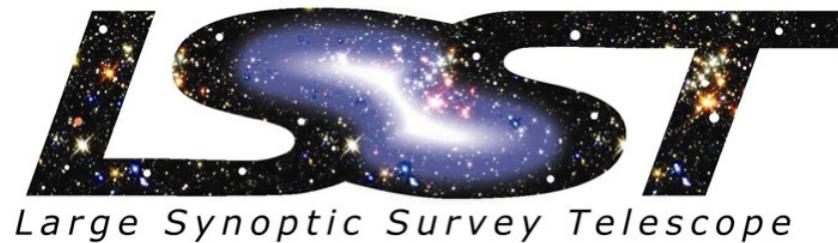


# Camera Control System (CCS) Filter Changer System (FCS)

Réunion LSST France, APC, Paris

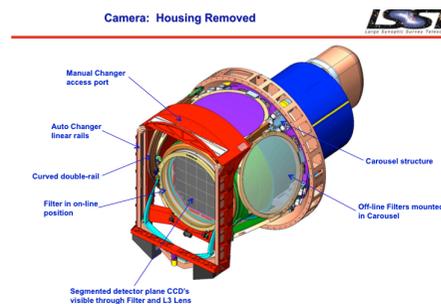
18 décembre 2013



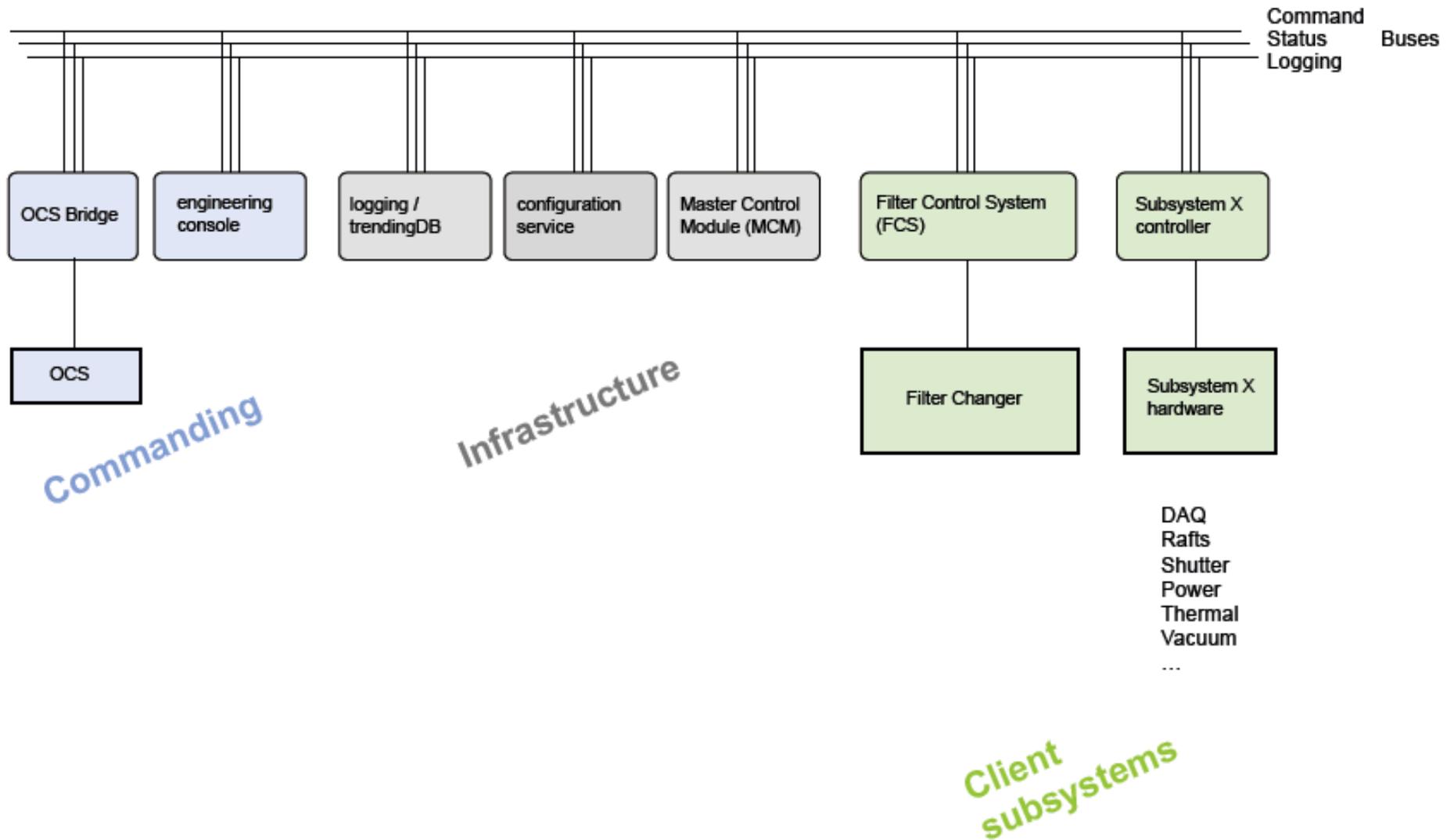
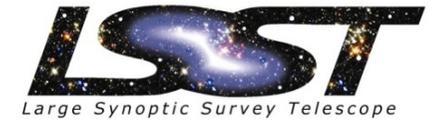
Eric Aubourg, Bernard Amade,  
Françoise Virieux, IN2P3-APC, Paris, France

- **Présentation générale du CCS et du FCS**
- **Ressources humaines**
- **CCS**
  - **Principales réalisations en 2013**
  - **Plans pour 2014**
- **FCS**
  - **Principales réalisations en 2013**
  - **Plans pour 2014**
- **Les revues en 2013**

- **CCS : Camera Control System** : logiciel de contrôle et monitoring de toute la caméra : une quinzaine de sous-systèmes autonomes qui communiquent entre eux par l'intermédiaire de 3 bus :
  - Qui reçoivent des ordres de haut niveau en provenance du télescope ou des consoles (bus de commande), les exécutent et rendent compte de leur exécution
  - Qui enregistrent des données sur leur état dans la base de données de télémétrie (bus de status)
  - Qui journalisent leurs activités (bus de log)
  - Qui sont configurables de manière plus ou moins conviviales par l'utilisateur final
- **FCS : le logiciel de contrôle du changeur de filtres, un des sous-systèmes de la caméra**
  - **Carousel**
  - **Auto-changeur**
  - **Loader**



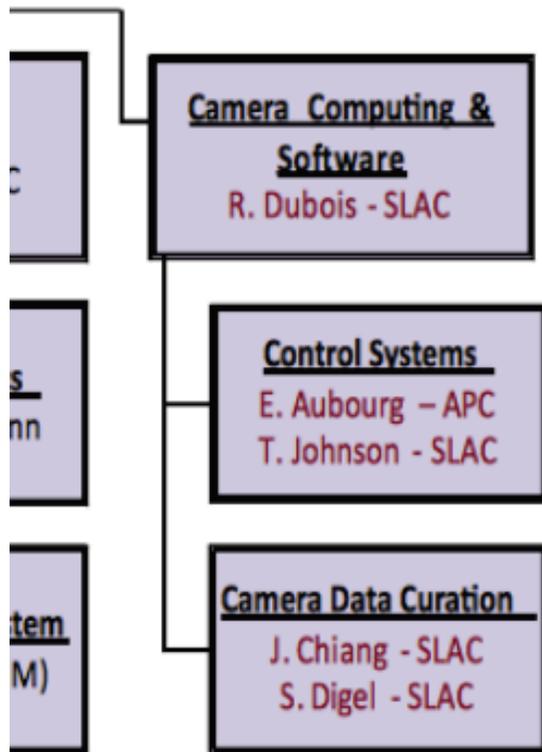
# Description générale du CCS et du FCS



- **CCS : fournir des outils pour les développeurs de sous-systèmes**
  - Les bus de communication
  - La configuration des sous-systèmes
  - La base de données de télémétrie
  - Des outils pour tester
  - De la documentation
  - Du support aux développeurs des sous-systèmes
- **FCS : fournir le logiciel final et celui pour les bancs de test**



- **Nouveau en 2013 : Eric Aubourg, big boss du CCS**



New camera computing group (R. Dubois)

New Camera Control System management :

Éric Aubourg — CCS Subsystem Manager (Physicist)

Tony Johnson — CCS Architect / Engineering Manager

- **A SLAC**

- Tony Johnson : chef de projet technique pour tout le CCS consoles, “trending data”
- Max Turri : GUI pour le FCS, “trending data”
- Stuart Marshall : system engineering
- Owen Saxon : développeur des logiciels de contrôles pour le “shutter”, le refrigeration system, les bancs de test de l’électronique
- Heather Kelly : banc de test des CCD

- **A l’APC**

- Eric Aubourg : physicien référent, auteur des concepts de base du CCS
- Bernard Amade : développeur du “framework”
- Françoise Virieux : développeur du FCS

- **Au LPNHE**

- Eduardo Sepulveda, Diego Terront : banc de test des CCD

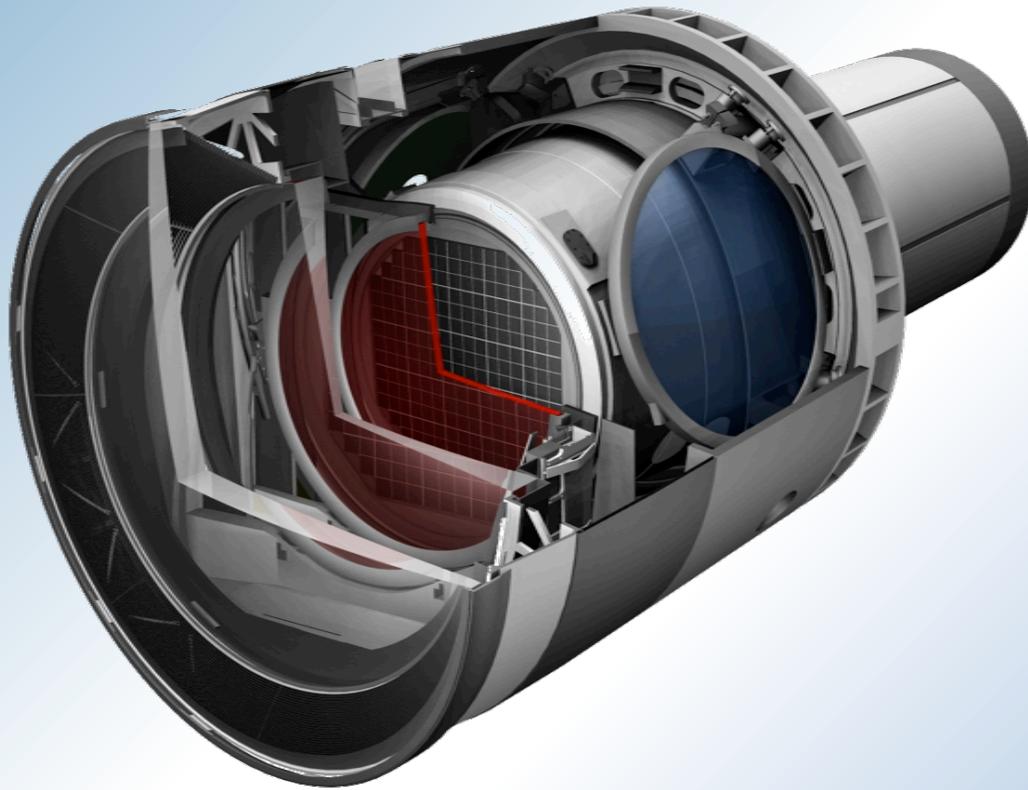
- **Au CPPM, Marseille : Patrick Breugnon**
  - single-filter-test,
  - prototype changeur de filtres échelle 1
- **Au LPSC, Grenoble : Galdric Marcotte, Eric Lagorio**
  - banc de test du loader
  - prototype changeur de filtres échelle 1
- **Au LPC, Clermont : Philippe Chassagny, Patrick Lafargette**
  - Prototype changeur de filtres échelle 1
- **A l'APC, Paris : Françoise Virieux**
  - développement du FCS
  - support aux bancs de test

- **Contrat de Bernard Amade, principal développeur du framework, prolongé jusque fin décembre 2014 sur des crédits US (projet LSST)**
- **L'APC qui a financé les 2 premières années du contrat avait eu la promesse d'obtenir un CDD IN2P3 avec possibilité de titularisation : affaire à suivre !**





**CCS**



- **Base de données “Telemetry” et “trending data”**
  - Outils pour stocker des données sont opérationnels
  - Outils pour relire ces données (Rest server) et tracer des “plots” dans la console JAS sont opérationnels
- **Configuration des sous-systèmes**
  - Conception du système terminée
  - Réalisation faite en grande partie
    - Un sous-système est décrit en langage groovy
    - Les propriétés modifiables sont décrites dans un fichier `.properties`
    - Les variables peuvent être changées par l'utilisateur au cours de l'exécution
  - Elimination du framework Spring
- **Bus de commandes : commande Halt avec un délai**

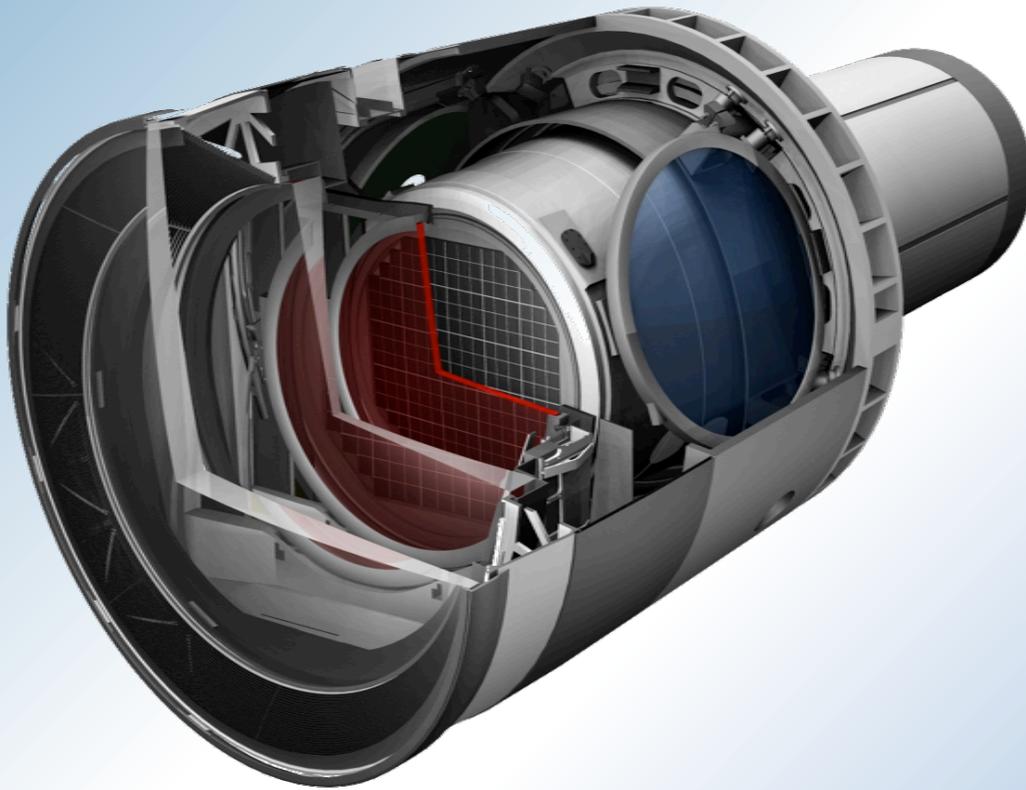
- **Amélioration de la qualité du code :**
  - **Réécriture pour correction de bugs, plus de lisibilité, plus de fiabilité**
  - **Ajout de Findbugs dans notre chaîne d'intégration continue**
- **Conception du dictionnaire de commandes**
  - **Pour éviter à l'utilisateur d'apprendre toutes les commandes par coeur !**
- **Intégration de nos outils de test dans la console JAS**
- **Nouveau système de distribution du logiciel pour faciliter les livraisons de nouvelles versions aux bancs de test**

- **Support aux bancs de test**
  - **Beaucoup de drivers ont été écrits pour les bancs de test des CCD :**
    - on sait piloter le reflex controller
    - on sait piloter les REB
  - **On reste dans une optique évolutive :**
    - Ccs test bench → raft testing -> raftcam -> commissioning caméra
- **De la documentation : il n'y en a jamais assez !!!**

- Réaliser un prototype du dictionnaire de commandes
- Réaliser un prototype du MCM, Master Control Module
- Conclure sur la réflexion sur les changements d'état du CCS et réaliser un prototype
  - Normal
  - Engineering
- Mettre en oeuvre les scripts
- Encore et toujours de la documentation
- Support aux développeurs de sous-systèmes

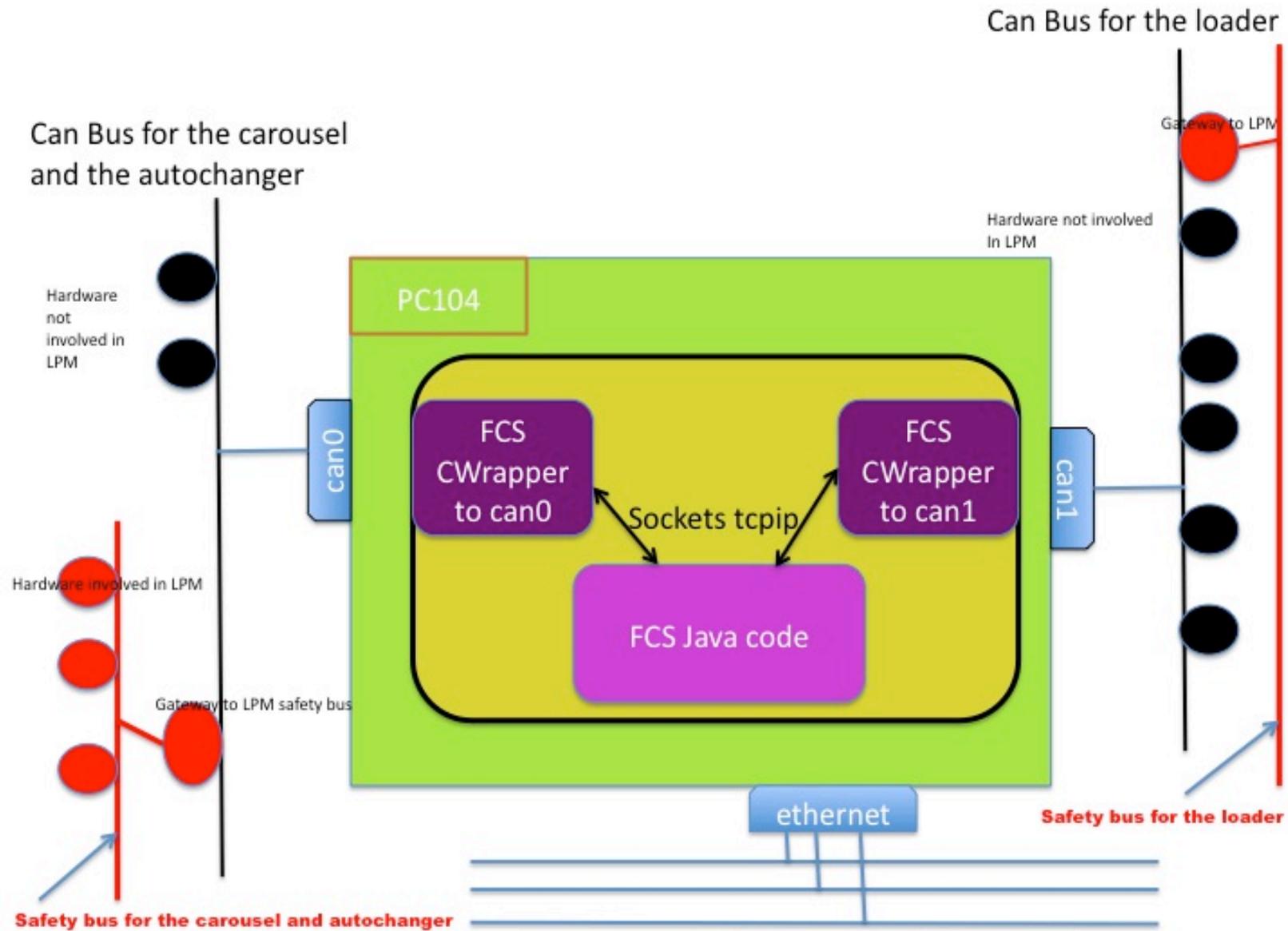


**FCS**



- The hardware is connected on a CANbus and communicate via a CANOpen protocol
- The list of hardware to control and monitor is on Confluence :
  - <https://confluence.slac.stanford.edu/display/LSSTCAM/List+of+Hardware+to+Control+and+Monitor>
  - We control motors via industry standard motor controller (EPOS controller)
  - The sensors involved in the Protection System will be read through the safety PLC gateway

# FCS Architecture : communication with the hardware



- **Single-filter-test :**
  - plusieurs versions du logiciel de contrôle du single filter test ont été livrées, installées et testées
  - Une démonstration a été faite à Nadine et Martin lors de la PDR Changeur de filtres en novembre
  - Documentation sous Confluence, exemple : (mais c'est jamais assez !)

## Available commands in engineering mode

### Available commands in normal mode for the Main module

This commands can be sent to the single-filter-test subsystem, from the console :

```
invoke single-filter-test COMMAND NAME
```

example :

```
invoke single-filter-test updateStateWithSensors
```

example to execute many times the Sequence1 of action (12 in the example below)

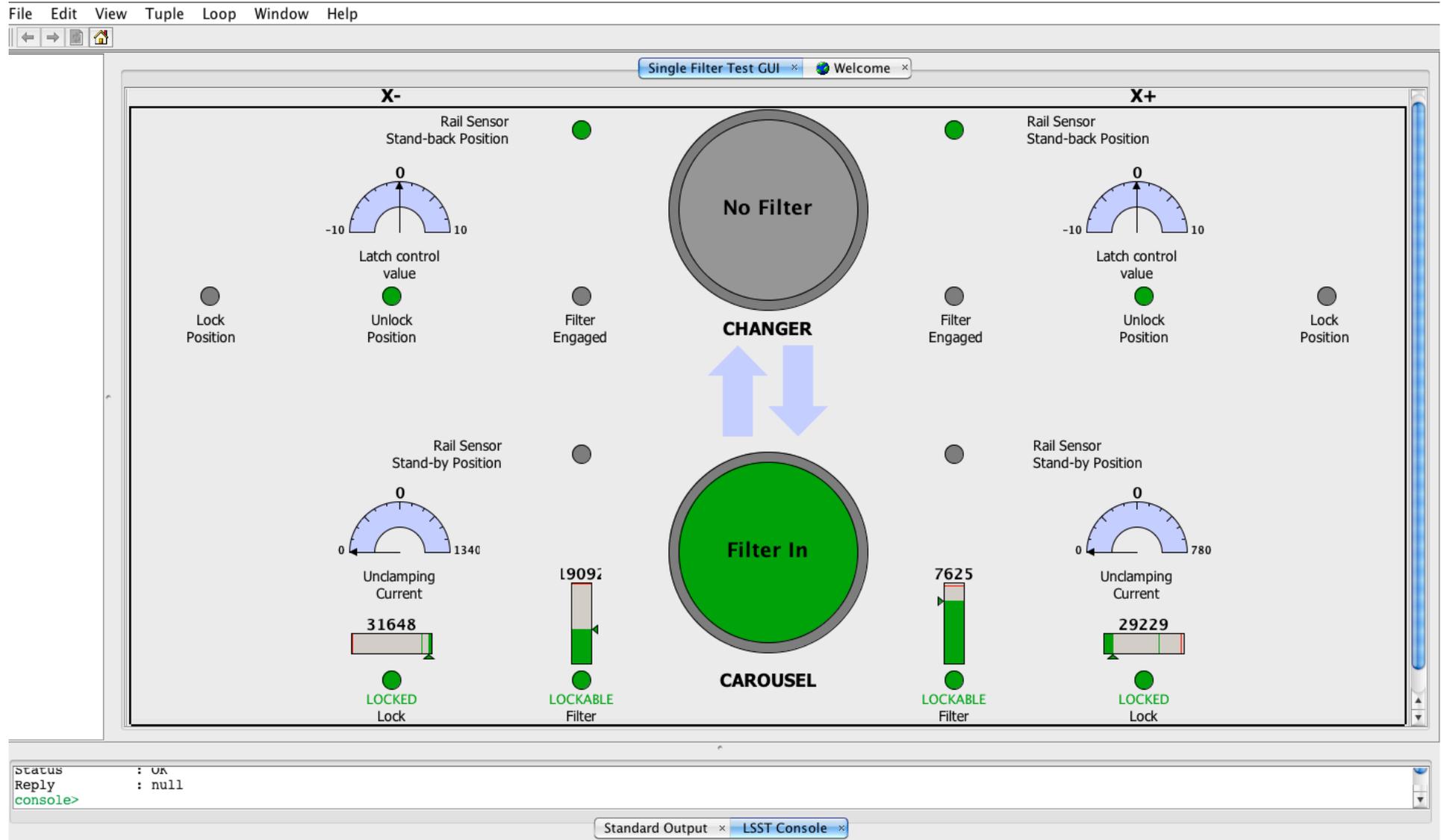
```
invoke single-filter-test executeSequence1 12
```

NB. If the hardware is not in the initial position for the Sequence1, the command above won't work, you'll get an ERROR message.

The commands available in normal mode are some methods of the SFTMainModule, mainly this methods are set of pre-defined sequences of operations :

- **goToInitialState**
  - this command moves the hardware to the initial state (in the initial state the autochanger trucks are empty at stand back position)
- **updateStateWithSensors**
  - this updates the carousel and the autochanger in reading the sensors and publishes on the STATUS bus.
- **locateFilter :**
  - this command reads all the sensors and updates the subsystem data which the new value return by the sensors (clamps state, latches state, autochanger trucks location), it initializes the software and makes it ready to execute the sequence1 of actions.
  - it has to be executed when we start the subsystem, prior the command executeSequence1
- **executeSequence1 :**
  1. this sequence starts when the autochanger trucks are empty in the higher position
  2. then the autochanger grabbes the filter at standby position
  3. the carousel ungrabbes the filter at standby position,

# FCS: Jas console et GUI pour single-filter-test

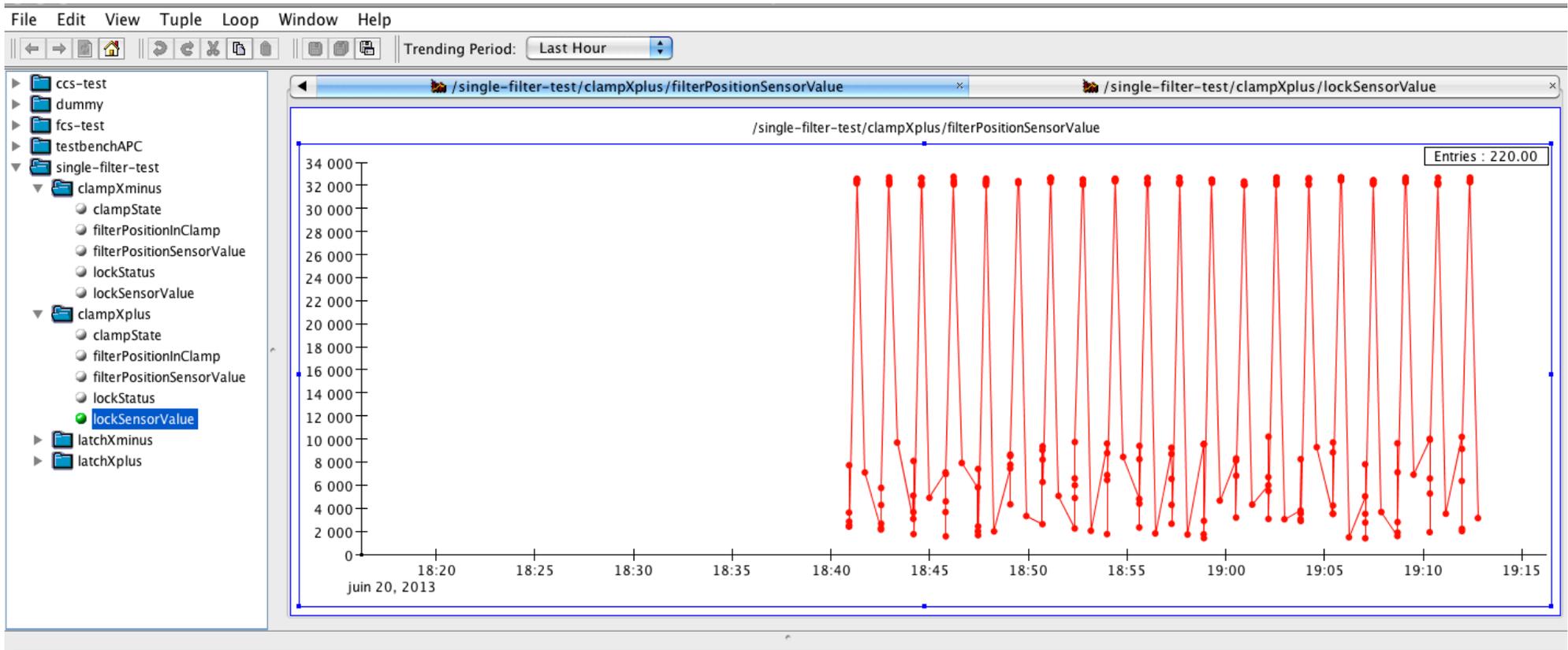


The screenshot displays the 'Single Filter Test GUI' interface. At the top, a menu bar includes 'File', 'Edit', 'View', 'Tuple', 'Loop', 'Window', and 'Help'. Below the menu is a toolbar with navigation icons. The main window contains two side-by-side control panels for 'X-' and 'X+' rail sensors. Each panel features a 'Rail Sensor Stand-back Position' gauge (range -10 to 10), a 'Latch control value' gauge (range 0 to 10), and a 'Lock Position' indicator. The 'X-' panel also shows an 'Unlock Position' indicator and a 'Filter Engaged' indicator. The 'X+' panel shows a 'Filter Engaged' indicator and a 'Lock Position' indicator. In the center, a 'CHANGER' indicator (grey circle) is positioned above a 'CAROUSEL' indicator (green circle), with two blue arrows pointing between them. Below the carousel, there are two vertical bar graphs: 'Unclamping Current' (left, value 31648, indicator 'LOCKED Lock') and 'Filter' (right, value 7625, indicator 'LOCKABLE Filter'). The 'X+' panel also has an 'Unclamping Current' gauge (range 0 to 780) and a 'Lock Position' indicator. At the bottom, a console window shows the following text: 'status : OK', 'Reply : null', and 'console>'. The bottom of the window has a 'Standard Output' and 'LSST Console' tab.

# FCS: Jas console et Trending data pour single-filter-test



- Courbe des valeurs mesurées par le capteur de position d'un filtre dans le verrou qui fixe le filtre sur le carrousel (Filter Clamp Position Sensor Values)



- **Prototype du changeur de filtres à l'échelle 1**
  - **Les séquences des opérations ont été décrites dans Confluence:**
    - <https://confluence.slac.stanford.edu/display/LSSTCAM/Filter+Exchanger+Sequences>
    - Exemple ci-dessous
  - **Le FCS a été distribué pour tous les bancs de test et des premiers tests ont été réalisés**

## To load a filter in the camera from the loader

"Position naming" is given in the doc LCA-280 (§ 8.2)

Conditions in which this operation is authorized :

- the camera is in the position "horizontal"
- the auto changer trucks are empty at "Hand-Off" position
- the carousel socket at "Stand-by" position is empty
- The Loader carrier is retracted at "Storage" position and the filter is clamped

Sequence : "Loader stores Filter in Camera"

1. The Loader is clamped on the Autochanger (manual operation)
2. Move the filter on the loader carrier to "Hand-Off" position : (Loader Carrier goes to Hand-off position)
3. Check the AutoChanger latches sensors in state : **FILTER ENGAGED**
4. Close the Autochanger latches on the filter at "Hand-Off" position: (Close AutoChanger Latches)
5. Check the AutoChanger latches sensors in state : **AUTOCHANGER LATCHES LOCKED**
6. Unclamp the Loader filter clamp : (Unlock Loader Clamp) (on ouvre les pinces)
7. Check the Loader Clamp sensors in state : **UNCLAMPED**
8. Move back the filter carrier to "storage" position : (Loader Carrier go to Storage position) (on retire le chariot)
9. Check the sensors of the carrier in state : **STORAGE POSITION**
10. The filter can be stored in the carousel : Sequence "AutoChanger stores filter in Carousel"
11. The AutoChanger empty trucks are at the "Hand-off" position
12. Manual action if needed : Unclamp the Loader from the AutoChanger and move it back

- **Développer, installer, tester le logiciel pour tous les bancs de test :**
  - **Grenoble,**
  - **Marseille,**
  - **Clermont**
- **Livrer le FCS complet pour le prototype à l'échelle 1**
- **Revoir le simulateur pour le système complet**
- **Développer les interfaces graphiques**
- **Formaliser et écrire le plan de test et de validation du logiciel**

- **LSST DOE Status Review 18-19 juin 2013 à SLAC pour le CCS et les sous-systèmes**
- **Filter Exchange System Preliminary Design Review 4-5 novembre au CPPM**
  - **Principales recommandations pour le FCS :**
    - Mieux décrire les conditions d'exécution des différentes commandes
    - Mieux définir les niveaux de sécurité des différentes commandes (utilisateurs avec plusieurs niveaux)
    - Fournir un plan de validation du logiciel pour chaque nouvelle version
- **Revue interne du CCS prévue en mars 2014**



**End of presentation**

