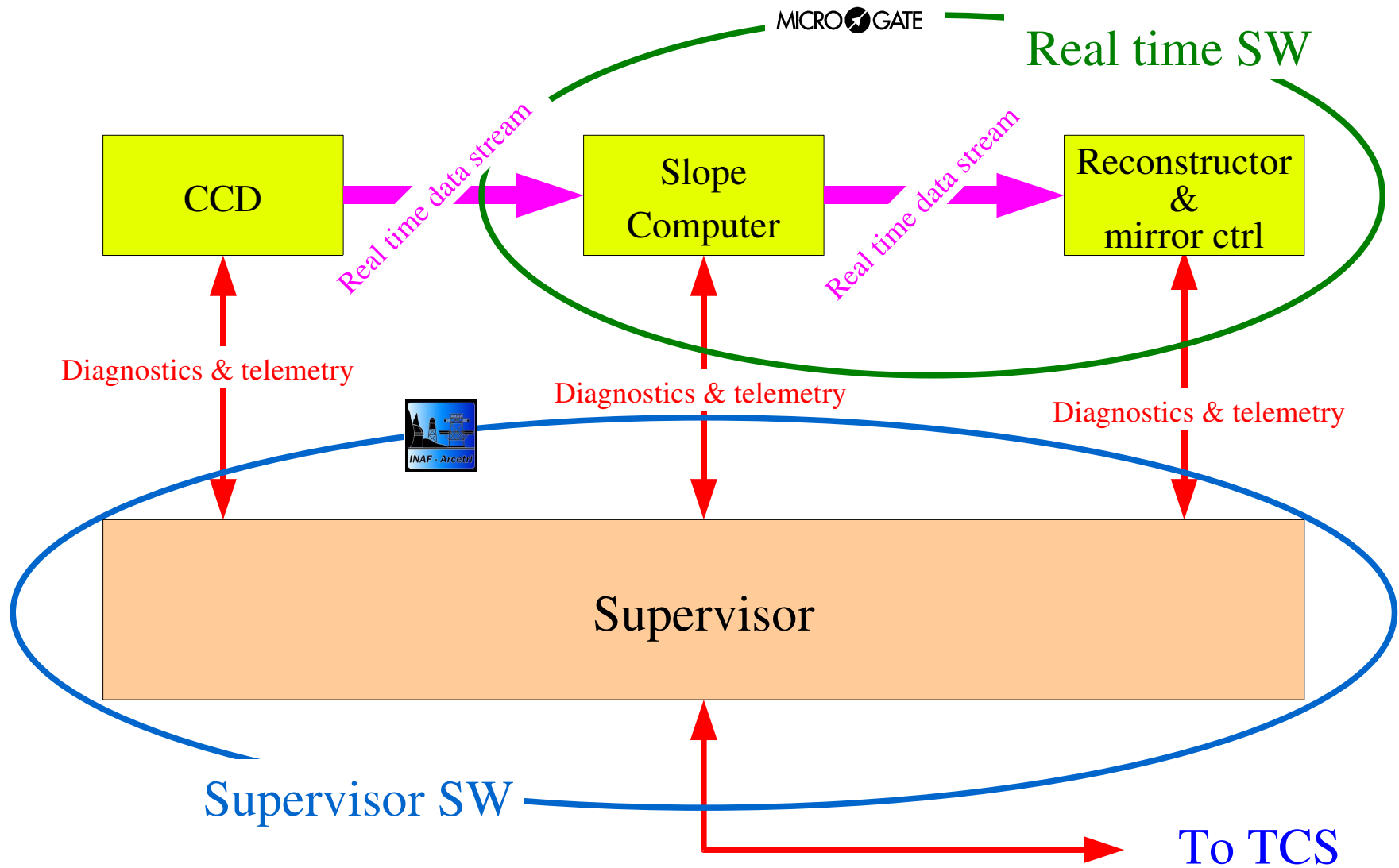

LBT-AO Progress Report Meeting

Arcetri, 10-11 November 2005

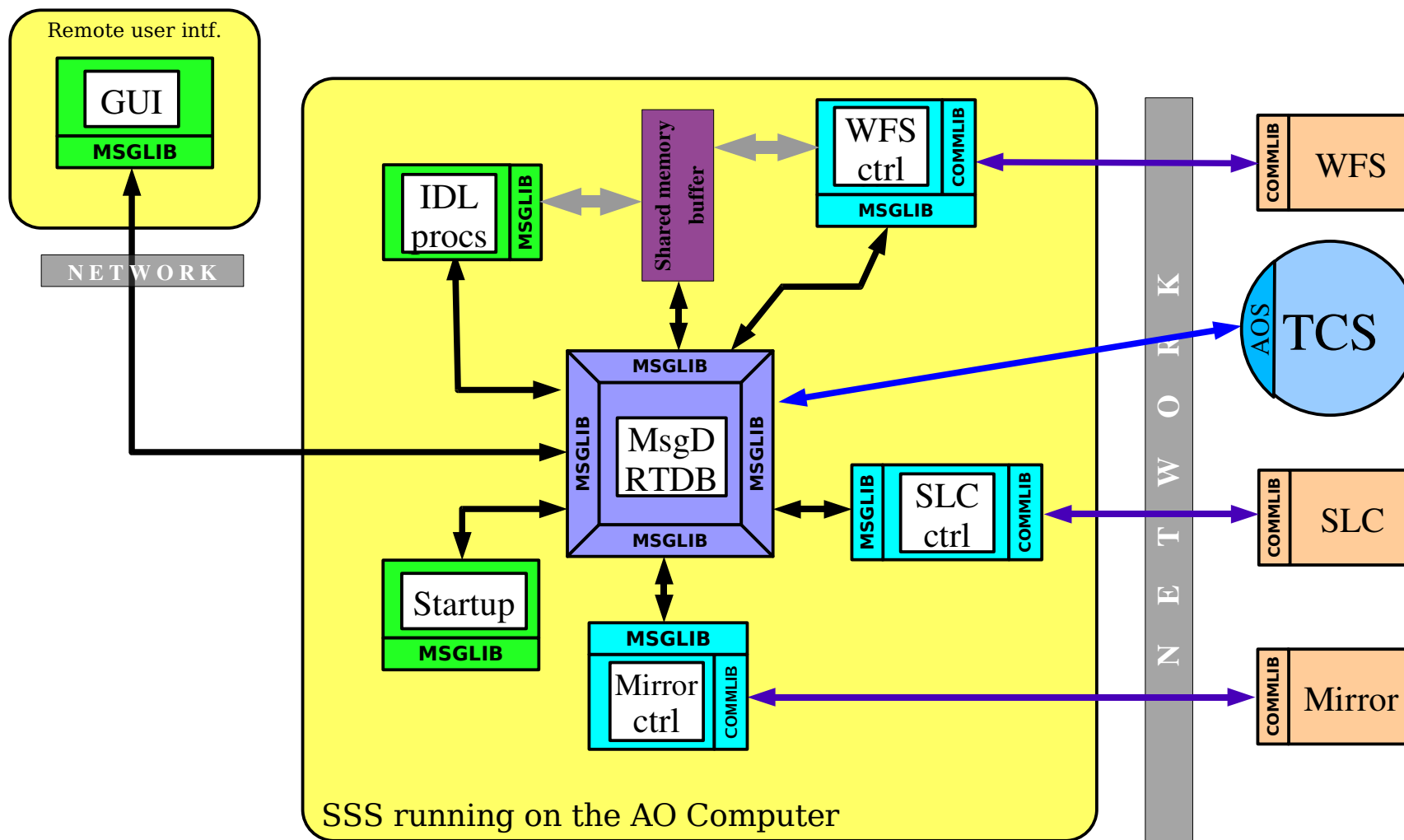
LBT-AO Software Status and Plans

Luca Fini, Alfio Puglisi, Lorenzo Busoni

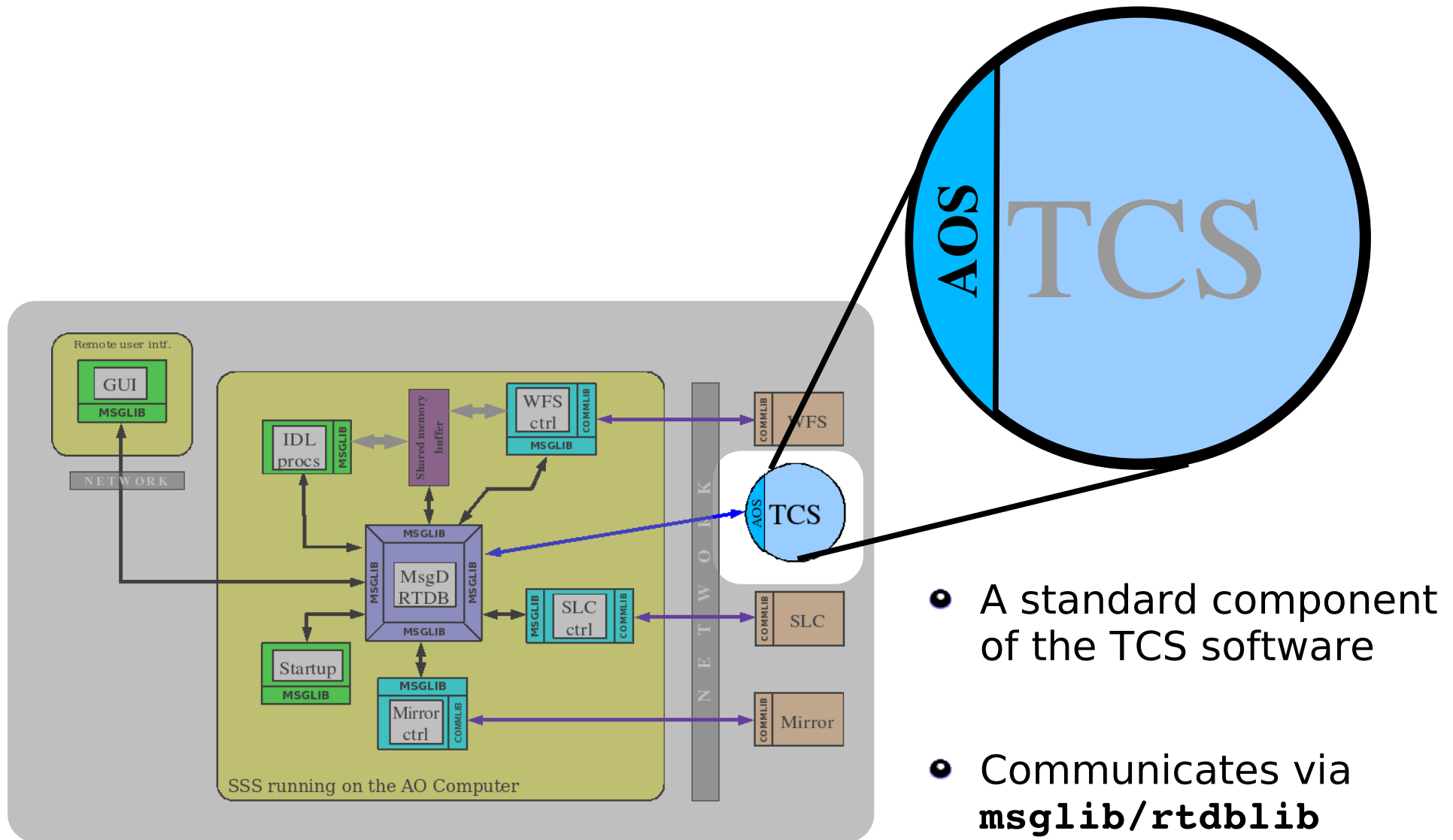
System architecture



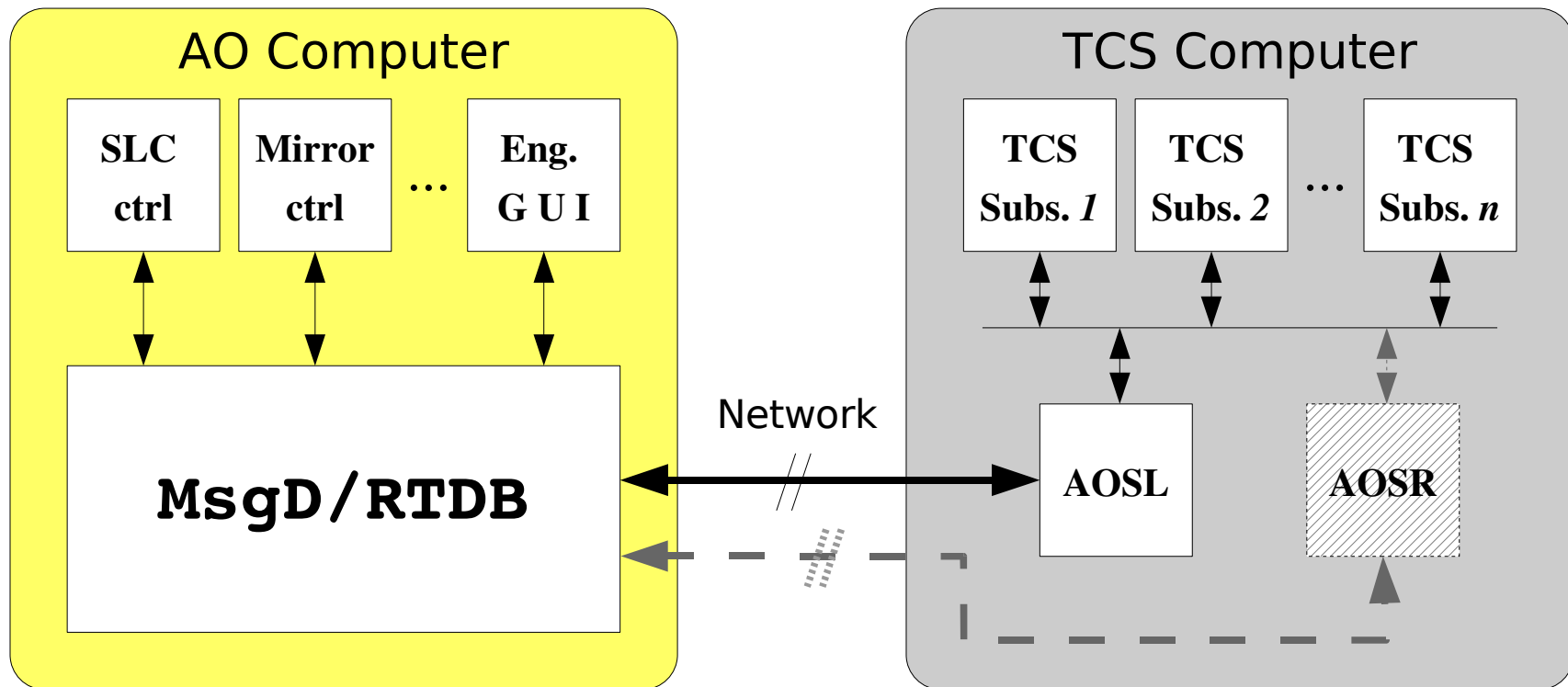
Supervisor architecture



AOS: the TCS to AO Interface



AOS



AOS Design Key Points

- AO-SW must be able to run standalone
 - Except for operations depending on TCS
- AO-SW must compile with no need of external libraries
- The AO-SW will run on a dedicated server
- The AO-SW will have a dedicated console for engineering tasks
- Interaction with TCS is provided via a TCS subsystem (AOS)
 - All dependencies are concentrated in the AOS
- AOS communicates with AO-SW through `msglib/rtdb1ib`

AO Supervisor states

- **ENGINEERING**

Used for non-observation tasks which need support from TCS. The AO Supervisor operates as **master** when requesting TCS services. Status and emergency requests from TCS are honored and properly managed.

- **STANDALONE**

Similar to engineering state, but no interaction with TCS is allowed or required. It may be considered a “protected” mode to allow maintenance operations while the TCS is operating on its own.

- **OBSERVATION**

Used during observation. The AO Supervisor operates as a **slave** and obeys to requests issued by the TCS. When in observation state the supervisor may also asynchronously send to TCS requests such as: mode offload, emergency status notifications and log information.

Supervisor functions - ENGINEERING state

- Housekeeping:
 - Startup,
 - Program and configuration data upload,
 - Hardware checkup,
 - Safe shutdown.
- Support calibration procedures.
- Support maintenance operations.

Supervisor functions - OBSERVATION state

- Set up observation modes
 - Diffraction limited
 - Adaptive
 - Extern tip-tilt
 -
- Communicate with TCS
 - Mode offload
 - Log status
 - Emergency
- Gather diagnostic data
- Perform system health checks.

Human interface

- **STANDALONE/ENGINEERING state**

An engineering GUI is provided for non-observation tasks. It operates on the AO Console and is independent from TCS. When in OBSERVATION state it will be essentially non-operational but may show engineering information on AO System status.

- **OBSERVATION state**

A panel of the standard TCS observation GUI will provide information related to AO status and functioning at the astronomer's console. It will also allow selection of operating modes and ways to modify selected parameters for an astronomer with "AO skills".

AOS Functionalities - 1

Querying TCS for information

- Telescope status
 - E.g.: observing, or AO-Engineering
- Derotator status
 - Including current position
- Hexapode status
- Telescope pointing status
 - Includes current coordinates
- Instrument status
 - Info on operating instrument
- Environment data
 - Temperature, humidity, air pressure, wind speed/direction, etc.

AOS Functionalities - 2

Information provided to TCS

- AO status
 - NOT READY, STANDALONE, ENGINEERING, OBSERVATION
- Observation mode
 - Provides detailed information on current mode and on individual devices (E.g.: flat mirror, WFS Ready, etc.)
- Image quality
 - During observation some image quality indicators will be provided to the observer's console

AOS Functionalities - 3

Commands issued to TCS

- Move Hexapode
- Rotate Derotator
- Offset Pointing
 - To support mapping operations
- Offload Modes
 - This is used during observation when some limit is reached
- Log Item
 - To leave traces into the telescope history
- Emergency Shutdown
 - When something dangerous happens

AOS Functionalities - 4

Commands accepted from TCS

- Prepare observation
- Start observation
- Stop observation
- Set Tip-Tilt
- Reset
- Shutdown
- Emergency Shutdown

AOS – Who, When, Where

- AOS design and implementation: Arcetri
 - With a local copy of TCS software
- AOS design and implementation support: Tucson
 - Hands on session fall 2005: preliminary
 - Hands on session fall 2006: test against TCS @ telescope
- AO observer's Interface: Tucson
- Solar Tower Tests: Arcetri
 - Against working AO System
- Commissioning: Arcetri
 - With local staff support

AO Supervisor State of Art

MsgD/RTDB	Release	The Message Dispatcher and Real-Time Database was the first component developed and is intensively used in all running programs.
BCUCtrl	Release	Device controllers for BCU based hardware (the Adaptive Secondary and the Slope Computer).
Engineering GUI	Beta	Graphical User interfaces for laboratory use (and engineering operations at the telescope).
PowerCtrl	Release	Control programs for power switches and similar devices.
SimpleMotorCtrl	Beta	Control program for stepper motors. Used for: Rerotator, Filter Wheels, ADC.
StageCtrl	Beta	Control program for bayside linear stages.
MirrorDiagnostics	Alpha	Adaptive Mirror diagnostic programs.
StartUp	Alpha	Startup procedures.
TTCtrl	Release	Control program for the Tip-Tilt signal generator board.
WFSCtrl	Release	Control program for the WFS CCD camera.
AOS	Alpha	Interface to TCS.
Libraries	Release	The complete set of AO-SW libraries.
IDL	Beta	Support for IDL procedures.
PyModules	Beta	Wrappers for Python code and high level Python library.



AO-SW Development Schedule - 1

- November 2005: Preliminary AOS integration test in Tucson
- December 2005: continuing development
 - Engineering GUI update
 - High level automation scripts
 - Prototype AOS with full functionality
 - Documentation
- July 2006: solar tower AO System tests support
- November 2006: AOS integration test at the telescope
- December 2006: AOS integration test at Arcetri's solar tower
- February 2007: Commissioning begins
 - ENGINEERING functions test as part of commissioning
- April 2007: Commissioning continues
 - OBSERVATION functions test as final part of commissioning
- May 1st 2007: AO-SW operational

AO-SW Development Schedule - 2

- May 2007: AO-SW upgrade starts
 - Using lessons learned during commissioning and astronomical usage
- November 2007: AO-SW + AOS Vers. 2.0 at the telescope
 - To support unit #2 commissioning
- February 2008: AO-SW + AOS Vers. 2.1 operational
 - Together with AO System unit #2