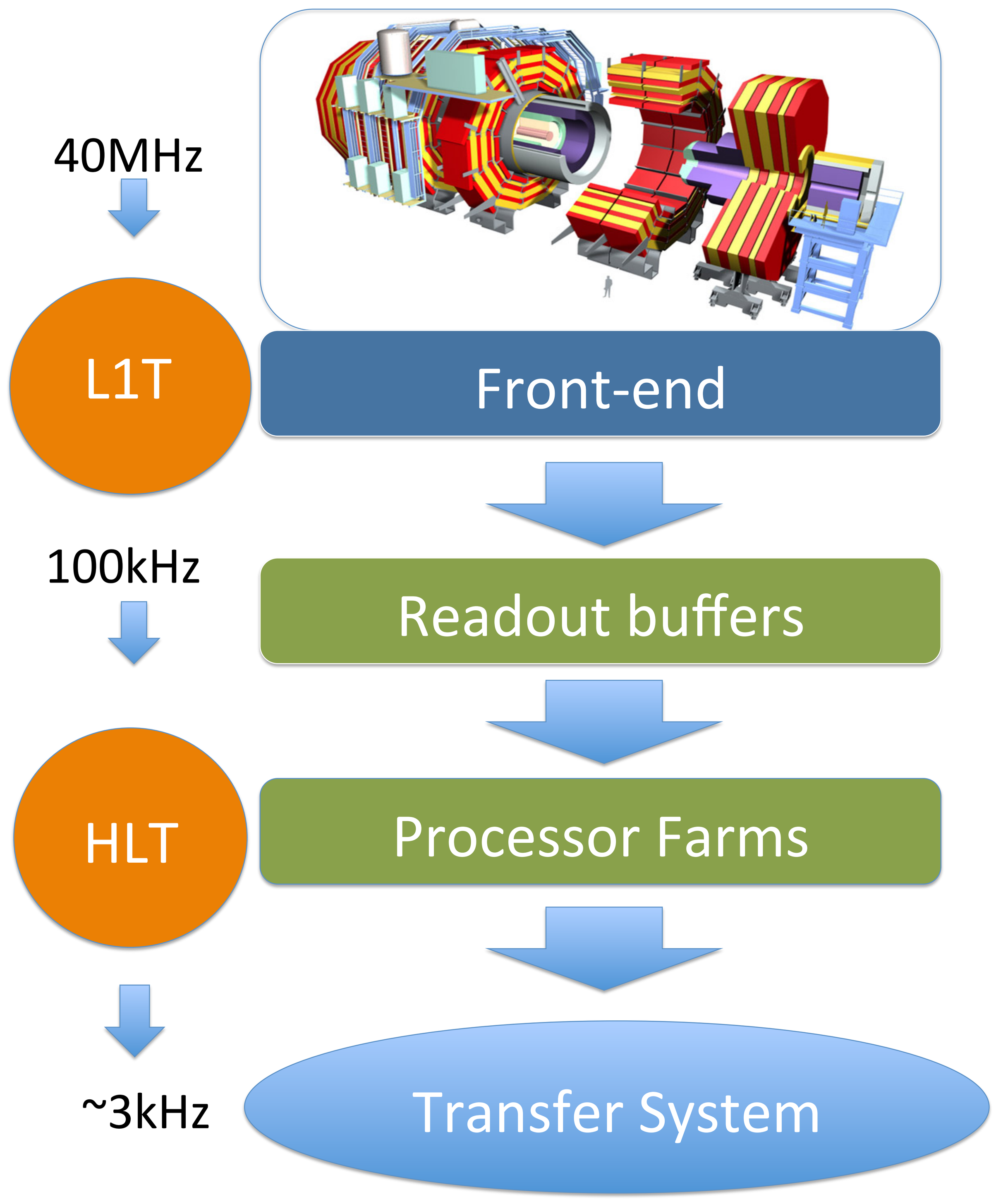
Upgrade of CMS non-event data infrastructure for the Run3 High Level Trigger

*A. Bocci, G. Franzoni, G. Govi, **R. Mommsen**, S. Morovic*

For the CMS Collaboration

The CMS Trigger System

The CMS Trigger system comprises two levels, the Level 1 Trigger (L1T) and the High Level Trigger (HLT).

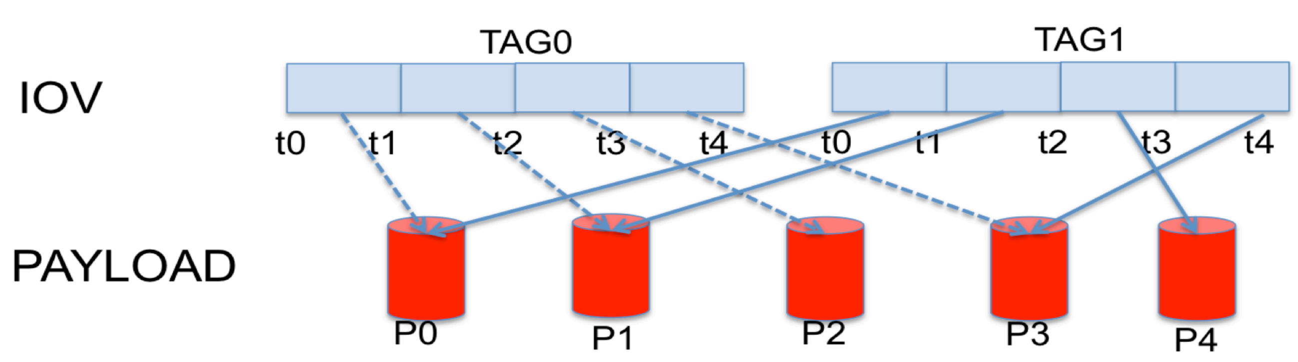


- **L1**: Hardware, Firmware (FPGAs+ASICs. Quick read-out of specific detectors: Calorimeter (Ecal, Hcal), Muon (CSC, RPC, DT).
- **HLT**: Software, running on ~1000 Nodes, 30K CPUs. Look at the full content of L1 selected events.

The Conditions Database System

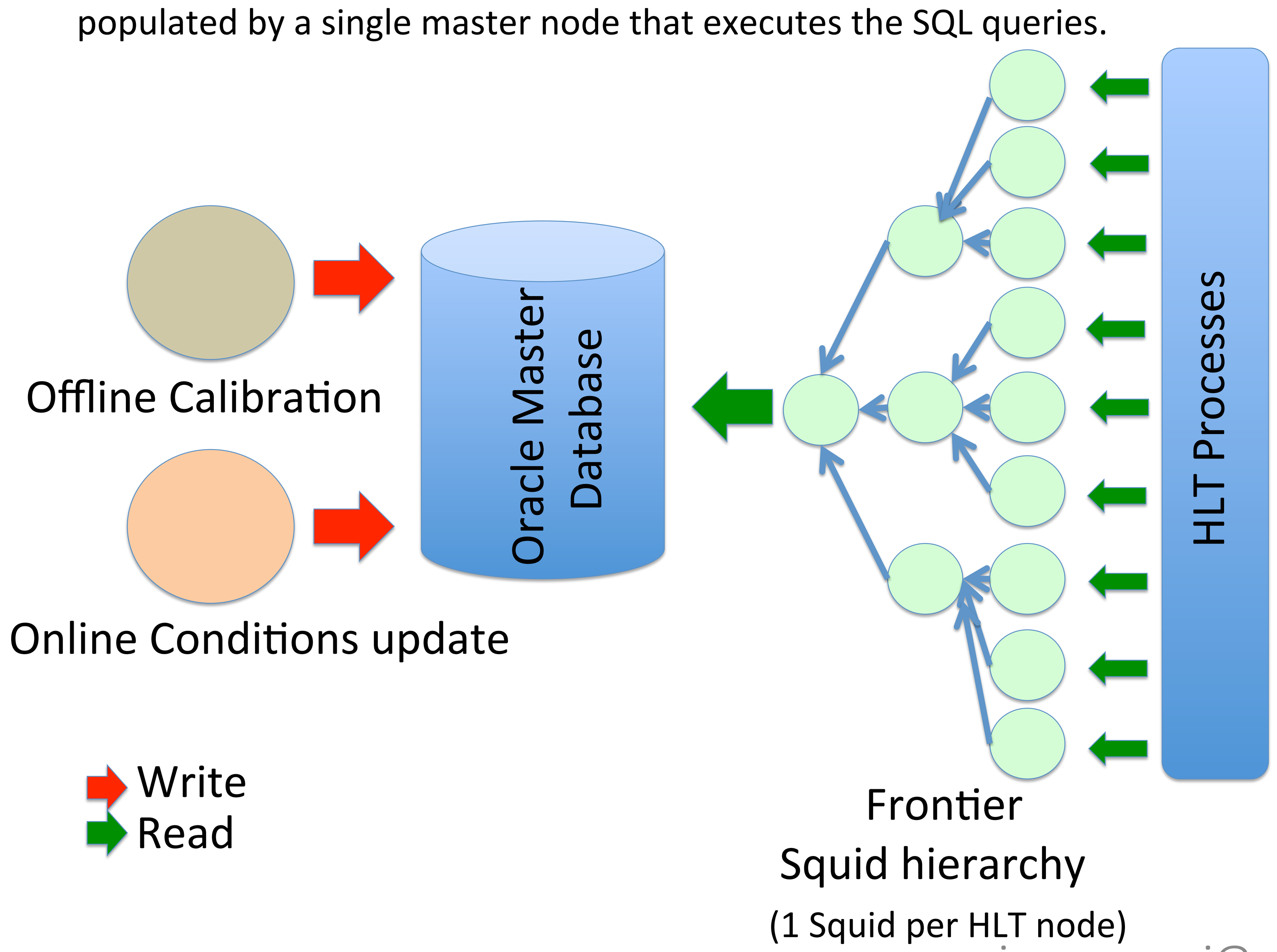
The non-event information (“Conditions”) is stored in the Conditions Database.
The data model can be described as:

- Data:
  - Payload: Values consumed as an aggregated data set.
- Metadata:
  - Interval Of Validity (IOV)
    - Time information (Run number, Luminosity block ID, Timestamp).
    - Open IOV, defined only by its starting point (“since” value).
  - Tag
    - Label identifying/categorizing a specific set of IOVs.
    - Defined for a specific payload type.
  - Global Tag
    - A consistent set of Tags specific to a given workflow.



Architecture

- Storage based on an Oracle RAC cluster sitting at the detector area.
- Offline updates serving the detector calibration – via centralized service.
- Online updates serving the Conditions critical for HLT - via automated jobs.
- Reading in HLT via Frontier: a distributed, Squid-based cache hierarchy that is populated by a single master node that executes the SQL queries.



Conditions Data and HLT until 2018

The event selection operated by the High Level Trigger consists of a simplified event reconstruction, involving the information coming from most of the detector sub-systems. This reconstruction process involves several calibration and other non-event data - generally categorized as “Conditions”. The variation of most of these parameters on the time scale of a typical data taking run (~12-18 hours) has a low impact on the selection efficiency. Therefore, until the 2018 data taking, the HLT workflow has loaded the Conditions at the run start, and it has kept them frozen for the duration of the entire run.

Updating Conditions for the HLT: the new challenge

For some specific Conditions a more precise tracking of the evolution during the data collection run could provide an improvement on the HLT efficiency, in particularly within long LHC fills:

- The LHC luminous region parameters (called “Beam Spot”), changing frequently during the fill cycle.
- The radiation-induced effects on the detectors. This includes, for instance the Calorimeter Cristal transparencies and the Calorimeter Pedestal Calibration.

For these Conditions, more updates must be made available to the HLT during the data collection run, with the update period ranging from once every lumi-section, which is a minimal time unit for Conditions data validity (spanning ~ 23 seconds), to hours.

Requirements

Latency

The goal of the updates is to make the Conditions available to the HLT jobs with a minimal latency, to guarantee that the parameters are well representing the recent state of the detector.
For the Beam Spot case, a latency of few lumi-sections, ideally one or two, is required.

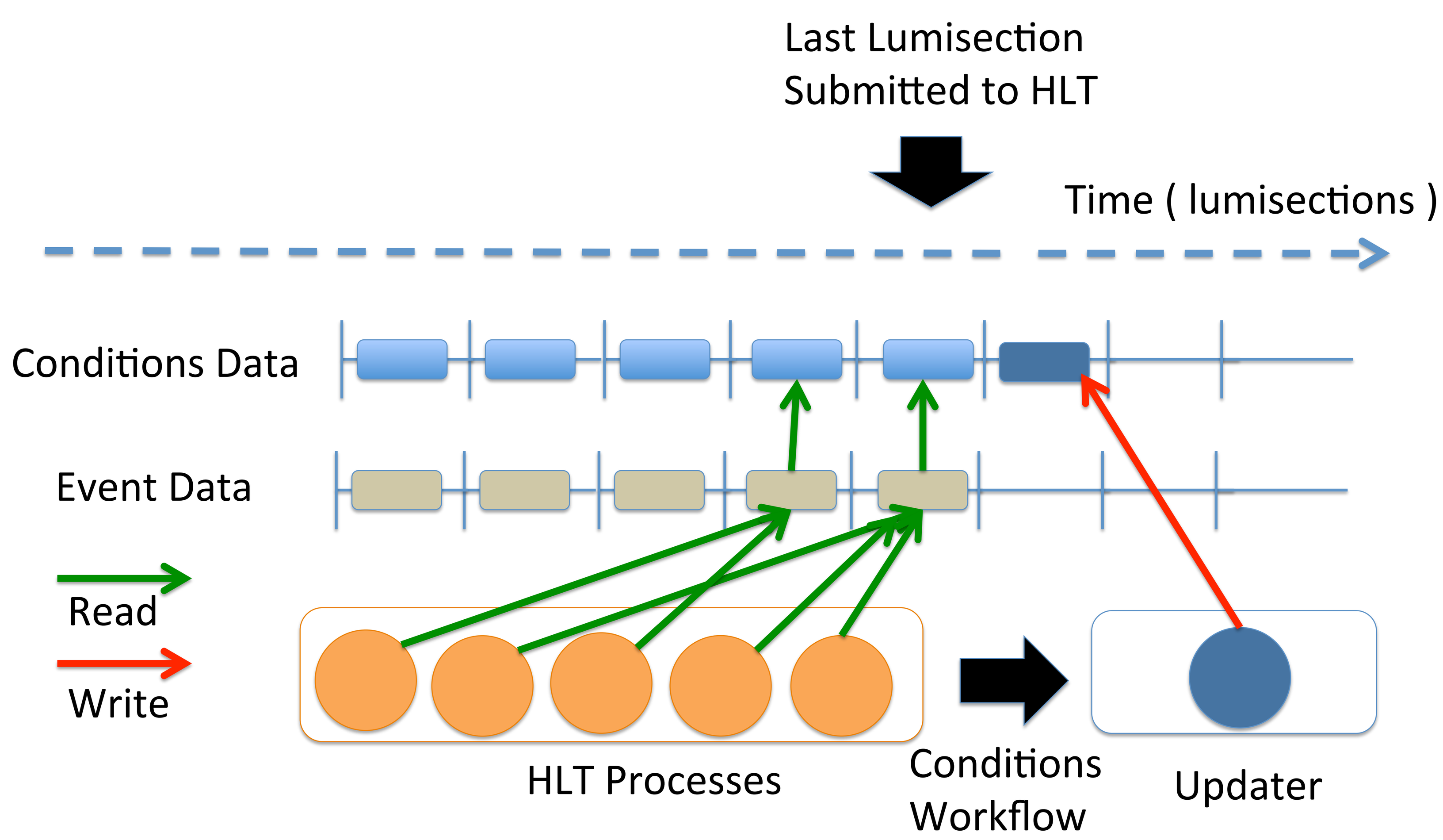
Consistency

One key factor for the efficiency of the event selection, is that all distributed HLT processes should associate the same Conditions data set to a given time window. This implies that the distribution of the cached Conditions data must be well synchronized across the HLT node hierarchy.

Reproducibility

To provide an optimal control of the HLT selection, the result of the workflow execution on a given raw data set must be entirely reproducible. This means that the Conditions associated to a given event must be guaranteed to be frozen once the original selection during the data taking is performed.

The Strategy: the updater



- **Steering process** (python based): polls the DAQ system to detect the start of a new lumi-section, and the Sub-system workflow for the availability of an update. When a match is found, the update procedure is called.
- **Update of the IOV sequence** (based on the CMSSW framework): The new payload is inserted with an validity window (IOV) starting from the next lumi-section (minimal latency). To ensure reproducibility, any update committed late with respect to the consumption time must be identified and invalidated

Outlook

A fully functional prototype has been finalized and tested with a simplified Conditions payload. The measured latency for the update query was ~200-300ms. The integration test with the latest production version of the HLT workflow has proven that the requirements of Latency, Consistency and Reproducibility have been fully met. A further validation step will be performed within a realistic global run in cosmic data taking.