Run Control Shifter Training

April 2015

DAQ on-call: 162772
atlas-tdaq-daqhlt-ops@cern.ch

Speaker: Giuseppe Avolio
Material by A.Corso-Radu on behalf of the DAQ Training Team

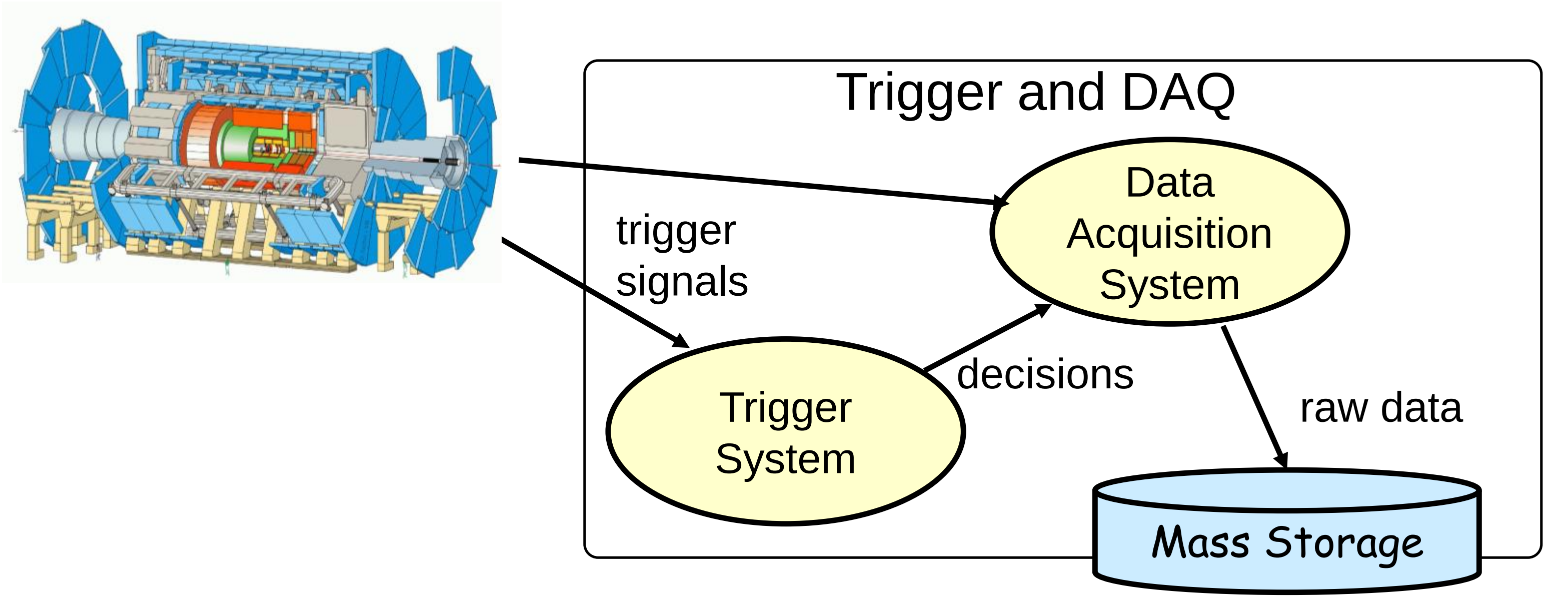
Run Control and DAQ

DAY 1 Session Outline

- DAQ overview and terminology
  - Backpressure, dead-time
- Configuration and Control overview
  - Partition, Segments, Run Control Tree, FSM
- IGUI tool description
- Shifter Assistant
- Recovery concepts
  - Stop-less removal and recovery
  - TTC restart

DAQ overview and terminology

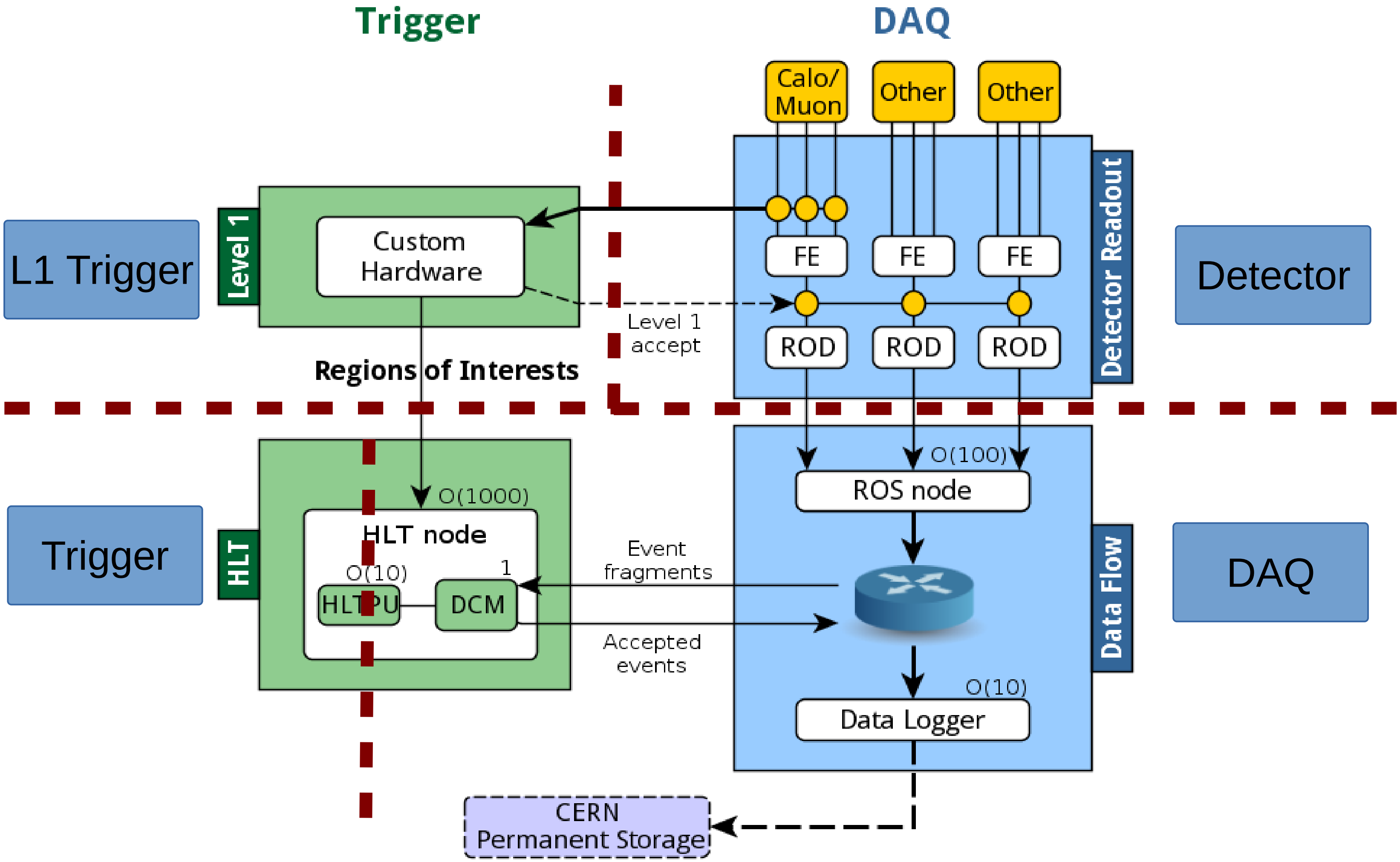
- Overall the main role of Trigger & DAQ is to process the signals generated in a detector and storing the interesting information on a permanent storage



DAQ overview and terminology

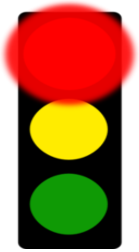
- Trigger
  - Selects interesting events in real time (i.e. with minimal controlled latency)
  - **Level-1**: hardware based, rejection factor ~400
  - **High-Level Trigger**: software based, rejection factor ~100
- DAQ
  - Gathers the data produced by the detectors (**Readout: ROS**)
  - Feeds high level triggers (**HLT**)
  - Builds complete events
  - Stores event data (**SFO**) to permanent storage
  - Provides control, configuration and monitoring facilities (**RunControl**, **Configuration**, **Monitoring**)

DAQ overview and terminology



DAQ Backpressure

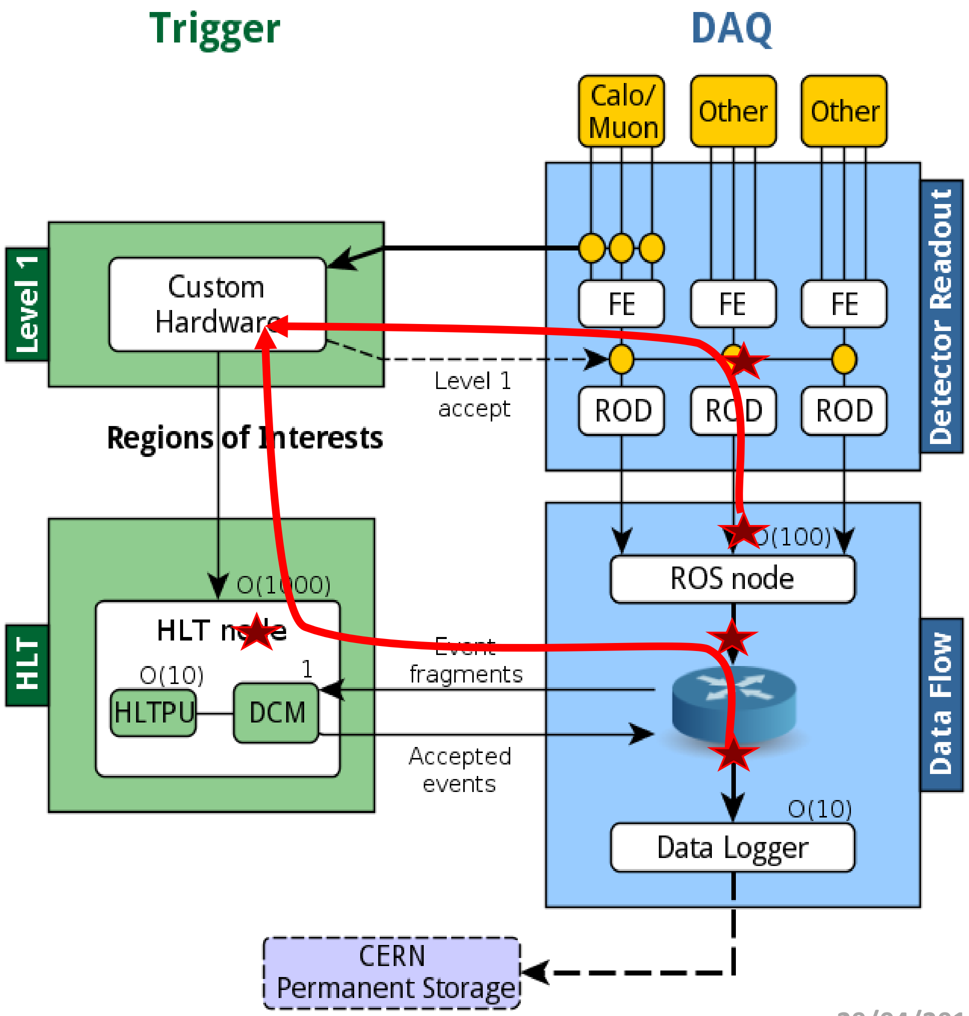
- Dead-time
  - Fraction of time when new triggers are vetoed
  - To be minimized, compatibly with the ATLAS capabilities
- Busy signal is the mechanism to stop triggers -> causes dead-time
  - Busy can be generated by many components
  - Applied centrally by the Level-1 trigger (CTP)



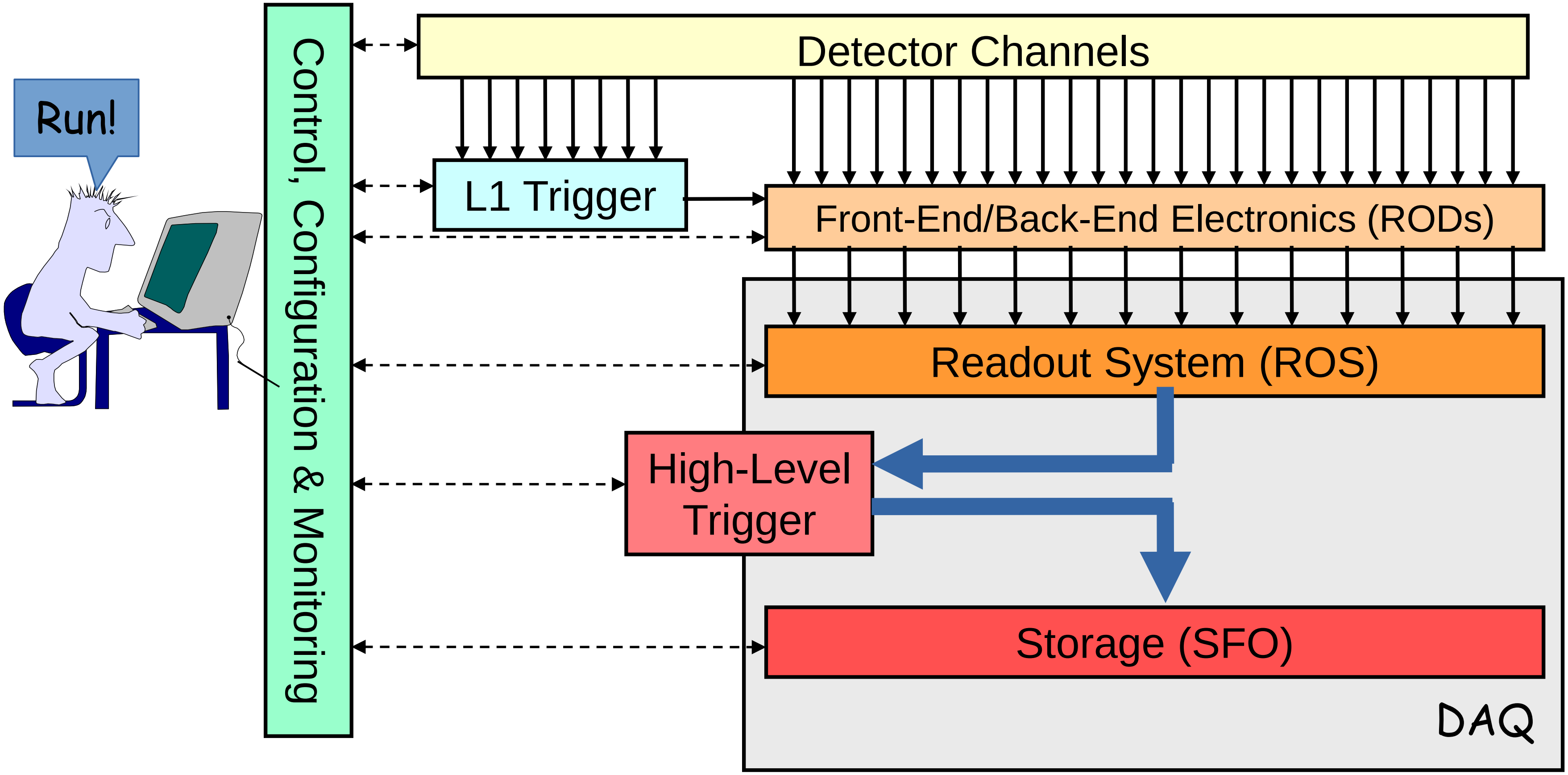
- The system provides bookkeeping of lost luminosity and sources of the busy for latter analysis.

DAQ Backpressure (cont.)

- Busy in DAQ is called "backpressure"
  - Limited capabilities at each level (Readout, HLT, SFO)
  - If a system is saturated, the "pressure" is propagated upstream
- Typically due to indirect causes, e.g.
  - Noisy detector
  - Misconfigurations
  - Hardware failures
  - …



DAQ framework

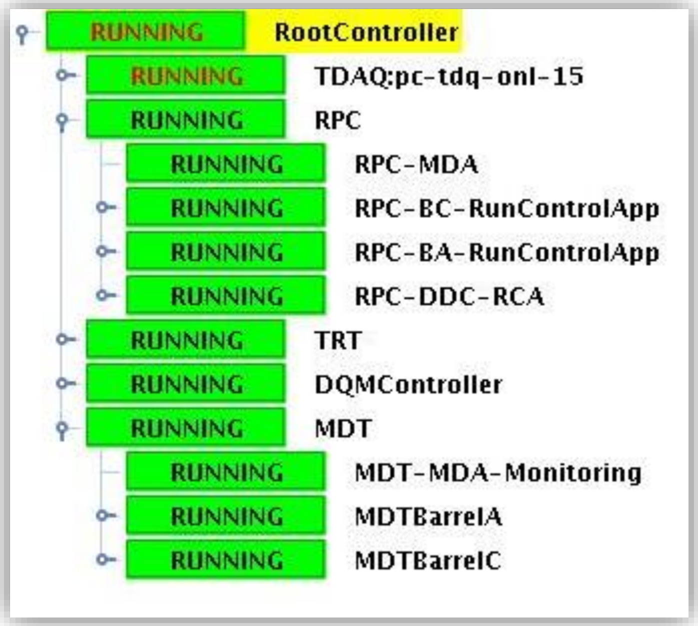


The DAQ Configuration Database

- It defines the content and the behavior of any data taking session
- The database used in the DAQ system is called **OKS** (**O**bject **K**ernel **S**upport)
  - Configuration is stored in XML files on a usually shared file system
- Applications usually access the OKS data via a **R**emote **D**atabase **S**ervice (**RDB**)
  - Avoid scalability issues with thousands of applications accessing concurrently the file system

Partitions & Segments

- A **Partition** is a set of ATLAS systems and detectors taking part to a data-taking session
  - The **ATLAS** partition is the one used for physics
  - Several other partitions used by sub-systems (i.e., detector calibrations)
- Systems and detectors are described and represented by **Segments**
  - A Partition contains several Segments
  - Segments can be nested
  - Leaf nodes are **Applications**



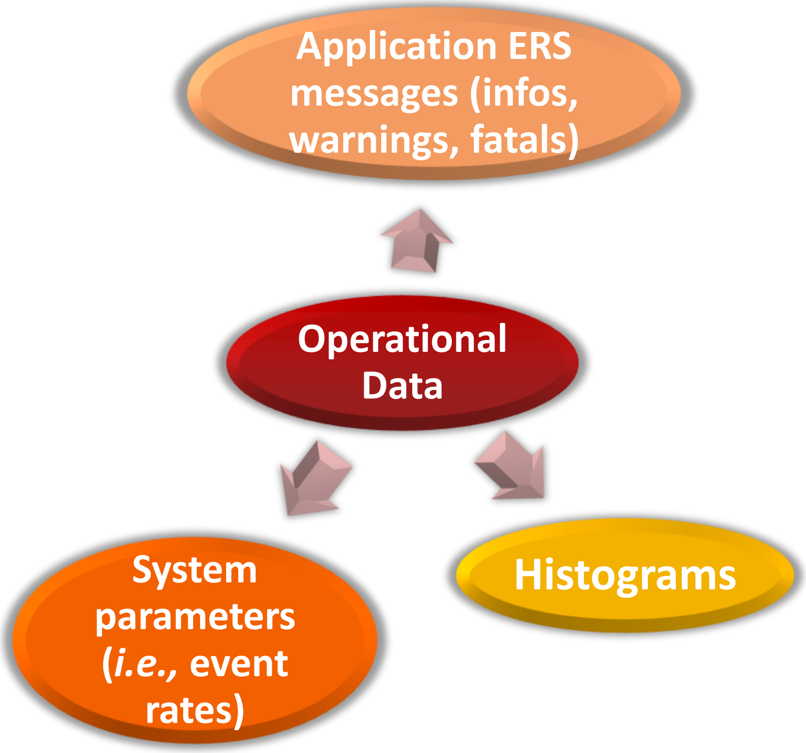
```
RUNNING   RootController
  RUNNING   TDAQ:pc-tdq-onl-15
  RUNNING   RPC
    RUNNING   RPC-MDA
    RUNNING   RPC-BC-RunControlApp
    RUNNING   RPC-BA-RunControlApp
    RUNNING   RPC-DDC-RCA
  RUNNING   TRT
  RUNNING   DQMController
  RUNNING   MDT
    RUNNING   MDT-MDA-Monitoring
    RUNNING   MDTBarrelA
    RUNNING   MDTBarrelC
```

Some Basic Services

- **R**esource **M**anager (RM)
  - It allows to lock resources for exclusive or limit-shared usage as described in the configuration database
    - Example: allow a limited number of application instances to run globally, or per partition, or per host
- **A**ccess **M**anager (AM)
  - It allows to introduce authorization for actions based on users/roles
- **P**rocess **M**ana**g**er (PMG)
  - It is the only service in the system that starts and stops processes
    - It uses the AM and RM services to evaluate whether an operation can be carried out
  - <u>Runs as a system service on every node</u>

Operational Data

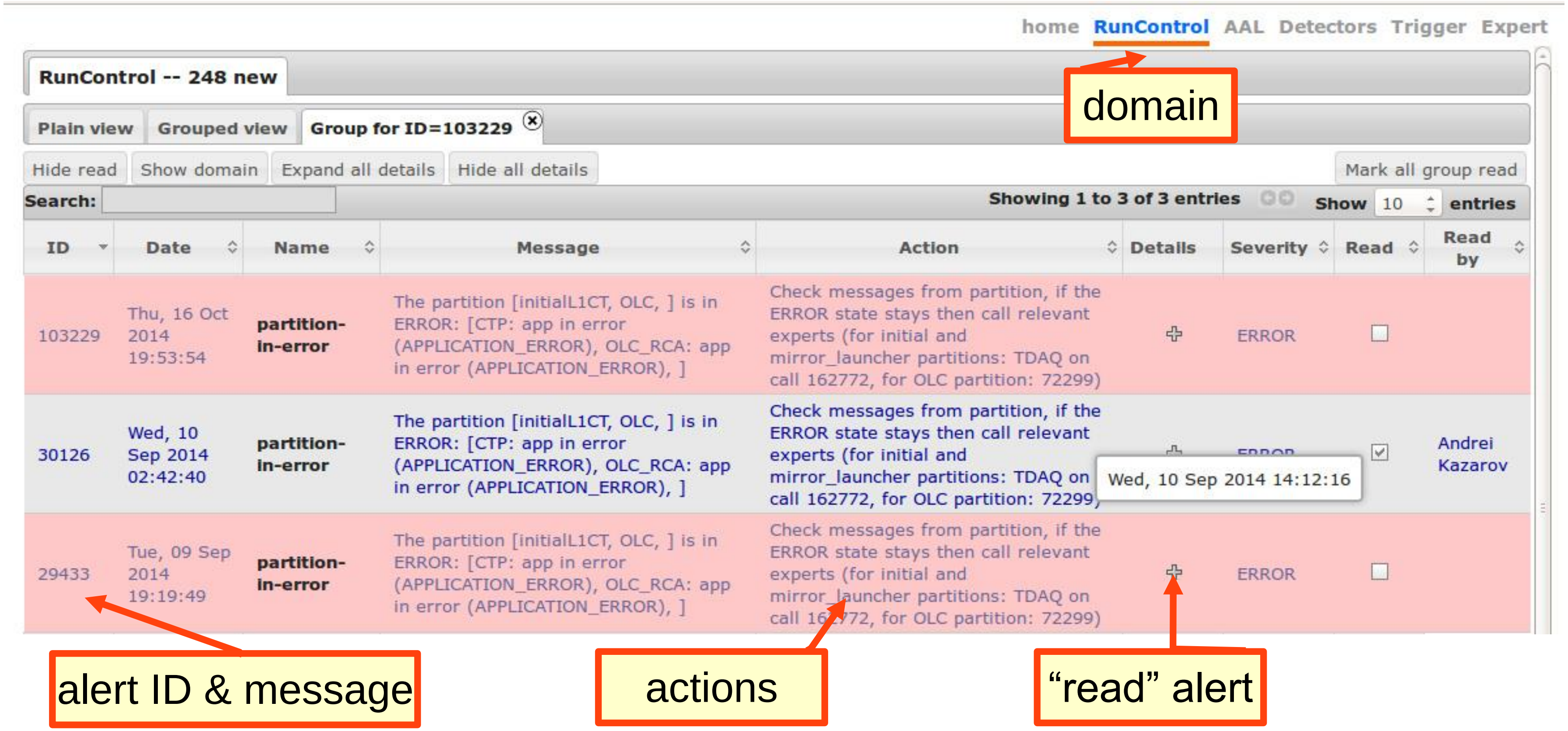
- Applications in the system produce different kind of **operational data** in order to
  - Monitor the status of the data taking session
    - Report anomalies (*i.e.*, detect backpressure)
  - Asses the quality of the recorded data
- Operational data are made available (distributed) to any consumer in the system
  - Different GUI tools available to operators and experts (RunControl specific tutorial)
  - The **Shifter Assistant** being one of the major consumers
- Operational data are archived for latter analysis.


Application ERS messages (infos, warnings, fatals)
Operational Data
System parameters (*i.e.*, event rates)
Histograms

Shifter Assistant

- A service to **reduce and simplify** shifters tasks
  - Subscribe to all sorts of operational data sources.
  - Provide automated monitoring procedures
  - Automatically recognize unexpected behavior of the system as anomaly
  - Suggest actions to take
- Configured by experts to access and analyze any relevant information
- **Alerts** are generated when defined conditions are met
  - Application status outside parameters
  - Start of Run check errors, End of Run
- Alerts are available in a Web application (next slide) via a browser and requiring CERN SSO log-in (and log-out at the end of the session).
- Every desk in ACR has a different "domain" where alerts are presented.
- Once alert is read (i.e. "acknowledged") by the shifter, it is marked as "read" in the SA web page (and database), having timestamp and user ID recorded – making it traceable by the experts.

Shifter Assistant – Web page



home **RunControl** AAL Detectors Trigger Expert

RunControl -- 248 new

Plain view | Grouped view | **Group for ID=103229** ⊗

Hide read | Show domain | Expand all details | Hide all details | Mark all group read

Search: | **Showing 1 to 3 of 3 entries** | **Show** 10 **entries**

ID	Date	Name	Message	Action	Details	Severity	Read	Read by
103229	Thu, 16 Oct 2014 19:53:54	**partition-in-error**	The partition [initialL1CT, OLC,] is in ERROR: [CTP: app in error (APPLICATION_ERROR), OLC_RCA: app in error (APPLICATION_ERROR),]	Check messages from partition, if the ERROR state stays then call relevant experts (for initial and mirror_launcher partitions: TDAQ on call 162772, for OLC partition: 72299)	✚	ERROR	□	
30126	Wed, 10 Sep 2014 02:42:40	**partition-in-error**	The partition [initialL1CT, OLC,] is in ERROR: [CTP: app in error (APPLICATION_ERROR), OLC_RCA: app in error (APPLICATION_ERROR),]	Check messages from partition, if the ERROR state stays then call relevant experts (for initial and mirror_launcher partitions: TDAQ on call 162772, for OLC partition: 72299)	✚	ERROR	☑	Andrei Kazarov
29433	Tue, 09 Sep 2014 19:19:49	**partition-in-error**	The partition [initialL1CT, OLC,] is in ERROR: [CTP: app in error (APPLICATION_ERROR), OLC_RCA: app in error (APPLICATION_ERROR),]	Check messages from partition, if the ERROR state stays then call relevant experts (for initial and mirror_launcher partitions: TDAQ on call 162772, for OLC partition: 72299)	✚	ERROR	□	

Wed, 10 Sep 2014 14:12:16

domain

alert ID & message

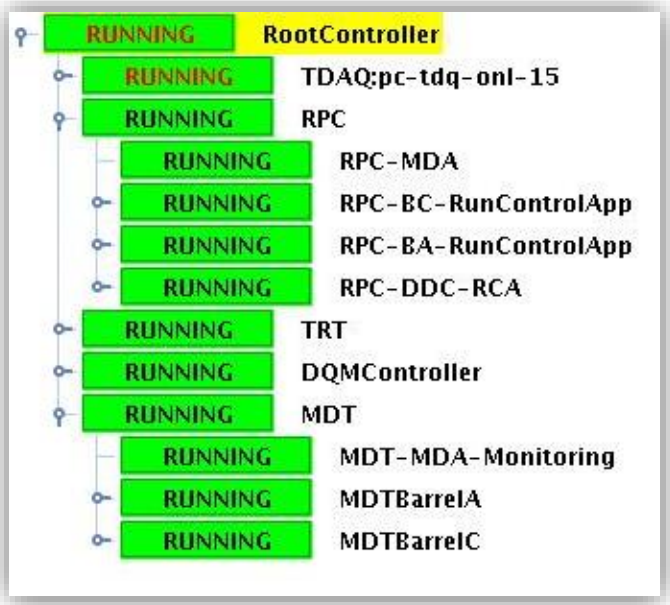
actions

“read” alert

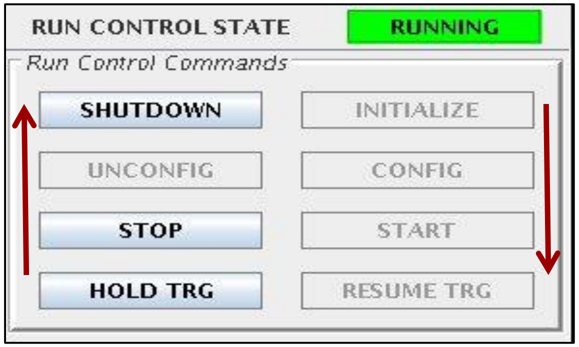
The Run Control

- The main Run Control’s task is to steer the data acquisition system in a coherent and uniform way in order to assure healthy data-taking conditions
- Hierarchical tree of applications
  - Controllers
    - Nodes “controlling” some children
    - *RootController* being the root node
  - Applications
    - Leaf nodes in the tree
- **F**inite **S**tate **M**achine (FSM) pattern
  - Reach a state suitable for data-taking operations

(More details on RunControl specific tutorial)



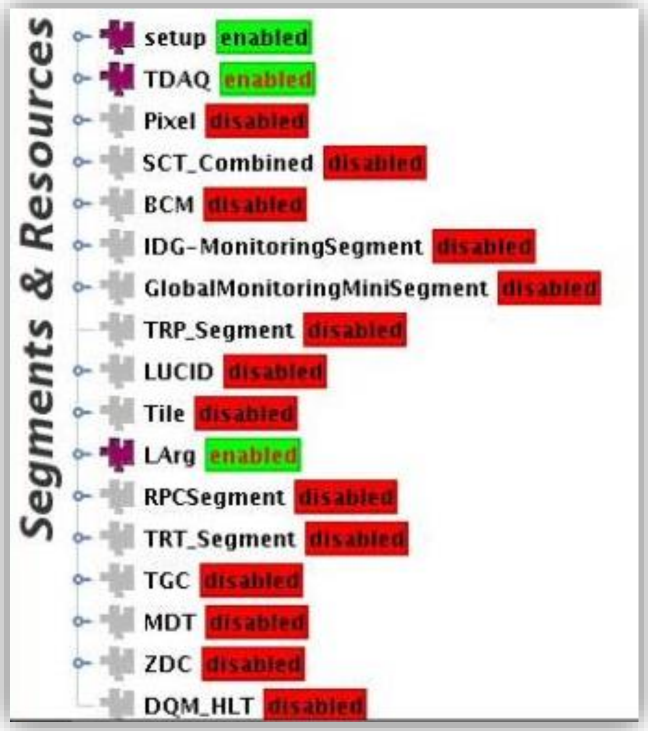
- RUNNING RootController
  - RUNNING TDAQ:pc-tdq-onl-15
  - RUNNING RPC
    - RUNNING RPC-MDA
    - RUNNING RPC-BC-RunControlApp
    - RUNNING RPC-BA-RunControlApp
    - RUNNING RPC-DDC-RCA
  - RUNNING TRT
  - RUNNING DQMController
  - RUNNING MDT
    - RUNNING MDT-MDA-Monitoring
    - RUNNING MDTBarrelA
    - RUNNING MDTBarrelC



RUN CONTROL STATE RUNNING

Changing System Configuration

- Segments in the Partition have two states: **enabled** or **disabled**
  - The corresponding detector/system **is** or **is not** not part of the data session
- The Segment's state is stored in the configuration database
- When the configuration is modified, all the applications in the system have to be notified
  - This operation is called database's **Commit & Reload**
  - Allowed only in the FSM's **NONE** state



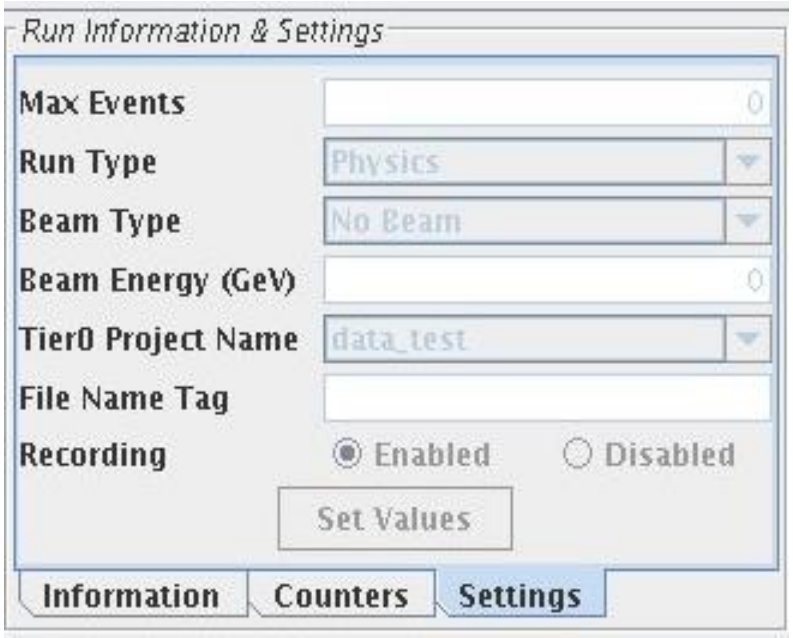
Segments & Resources

setup enabled
TDAQ enabled
Pixel disabled
SCT_Combined disabled
BCM disabled
IDG-MonitoringSegment disabled
GlobalMonitoringMiniSegment disabled
TRP_Segment disabled
LUCID disabled
Tile disabled
LArg enabled
RPCSegment disabled
TRT_Segment disabled
TGC disabled
MDT disabled
ZDC disabled
DQM_HLT disabled

What is a Run?

- A **Run** is an **uninterrupted** time period in which ATLAS is able to take data
  - All the applications are in the FSM’s **RUNNING** state
- Each Run is identified by an unique **Run Number** assigned at the start of run.
- When a Run is **stopped** (*e.g.,* no more collisions from the LHC), applications are asked to leave the RUNNING state and triggers are inhibited
  - A new transition to the RUNNING state will lead to the start of a new Run with a new Run Number.

Run Settings



Run Information & Settings

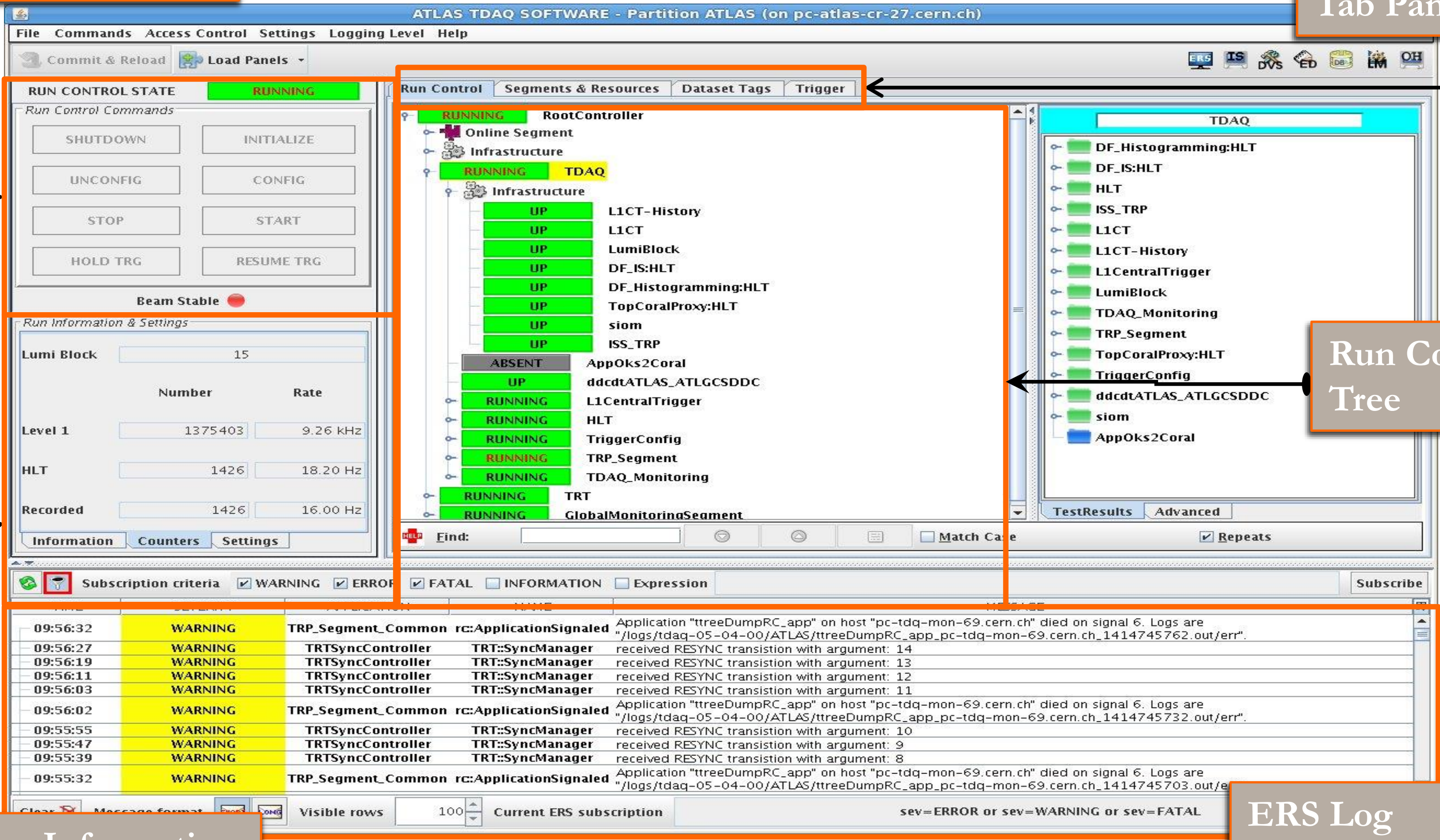
Max Events	0
Run Type	Physics
Beam Type	No Beam
Beam Energy (GeV)	0
Tier0 Project Name	data_test
File Name Tag	
Recording	● Enabled ○ Disabled

- Few parameters have to be mandatorily set before a run is started
  - Run Type
    - Usually **Physics**
  - Beam Type
    - Protons, ions
  - Recording
    - If set to **Disabled** no data will be recorded
  - Tier0 Project Name
    - Crucial for data reconstruction at T0 i.e. **data14_cos**
- They are **specific** to each single run
- Max Events = ‘X’ means run will be stopped automatically after X events. Default value (0) means that the run will continue until operator’s intervention
- Changes allowed up to the FSM’s **CONNECTED** state

IGUI tool description

Run Control Commands

Tab Panels



RUN CONTROL STATE: RUNNING

Run Control Commands

Beam Stable ●

Run Information & Settings

Lumi Block: 15

	Number	Rate
Level 1	1375403	9.26 kHz
HLT	1426	18.20 Hz
Recorded	1426	16.00 Hz

□ Match Case ☑ Repeats

Subscription criteria ☑ WARNING ☑ ERROR ☑ FATAL □ INFORMATION □ Expression

```
09:56:32  WARNING  TRP_Segment_Common  rc:ApplicationSignaled  Application "ttreeDumpRC_app" on host "pc-tdq-mon-69.cern.ch" died on signal 6. Logs are
                                                               "/logs/tdaq-05-04-00/ATLAS/ttreeDumpRC_app_pc-tdq-mon-69.cern.ch_1414745762.out/err".
09:56:27  WARNING  TRTSyncController   TRT::SyncManager        received RESYNC transistion with argument: 14
09:56:19  WARNING  TRTSyncController   TRT::SyncManager        received RESYNC transistion with argument: 13
09:56:11  WARNING  TRTSyncController   TRT::SyncManager        received RESYNC transistion with argument: 12
09:56:03  WARNING  TRTSyncController   TRT::SyncManager        received RESYNC transistion with argument: 11
09:56:02  WARNING  TRP_Segment_Common  rc:ApplicationSignaled  Application "ttreeDumpRC_app" on host "pc-tdq-mon-69.cern.ch" died on signal 6. Logs are
                                                               "/logs/tdaq-05-04-00/ATLAS/ttreeDumpRC_app_pc-tdq-mon-69.cern.ch_1414745732.out/err".
09:55:55  WARNING  TRTSyncController   TRT::SyncManager        received RESYNC transistion with argument: 10
09:55:47  WARNING  TRTSyncController   TRT::SyncManager        received RESYNC transistion with argument: 9
09:55:39  WARNING  TRTSyncController   TRT::SyncManager        received RESYNC transistion with argument: 8
09:55:32  WARNING  TRP_Segment_Common  rc:ApplicationSignaled  Application "ttreeDumpRC_app" on host "pc-tdq-mon-69.cern.ch" died on signal 6. Logs are
                                                               "/logs/tdaq-05-04-00/ATLAS/ttreeDumpRC_app_pc-tdq-mon-69.cern.ch_1414745703.out/e
```

Visible rows: 100

Current ERS subscription: sev=ERROR or sev=WARNING or sev=FATAL

Run Control Tree

ERS Log Window

Run Information & Settings

IGUI Tool

- IGUI can work in two modes: Control and Status Display (can switch between them from top menu bar “*Access Control*”)
  - At RunControl desk you should always have the IGUI in control mode (all buttons are enabled)
  - In case of errors when asking back the control for IGUI, you will get information on the user holding the expert token (look for that user or call the DAQ on-call)
- IGUI will enable only those actions (buttons) that are allowed for a given system state (don’t need to know details about the FSM). Only “*Shutdown”* button is enabled at every moment. Do not abuse using it!
- Buttons in *Run Control commands* area will send commands to the top node RootController and the commands will be distributed to its children until all tree components will reach the same state. In case of errors no other state transition is possible, until the error is fixed and the RunControl Tree is no more in error state (is green)
- Not all the IGUI panels are loaded by default. Use “*Load Panels*” top menu item to load/unload the panel needed.

Recovery Procedures

- Recovery procedures handle abnormal situations with the aim to
  - Maximize the data-taking efficiency
  - Minimize the work to be carried out by the operator
- They are orchestrated by the **C**entral **H**ints and **I**nformation **P**rocessor (**CHIP**) service, known as **Expert System** as well
- Some of them crucial in order to have a smooth data-taking
  - Stop-less removal
  - Stop-less recovery
  - TTC restart

Stop-less Removal and Recovery

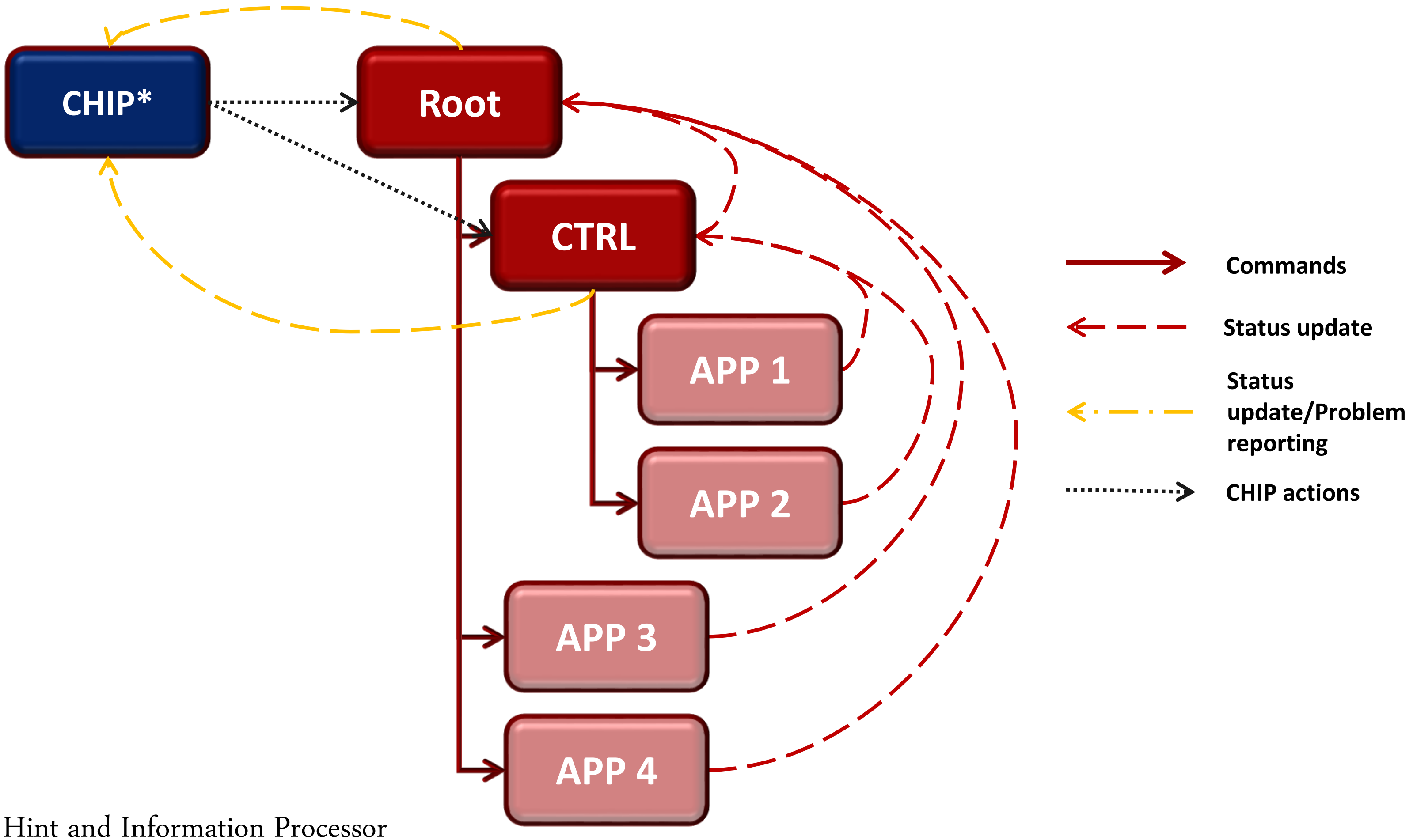
- Problems in the detector readout may cause some parts of it to be dynamically excluded from data taking during a run
  - A single busy readout channel can block the full flow of data
  - A persistently busy readout channel can be **stop-less removed**
    - It is ignored and no more acquired
  - A no more busy readout channel can be **stop-less recovered**
    - It is acquired again
- Operations are *stop-less* because they are performed without stopping the run
  - Completely automatized
  - Trigger is held for a short time period
- This kind of changes in a readout channel's status have the **lifetime of a run**

TTC restart

- The **TTC restart** recovery procedure allows to completely restart and reconfigure a part of the detector during a run
  - The run is not stopped
    - Restarting a run would imply to reconfigure the full detector
  - The trigger is kept on hold until the end of the procedure
- It is a *last resort* action
  - Even if faster than restarting the run, it can potentially introduce a large dead-time
- **Initiated by the ShiftLeader** and executed by the RunControl shifter.

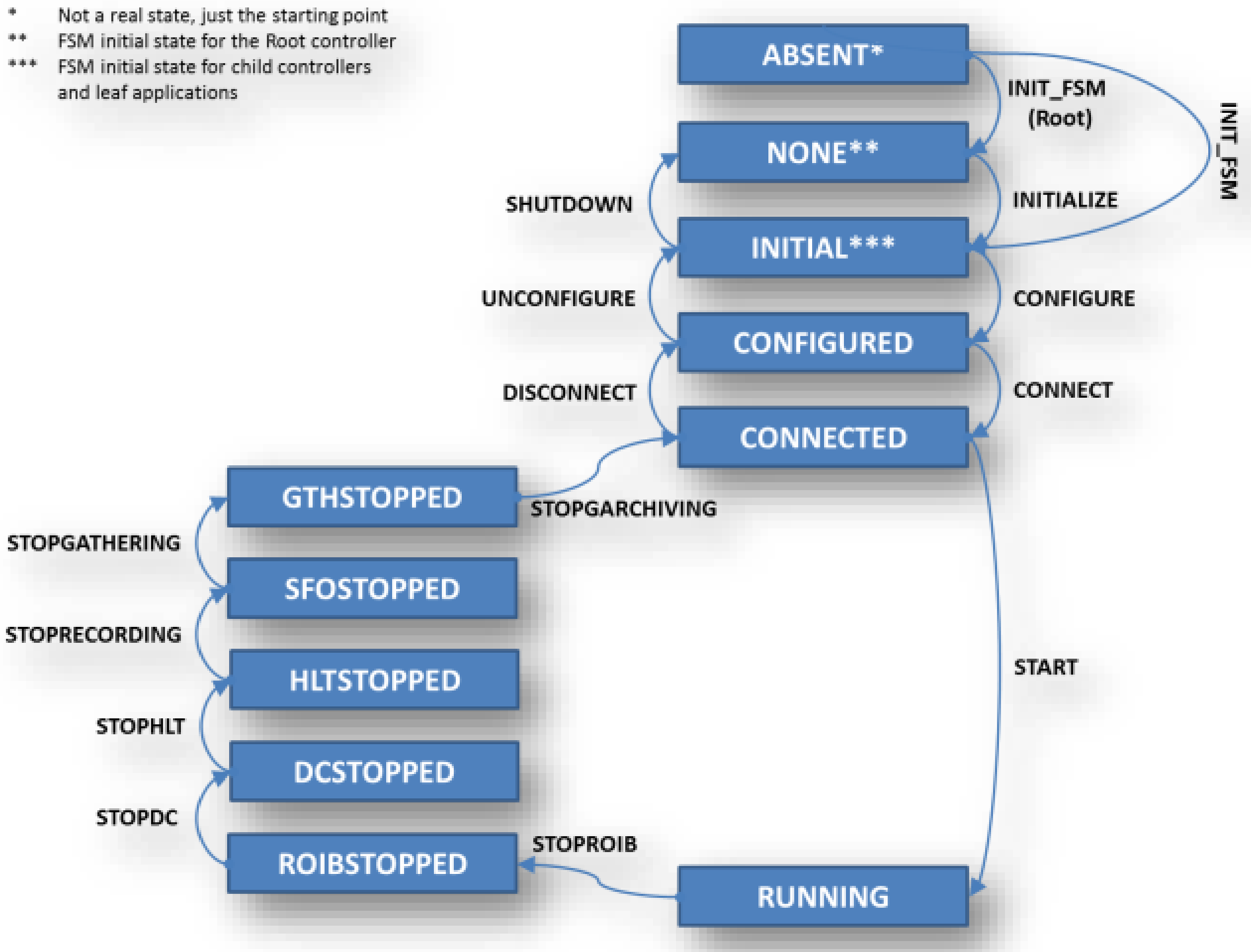
Back-up Slides

The Run Control Architecture



* Central Hint and Information Processor

The Run Control FSM



* Not a real state, just the starting point

** FSM initial state for the Root controller

*** FSM initial state for child controllers and leaf applications