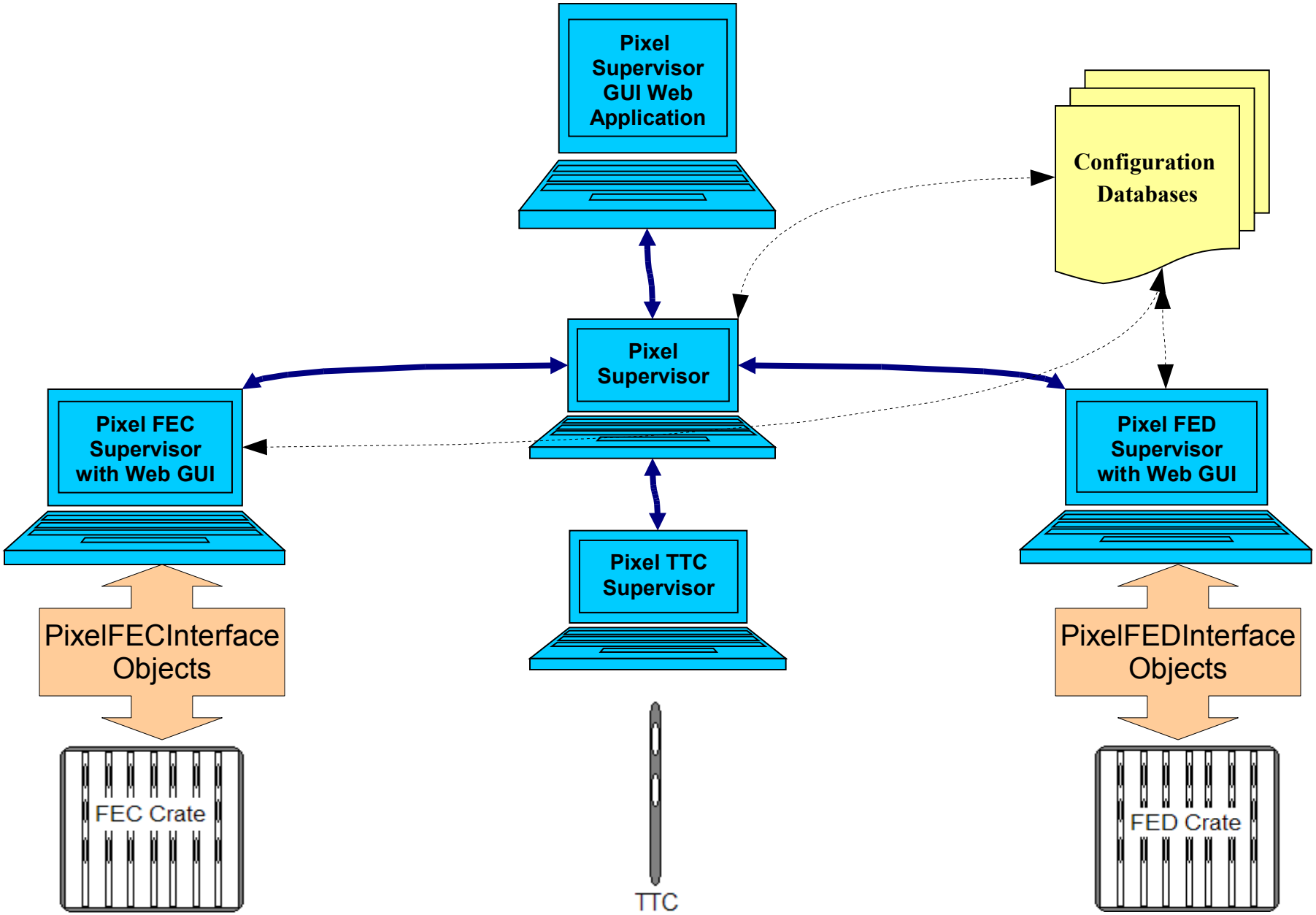
Pixel Online Software

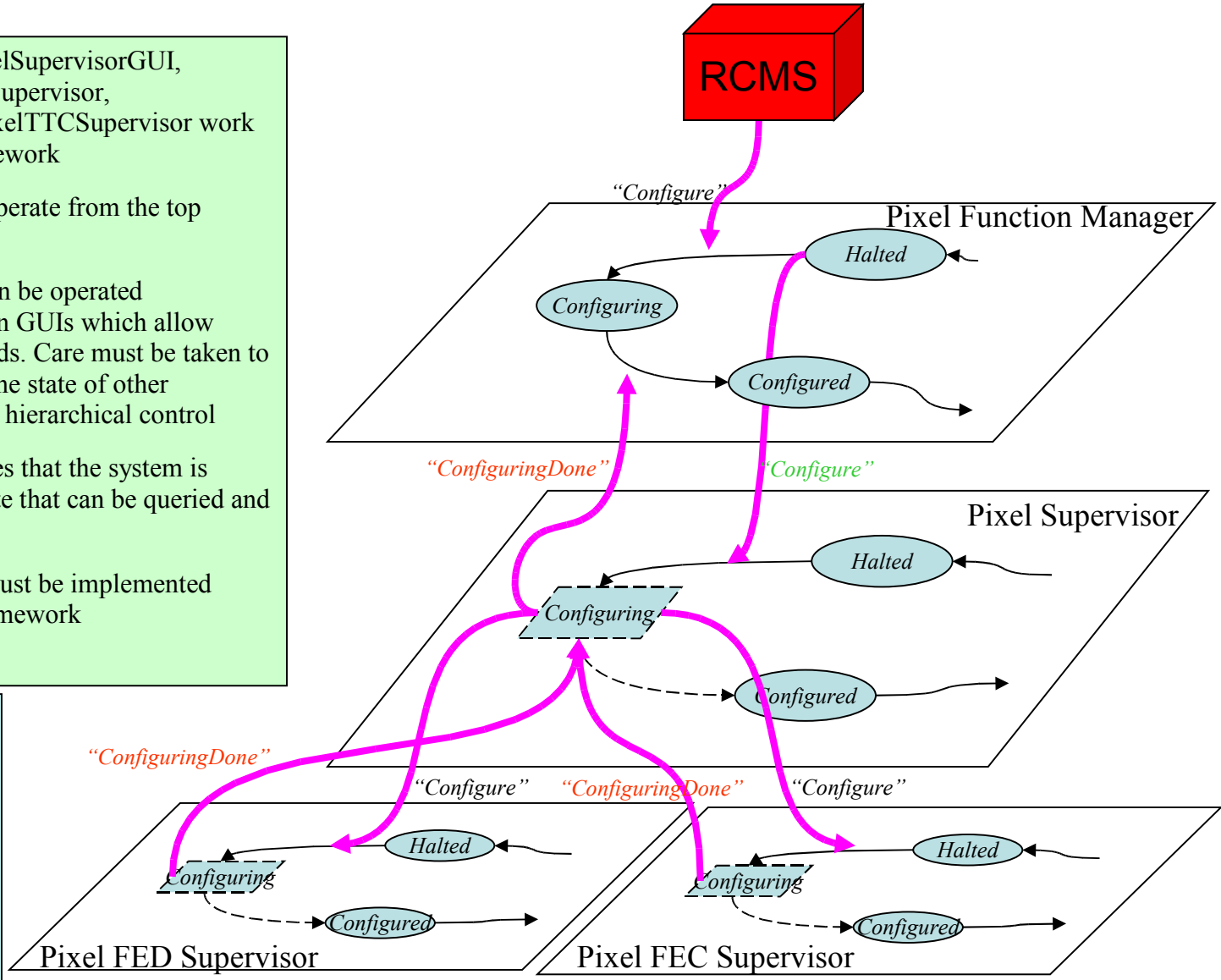
Souvik Das
(Cornell University)

Overview of Pixel Online Software Version 1.1.0



Layers of State Machines

- PixelFunctionManager, PixelSupervisorGUI, PixelSupervisor, PixelFEDSupervisor, PixelFECSupervisor and PixelTTCSupervisor work within a state machine framework
- Hierarchically arranged to operate from the top (PixelFunctionManaer)
- However each supervisor can be operated independently with their own GUIs which allow access to low level commands. Care must be taken to restore its state machine to the state of other supervisors before resuming hierarchical control
- Why state machines? Ensures that the system is always in a well-defined state that can be queried and recovered
- All calibration algorithms must be implemented within the state machine framework



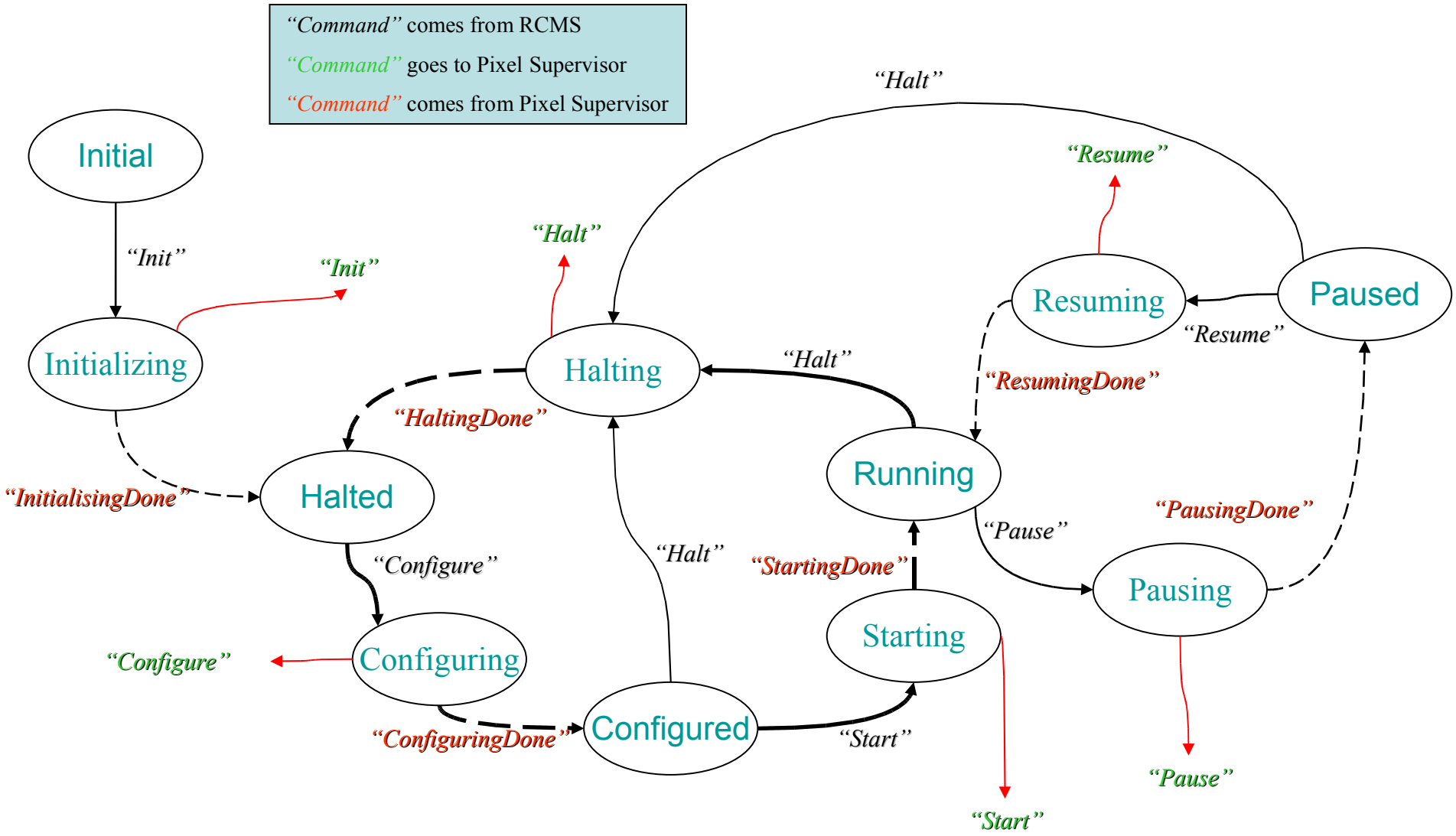
Legend

SOAP message

Configuring — State

Configuring — Transition (Entry) Function

State Machine of PixelSupervisorGUI Version 1.1.0



HyperDAQ

http://mp72.lns.cornell.edu:1973
urn:xdaq-application:service=hyperdaq


Control Panel

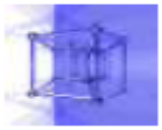

Cluster Explore


Executive
urn:xdaq-application:ser


XRelay


PeerTransportHT[...]
pplication:lid=1


PeerTransportFi[...]
urn:xdaq-application:lid=


HyperDAQ
urn:xdaq-application:ser


PixelSupervisor
ication:lid=


PixelSupervisor[...]
urn:xdaq-application:lid=


PixelTKFECSuper[...]
urn:xdaq-application:lid=55


PixelFECSupervi[...]
urn:xdaq-application:lid=


PixelFEDSupervi[...]
urn:xdaq-application:lid=


PixelLTCSupervi[...]
urn:xdaq-application:lid=

PixelTTCSupervi[...]
urn:xdaq-application:lid=


PixelDCSSupervi[...]
urn:xdaq-application:lid=100

Contact Us | Documentation Site | Web Site

Road to Version 1.1.0

- Implemented the layered state machine structure of Supervisors
- Communicated with the FEC and FED Boards from XDAQ using PixelFECInterface and PixelFEDInterface classes
- Each Supervisor given its independent GUI for low level control
- Established a working file based system for storing configuration and calibration information (will be replaced with the DataBase for later versions)
  - •Every configuration is associated with a Global Key that is passed from PixelSupervisor or above
  - •Each Global Key points to a

    Detector Configuration Key,

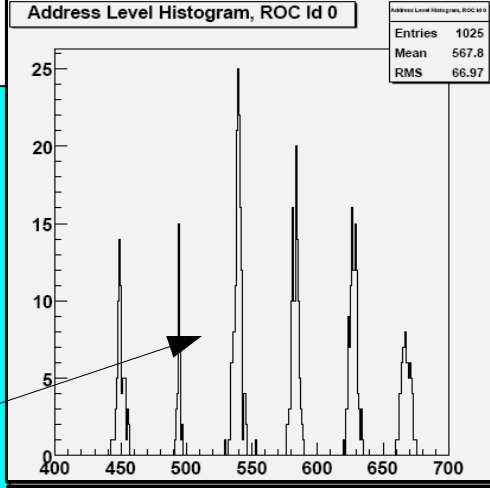
    Name Translation Table Key,

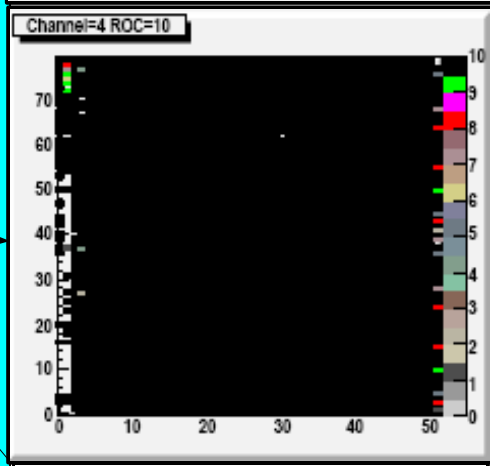
    FED Configuration Key, etc.
  - •Can be accessed from any Supervisor

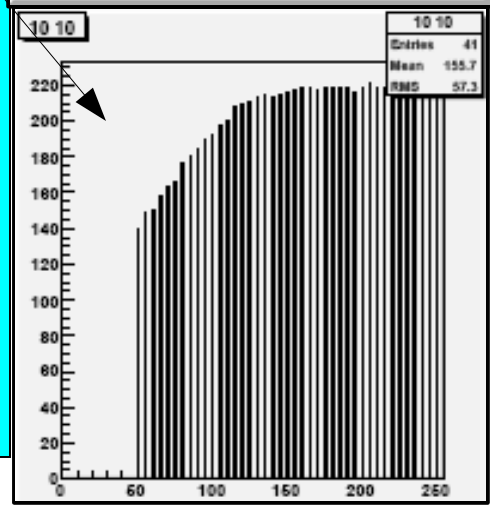
```
key 4                    # Gain Calibration
detconfig 2
nametranslation 0
fedconfig 0
fecconfig 0
fedcard 0
dac 2
mask 2
trim 2
calib 3
tbm 0
```

Road to Version 1.1.0

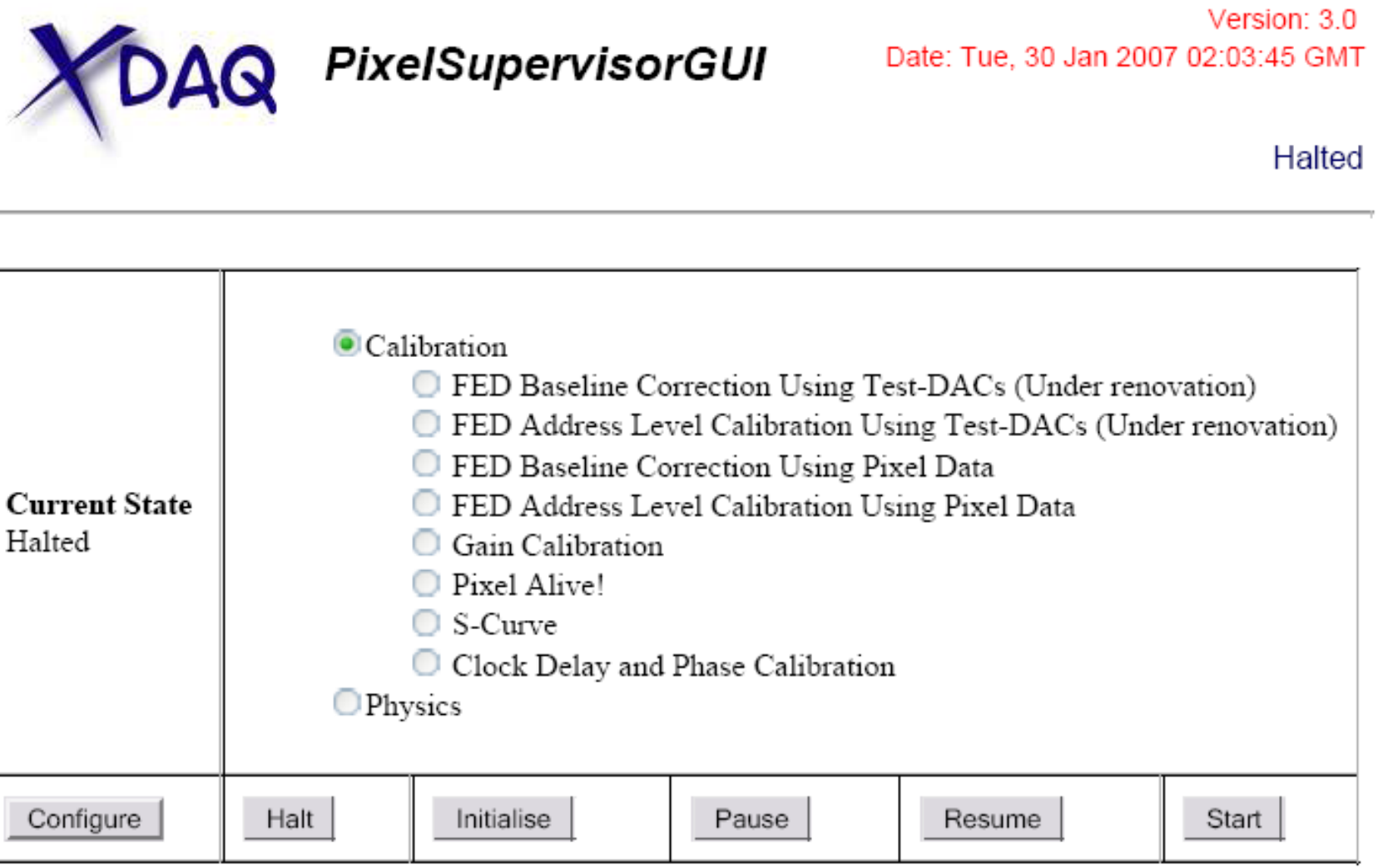
- Implemented some basic calibration routines within the framework:
  - **FED Baseline Calibration**: Stabilizes the Black level of TBM output using FED's channel offset parameters
  - **FED Address Level Calibration**: Collects statistics of address levels from each FED channel and sets threshold levels for ROCs and TBMs
  - **Pixel Alive Test**: Tells us which pixels in our detector are responding
  - **Gain Calibration** (Data taking): ADC vs VCal curve
  - **S-Curve Calibration** (Data taking): Fraction of responding pixels vs VCal
- Tested on the '07 detector at Fermilab
- Tested on the Cornell Test Stand
- Drafted a **manual**. Available at **DOC DB, Document #1560**
  or,
  http://pages.physics.cornell.edu/~souvik/CMS/PixelOnlineSoftwareManual.pdf
- **Version 1.1.0** Release







Screenshot of Pixel Supervisor GUI Version 1.1.0



Pixel FEC/FED Supervisor GUIs Version 1.1.0

FEC with Base Address 0x30000000

TBM Command

mFEC: 1 mFEC Channel: A TBM Channel: A
Hub Address: 0 Port Address: 0 Offset: 0
Data Byte: Direction: 0 (Write)
TBMCommand

Program DAC

mFEC: 1 mFEC Channel: A
Hub Address: 0 Port Address: 0
ROC Id: 0
DAC Address: Vdd DAC
Value:
Prog_DAC

Program Pixel

mFEC: 1 mFEC Channel: A
Hub Address: 0 Port Address: 0
ROC Id: 0
Pixel Column: 0 Pixel Row: 0
Pixel: Enable Trim (0-15): 0
Prog_Pix

Calibrate Pixel

mFEC: 1 mFEC Channel: A
Hub Address: 0 Port Address: 0
ROC Id: 0
Pixel Column: 0 Pixel Row: 0
Calibrate with: Sensor Bumps
Cal_Pix

Clear Calibration

mFEC: 1 mFEC Channel: A
Hub Address: 0 Port Address: 0
ROC Id: 0
ClrCal

FED with Base Address 0x1c000000

Reload Firmware

ReloadFirmware

Reset FEDs

ResetFEDs

Channel Offset

Channel	Capacitor Adjust	Input Offset	Output Offset	Offset DAC
1	0	0	0	

ChannelOffsets

Control and Mode Registers

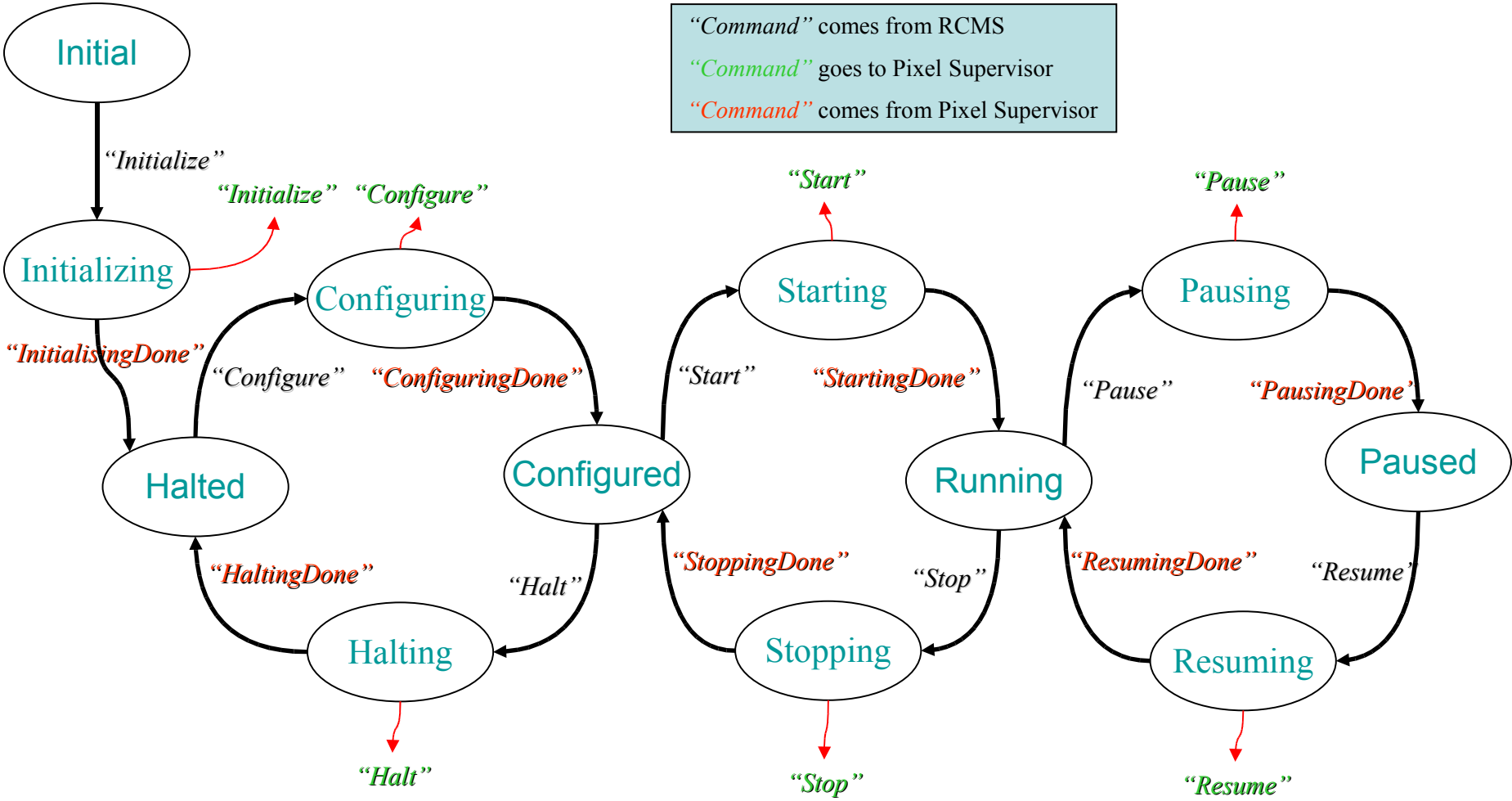
Control Registers
Transparent Mode (•) Disable ( ) Enable
Transparent Gate Start by ( ) L1A (•) VME or EFT (OPTO Module)
Use simulated test-DAC (•) Disable ( ) Enable
Event number generated by ( ) TTC (•) VME
L1A triggers from TTCrx (•) Disable ( ) Enable
EFT Signals from the OPTO Module (•) Disable ( ) Enable
TTSReady (•) Disable ( ) Enable
TTSError (•) Disable ( ) Enable
OUTofSYN (•) Disable ( ) Enable

Mode Registers
S-Link ( ) Disable (•) Enable
Write Spy Memory ( ) Disable (•) Enable
S-Link (•) Let it be, or ( ) Reset
SetControlRegister

Beyond Version 1.1.0

- ✔ Extended code to use multiple FEC/FED crates and multiple FEC/FED boards in each crate
- Testing and improving low level GUI-s.
  - •Health light for each FEC and FED board visible from Pixel FEC/FED Supervisor supervising a crate
  - •GUI for a FED board can update Spy Data FIFO, Last DAC FIFO, Error FIFO and TTS FIFO continuously or on demand in a text box
  - •FED board GUI can display TBM signals on demand
  - •Register settings on the FED updated live on the GUI
  - •Ideas for organisation of FEC board low level GUI?
- Cleaning up messy code, hard-coding and simplifying user's life
- Efficient configuration of pixels during “Configuring” state using new PixelFECInterface developed by Rutgers group. Tested once on '07 Detector.
- Efficient firing of pixels during calibration runs. On-going work at Rutgers.
- Writing of Tracker FEC Supervisor and integration with rest of Pixel Online Software. Working with Heng Li.
- New State Machine proposed by Alex Oh in Function Manager Document 1.5

New Function Manager State Machine



Beyond Version 1.1.0

- Integration with Error Handling. Working with Stephan Spanier.
- Integrating with Detector Control System. Working with Christian Veelken, Andres Carlos Florez and Angel Lopez. Several design issues:
  - •DCS has different state machine (close enough to be mapped to DAQ's)
  - •Not obvious which Supervisor data returning from DAQ to DCS must go to
  - •Not clear how it fits with the Trigger System
- Integration with the Database. Will work with Umesh Joshi, Anders, Zongru and Lorenzo once interaction with database is finalised.
- Integration with Run Control and Monitoring System. Will work with Alex Oh at CERN
- Integration with the Trigger Throttling System of the DAQ. Will work with Will Johns.
- Analysis of calibration data. Integration with CMSSW
- More calibrations types:
  - •UltraBlack levels for all ROCs must be close and close to TBM's UltraBlack. Gain of each ROC and TBM must be adjusted for maximum dynamical range.
  - •Trims calibration