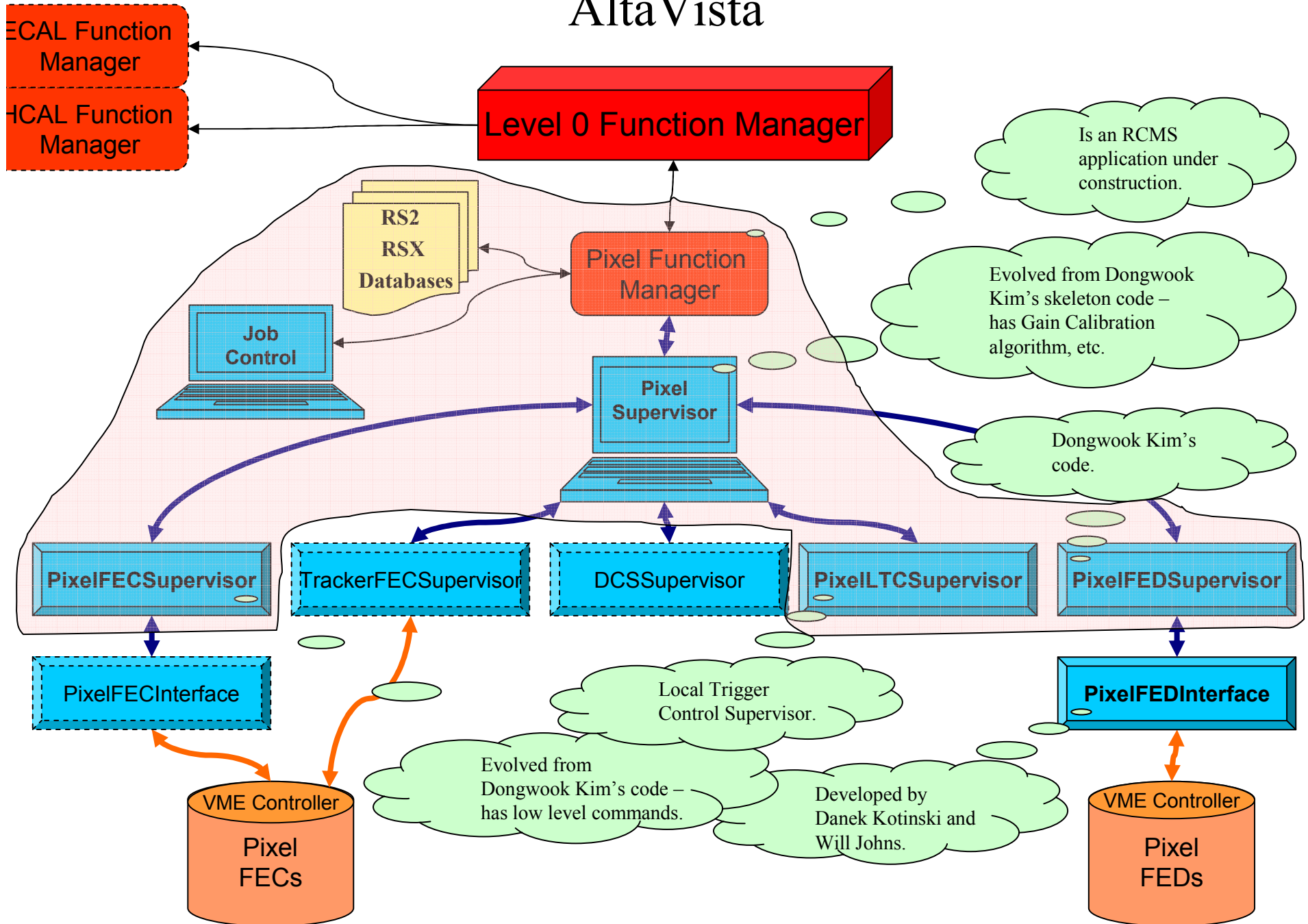


Pixel Run Control and Calibration

-Souvik Das

under Anders Ryd and Karl Ecklund

AltaVista



The Twin State Machine Layers Philosophy

Why two layers?

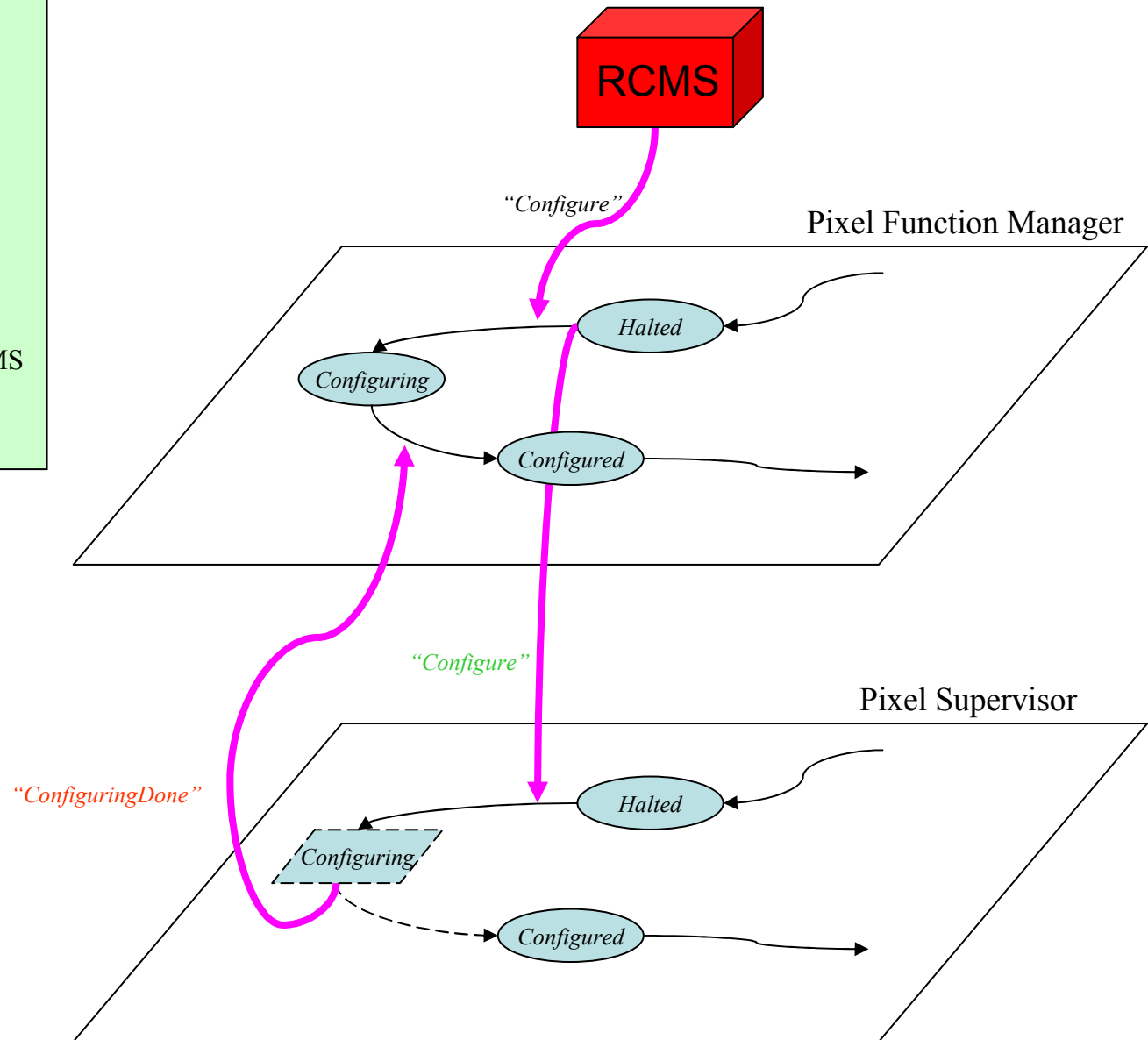
- Level 1 Function Manager (Version 1.0) proposed by Alexander Oh is a Moore State Machine.
- Pixel Supervisor has no reason to be Moore. Mealy is economic on states and easily implemented in XDAQ. (Alex Oh mentions this in email.)
- Definite states of Function Manager ensures that status queries from RCMS will be always answered.

Legend

→ SOAP message

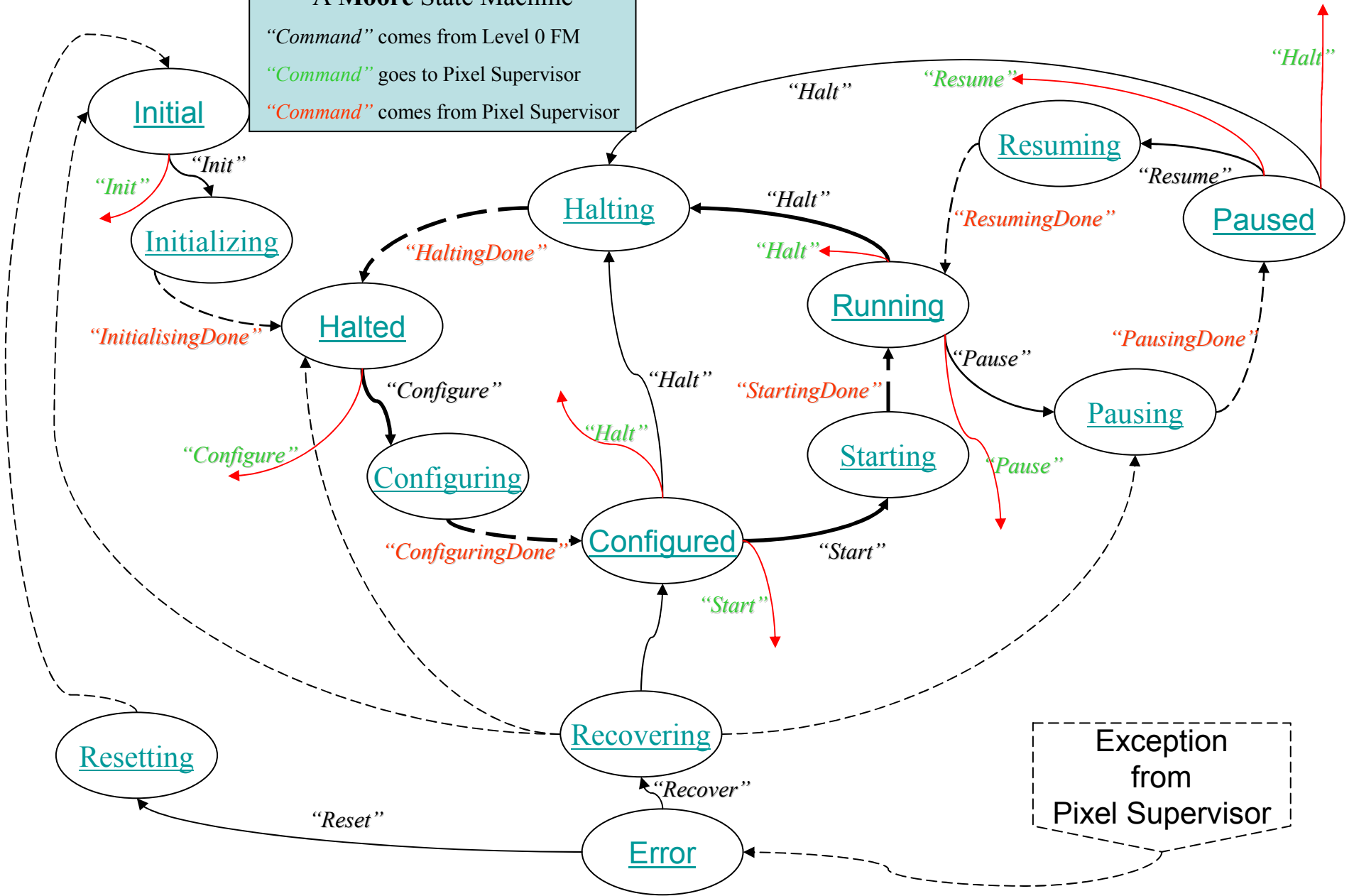
○ *Configuring* State

▭ *Configuring* Transition (Entry) Function



Pixel Function Manager State Diagram

A Moore State Machine
 "Command" comes from Level 0 FM
 "Command" goes to Pixel Supervisor
 "Command" comes from Pixel Supervisor



Pixel Function Manager

Done

- Installed DAQKit and replaced its packaged XDAQ with the latest XDAQ.
- Installed MySQL for the RS2 and RSX databases. These are filled by the Data Utility for Configuration Knowledge (DUCK or DUCK-CAD) tool. RS2 is filled with “control structure configurations” while RSX is stored with “XDAQ configurations”.
- Installed Apache Tomcat Server on which RCMS Java applets run.
- Implemented a simple Hello World example on RCMS with a simple state machine.
- Wrote a skeleton version of the Pixel Function Manager with a state machine in accordance to Alex Oh’s document.
- Sent a SOAP message from the Pixel Function Manager (RCMS application) to the Pixel Supervisor (XDAQ application). So we’re on our way of replacing WebStateMachine!

To Do

- Figure out how to receive SOAP from XDAQ applications.
- Get comfortable with designing the GUI because we will need radio buttons and all the capabilities of WebStateMachine before we phase it out.
- Participate in the Magnet Test Cosmic Challenge with our Pixel Function Manager (and PixelSupervisor maybe) to begin integrating into the larger RCMS framework.

Pixel Supervisor

Recent Changes

- A working prototype exists as a XDAQ application and may be checked out from the /TriDAS/pixel/PixelSupervisor area of our CVS repository. A Makefile has been included.
- Implemented a dummy Gain Calibration Algorithm within the “Running” state.

To Do

- Get back to development of the S-Curve algorithm. It stalled for a while because of my interest in the RCMS Function Manager.
- Need to interface with the Database Supervisor for configuration requirements.
- The “Recover” commands and state have not been implemented yet.

Pixel FEC Supervisor

Recent Changes

- The name has been changed from FastFECSupervisor to PixelFECSupervisor.
- A prototype exists as a XDAQ application and may be checked out from the /TriDAS/pixel/PixelFECSupervisor area of our CVS repository. A Makefile has been included.

To Do

- We are waiting for development of the PixelFECInterface so that we may implement our low level and high level commands that PixelFECSupervisor may receive.

Pixel FED Supervisor

Recent Changes

- The name has been changed from FEDSupervisor to PixelFEDSupervisor.
- A prototype exists as a XDAQ application and may be checked out from the /TriDAS/pixel/PixelFEDSupervisor area of our CVS repository. A Makefile has been included.

To Do

- I assume a more advanced version exists that Danek has from Dongwook. We need to commit it to the CVS repository.

Pixel LTC Supervisor

Done

- The name has been changed from LTCSupervisor to PixelLTCSupervisor.
- A prototype exists as a XDAQ application and may be checked out from the /TriDAS/pixel/PixelLTCSupervisor area of our CVS repository. A Makefile has been included.

To Do

- Refine and add to interfaces.

Summary and To Do List

- The XDAQ structure is falling slowly into place in the TriDAS/pixel area of our CVS repository. A lot of naming changes have taken place recently.
- We have begun to explore the RCMS territory. Have constructed a skeleton Pixel Function Manager to that end and are hoping to host it on the CVS repository too.
- Connect between PixelFECSupervisor and PixelFECInterface. (There seemed to be some spelling mistakes like “reciever” in PixelFECInterface at a first glance we should take care of because they are output messages.)
- Having read the SpyFIFO line, we need to decide on how we must channel the data for processing and how we must we must respond to the results of the processing. That is, close the calibration loop.
- Prod on with the SOAP timing tests to see how network topologies affect it and if we could run into possible bottlenecks. The wildcard * was introduced with such SOAPy situations in mind.
- Work in closer alliance with hardware especially if we are work on PixelFECSupervisor, PixelFEDSupervisor and PixelLTCSupervisor.
- Updates, notes, questions etc will be posted on the HyperNews forum:
<https://hypernews.cern.ch/HyperNews/CMS/get/pixelOnlineSW.html>