Open source to the stars:

How open source helps one of the biggest astronomical observatories in the world.

-

A more control software integration centered story

Federico Pellegrin, fpellegr@eso.org

About ESO

The European Southern Observatory is the pre-eminent intergovernmental science and technology organization in astronomy. It carries out an ambitious program focused on the design, construction and operation of powerful ground-based observing facilities for astronomy, in order to enable important scientific discoveries. ESO also plays a leading role in promoting and organizing cooperation in astronomical research.

ESO Projects

- La Silla Observatory
  - Operational since 1976 (first light of 3.6m), 8 still in operation today
  - Home to many generations of telescopes

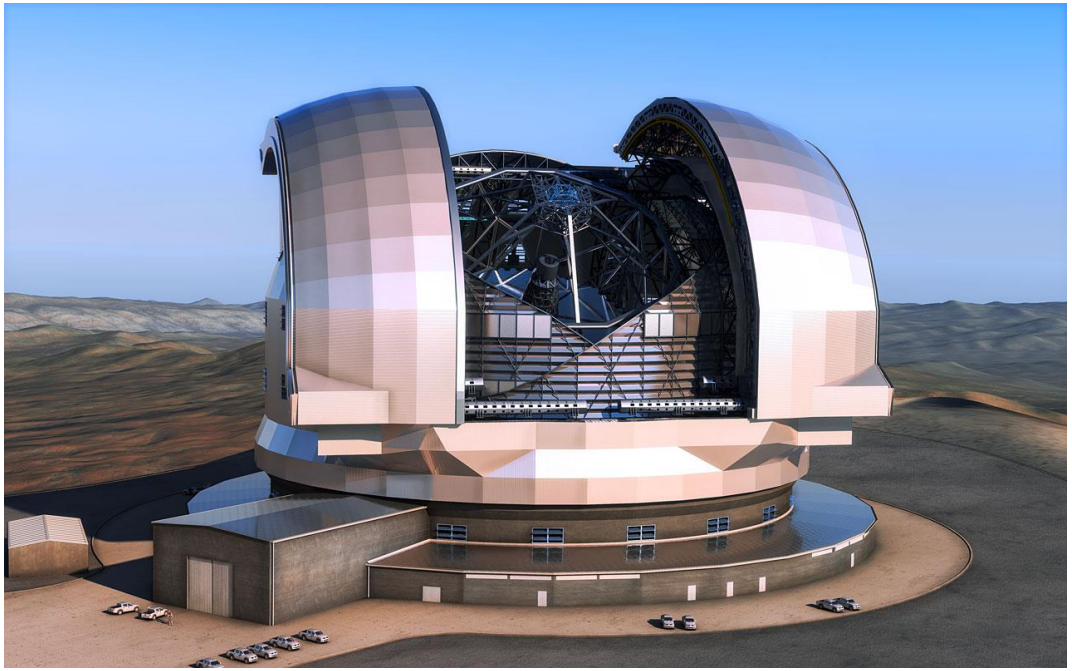
ESO Projects

- Paranal Observatory
  - Operational since 1999 (first light of UT1)
  - Very Large Telescope (4 x 8m visible)
  - VLT Interferometer (4 x 8m VLT + 4 x 1.8m AT)
  - Other telescopes on site (VISTA, VST)
  - Future home for Cherenkov Telescope Array

ESO Projects

- ALMA – Atacama Large Millimeter Array
  - Joint project with NRAO and NAOJ
  - Operational since 2011
  - 54 x 12m + 12 x 7m moveable antennas at 5000m
  - APEX nearby

The ELT



Extremely Large Telescope

- 39m ground-based
- Cerro Armazones
- First stone May 2017
- First light expected 2024
- Largest optical/near-IR
- Exoplanets, star formations, protoplanetary systems

- Five-mirror design
- M1: 798 segments 1.4 meters wide 5cm thick (3 PACT, 6 ES, 12 WH)
  - Figure loop at 500Hz ~ 1Gbit/s traffic
- M4: 4 meters (~6000 actuators)
- Alt-azimuth mount with 6 LGS

Commonalities of Projects

- Long time between design and start of operations
  - Usually between 5 to 10 years
- Long lifespan of projects: 30+ years
  - Both hardware and software obsolescence
- One-off projects with specific needs
  - There isn’t a big market for big telescopes
  - Related difficulties in testing and reproducing problems
- Mixed developer and user base
  - Internal / external development, scientific user base
- Environmental conditions, high uptime

Control Software at ESO

- Software (real-time and not) for:
  - Control of the structures
    - Rotation of the dome, hydraulic bearings, management of wind-shields
  - Control of the telescope pointing, guiding and tracking
  - Control of the optics
    - Active and adaptive optics, deformable mirrors
  - Control of the detectors
- Data processing and pipeline then follows
- Team composed by ~ 50 people
  - I manage VLT* SW maintenance, design and develop ELT development environment and follow CI ESO-wide

Code repository at La Silla

Languages and Technologies

- Went through various phases (HPUX, Solaris …)
- Mostly now Linux based
- Some real-time still based on VxWorks, some PLC
  - Hardware architectures MC68000, PPC, NEHALEM
- GNU Make with simplification layer
- Languages used:
  - C, TCL/TK, Fortran, C++, Java, Python
- X-Based UI for Control SW (TCL, moving to Python + Qt), Web based on data handling side
- Communication: DDS, CORBA based, custom

Distribution

- Mostly based on RPM based distributions, until recently Scientific Linux, recently CentOS.
- Not cutting edge but with experimental packages being used:
  - devtoolset-7 + ASAN
  - New kernels and preempt-RT
- Installation and version control is based on Puppet
  - Puppet in master-less configuration driven by Jenkins
  - Sub-classing per version, site and facility
  - Installation kickstart doing basic tasks and then Puppet
    - Optimizations such as single call to yum

Build and Test Infrastructure

- Jenkins, arriving from in-house Perl based solution:
  - Very customized using HTML-publisher for some projects
    - One job per series of modules (not very granular)
    - Build make is parallelized
    - Test entry point via make rule with environment filters
  - CI build on commit
  - Test runner is an in-house solution (in TCL)
    - Generated files converted to HTML for integration in Jenkins
  - Relatively fast (~20 min) smoke test
    - Basic infrastructure is tested (comms, database, events)
    - One LCU architecture is tested
  - Nightly execution full test (~10 hours total) in parallel:
    - Multiple architectures (68k, PPC, NEHALEM)
    - Multiple versions and branches

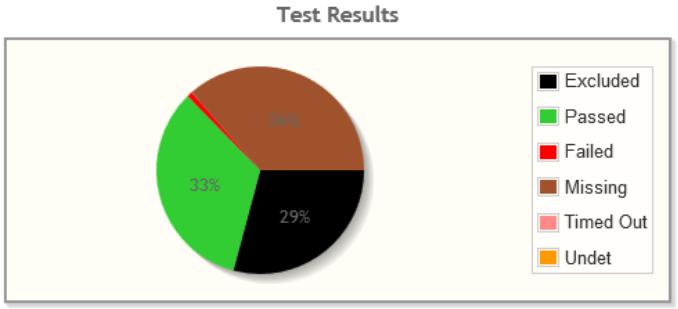
Jenkins customized jobs

Environment log: environment.txt.

SVN Revision URL: http://svnhq1.hq.eso.org/p1/trunk/VLTSW/Core@283185

Kernel version: Linux 3.10.0-327.10.1.el7.x86_64

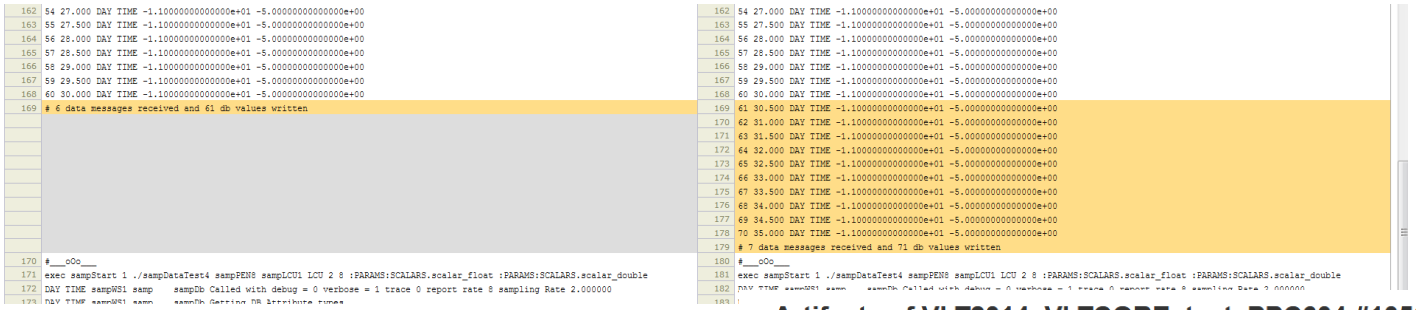
Modules total: 214
Source Build: EXCL 1, PASS 213, FAIL 0
Test Build: EXCL 0, PASS 112, FAIL 0, MISSING 102
Test Run: EXCL 68, PASS 78, FAIL 2, MISSING 85, TIMED OUT 0, UNDETERMINED 0



Name	Dir	Sinner	Arch/Opt	Build	Warn	Test Build	Test Result
fnd	CCS	bgilli	-	PASSED	0	PASSED	PASSED
eccs	CCS	bgilli	-	PASSED	0	PASSED	PASS-PPC604 / PASS-MC68040
eccs	CCS	bgilli	PTHREADS	PASSED	0	NORUN	NORUN
evh	CCS	bgilli	-	PASSED	0	PASSED	PASSED
evh	CCS	bgilli	PTHREADS	PASSED	0	NORUN	NORUN
evhEt	CCS	bgilli	-	PASSED	0	PASSED	PASSED
evhEt	CCS	bgilli	PTHREADS	PASSED	0	NORUN	NORUN
alrm	CCS	bgilli	-	PASSED	0	PASSED	PASS-PPC604 / PASS-MC68040
ccs	CCS	bgilli	-	PASSED	0	PASSED	PASS-PPC604 / PASS-MC68040
cmd	CCS	bgilli	-	PASSED	0	PASSED	PASSED
db	CCS	bgilli	-	PASSED	0	PASSED	PASS-PPC604 / PASS-MC68040
err	CCS	bgilli	-	PASSED	0	PASSED	PASS-PPC604 / PASS-MC68040

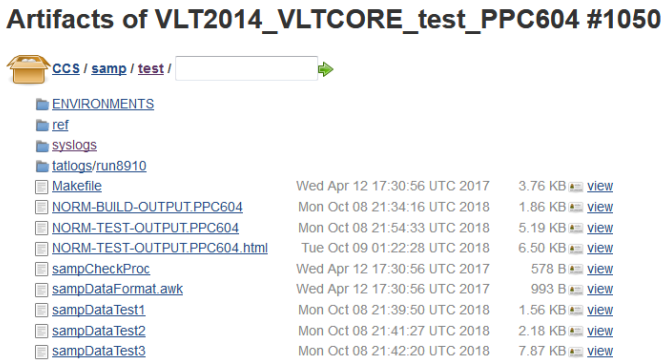
Jenkins customized jobs

```
TEST sampTest2 PASSED.  00:01:29
Executing sampTest3 (timeout: 5400 s.)
No filter applied: grepFile does not exist
Cleaning with TestList.sed: ./tatlogs/run8910/sampTest3.out
TEST sampTest3 PASSED.  00:00:54
Executing sampTest4 (timeout: 5400 s.)
No filter applied: grepFile does not exist
Cleaning with TestList.sed: ./tatlogs/run8910/sampTest4.out
Differences found in /diska/NRI/jenkins/workspace/VLT2014_VLTCORE_test_PPC604/CCS/samp/test/./tatlogs/run8910/sampTest4.diff [VISUAL DIFF]
TEST sampTest4 FAILED.  00:02:12
Executing sampTest5 (timeout: 5400 s.)
No filter applied: grepFile does not exist
Cleaning with TestList.sed: ./tatlogs/run8910/sampTest5.out
TEST sampTest5 PASSED.  00:01:20
Executing sampTest6 (timeout: 5400 s.)
No filter applied: grepFile does not exist
```



```
54 27.000 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
55 27.500 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
56 28.000 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
57 28.500 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
58 29.000 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
59 29.500 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
60 30.000 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
# 6 data messages received and 61 db values written
#___oOo___
exec sampStart 1 ./sampDataTest4 sampPEN8 sampLCU1 LCU 2 8 :PARAMS:SCALARS.scalar_float :PARAMS:SCALARS.scalar_double
DAY TIME sampWS1 samp    sampDb Called with debug = 0 verbose = 1 trace 0 report rate 8 sampling Rate 2.000000
```

```
54 27.000 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
55 27.500 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
56 28.000 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
57 28.500 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
58 29.000 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
59 29.500 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
60 30.000 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
61 30.500 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
62 31.000 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
63 31.500 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
64 32.000 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
65 32.500 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
66 33.000 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
67 33.500 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
68 34.000 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
69 34.500 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
70 35.000 DAY TIME -1.1000000000000e+01 -5.0000000000000e+00
# 7 data messages received and 71 db values written
#___oOo___
exec sampStart 1 ./sampDataTest4 sampPEN8 sampLCU1 LCU 2 8 :PARAMS:SCALARS.scalar_float :PARAMS:SCALARS.scalar_double
```



Build and Test Infrastructure

- Usage of virtual machines and containers for tests
  - VM switching with command line tools scripting via Jenkins job
  - Containers using standard Jenkins plugins
- Code checkers (cppcheck, Nagelfar)
- Special weekend builds:
  - Code coverage (gcov, only workstation side)
  - Debug kernels, ASAN runs, builds with optimization options
- Presence of a Control Model for special tests
  - Additional hardware resembling final installation
  - Open source booking system
- Some other code metrics available
- Generation of release packages (RPM) on demand
- Triggers machine configuration verification via Puppet

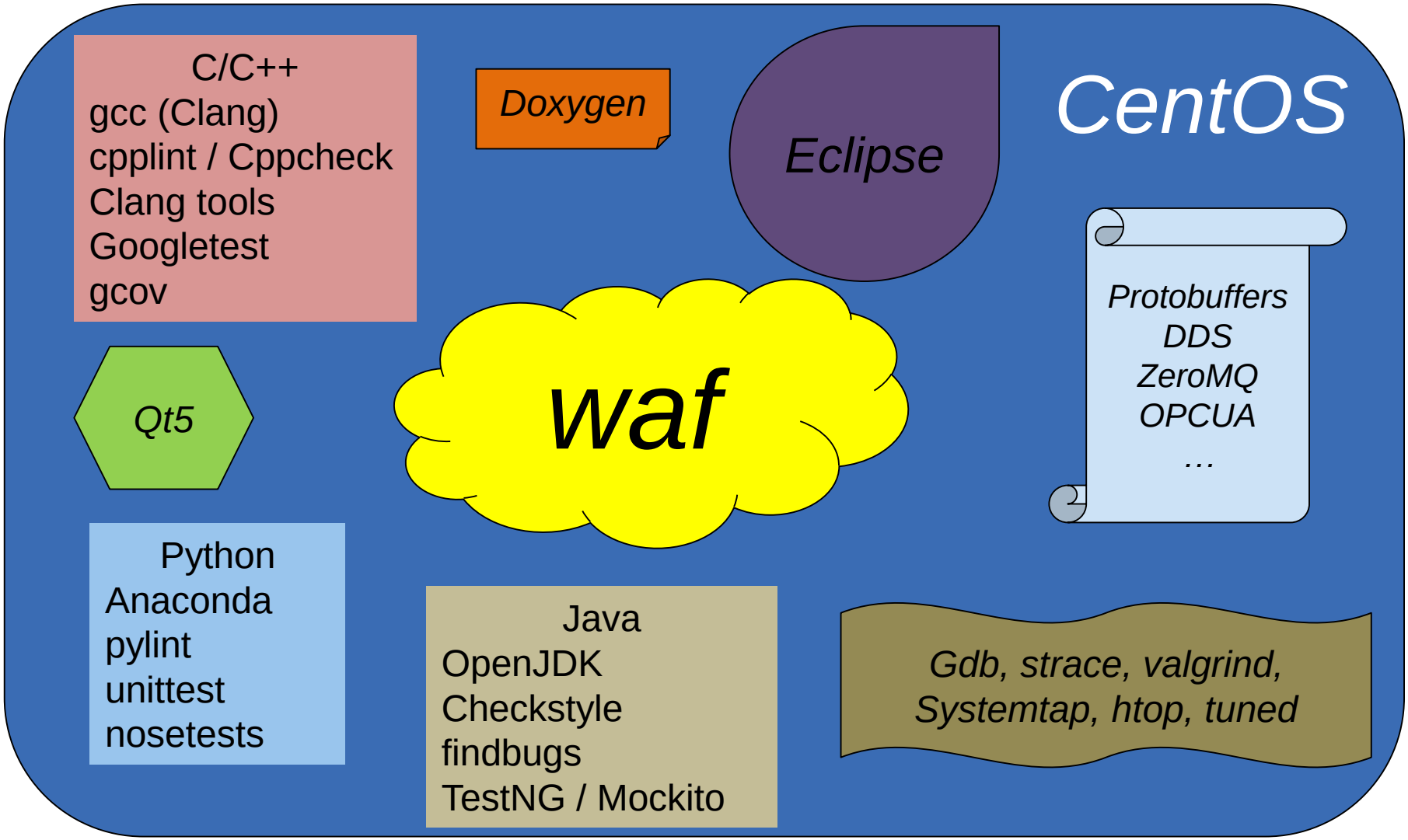
ELT Development Environment

- The next project that will see the light in 2024 and will be in operation until 2060+
- Opportunity to update technologies used
- Sensibility for Open Source has grown since previous projects
  - Lessons learned from closed software
  - In general major adoption globally
- Newer technologies introduced:
  - Linux RT, DPDK, OpenBLAS
  - Waf, CLANG tools, Anaconda Python
  - Docker, Terraform, Nomad

Build system challenges

- Single build system for C++ / Python /Java
  - Reliable partial builds
  - Full parallelization
  - Requires less specific knowledge
- Automatic dependency management
- Efficient and parallel
- Off-tree builds
- Ease of integration with new tools
- Logging and debugging support

DevEnv Overview



waf

- Open source project started in 2005
- Entirely Python based (2.5 -> 3.6)
- Focus on:
  - Portability
  - Speed of execution
- Efficiency on condition of rebuilds
- Supports many languages and tools; expandable
- Users: Samba, RTEMS, Ardour, game companies

https://waf.io/

waf

- wscript: build scripts defining configuration, options and build steps
  - Python code
  - Interaction with the waf framework
- Command line execution of phases
  - configure
  - build
  - test
  - install / dist
  - Custom commands

waf: an example

```
def options(opt):
        opt.load('compiler_cxx python pyqt5 ')
def configure(conf):
        conf.load('compiler_cxx python pyqt5 ')
        conf.check(header_name='stdio.h', features='cxx')
        conf.check_python_version((3,5,0))
def build(bld):
        bld.shlib(source='a.cpp inc/a.h', target='alib', export_includes='inc')
        bld.program(source='m.cpp', target='app', use='alib')
        bld.stlib(source='b.cpp', target='foo')
        bld(features="py pyqt5",  source="src/test.py src/gui.ui",
                install_path="${PREFIX}/play/", install_from="src/")
```

wtools

- wscripts are readable and easy but still…
- wtools as a layer for:
  - Simplification for common tasks for users
  - Centralized maintenance and roll-out of new features
  - Easier to enforce certain practices
- Can reduce wscript to a single line:

```
declare_cprogram(target=”foo”, use=”bar”)

declare_jar(target='jarEx', manifest='src/manifest’)

declare_pyqt5program(target='pyqt5example')
```

- Tasks for primary artifacts and additional ones are created: tests, installation, linting …

wtools

- Based on a set on conventions:
  - Directory structure, file positioning, file naming
- doc/
- interface/
- resources/
  - resources/audio
  - resources/config
  - resources/images
  - …
- src/
  - src/include
  - src/resources
- test/

wtools

- Currently supporting:
  - C/C++ program, shared and static library,
  - Python program and package,
  - Qt5 C++ or Python programs and libraries
  - Java JAR packages
  - Protobuffer / DDS / internal IDL
  - Configuration only modules
- Custom modules that leverage full waf can be created for specific needs not included in wtools
  - Mixed languages

ESO Open Source Contributions

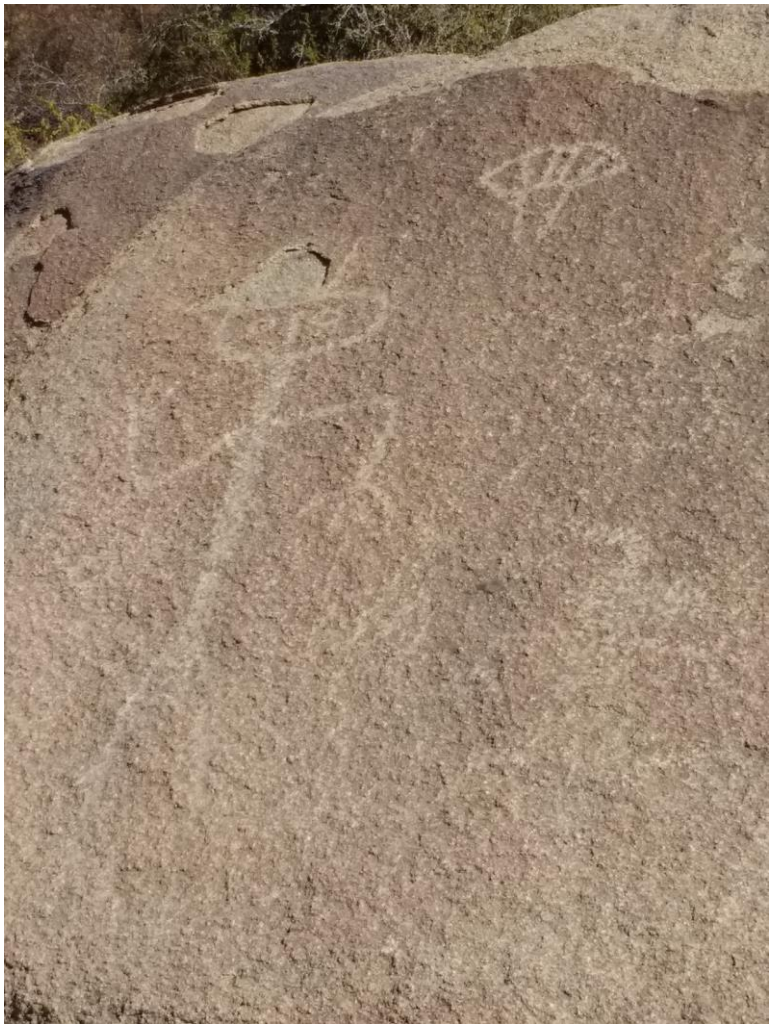
- Software:
  - ACS: Alma Common Software
  - Astronomical/scientific software under GPL:
    - ESO-MIDAS, CPL
  - IAS: Integrated Alarm System
  - Contributions to projects used
- Science archive data:
  - Data available to everyone after a period of exclusive usage by the Principal Investigator (usually 1 year)
- Images and videos (including UltraHD, fulldome and VR 16k) released under Creative Commons 4.0

FAQ

So when are you going to Mars?

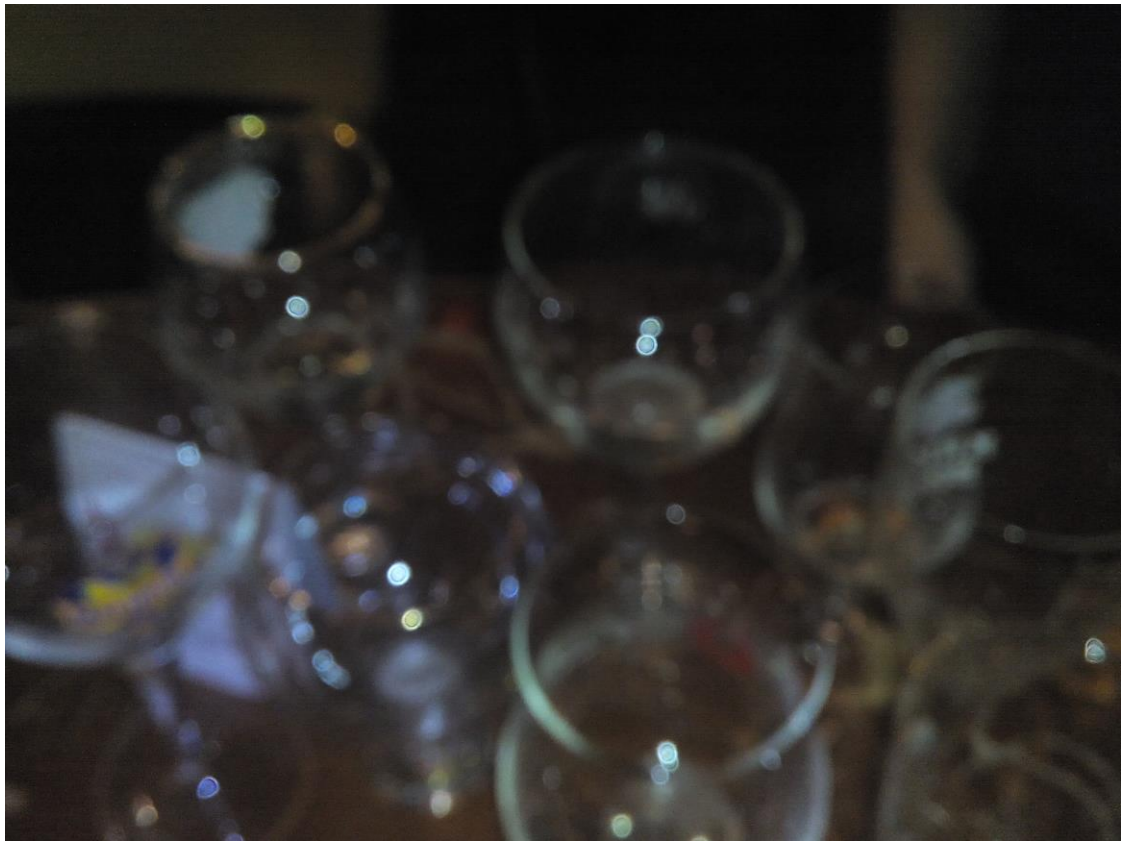
FAQ

Is there life out there?



FAQ

I’m interested in astrology too, can you tell fortune?



Questions?

- Our observatories can be visited, so you can visit our HQ and the newly opened planetarium.

Thank you Open Source Community!