Control Software for the LBT interferometer LINC-NIRVANA

27. March 2012
Technical Seminar
DESY Zeuthen

Jan Trowitzsch [trowitzsch@mpia.de]
Florian Briegel [briegel@mpia.de]
Max-Planck-Institute for Astronomy, Heidelberg

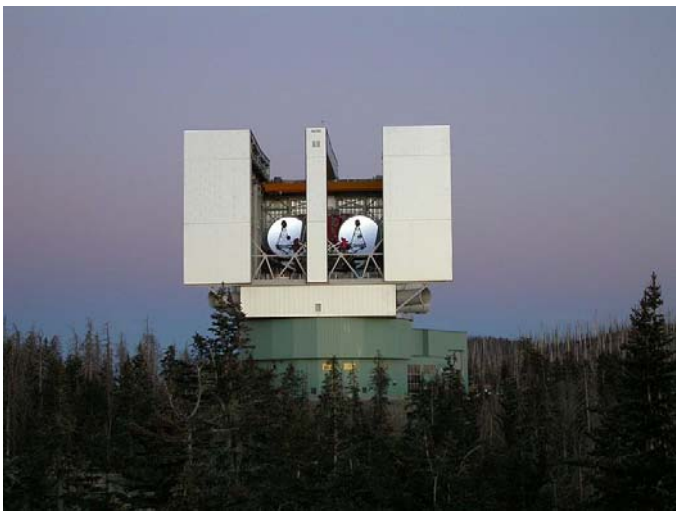
Outline

- Introduction to LINC-NIRVANA
- MPIA common software
  - Development tools and frameworks
  - Components and features
- Control Software for LINC-NIRVANA

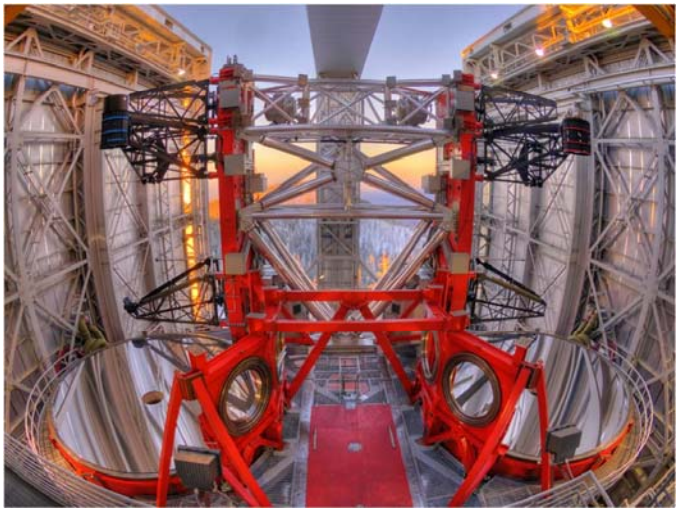
LINC-NIRVANA

- **L**BT **In**terferometric **C**amera and **N**ear-**I**nfra**r**ed / **V**isible **A**daptive **In**terferometer for **A**stronomy
  - Imaging beam combiner for the LBT
- Multi-conjugated Adaptive Optics: MCAO
  - High-layer and ground-layer sensing and correction using adaptive mirrors
  - Guidestars for wavefront measurement
- Several rotating fields

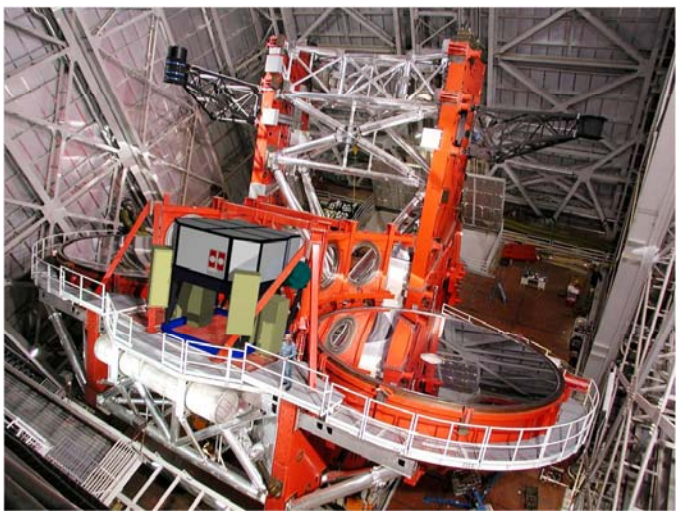
Large Binocular Telescope



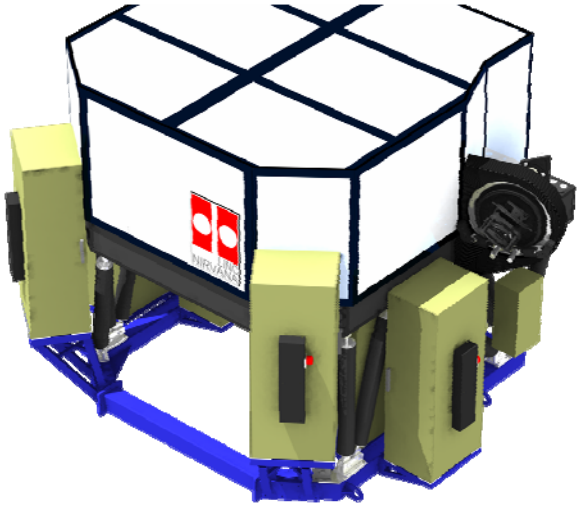
Large Binocular Telescope



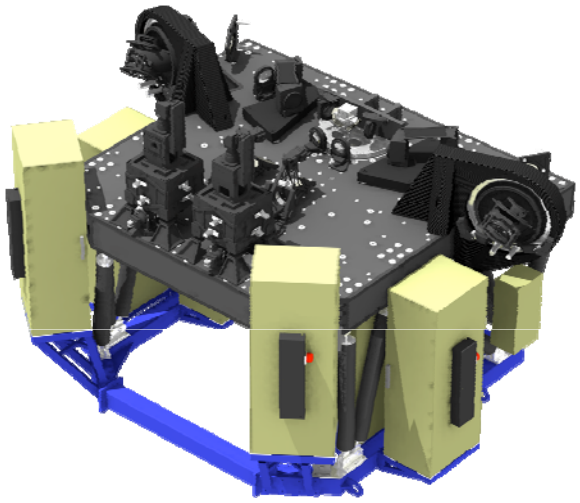
LINC-NIRVANA



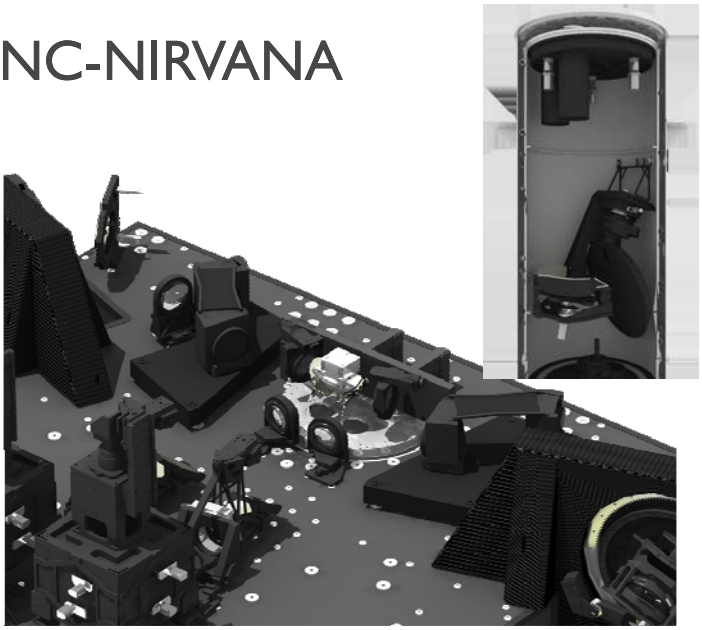
LINC-NIRVANA



LINC-NIRVANA



LINC-NIRVANA



What needs to be controlled?

- 4 Wavefront sensors → AO Loops
- 6 De-rotating elements
- Detector positioning and readout
- Many movable elements on motorized stages (ca. 120)
- Interaction with the telescope

- We will have multiple services running on 8 different computers

MPIA Common Software

- TwiceAsNice
  - Framework for developing service oriented architectures – SOA
  - Linux only
  - Based on Ice and Qt
1. Neubau
2. Nice
3. Basda
   - Mocca, Tempo, Vino, IO, Misda…

Third party software

- Internet Communications Engine (Ice) from ZeroC – *www.zeroc.com*
  - Distributed computing platform
  - Provides powerful network infrastructure
  - Supports many languages: C++, Java, Python, Ruby…
- Qt and Qwt – *qt.nokia.com*
  - Crossplatform application and UI framework
  - Widgets for technical applications
- Boost – *www.boost.org*
  - Widely useful C++ source libraries

Development Organization

- Version control using subversion – SVN
- Using TRAC on top of SVN
  - Web-based software project management
  - Issue tracking system – Tickets
  - Enhanced wiki system (documentation)
    - Allows linking of sources or tickets to wiki pages
  - Plugins and enhancements:
    - Blog, Discussion, Roadmap based on tickets, Source browser … extendable
  - *trac.edgewall.org*

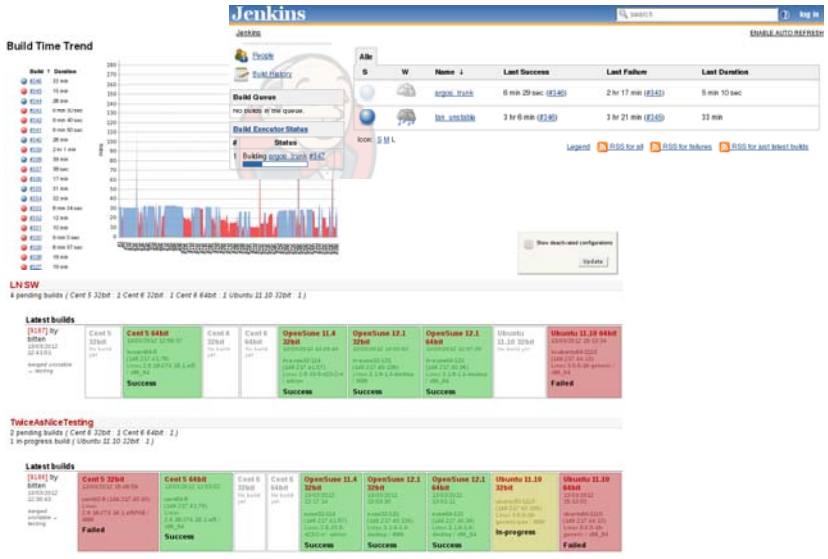
Development Organization

Development Organization

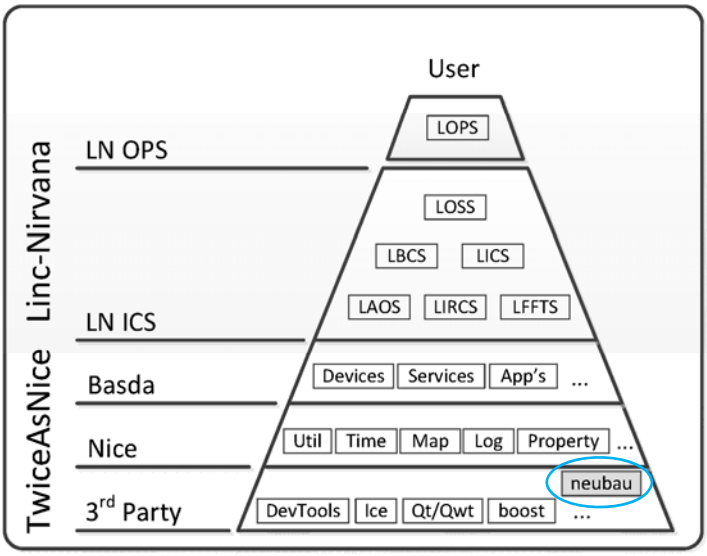
- Continuous integration testing using:
  - *Bitten*:
    - continuous integration plugin for Trac
    - Python-based framework for collecting various software metrics via continuous integration
  - *Jenkins*:
    - Extendable open source continuous integration server
    - Several plugins available

  - *bitten.edgewall.org*
  - *jenkins-ci.org*

Continuous Integration



Software Overview



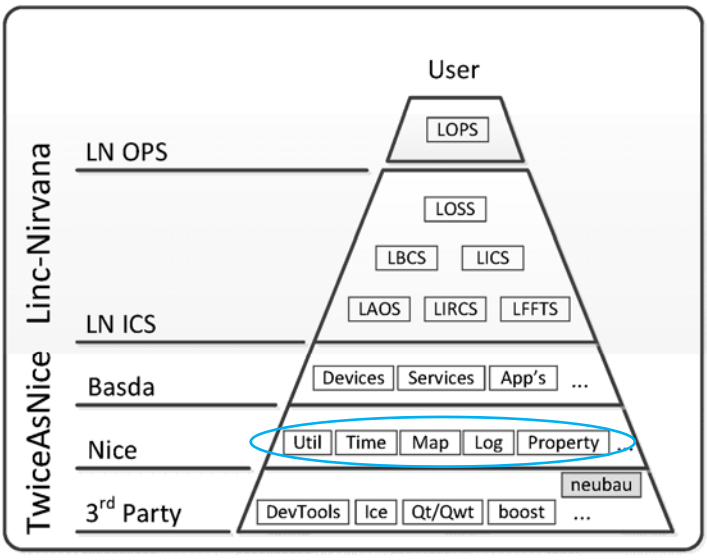
Neubau

- Perl script for creating auto tools based projects
- Helps to manage projects
  - Adding subdirectories
  - Keeping the *subdir* tree of *configure.in* an *Makefile.am* synchronized
  - Code template support, for creating pieces of code
- *Autoconf* macros for finding various software packages (boost, qt4, ice, …)
  - *gnu.org/software/autoconf*

Neubau

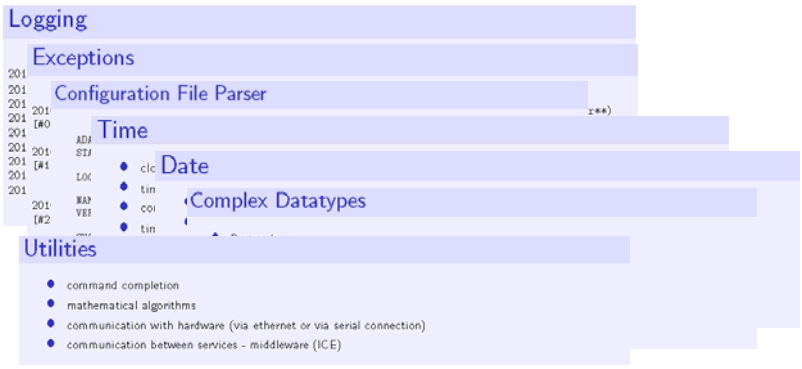
- *Automake* build rules for administration
  - Automatic subversion add
  - Building of rpm packages
  - Doxygen documentation …
- *Automake* build rules for compiling not supported meta languages and formats
  - Qt, ice, png, nice, …
  - *gnu.org/software/automake*

Nice



Nice

- **N**ice **I**nstrument **C**ontrol **E**nvironment
- Basic functionalities:

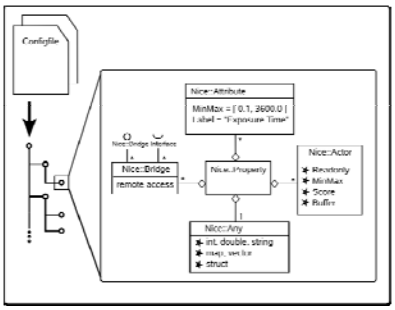


Nice

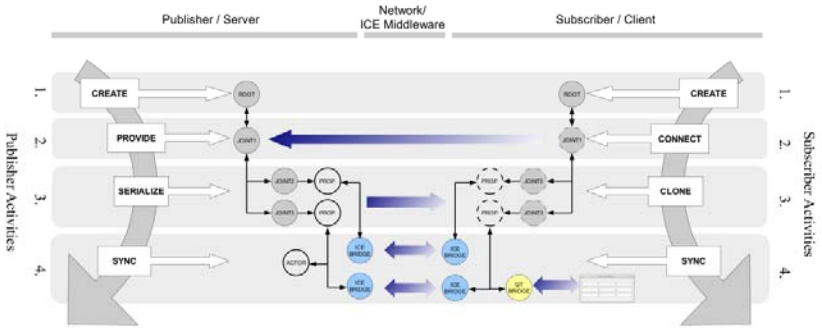
- Integration of Qt/Qwt
- *Ranger* tool for engineering GUIs
- Service/Device architectures with dedicated property trees
- Code generation
- Python interface

Heart and Soul - Properties

- Property:
  - Weak typed data container attached to a Joint
  - All supported data types of our framework can be placed here
  - May contain a set of bridges to communicate with outside world
  - May contain a set of actors to trigger actions whenever the internal value is about to change

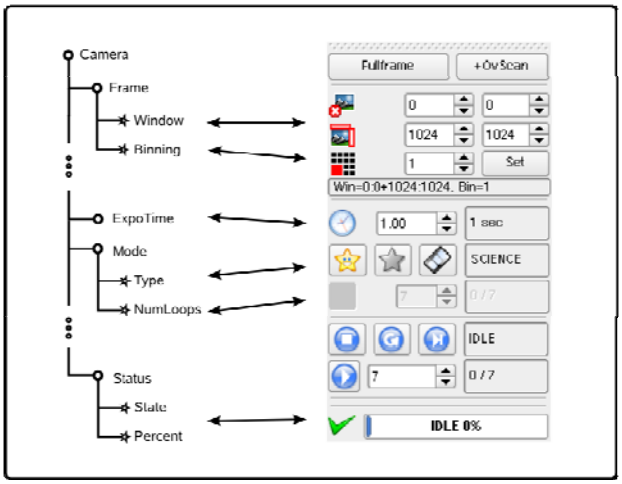


Service – Client Connections

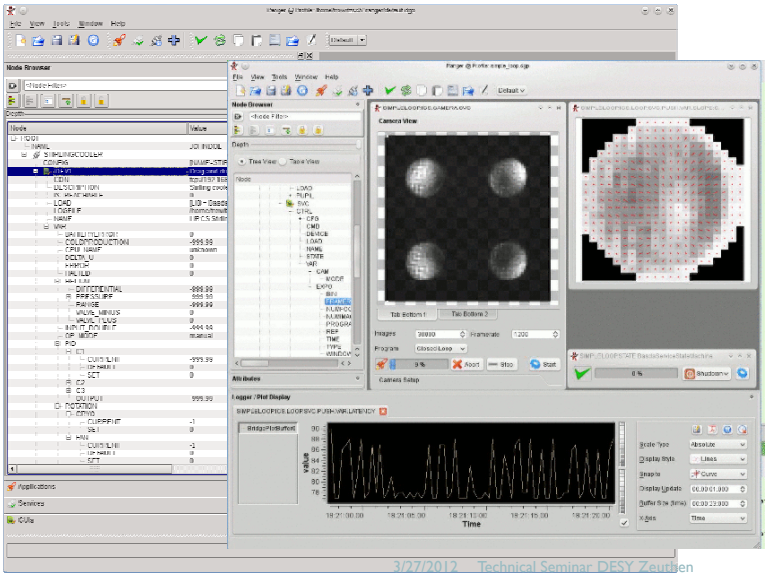


1. Server & Client create a Joint root node
2. Client connects to publisher's Joint node
3. Server sub-tree is cloned at client side and an Ice bridge is attached to each property
4. Property value changes are distributed via attached Ice bridges to all subscribers

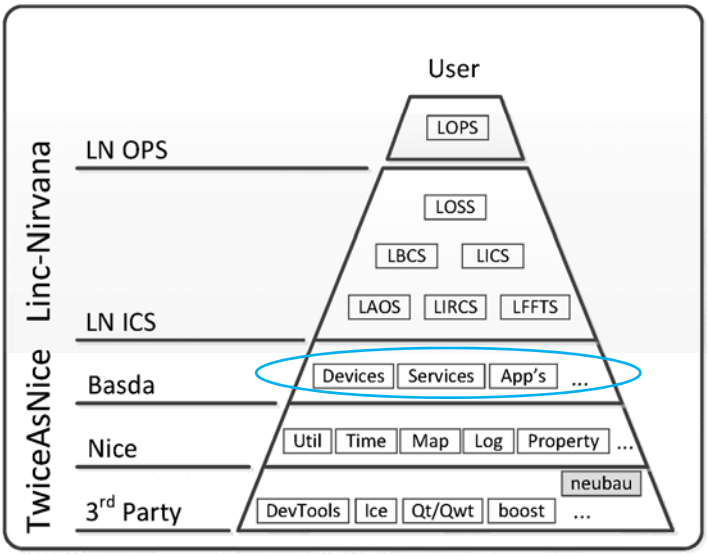
More Heart and Soul



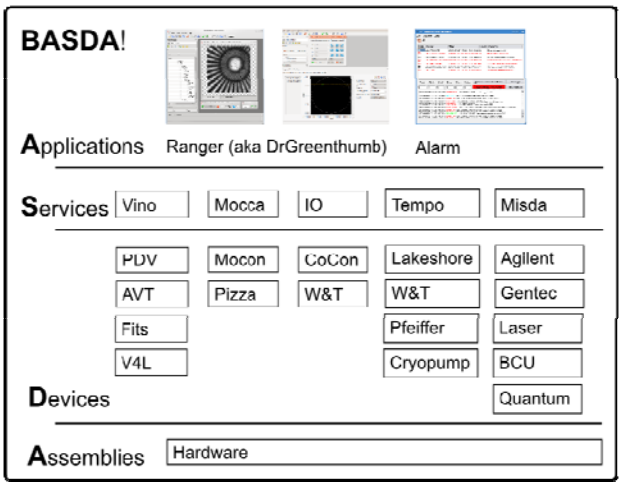
Ranger – Engineering Interface



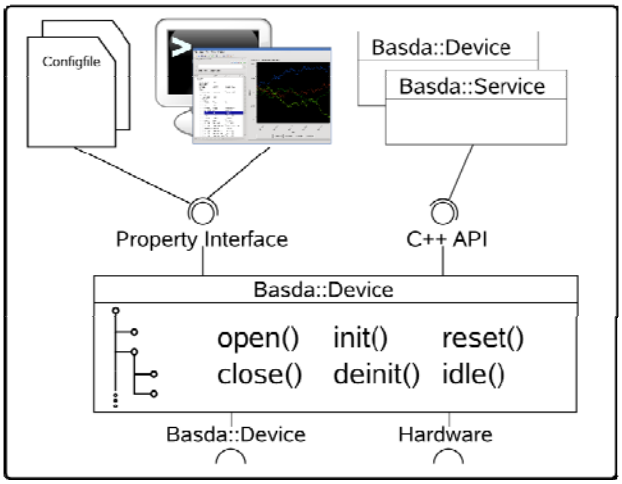
Basda



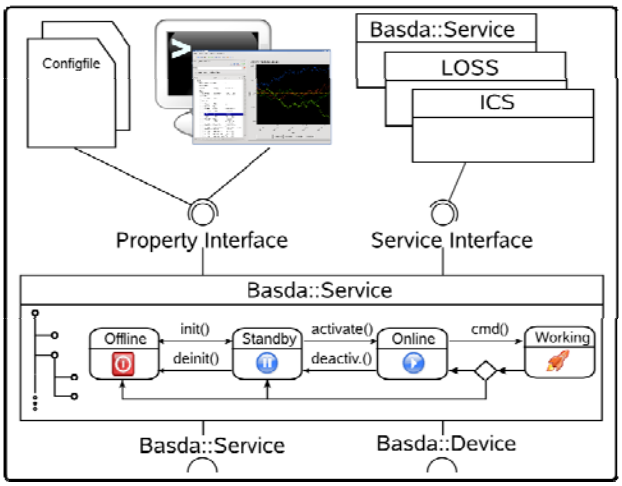
Basda - Overview



Basda - Device



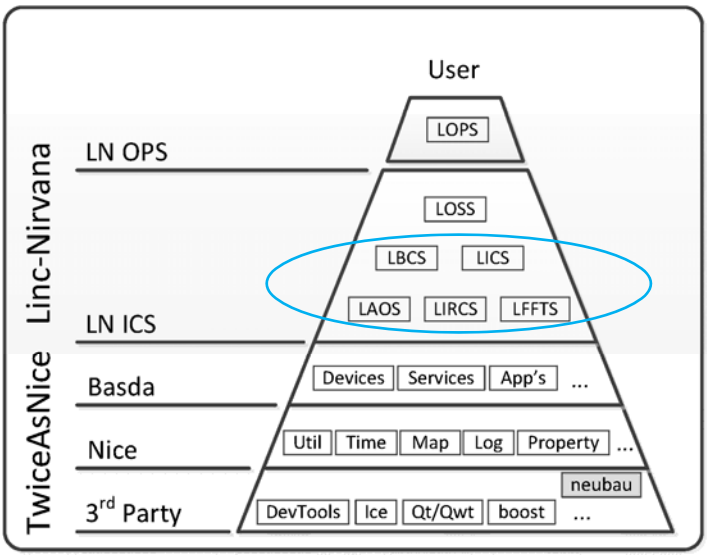
Basda - Service



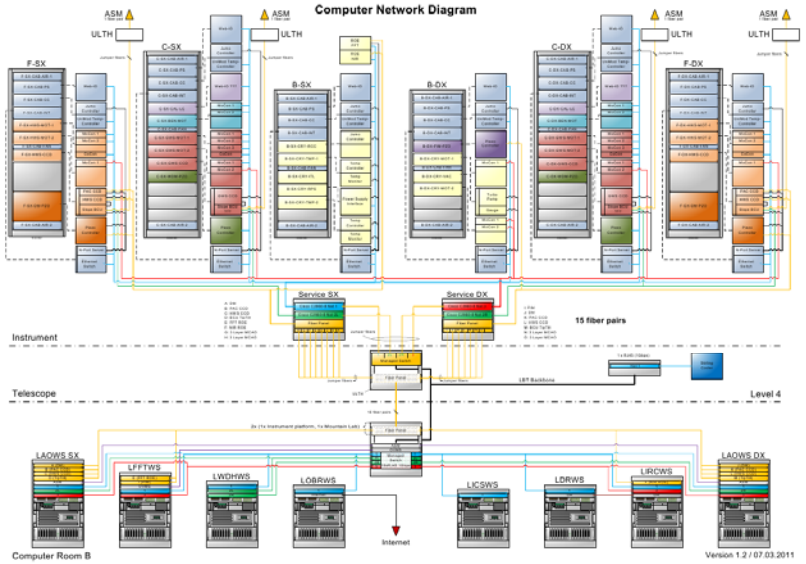
Basda – Interface Def. Language

- Asynchronous communication via bridge member functions
- What about synchronous stuff?
- General programming language independent interface
  - Similar to Corba or Sun RPC → **.basda*
- Basda2xx
  - Automatic generation of skeleton and stub code for the target programming language
  - Special property sub-tree *CMD* for synchronous invocations

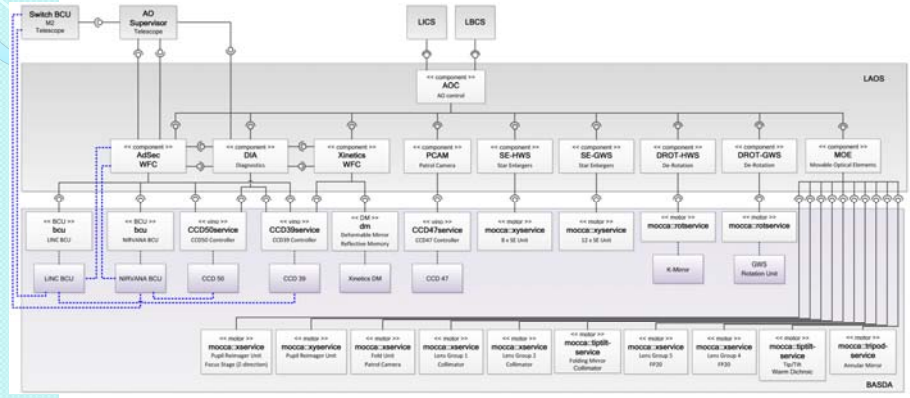
LN Instrument Control Software



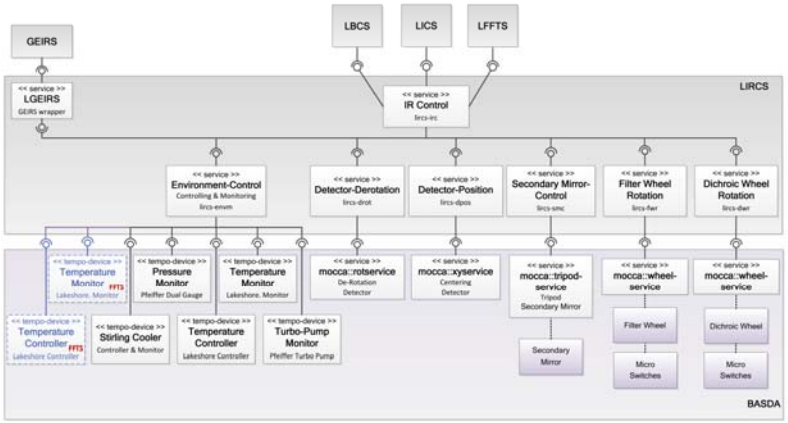
Computer Architecture



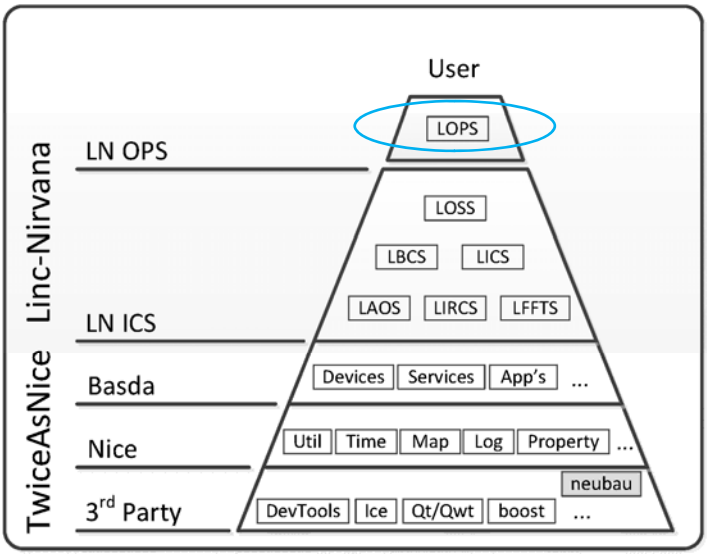
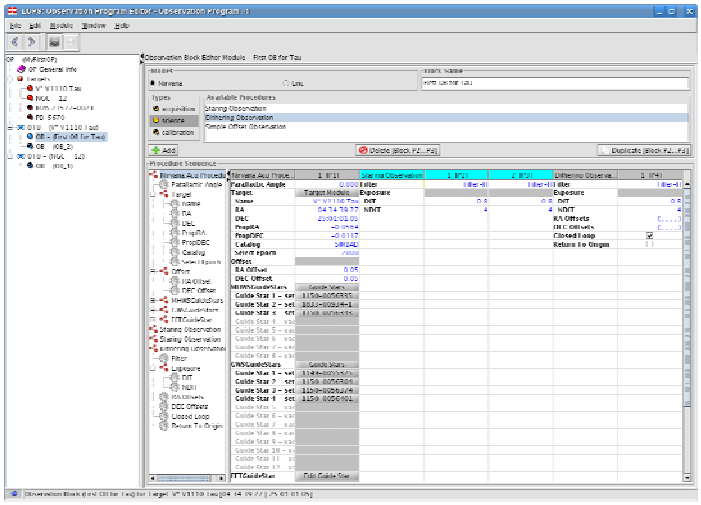
LAOS Sub-System



LIRCS Sub-System



Observation Preparation Software



Observation Preparation Software