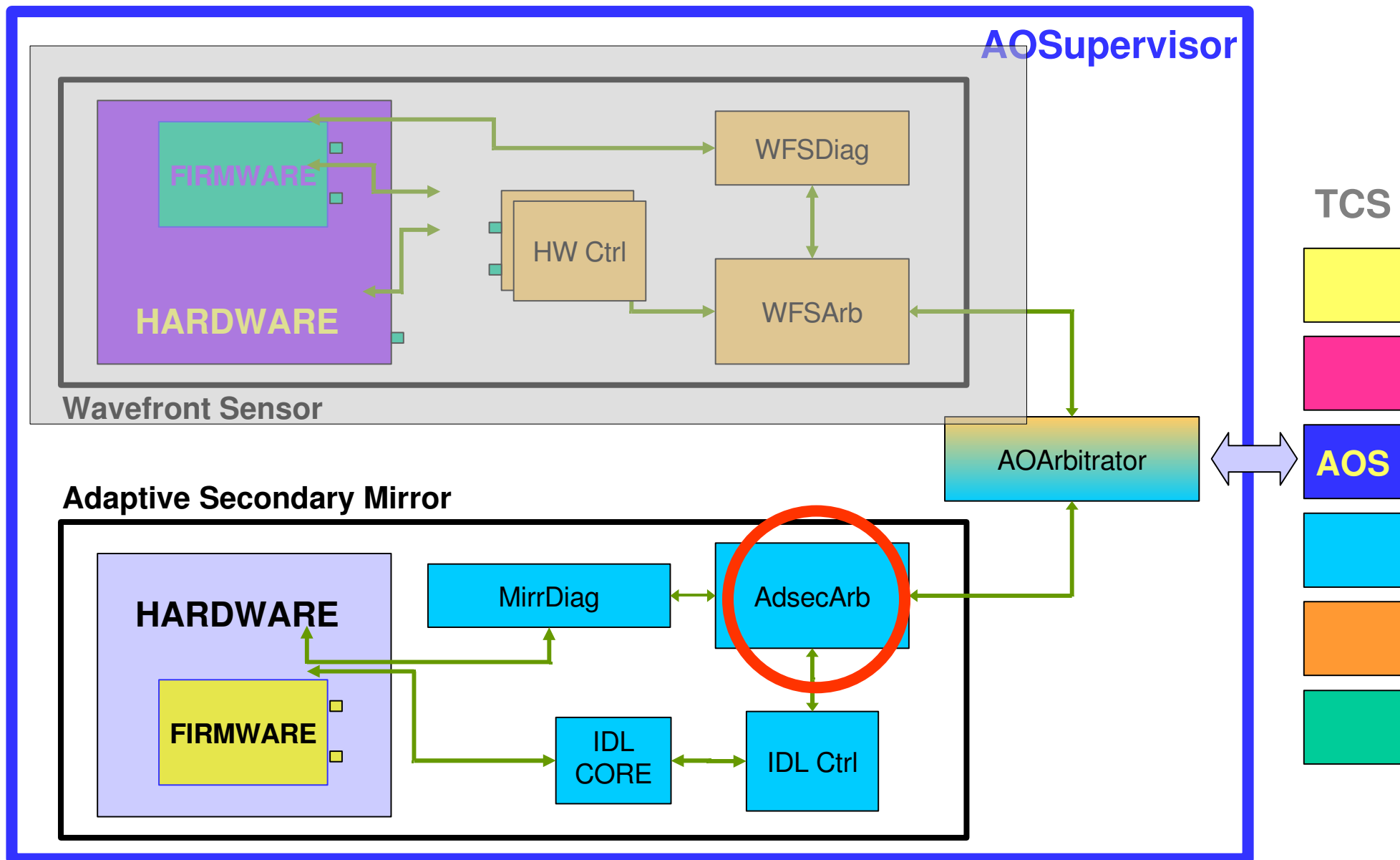
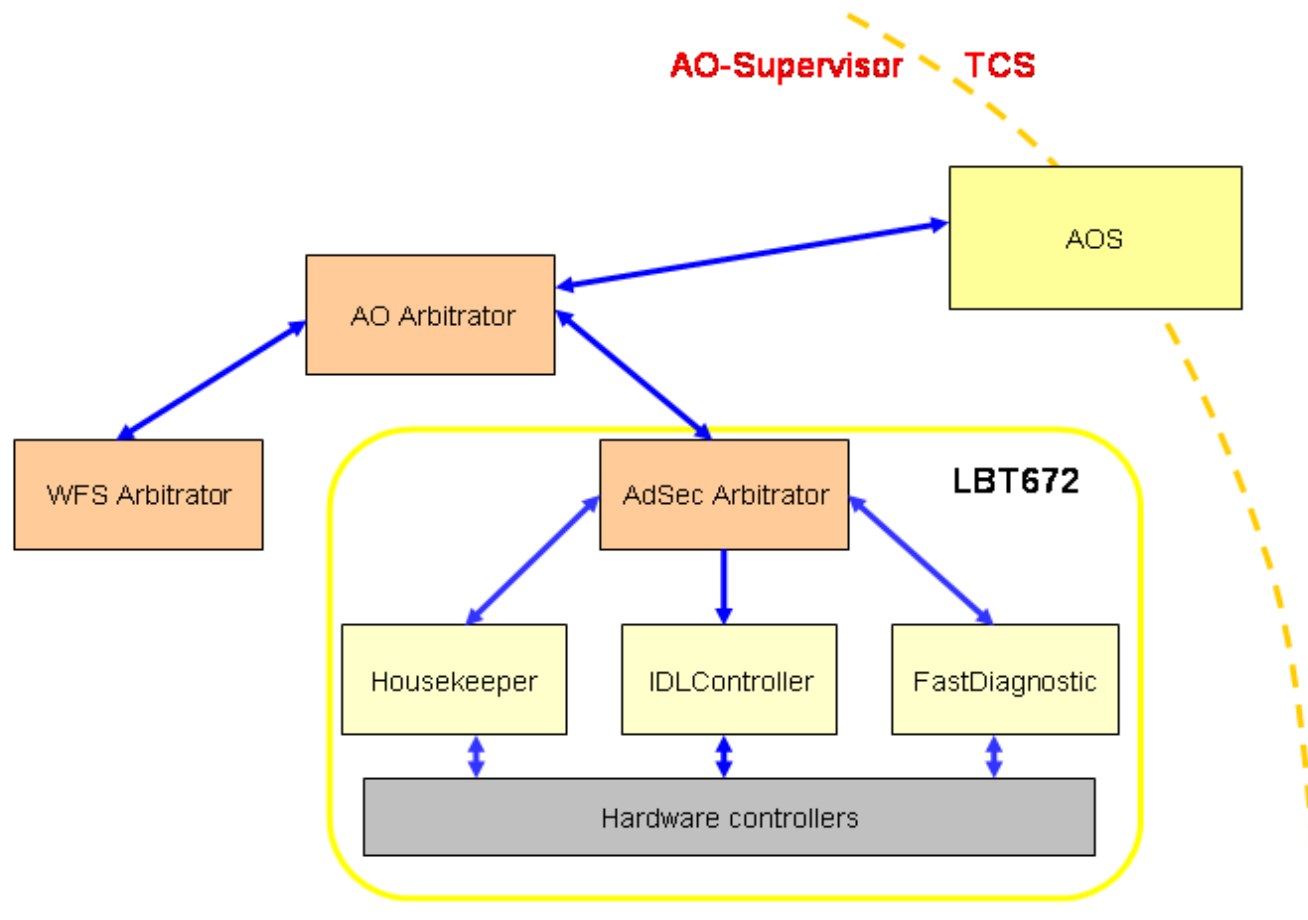

LBT Adaptive Secondary Finite State Machine

L. Busoni, M. Xompero, D. Zanotti, A. Riccardi

Where?



AdSec hierarchy



AdSec Arbitrator

- **Interface** of the Adaptive Secondary system toward higher Arbitrators.
- Complete control of the AdSec system through ~15 RTDB parameters.
- Implemented as a finite state machine with 10 states that correspond to 10 different hard/software configurations of the AdSec unit.
- **Centralize** all the commands directed to AdSec as well as the notifications from AdSec to the AOS or to the WFS.
- FSM coded in Python, actions scripts in IDL.

Example: setting up the mirror



in GUI, AOS, AOArbitrator, console, ...

```
SetVar(L.ADSEC.REQ_STATUS, 'SET_AO')  
SetVar(L.ADSEC.REC_MAT_FILE, 'conf/a_matrix.fits')  
SetVar(L.ADSEC.TRIG_SWITCH_STATE, 1)
```

Outside AdSec: as easy as writing 3 strings in the RTDB

Inside AdSec: ~70 IDL procedures are called

```
if self.__Y == AdSecFSM.Operating:  
    if e == self.SwitchStateEvent and self.ReqStateIsSetAO():  
        try:  
            self.SetAOAction();  
        except:  
            return  
        self.__Y = AdSecFSM.SetAO
```

```
Fast Diagnostic parameter set to restrictive values  
Reload Feed Forward Matrix... done  
Reload Reconstructor parameters... done  
Control stopped and DACs cleaned. Coils enabled.  
Mirror setting procedure... done  
Set derivative and proportional gain... done  
Flattening mirror... done  
Fast Diagnostic parameter reset to no restrictive values
```

in IDL lbt_set_ao.pro