user comments
- Simulator mode to test new development before deployment



Oct. 2015

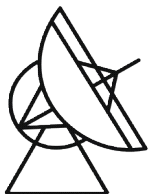
APECS, ERATec 2015

6



# Generic Interfaces

- Developed from a survey of observatory interfaces (100m, 30m, SEST, KOSMA, HHT)
- Identical interface for devices of the same kind (frontends, backends, etc.) make adding new instruments very simple
- Focussing on core functionality necessary for observing, e.g.:
  - *frequency, sideband and frequency switching offsets* for frontends
  - *bandwidth, number of channels, dump time* for backends
- Hide instrument details from APECS
- Interfaces defined in CORBA IDL
- Component code auto-generated with modified generator from U of Bochum



Oct. 2015

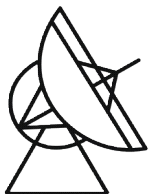
APECS, ERATec 2015

7



# UDP SCPI

- Instrument interfaces use simple ASCII commands and acknowledgements via network sockets
- This relieves the (often lightweight) embedded systems from having to use CORBA or field buses
- UDP sockets allow simple debugging from several different command origins
- UDP can also serve as distributed communication layer (→ Effelsberg)



Oct. 2015

APECS, ERATec 2015

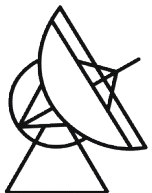
8





# MBFITS

- Based on ALMA TI-FITS
- Added keywords for multi-beam receivers
- WCS compliant
- Frequency axes for TOPO and Doppler shifted source frames
- Backend data are associated with telescope positions by their exact time stamps
- MBFITS is a NASA registered FITS convention (<http://fits.gsfc.nasa.gov/registry/mbfits.html>)
- Hierarchical grouping exploits file system efficiency when writing several tables concurrently



Oct. 2015

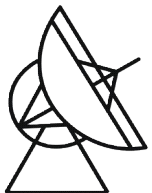
APECS, ERATec 2015

9



# apecs CLI

- Split in modules to configure
  - General information (IDs, scripts)
  - Catalogs
  - Complex Frontend / Backend combinations
  - Switch mode (TP, wobbler, frequency switching)
  - Stroke mode (linear, spiral, Lissajous)
  - Source and calibration / observing patterns
- Observation setup is stored in a Scan Object
- Python interface allows for scripting / macros
- Remote control mode for VLBI field system is available



Oct. 2015

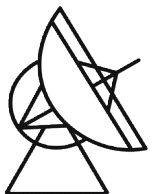
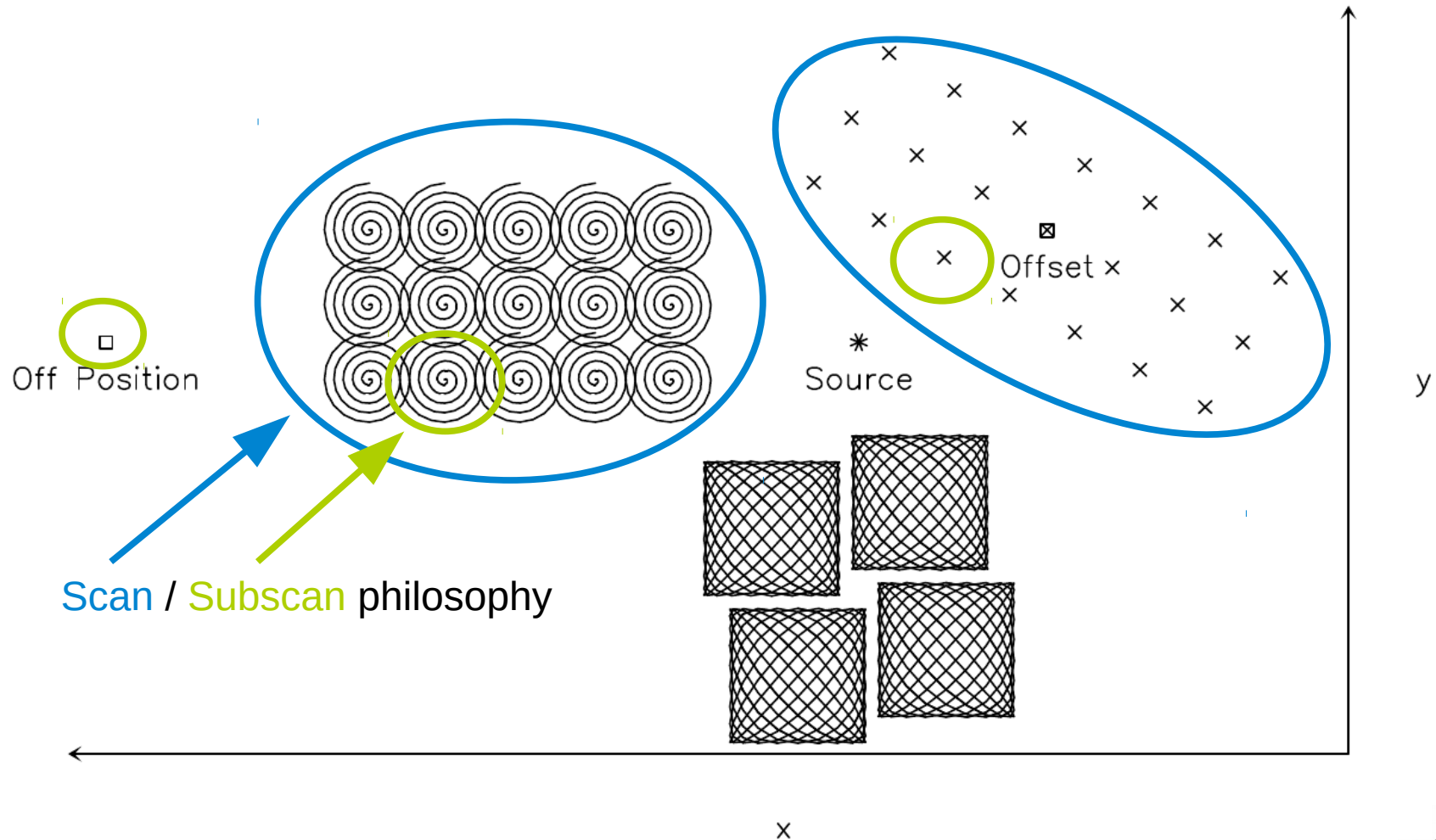
APECS, ERATec 2015

10



# Patterns

## APECS Observing Pattern Examples



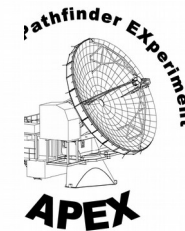
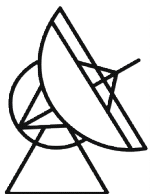
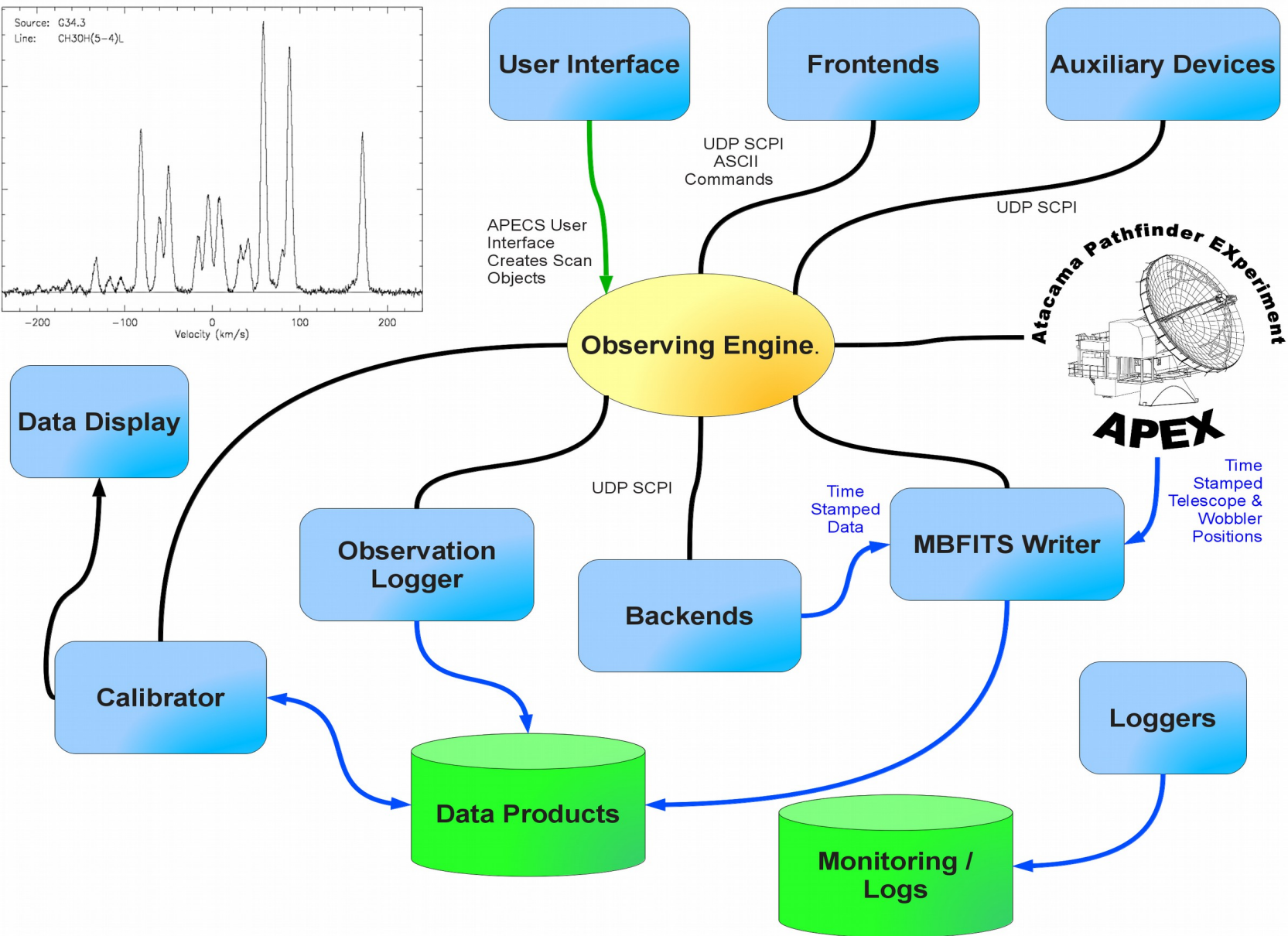
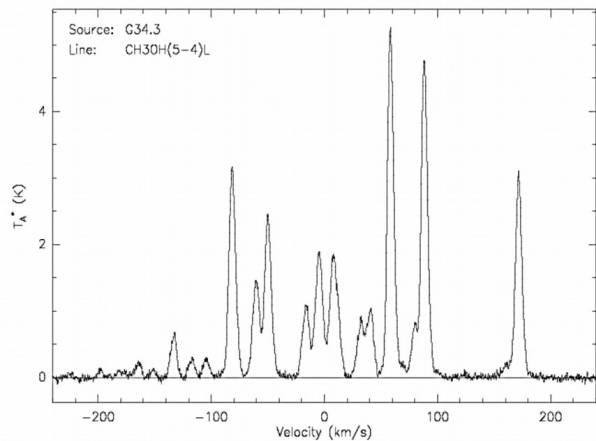
Oct. 2015

APECS, ERATec 2015

11

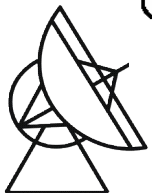
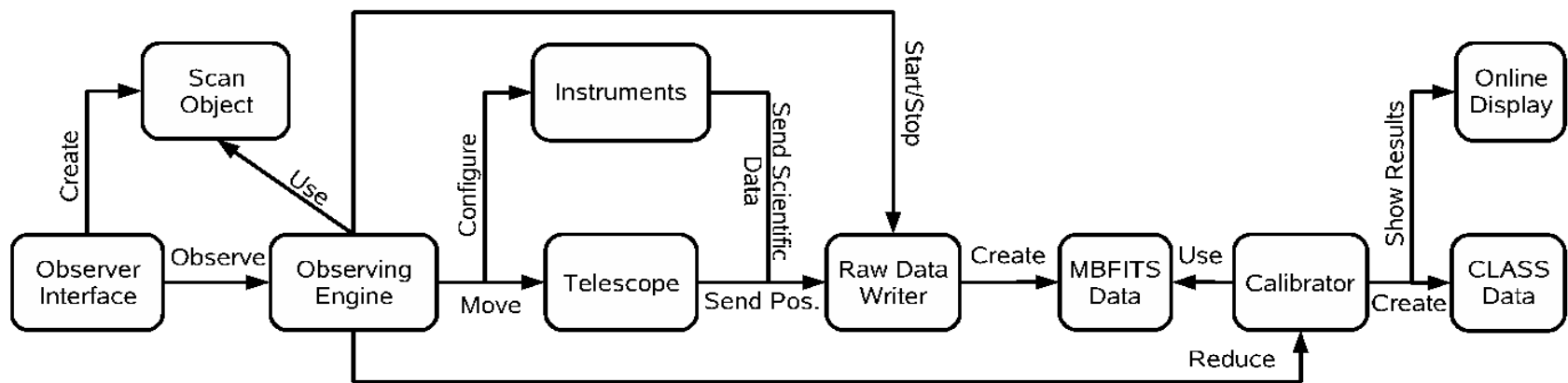


# APECS Design



# Observing Engine

- Central process coordinating the observations
- Uses Scan Objects containing the observation description
- Uses ACS / CORBA to communicate with all devices and instruments



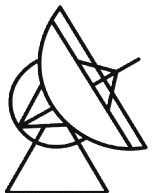
# Pros and Cons

- Pros

- Distributed, scalable system
- Generic, reusable interfaces, easy to add new instruments
- Simple text (SCPI) interface to embedded systems
- Simple monitoring and logging
- Auto-generated interface code
- Localized real-time usage; time stamping to associate data with sky positions
- Simulator

- Cons

- ACS alarm system came late and is complicated
- Dependence on ALMA, esp. in real-time area



Oct. 2015

APECS, ERATec 2015

14

