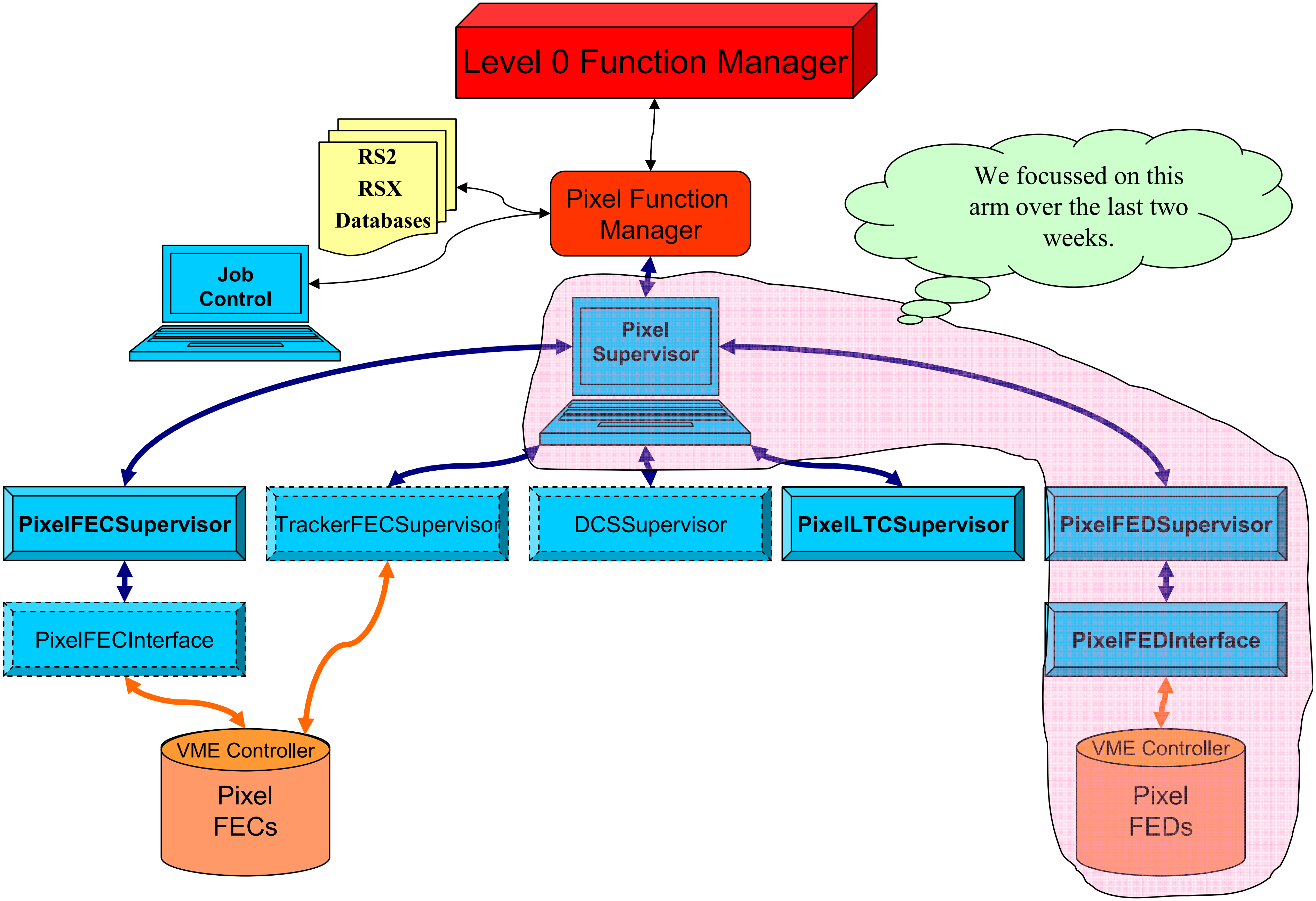
Pixel Run Control and Calibration

-Souvik Das

under Anders Ryd and Karl Ecklund

The View from Above



We focussed on this arm over the last two weeks.

Recent Progress

Pixel FED Interface

- Danek Kotlinski and Will Johns developed and uploaded this component into our CVS repository under TriDAS/pixel/PixelFEDInterface.
- We modified its Makefile so as to build static XDAQ libraries and uploaded the Makefile into CVS.

SOAPCommander

- A new component called SOAPCommander was developed and uploaded into the CVS repository area TriDAS/pixel/PixelUtilities/PixelSOAPUtilities/SOAPCommander.h
- Designed on the lines of appCommandSender, it provides a user-friendly interface for creating, sending, receiving and extracting messages from SOAP messages.
- For a discussion on its abilities, please see the thread on Hypernews.

Pixel FED Supervisor

- Danek Kotlinski emailed me his version of PixelFEDSupervisor which could use the methods defined in PixelFEDInterface.
- The state machine was removed. PixelFEDSupervisor is now a set of methods bound to incoming SOAP messages from the PixelSupervisor like “Configure”, “Start” etc.
- It can now be operated with or without hardware by changing one line in PixelFEDSupervisor.h
- It uses SOAPCommander for its SOAP transactions.
- The Makefile was modified to account for CAEN and HAL libraries, and SOAPCommander.
- Some minor changes to its console output were made for readability.
- It was uploaded to the TriDAS/pixel/PixelFEDSupervisor area of the repository.

Recent Progress

Pixel Supervisor

- Uses SOAPCommander for its SOAP transactions (as does WebStateMachine).
- It now checks for replies from its subordinate components before updating its state. Since error handling protocols haven't been established yet, it just prints out a message in case some components don't work as expected and carries on.
- The Makefile was modified a little to account for the inclusion of SOAPCommander.
- The changes were uploaded to the TriDAS/pixel/PixelSupervisor area of our repository.

- We were at Fermilab on Monday and Tuesday, and successfully tested our software with the test-stand FED. We were able to communicate with the FED right from the GUI of WebStateMachine!
- For testing purposes, we introduced a new function in PixelFEDSupervisor called testingFEDInterface() and copy-pasted into it chunks of code from pixel/PixelFEDInterface/test/testPixelFED.cpp. This allowed us to test the following:
  - ✓ Loading of FED configuration data from the file “params_fed.dat”.
  - ✓ Generation of VME triggers.
  - ✓ Reading out of data via the SpyFIFO.
  - ✓ Correlating the input DAC signals with the output ADC signals. A linear correlation was observed. (Will be quantified later.)
  - ✓ Decoding of the ADC signal pulse train for correct identification of a (simulated) pixel hit. This did not work initially. We fixed it by changing the level “Black” was identified with from 300 to 200.