CMS DAQ Online Frameworks, XDAQ + RCMS

Ichiro Suzuki
PPD/EPP/CMS
Fermi National Accelerator Laboratory

KEK Seminar, 2007/10/15

Contents

- Introduction
- XDAQ: Online application framework
    - User application
    - Messaging
    - Tools
- RCMS: Configuration and control framework
    - Control structure
    - Back-end services
- Status/summary

LHC

- A machine for Higgs and beyond.

Beam Energy: 7TeV

Circumference: 26.7km

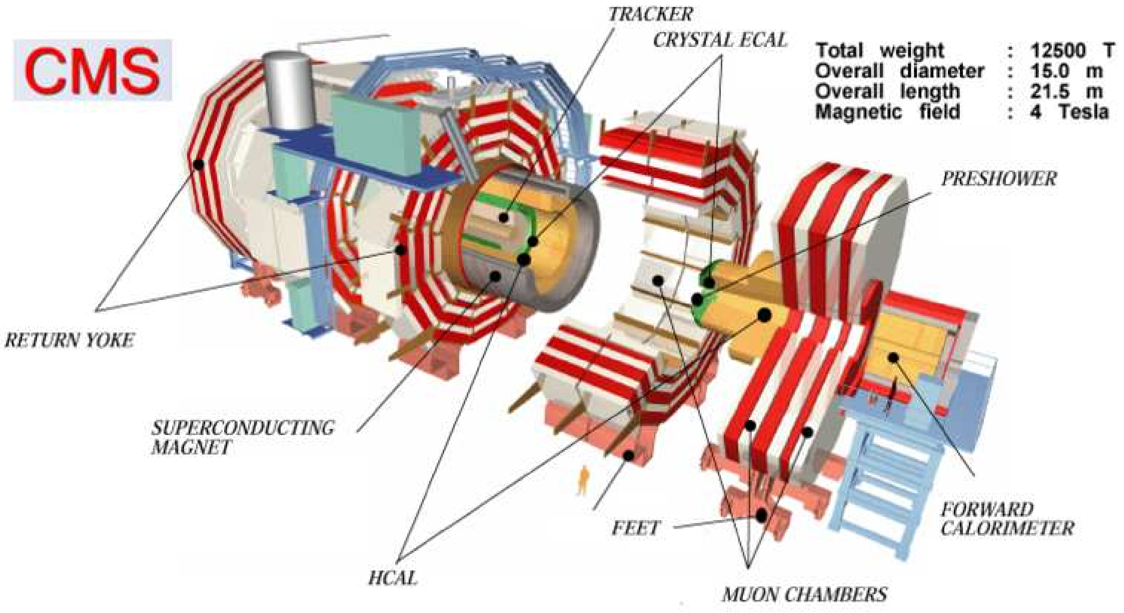
Luminosity: $10^{34}cm^{-2}s^{-1}$

#Bunches: 2835

p/bunch: $1.1x10^{11}$

CMS Experiment

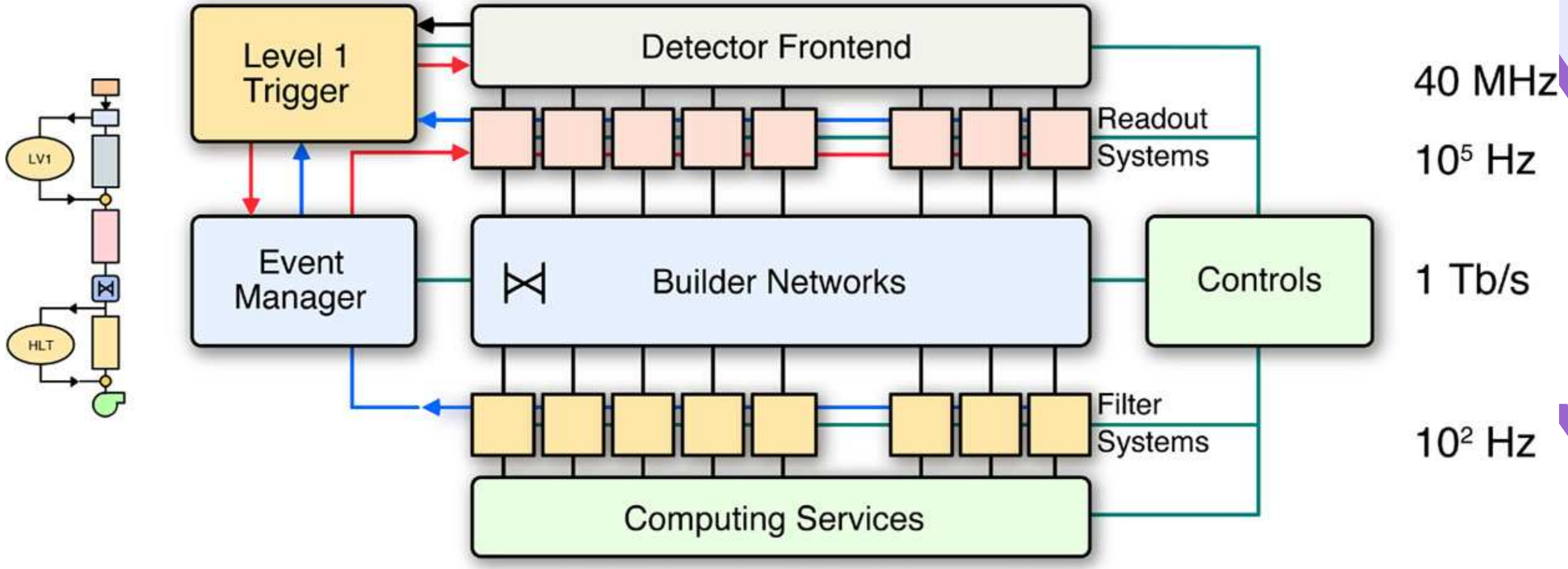
- One of four experiments at LHC
- 36 nations, 169 institutions, 2300 scientists
- 'Compact' Muon Solenoid - ½ the size of Atlas



Total weight	:	12500 T
Overall diameter	:	15.0 m
Overall length	:	21.5 m
Magnetic field	:	4 Tesla

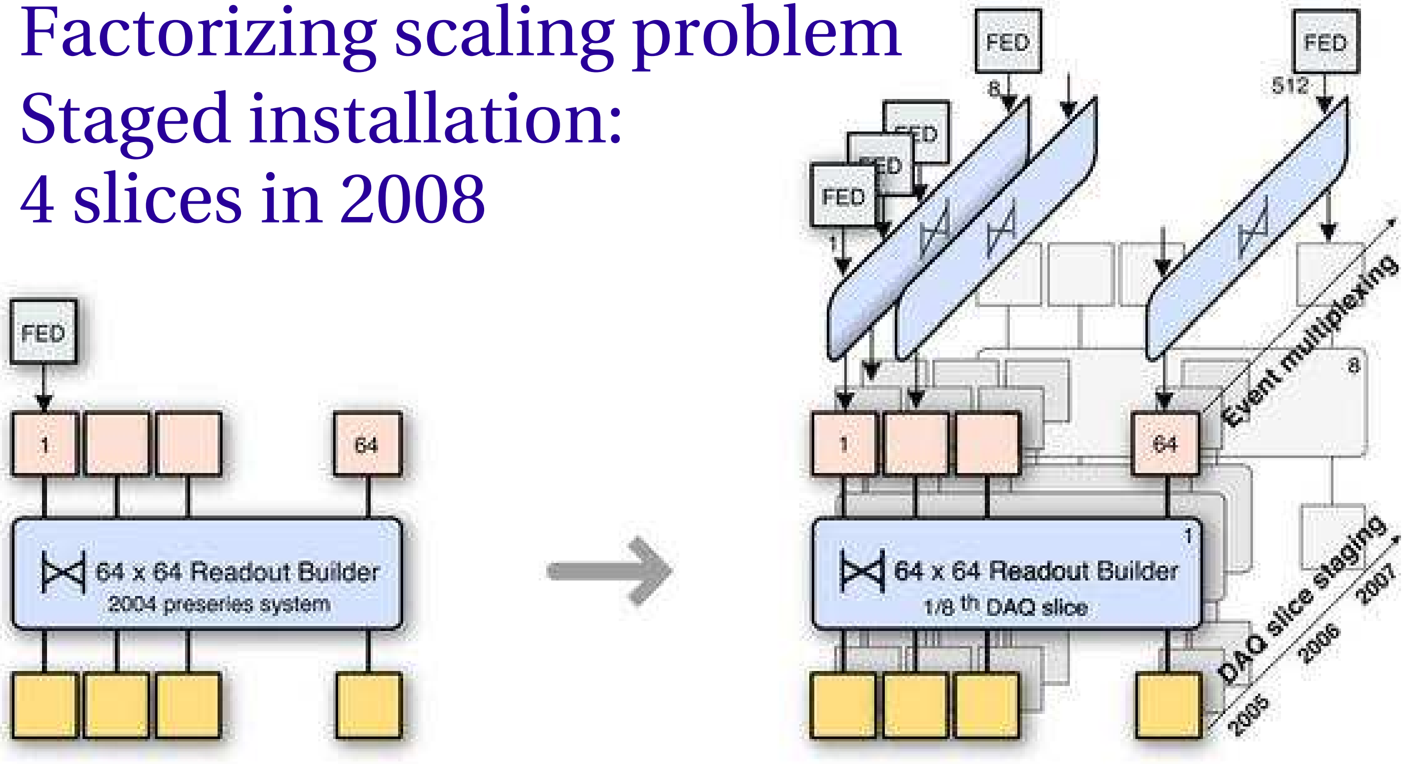
CMS Data Acquisition System

- Two stages design
  - Level-1: synchronous/hardware
  - HLT: asynchronous/PC-farm
- ~700 inputs, 1MB/event

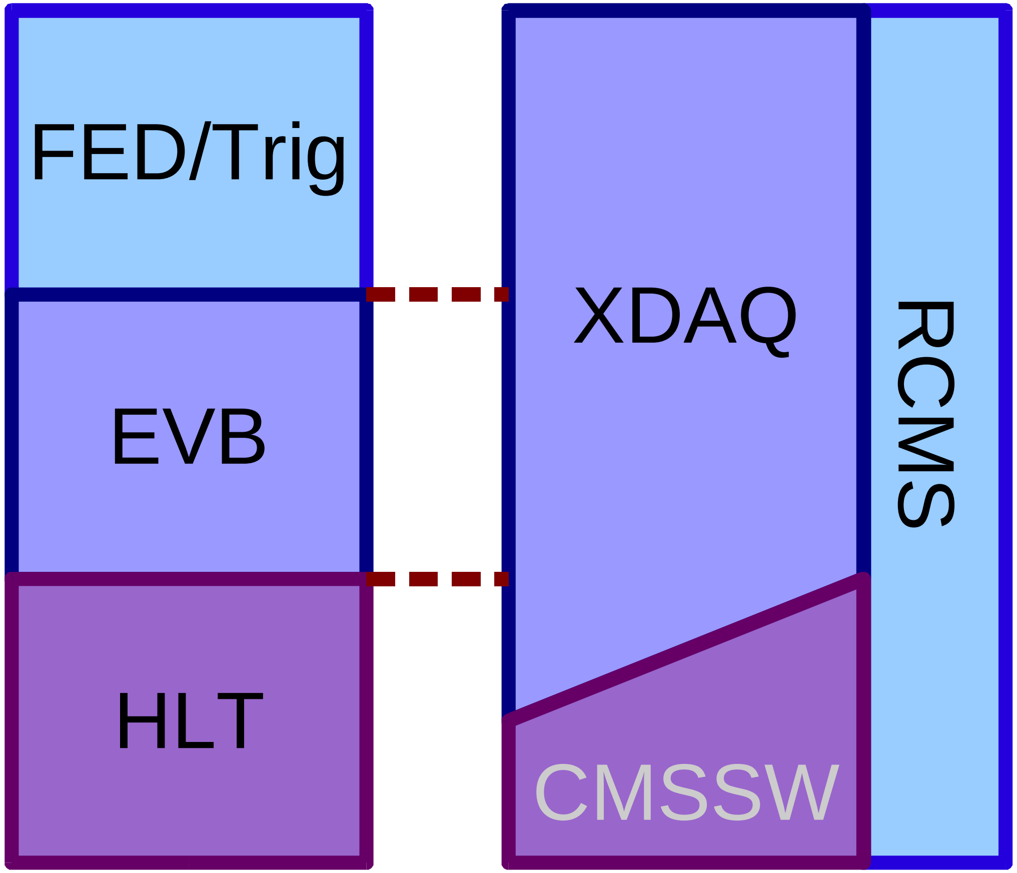


CMS DAQ System (cont'd)

- Sliced architecture
- 'Slice': 1/8 of the DAQ
- Factorizing scaling problem
- Staged installation:
  4 slices in 2008



Online Frameworks



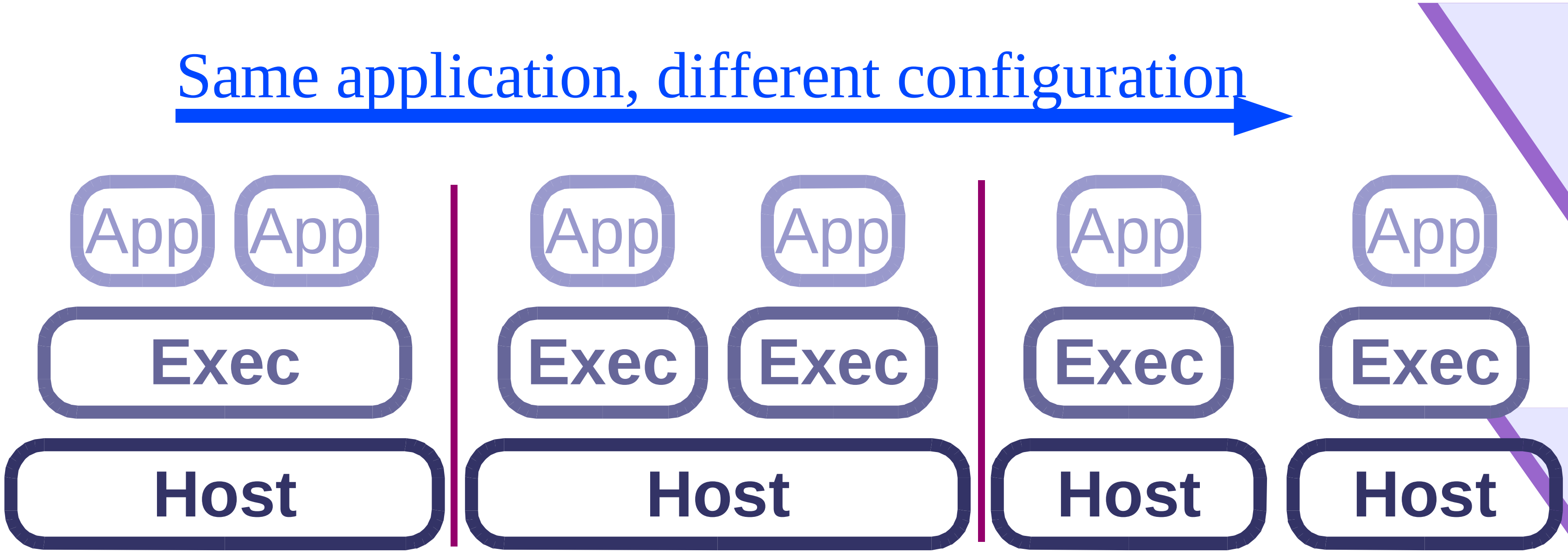
- XDAQ:
  Online FW
- RCMS:
  Control FW
- CMSSW:
  Offline FW

XDAQ

- CMS online software framework in C++.
  (Configuration, messaging, event handling...)
- It's a toolkit, too.
  (FSM, threads, logging, monitoring...)
- Extensive use of XML for configuration/messaging
- Both fast-binary and slow-XML communications
- Scalable: from small test stand to the CMS DAQ.
- First tagged release in 2000.
- J. Gutleber and L. Orsini
  http://xdaqwiki.cern.ch/

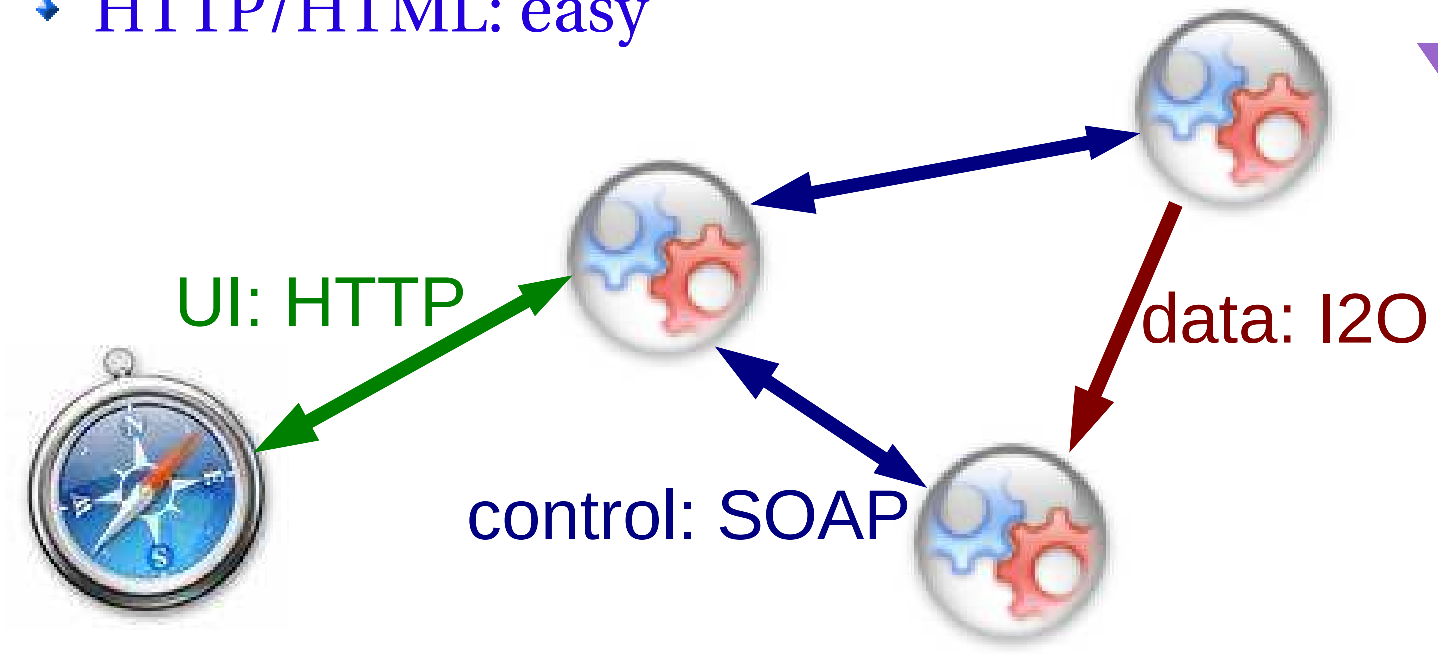
XDAQ: User Applications

- XDAQ user application is a collection of call-back functions, typically attached with FSM transitions.
- Applications run on distributed platforms



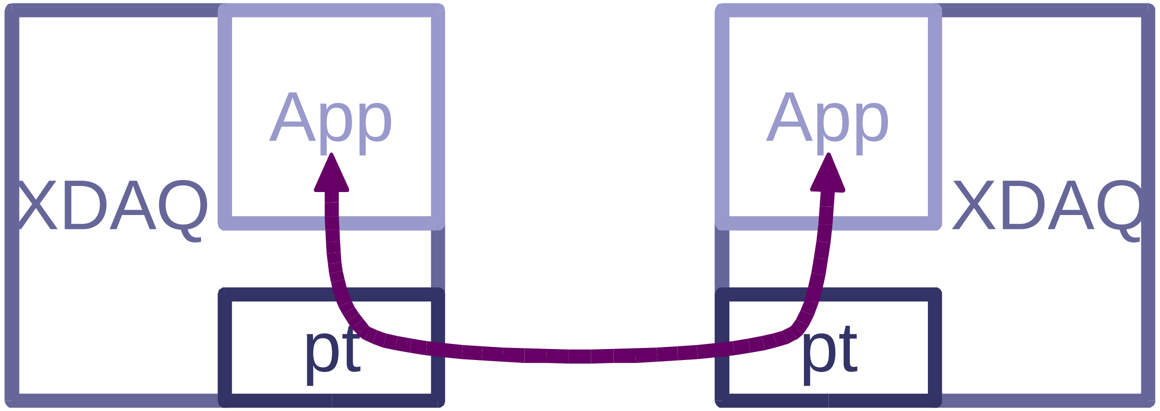
XDAQ: Message Types

- Flexible choices of message type
  - Binary I2O messages: fast, efficient
  - XML(SOAP-on-HTTP): slow, flexible
  - HTTP/HTML: easy



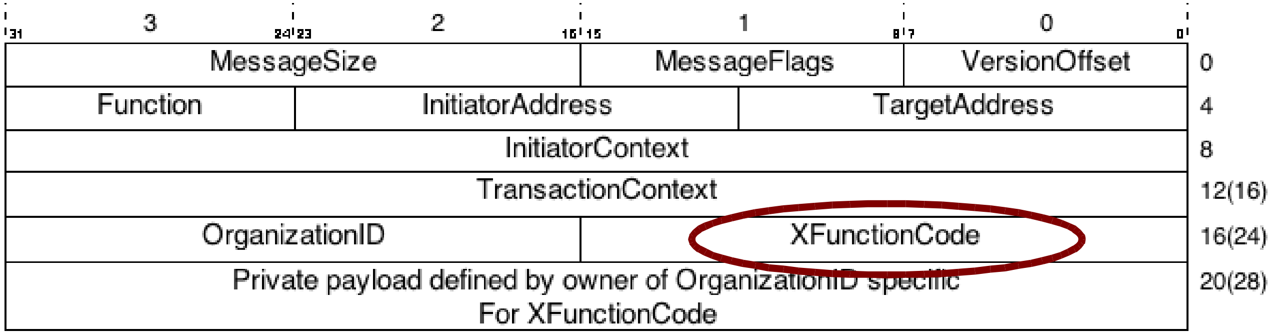
XDAQ: Peer-Transport

- Pluggable abstraction layer of various networking medium
  - I2O: TCP, aTCP, Myrinet, FIFO
  - SOAP/HTML: HTTP
- e.g., Application code doesn't change when you switch from Ethernet to Myrinet.



XDAQ: I2O Messages

- I2O header format was adopted for CMS use
- Binary data format
- In XDAQ,
  - used to pass data fragments
  - applications can register call-backs for XFunctionCode



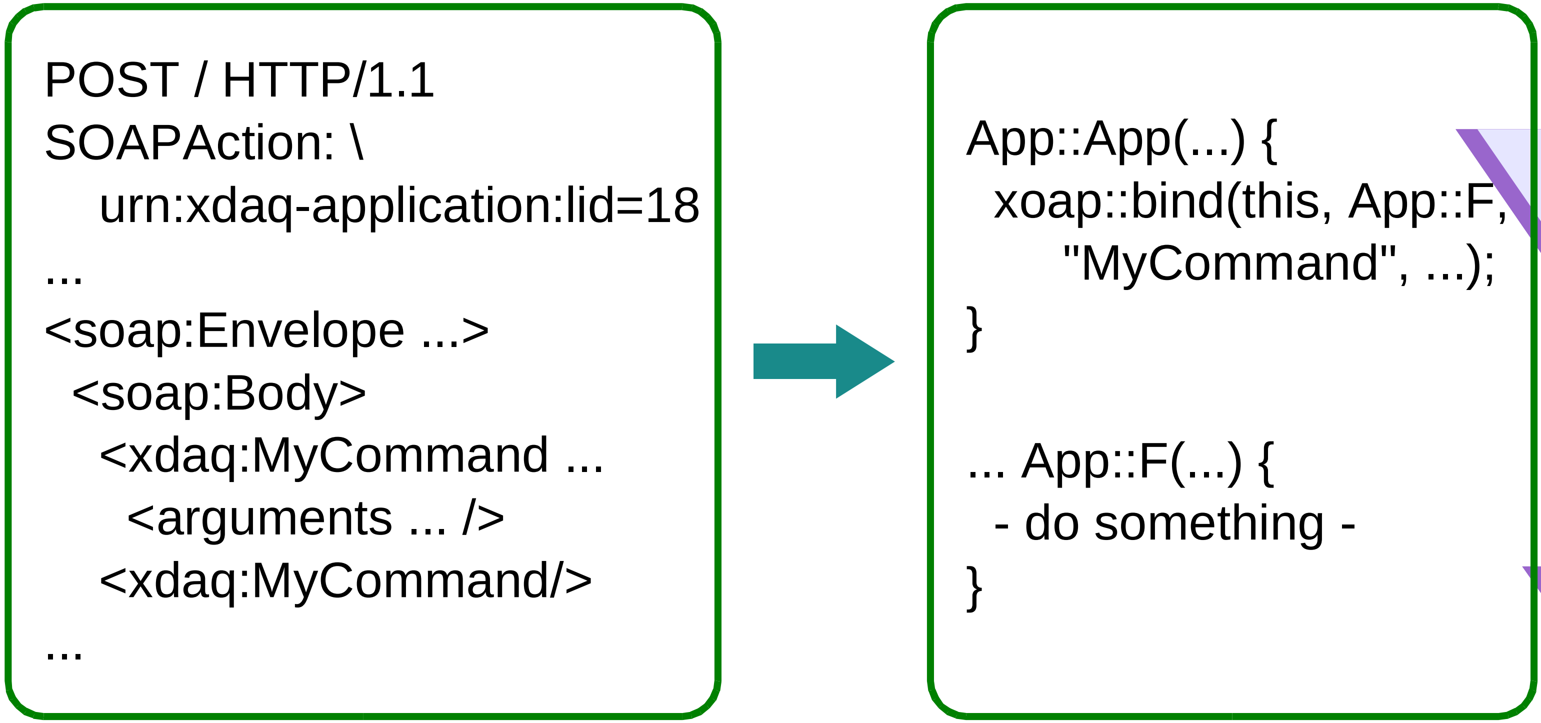
<table>
<tr><th colspan="2">3
31 … 24</th><th colspan="2">2
23 … 16</th><th colspan="2">1
15 … 8</th><th colspan="2">0
7 … 0</th><th></th></tr>
<tr><td colspan="4">MessageSize</td><td colspan="2">MessageFlags</td><td colspan="2">VersionOffset</td><td>0</td></tr>
<tr><td colspan="2">Function</td><td colspan="3">InitiatorAddress</td><td colspan="3">TargetAddress</td><td>4</td></tr>
<tr><td colspan="8">InitiatorContext</td><td>8</td></tr>
<tr><td colspan="8">TransactionContext</td><td>12(16)</td></tr>
<tr><td colspan="4">OrganizationID</td><td colspan="4">XFunctionCode</td><td>16(24)</td></tr>
<tr><td colspan="8">Private payload defined by owner of OrganizationID specific
For XFunctionCode</td><td>20(28)</td></tr>
</table>

XDAQ: SOAP Messages

- W3C standard to send information in XML
- Mostly used on HTTP
- Synchronous protocol
- Many libraries/parsers available
- In XDAQ;
  - Send a command (w/ arguments)
  - Get/set application's parameters
  - WS-eventing for error-report / monitoring

Command Message

- SOAP message drives application's call-back.
- Xerces-C based SOAP library to help users.



```
POST / HTTP/1.1
SOAPAction: \
    urn:xdaq-application:lid=18
...
<soap:Envelope ...>
  <soap:Body>
    <xdaq:MyCommand ...
      <arguments ... />
    <xdaq:MyCommand/>
...
```

```
App::App(...) {
  xoap::bind(this, App::F,
       "MyCommand", ...);
}

... App::F(...) {
  - do something -
}
```

Parameter Access

- 'Exported' parameters are accessed via SOAP messages.

```
class App {
  xdata::String mode;
  xdata::Integer count;
}
...
App::App(...) {
  fireItemAvailable(
    "mode",  &mode);
  fireItemAvailable(
    "count",  &count);
}
```

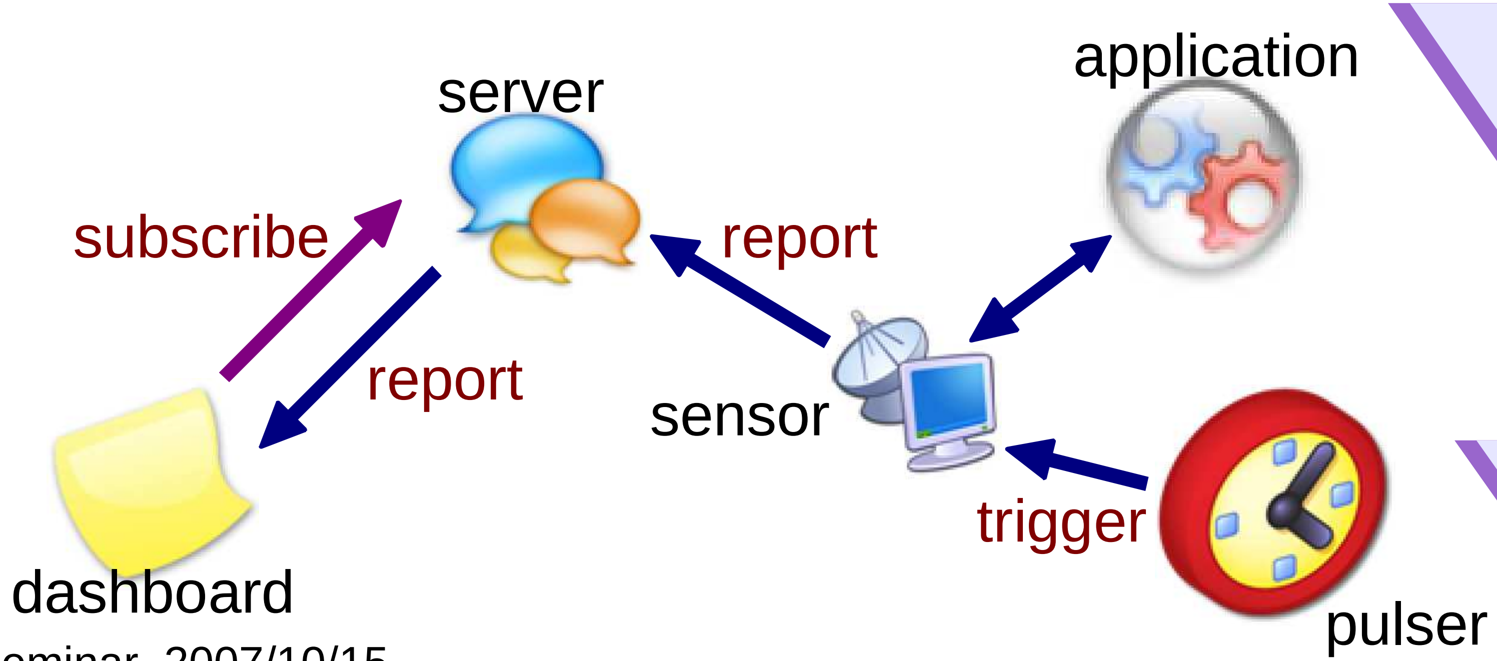
```
<soap:Body>
  <xdaq:ParameterGet ...
    <App:properties>
      <mode type="string"...
      <count type="integer"...
...
```

WS-Eventing

- XML based publish/subscribe messaging.
- Clients sends messages to a server.
- Server distributes messages to subscribers.
- In XDAQ,
  - used for monitoring and exception propagation
  - server: in-house development
  - helper applications to feed parameters or exceptions to the WS-E system

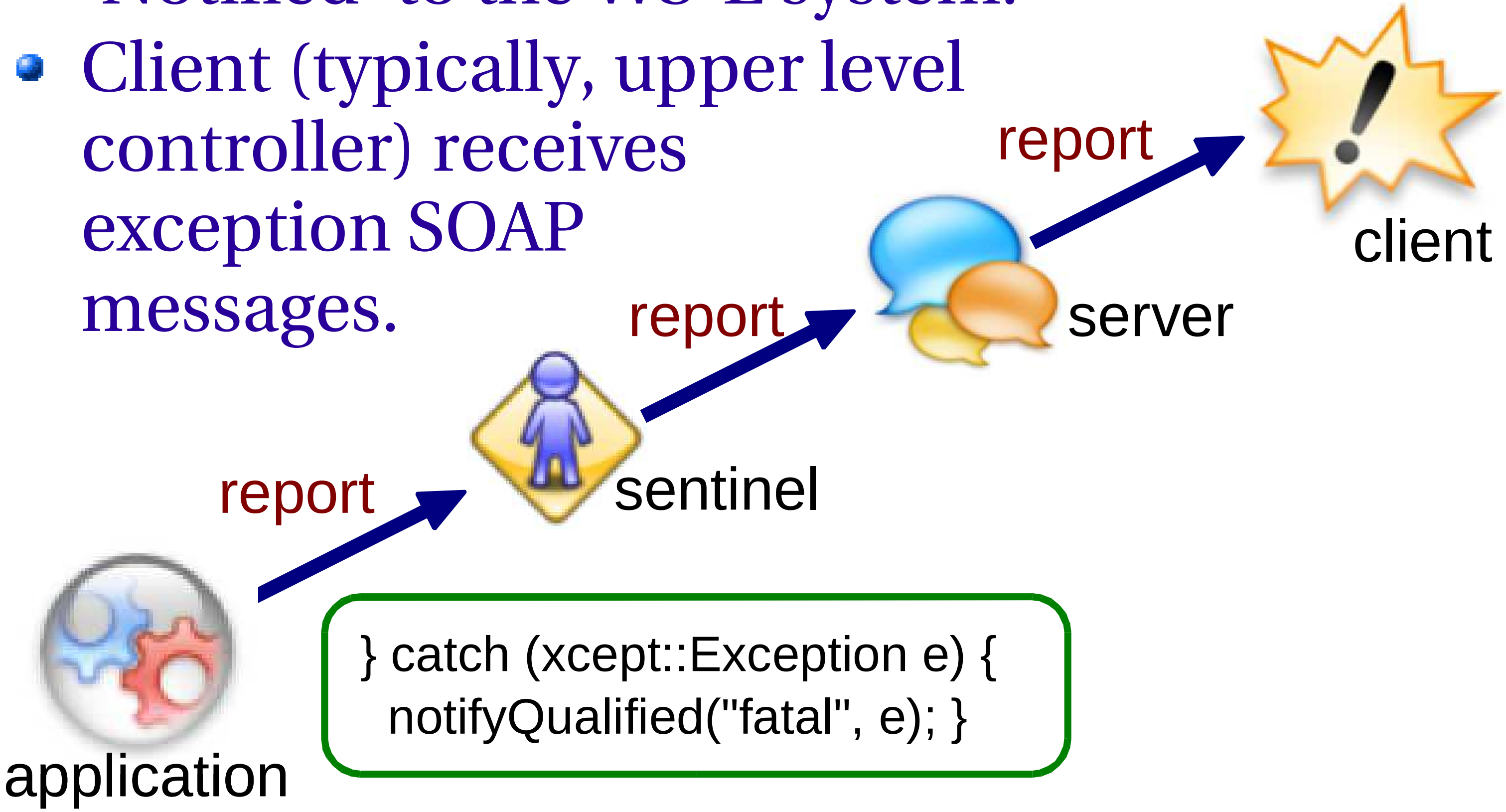
Monitoring Scheme

- Monitoring information 'table' is sent to WS-eventing server as a SOAP message with binary attachment (type-tagged XDR).
- Clients subscribes to the server, with XPath query.



Exception Handling Scheme

- C++ exception handling inside applications.
- 'Notified' to the WS-E system.
- Client (typically, upper level controller) receives exception SOAP messages.



```
} catch (xcept::Exception e) {
  notifyQualified("fatal", e); }
```

XDAQ: Data Types

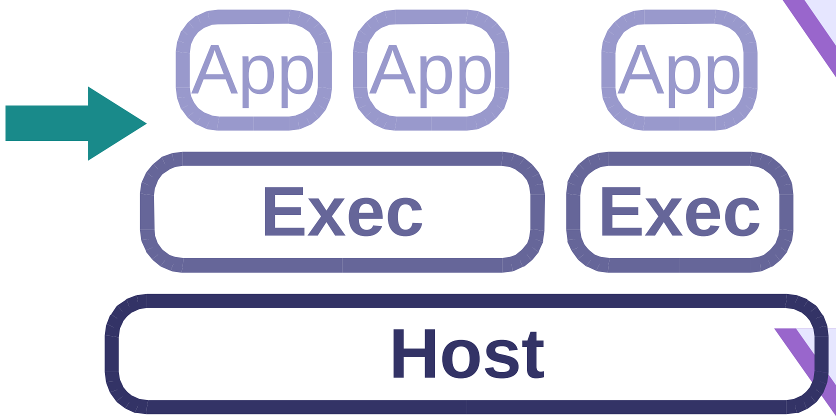
- Serializable data objects
  - Simple types and container types
  - Serializable to XML or type-tagged XDR.
- Used in,
  - Application's exported parameters.
  - Contents of monitor report.

Boolean, Integer, Long, Integer32, Integer64, UnsignedInteger, UnsignedInteger32, UnsignedInteger64, UnsignedShort, UnsignedLong, Float, Double, Timeval, Bag, Vector, Table, Properties, Mime

XDAQ: Configuration by XML

- Sending 'Configuration' XML to the platform (executive) loads applications dynamically.

```
<x:Partition ...>
  <x:Context url=”...
    <x:Application class="App"
    <x:Application class="App"
    <x:Module>libApp.so</...
  </x:Context>
  <x:Context url=”...
    <x:Application class="App"
    <x:Module>libApp.so</...
  </x:Context>
</x:Partition>
```



XDAQ: Miscellaneous Tools

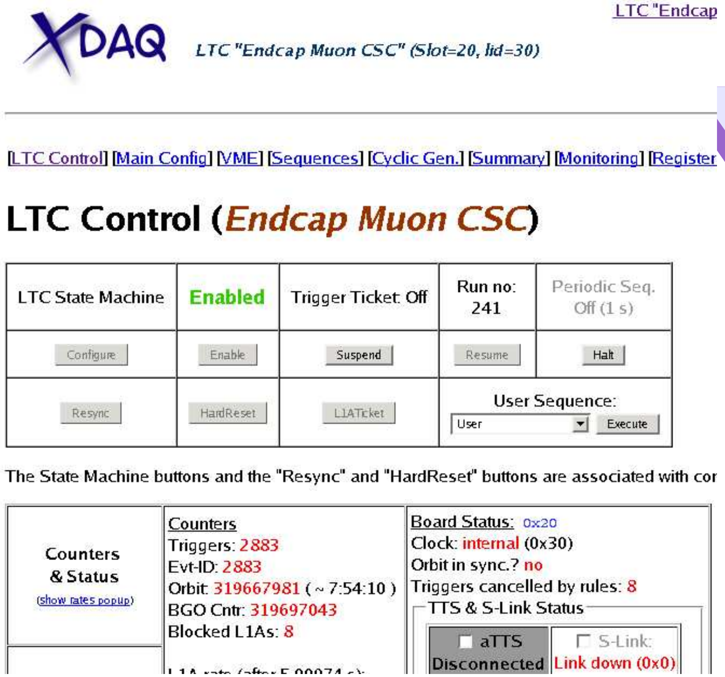
- Web interface: HyperDAQ
- Finite State Machine
- Logging
- Worker thread
- Service discovery via SLP
- Helper applications
- ... and a lot more ...

XDAQ: HyperDAQ

- Enables users to access applications with Web browsers
- Binding HTML call-backs to URLs

```
App::App(...) {
  xgi::bind(this,
       App::Page, "Page");
}


... App::Page(...) {
  - write out HTML -
  - using cgicc -
}
```



LTC "Endcap

LTC "Endcap Muon CSC" (Slot=20, lid=30)

[LTC Control] [Main Config] [VME] [Sequences] [Cyclic Gen.] [Summary] [Monitoring] [Register

LTC Control (*Endcap Muon CSC*)

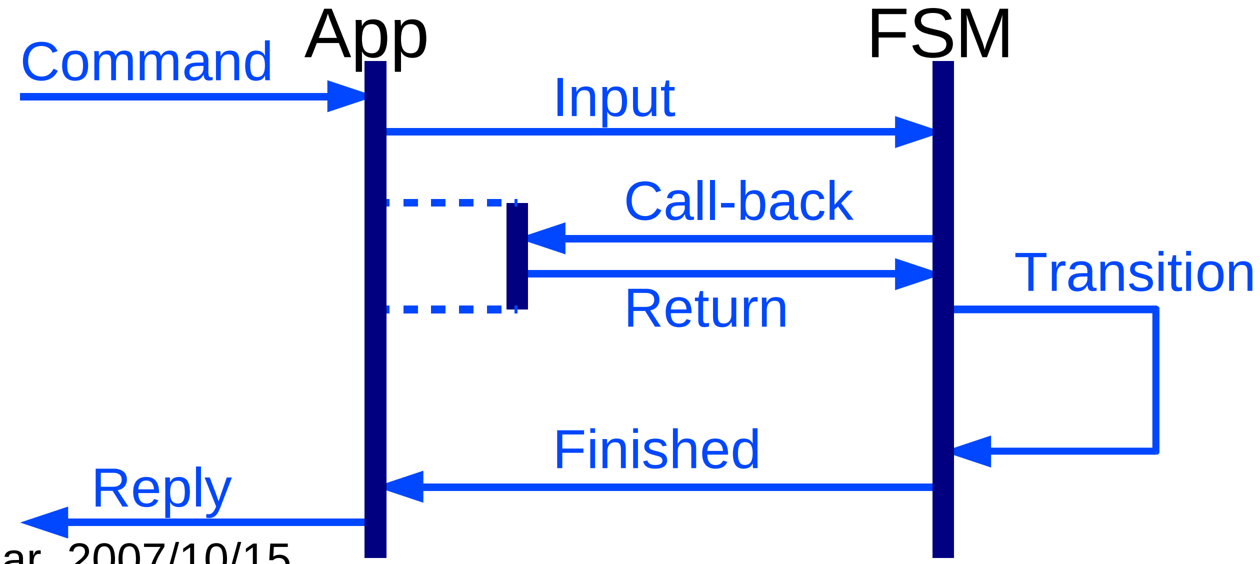
LTC State Machine	Enabled	Trigger Ticket: Off	Run no: 241	Periodic Seq. Off (1 s)
Configure	Enable	Suspend	Resume	Halt
Resync	HardReset	L1ATicket	User Sequence: User ▾ Execute	

The State Machine buttons and the "Resync" and "HardReset" buttons are associated with cor

Counters & Status (show rates popup)	Counters Triggers: 2883 Evt-ID: 2883 Orbit: 319667981 (~ 7:54:10) BGO Cntr: 319697043 Blocked L1As: 8 L1A rate (after 5.00074 s):	Board Status: 0x20 Clock: internal (0x30) Orbit in sync.? no Triggers cancelled by rules: 8 TTS & S-Link Status □ aTTS Disconnected □ S-Link: Link down (0x0)

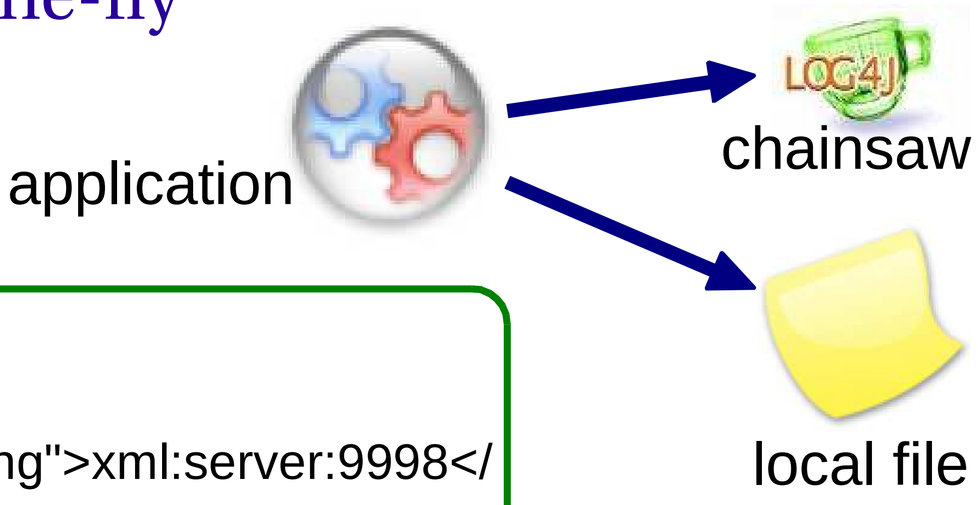
XDAQ: Finite State Machine

- In general, behavior of control applications are understood in a finite state machine model.
- User can bind XDAQ application functions to FSM state transitions.
- Typically, combined with SOAP call-backs.



XDAQ: Logging

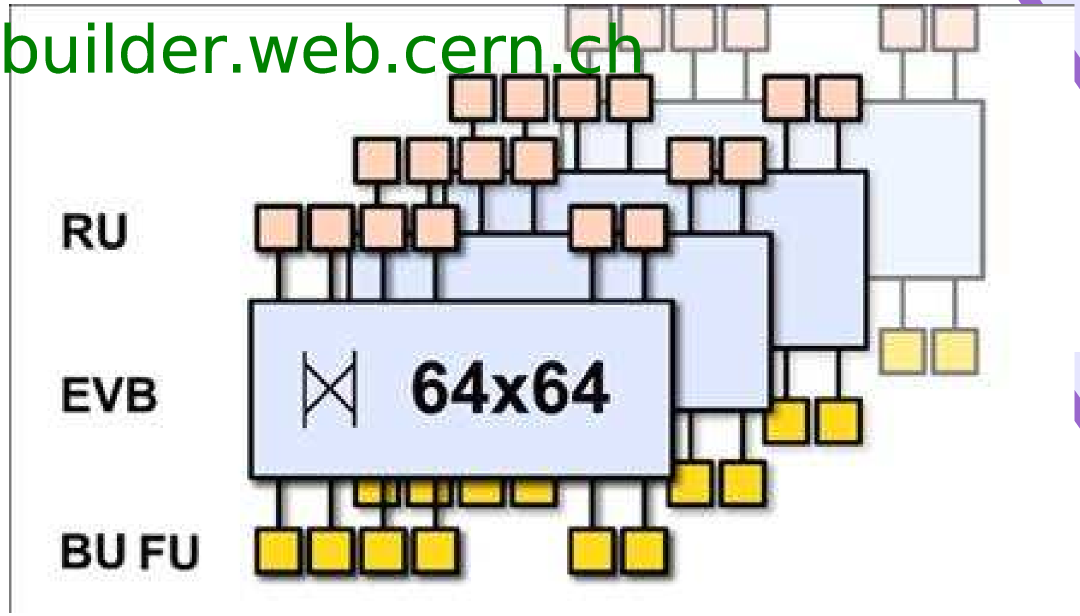
- log4cplus: C++ library compatible with log4j
- Using TTCC format everywhere
- Custom made appenders, XML and UDP
- Configurable by the application configuration file, as well as on-the-fly
(log level).



```
<x:Application ...
  <properties>
    <logUrl type="string">xml:server:9998</
    <logLevel type="string">DEBUG</
```

Use Case – Event Builder

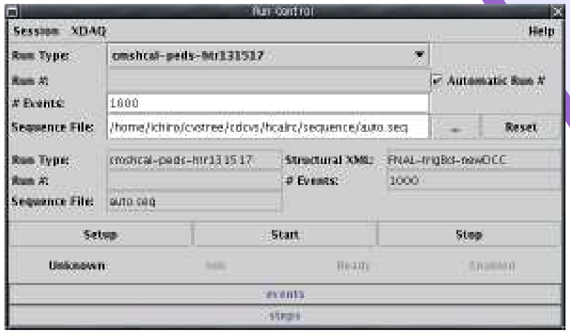
- PC farm interconnected by a big GbE switch.
- Input side: ~60 PCs with Myrinet cards
- Output side: O(10^2) PCs
- ~600 XDAQ applications in the current setup
- S. Murray
  http://cms-ru-builder.web.cern.ch



Use Case – HCAL Test Beam DAQ

- Java stand-alone GUI to control XDAQ applications.
- Stand-alone event builder using same I2O messages as CMS EVB

  → Switching to CMS EVB for combined runs (Muon, ECAL)
- Used also for commissioning, test benches and beam tests of other small detectors.
- J. Mans



Run Type: cmshcal-peds-htr131517

Run #:

☑ Automatic Run #

Events: 1000

Sequence File: /home/ichiro/cvstree/cdcvs/hcalrc/sequence/auto.seq

Run Type: cmshcal-peds-htr131517

Structural XML: FNAL-trigBit-newDCC

Run #:

Events: 1000

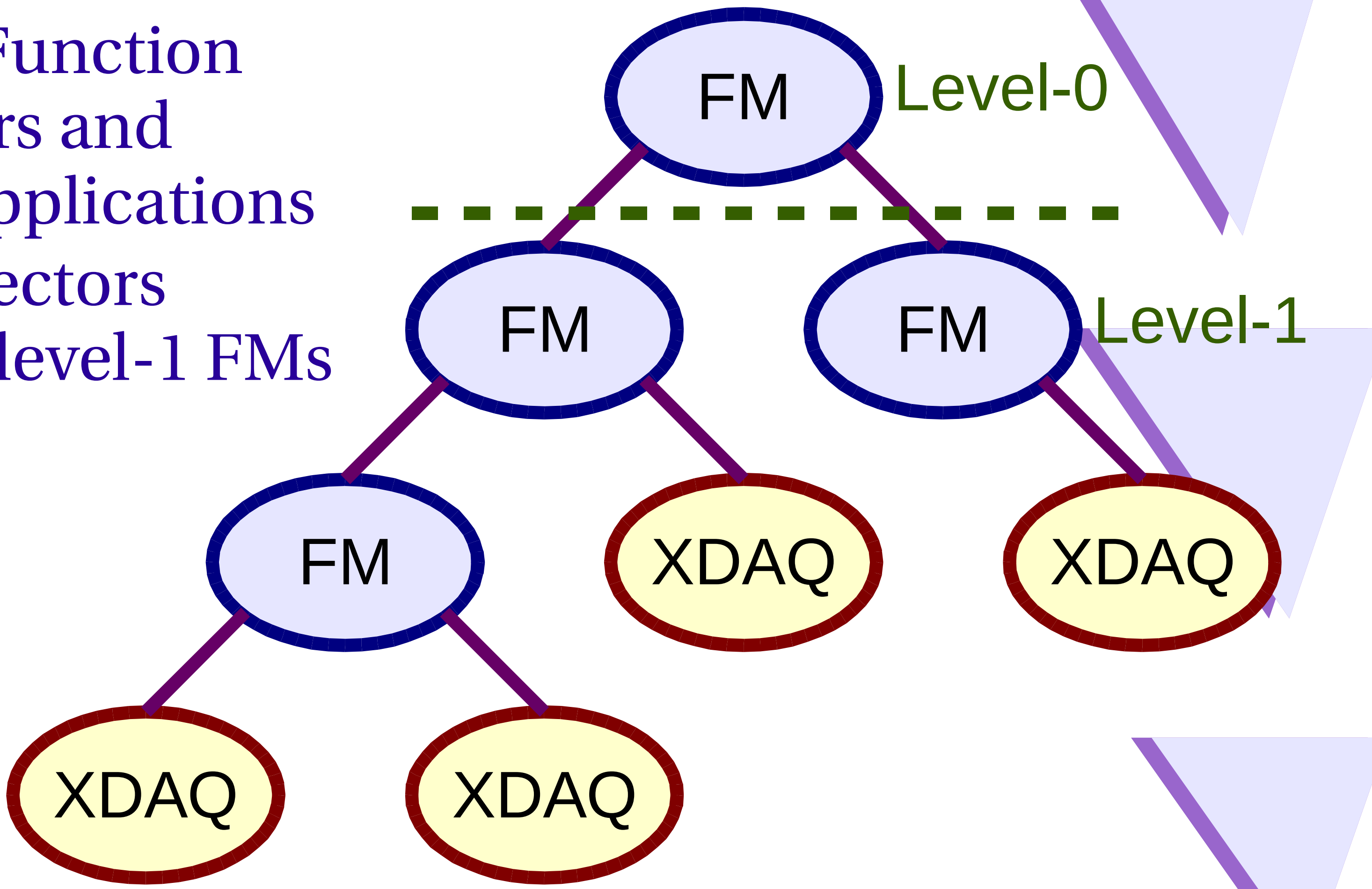
Sequence File: auto.seq

RCMS

- Run Control and Monitor System
- Configuration and control framework for CMS
- written in Java
- Function Manager: node of the control tree
  - Flexible structure for
  - Providing concrete DAQ-subdetector interface
- Uniform configuration DB for everyone
- First release in 2004.
- A. Oh, A. Petrucci, M. Gulmini et.al.
  http://cmsdoc.cern.ch/TriDAS/RCMS/

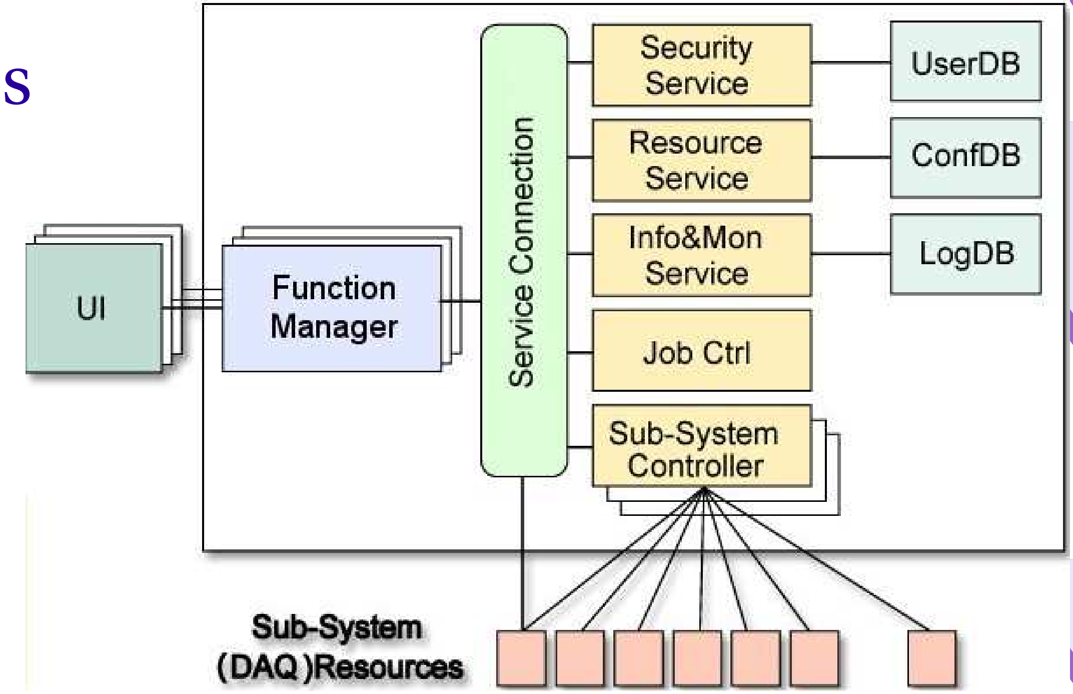
RCMS: Control Structure

- Tree of Function Managers and XDAQ applications
- Sub-detectors provide level-1 FMs



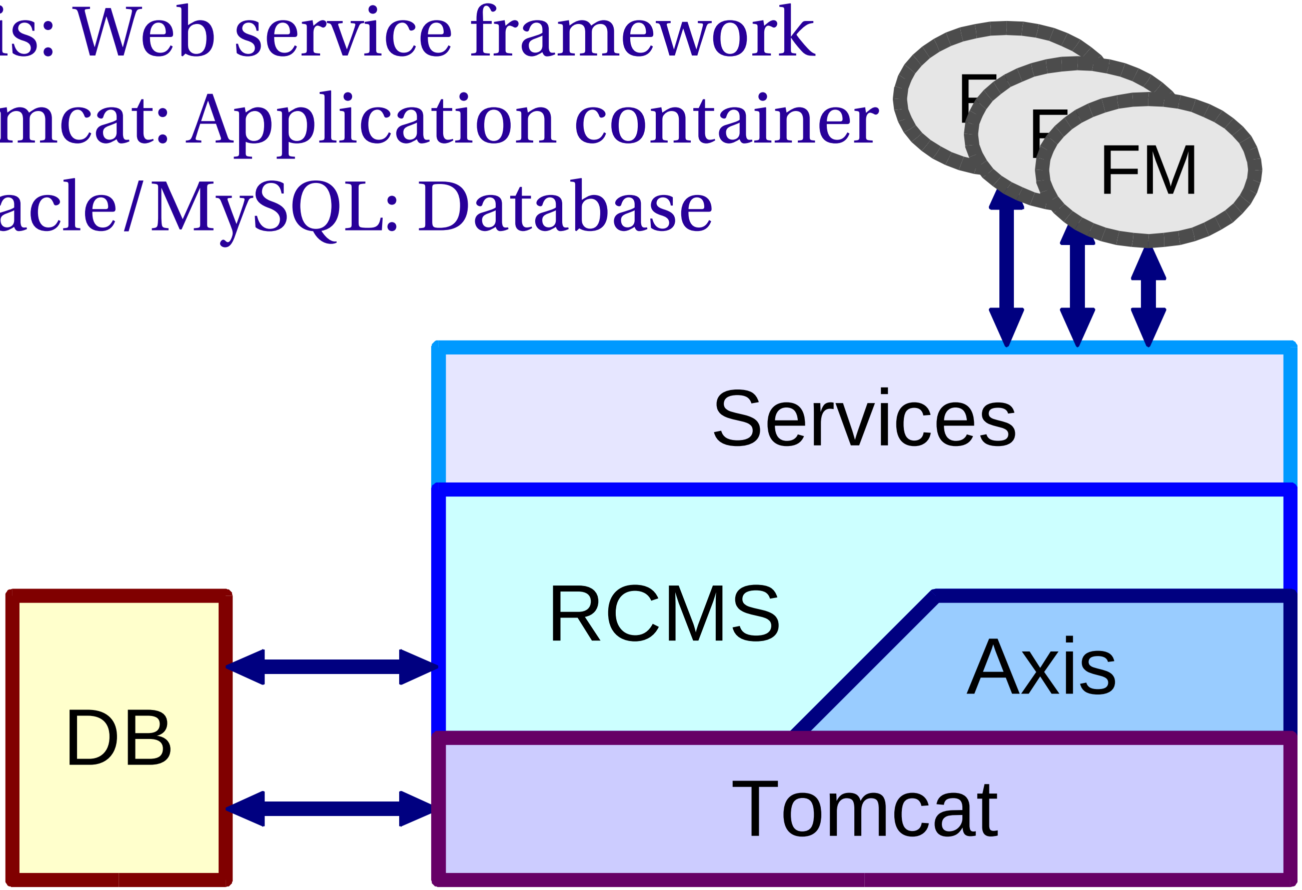
RCMS: Services

- Security: user account
- Resource: configuration
- Info & Monitor: status/messages
- Job control: start/stop applications

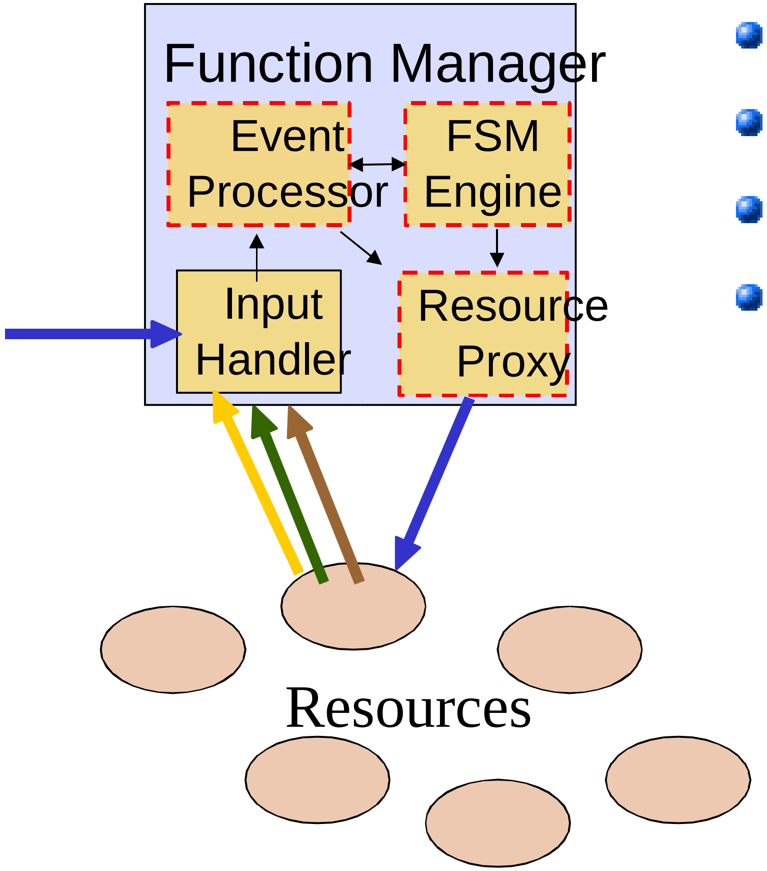


RCMS: Technologies

- Axis: Web service framework
- Tomcat: Application container
- Oracle/MySQL: Database



RCMS: Function Manager



- Input Handler
- Event Processor
- FSM
- Resource Proxy
- 'Resources': FM, XDAQ, JobControl, etc.

← monitor flow

← control flow

← state flow

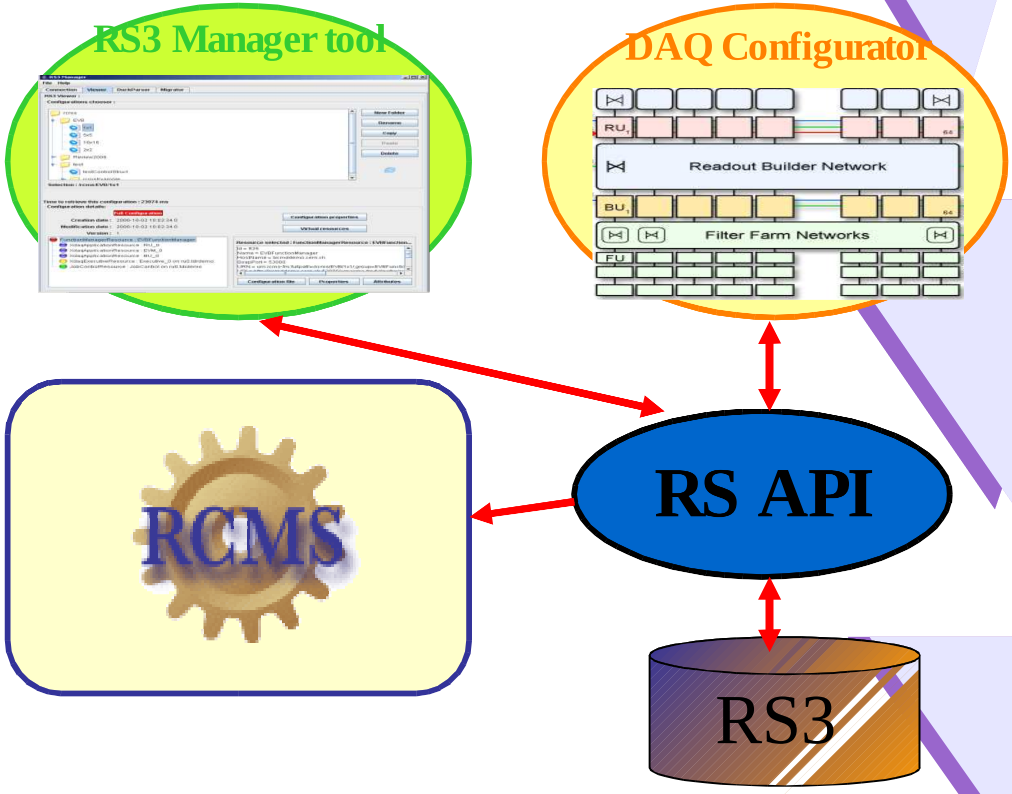
← error flow

▨ customizable

RCMS: Configuration DB

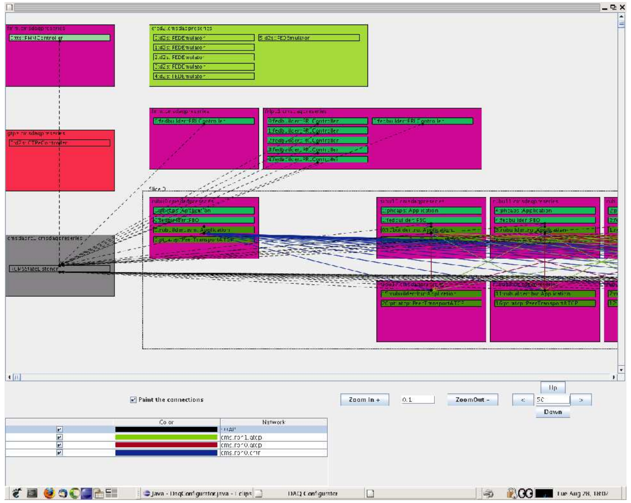
- Resource Service 3: DAQ configuration DB

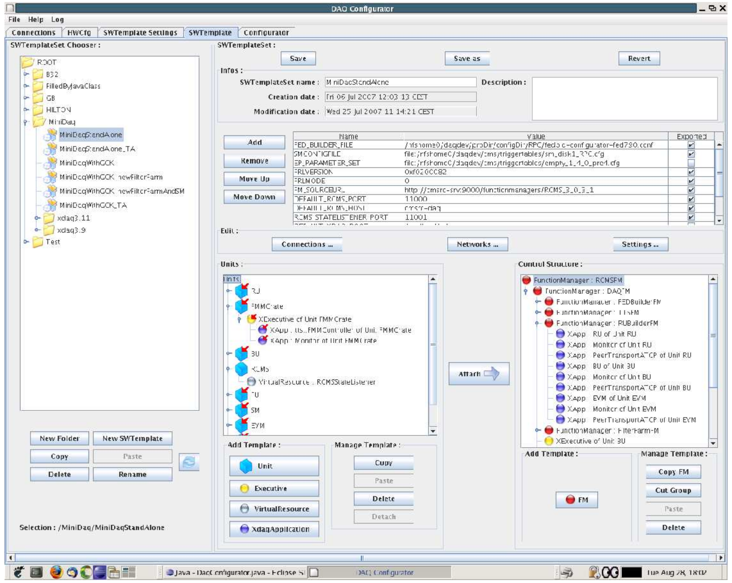
features
Fexible schema
Java API to R/W in RS3
Configuration documents can be built on the fly from relational schema
Versioning configuration system
Oracle and MySQL Compatible



RCMS: Configuration Tools

- DAQ Configurator
  - Tool for central DAQ group: for large scale system.
  - Can be used both graphically or programmatically.





SWTemplateSet Chooser :

- ROOT
  - B32
  - FilledByJavaClass
  - GB
  - HILTON
  - MiniDaq
    - MiniDaqStandAlone
    - MiniDaqStandAlone_TA
    - MiniDaqWithGCK
    - MiniDaqWithGCK_newFilterFarm
    - MiniDaqWithGCK_newFilterFarmAndSM
    - MiniDaqWithGCK_TA
  - xdaq3.11
  - xdaq3.9
  - Test

SWTemplateSet :

Infos :

SWTemplateSet name : MiniDaqStandAlone

Creation date : Fri 06 Jul 2007 12:03 13 CEST

Modification date : Wed 25 Jul 2007 11 14:21 CEST

Description :

Name	Value	Exported
FED_BUILDER_FILE	/nfshome0/daqdev/proDir/configDir/RPC/fedbuilder-configurator-fed790.conf	☑
SMCONFIGFILE	file:/rfshome0/daqdev/cms/triggertables/sm_disk1_RPC.cfg	☑
EP_PARAMETER_SET	file:/rfshome0/daqdev/cms/triggertables/empty_1_4_0_pre1.cfg	□
FRLVERSION	0xf020CC82	☑
FRLMODE	0	☑
FM_SOURCEURL	http://cmsrc-srv:9000/functionmanagers/RCMS_3_0_3_1	☑
DEFAULT_RCMS_PORT	11000	☑
DEFAULT_RCMS_HOST	cmsrc-daq	☑
RCMS STATELISTENER PORT	11001	☑

Edit :

Units :

- Units
  - RU
  - FMMCrate
    - XExecutive of Unit FMMCrate
      - XApp : tts::FMMController of Unit FMMCrate
      - XApp : Monitor of Unit FMMCrate
  - BU
  - CLMs
    - VirtualResource : RCMSStateListener
  - FU
  - SM
  - EVM

Control Structure :

- FunctionManager : RCMSFM
  - FunctionManager : DAQFM
    - FunctionManager : FEDBuilderFM
    - FunctionManager : TTSFM
    - FunctionManager : RUBuilderFM
      - XApp RU of Unit RU
      - XApp Monitor of Unit RU
      - XApp PeerTransportATCP of Unit RU
      - XApp BU of Unit BU
      - XApp Monitor of Unit BU
      - XApp PeerTransportATCP of Unit BU
      - XApp EVM of Unit EVM
      - XApp Monitor of Unit EVM
      - XApp PeerTransportATCP of Unit EVM
    - FunctionManager : FilterFarmFM
      - XExecutive of Unit FU

Add Template : Unit, Executive, VirtualResource, XdaqApplication

Manage Template : Copy, Paste, Delete, Detach

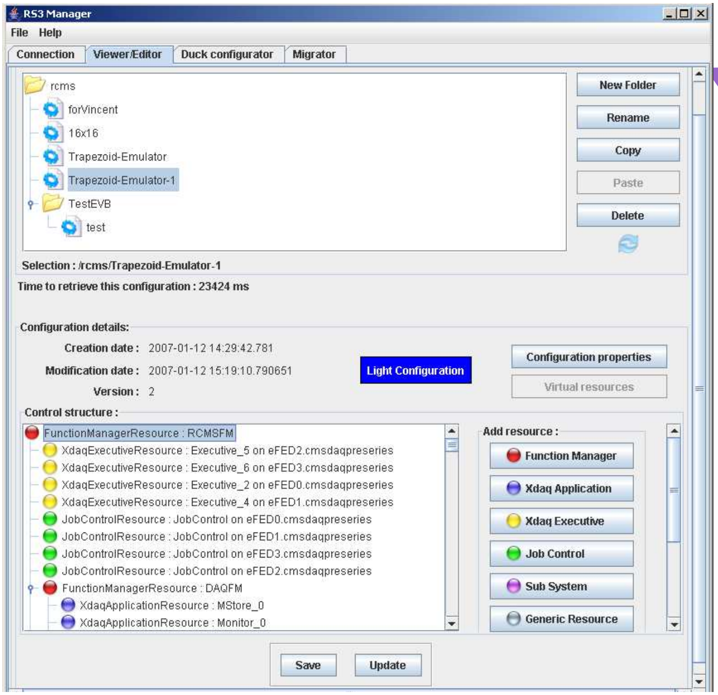
Add Template : FM

Manage Template : Copy FM, Cut Group, Paste, Delete

Selection : /MiniDaq/MiniDaqStandAlone

RCMS: Configuration Tools (cont'd)

- Resource Service Manager
- Reads XML configuration files and stores into the DB
- GUI
  - define structure
  - change parameters
  - organize in folders
  - associate with global key



rcms
- forVincent
- 16x16
- Trapezoid-Emulator
- Trapezoid-Emulator-1
- TestEVB
  - test

Selection : /rcms/Trapezoid-Emulator-1

Time to retrieve this configuration : 23424 ms

Configuration details:

Creation date : 2007-01-12 14:29:42.781

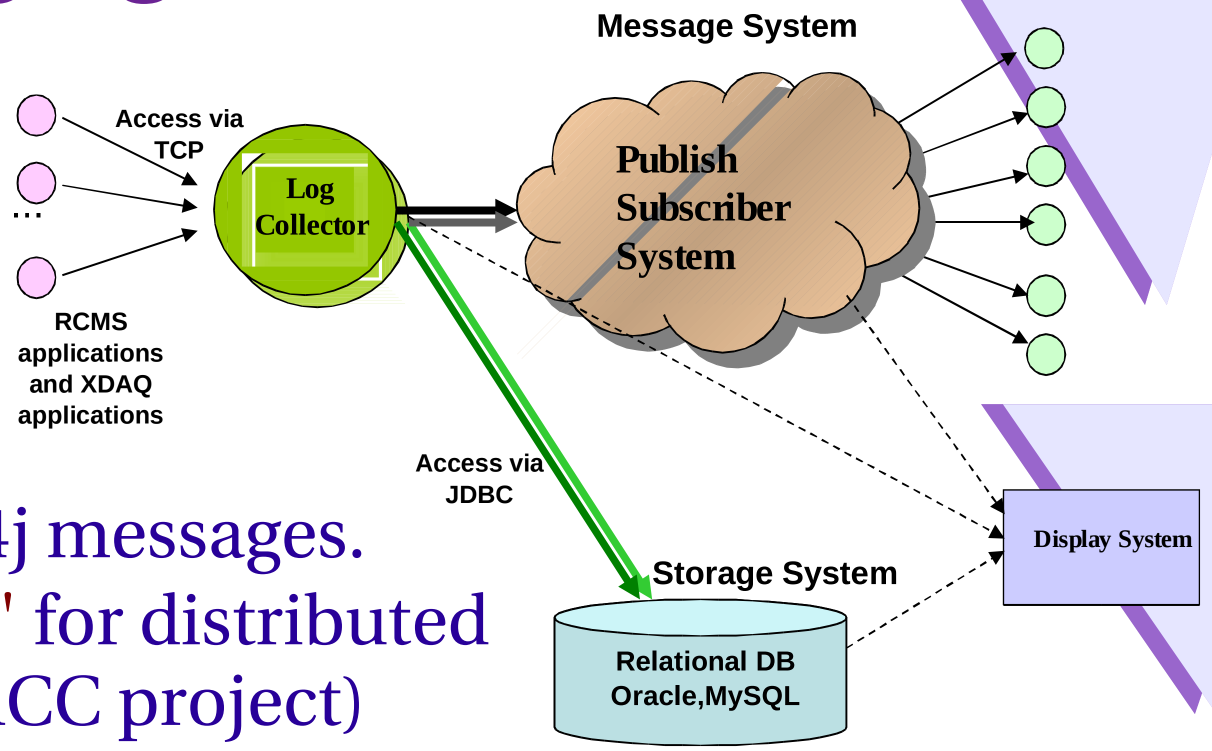
Modification date : 2007-01-12 15:19:10.790651

Version : 2

Control structure :

- FunctionManagerResource : RCMSFM
  - XdaqExecutiveResource : Executive_5 on eFED2.cmsdaqpreseries
  - XdaqExecutiveResource : Executive_6 on eFED3.cmsdaqpreseries
  - XdaqExecutiveResource : Executive_2 on eFED0.cmsdaqpreseries
  - XdaqExecutiveResource : Executive_4 on eFED1.cmsdaqpreseries
  - JobControlResource : JobControl on eFED0.cmsdaqpreseries
  - JobControlResource : JobControl on eFED1.cmsdaqpreseries
  - JobControlResource : JobControl on eFED3.cmsdaqpreseries
  - JobControlResource : JobControl on eFED2.cmsdaqpreseries
  - FunctionManagerResource : DAQFM
    - XdaqApplicationResource : MStore_0
    - XdaqApplicationResource : Monitor_0

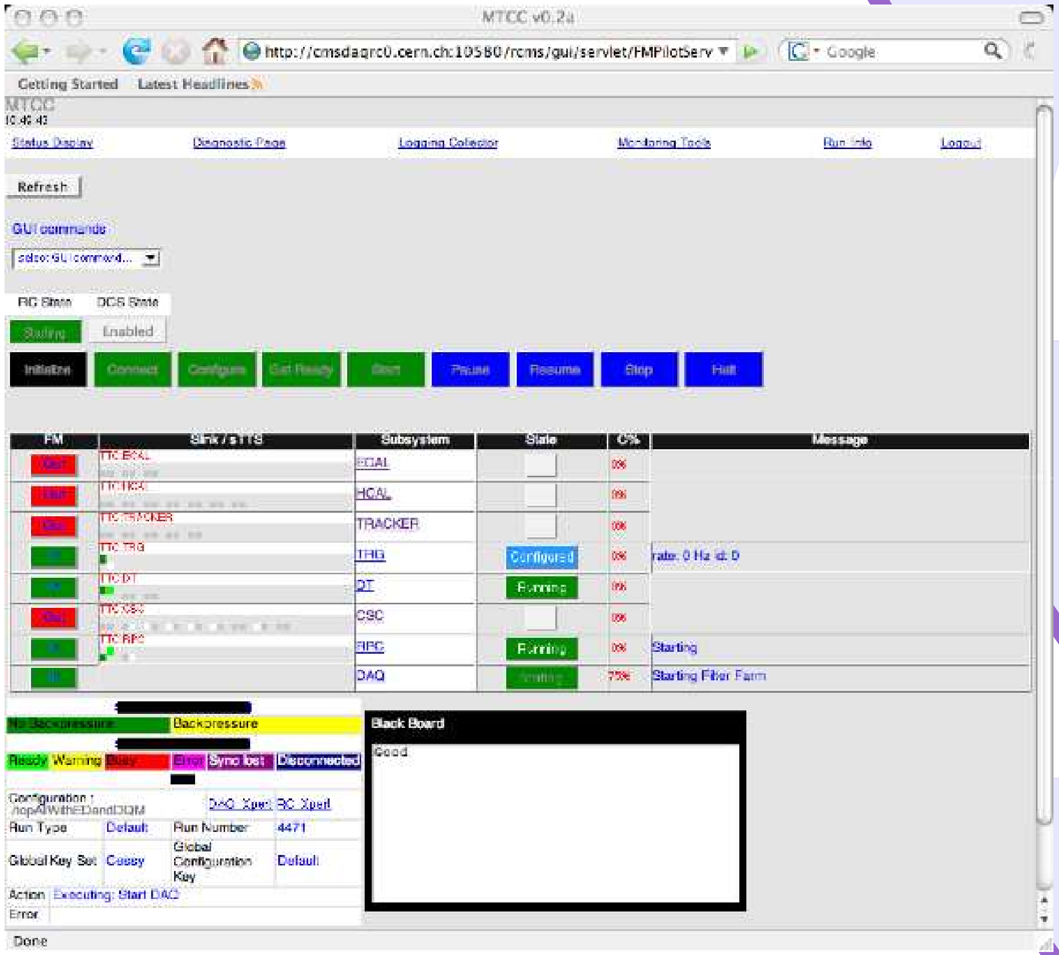
RCMS: Logging


Message System
Access via TCP
Log Collector
Publish Subscriber System
...
RCMS applications and XDAQ applications
Access via JDBC
Storage System
Relational DB Oracle,MySQL
Display System

- Based on log4j messages.
- 'LogCollector' for distributed systems (GridCC project)
- Chainsaw for user I/F
- 'LogDBViewer' for stored logs
- JMS for messaging

RCMS: GUI

- JSP based GUI
- Selection of sub-detector / part of sub-detector
- Used for >1 year



MTCC v0.2a

http://cmsdaqrc0.cern.ch:10580/rcms/gui/servlet/FMPilotServ

Getting Started Latest Headlines

MTCC
10:40:43

Status Display Diagnostic Page Logging Collector Monitoring Tools Run Info Logout

Refresh

GUI commands
select GUI command...

RC State	DCS State
Starting	Enabled

Initialize | Connect | Configure | Get Ready | Start | Pause | Resume | Stop | Halt

FM	Sink / sTTS	Subsystem	State	C%	Message
[illegible]	TTC ECAL	ECAL		0%	
[illegible]	TTC HCAL	HCAL		0%	
[illegible]	TTC TRACKER	TRACKER		0%	
[illegible]	TTC TRG	TRG	Configured	0%	rate: 0 Hz id: 0
[illegible]	TTC DT	DT	Running	0%	
[illegible]	TTC CSC	CSC		0%	
[illegible]	TTC RPC	RPC	Running	0%	Starting
[illegible]		DAQ	Starting	75%	Starting Filter Farm

No Backpressure | Backpressure

Ready | Warning | Busy | Error | Sync lost | Disconnected

Configuration : /topAllWithEDandDQM		DAQ Xpert	RC Xpert
Run Type	Default	Run Number	4471
Global Key Set	Cassy	Global Configuration Key	Default
Action	Executing: Start DAQ		
Error			

Black Board

Good

Done

RCMS: Miscellaneous Tools

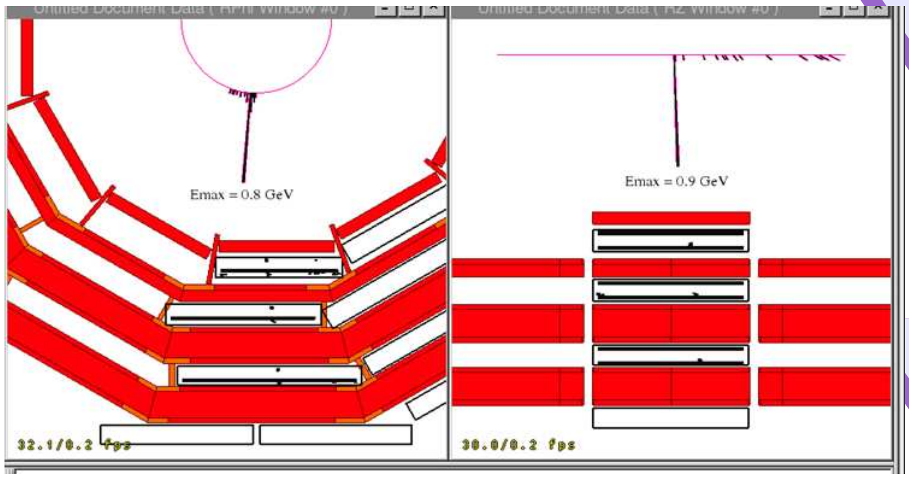
- XDAQ Command / Parameter access
- Thread pool
- Parallel/asynchronous communications
- General message receiver + dispatch for non-RCMS entities.
- additional DBs
  - run number management
  - run-related user information

Interfacing XDAQ-RCMS

- Most of the messaging is on SOAP
  → Easy to handle in Java (RCMS)
- Commands/Parameters: interfacing library
- Logging: log4j compatible XML schema
- Exceptions: mutual agreement on the schema (planned)
- Monitoring: mutual agreement on the schema + interfacing library (to handle binary format) (on-going work)

Operations: Global Runs

- MTCC on surface in 2006.
- Global runs every month since May 2007.
  - Real muon triggers, all components timed-in
  - Still ~3% of the readout. (no Tracker)
- Good for software integration and prioritize task list.
- Cosmic run in Nov.
  - All detectors?


Untitled Document Data (RPhi Window #0)
Untitled Document Data (RZ Window #0)
Emax = 0.8 GeV
Emax = 0.9 GeV
32.1/0.2 fps
30.0/0.2 fps

Operations: Testing

- Testing is important in a complex system, like CMS DAQ.
- Many components have unit tests.
- Most of the components have semi-automated functionality tests.
- Developers uses test clusters for acceptance tests.
- Test clusters are also used to test integrated systems.
- There are dedicated librarians + integration testers.
- Still, a few new problems show up in every global runs ...

Operations: Testing (cont'd)

- Test clusters with various scale/purpose
- Small: R&D cluster
  - Full chain of 8(FED)-8(RU)-8(BU)-8(FU)
- Medium: Surface integration cluster
  - EVB-FU farm test of 16(RU)-64(BU+FU)
  - Used also for surface cosmic-ray tests
- Large: Production system
  - Currently, 2-slice worth (~600 PCs)
  - Used for global runs

Status / Outlook

- Use of framework for all the online software
  - reduces user's effort.
  - defines interfaces among components.
- Use of open-standard/open-source products
  - reduces development effort.
  - makes integration easier.
- CMS online frameworks are used uniformly in the experiment.
- Further improvements are foreseen as the production system approaches nominal scale.
- CMS DAQ is (almost) ready for the beam runs.

Summary

- XDAQ
  - extensive use of XML → flexible, easy integration
  - extensive use of 'open-source' products
  - all necessary functionalities for DAQ, including fast/slow messaging and utilities
  - used by CMS DAQ + all the sub-detectors
- RCMS
  - extensive use of Web technologies
  - flexible configuration
  - plug-and-play integration with sub-detectors

Summary (cont'd)

- Integration going well, systems are stable
  - XDAQ + RCMS
  - central DAQ + sub-detectors
- Problem areas
  - Deployment scheme / Scalability
  - found slowly, solved slowly but steadily.
- Interactions with users are important.
  - Provide all the functionality, or users start their own development.
  - Quick turn-around time is a key to improve the framework.