



# A Web-based application for the collection, management and release of Alignment and Calibration configurations used in data processing at the Compact Muon Solenoid experiment

[Audrius Mecionis](#)<sup>1,2</sup> - [audrius.mecionis@cern.ch](mailto:audrius.mecionis@cern.ch), [Salvatore Di Guida](#)<sup>3</sup> - [salvatore.di.guida@cern.ch](mailto:salvatore.di.guida@cern.ch),  
[Giovanni Franzoni](#)<sup>1</sup>, [Marco Musich](#)<sup>4</sup>, [Gianluca Cerminara](#)<sup>1</sup>, [Andreas Pfeiffer](#)<sup>1</sup>, [Giacomo Govi](#)<sup>5</sup>  
**on behalf of the CMS collaboration**

1 - CERN, 2 - Vilnius University, 3 - Università degli Studi Guglielmo Marconi, 4 - Université catholique de Louvain, 5 - Fermi National Accelerator Laboratory

**Abstract:** The Compact Muon Solenoid (CMS) experiment makes a vast use of alignment and calibration measurements in several data processing workflows: in the High Level Trigger, in the processing of the recorded collisions and in the production of simulated events for data analysis and studies of detector upgrades. A complete alignment and calibration scenario is factored in approximately three-hundred records, which are updated independently and can have a time-dependent content, to reflect the evolution of the detector and data taking conditions. Given the complexity of the CMS condition scenarios and the large number (50) of experts who actively measure and release calibration data, in 2015 a novel web-based service has been developed to structure and streamline their management: the cmsDbBrowser. cmsDbBrowser provides an intuitive and easily accessible entry point for the navigation of existing conditions by any CMS member, for the bookkeeping of record updates and for the actual composition of complete calibration scenarios. This paper describes the design, choice of technologies and the first year of usage in production of the cmsDbBrowser.

## Conditions in the CMS experiment

The "Conditions" are non-event data, mainly consisting of detector state description, alignment and calibration constants. It is crucial for the optimal performance of reconstruction of collision events coming from simulated or real data, as well as for physics analysis.

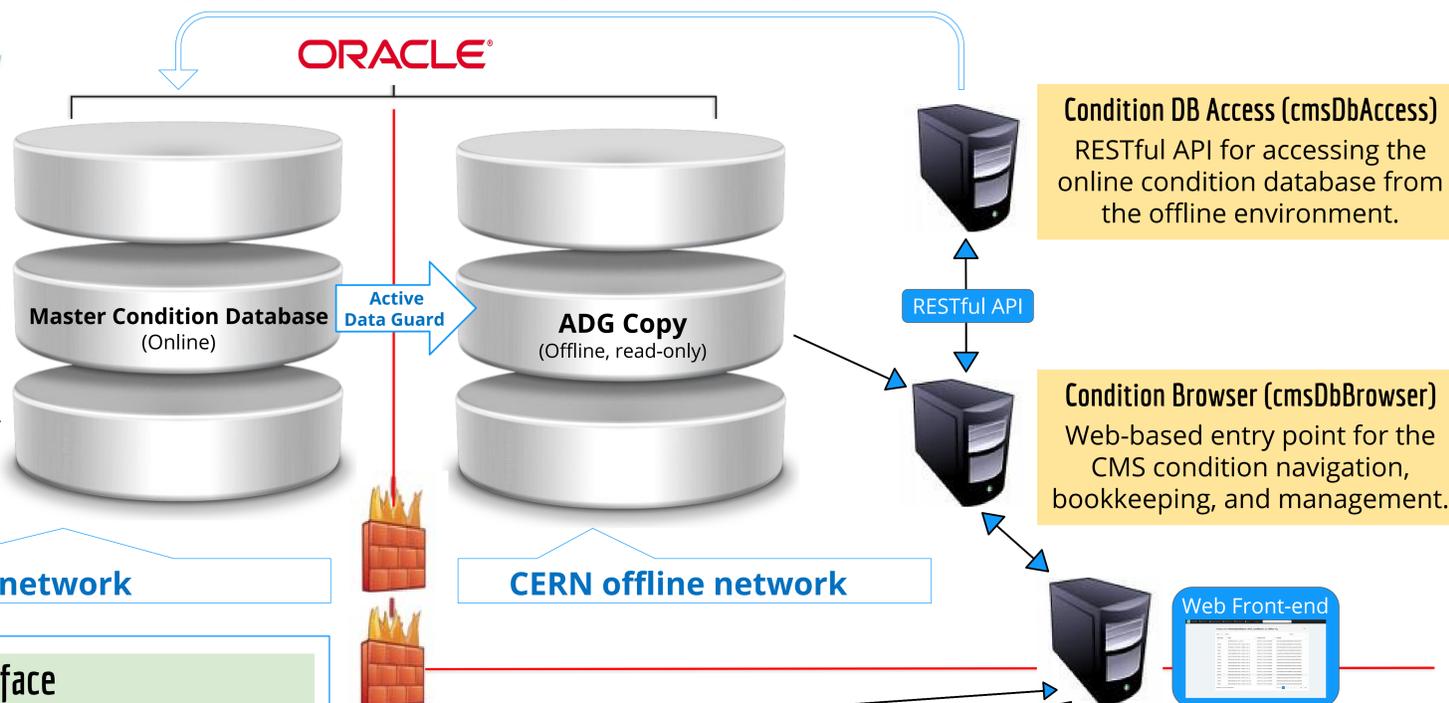


- High Level Trigger
- Online Data Quality (DQM)
- Detector Control System
- L1 Trigger
- Data Acquisition (DAQ)
- Data File Transfer

## Condition Data Model

- Payload** - holds calibration and alignments constants consumed in the various workflows of the physics data processing (e.g. reconstruction of raw data)
- Interval Of Validity (IOV)** - time interval during which a given Payload is consumed.
- Tag** - history of a given calibration or alignment content comprising a set of IOVs and their associate Payloads.
- Global Tag** - a consistent set of Tags assigned to be consumed by a given workflow (e.g. simulation, data).

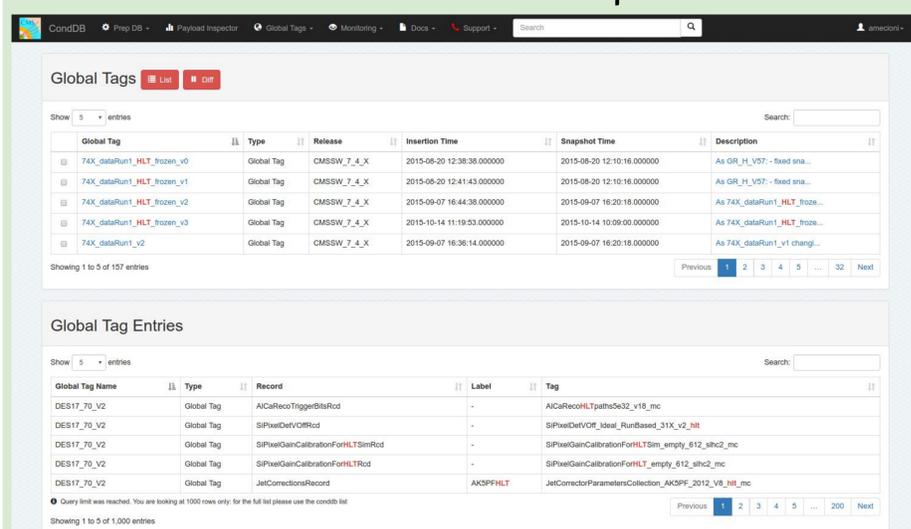
## The CMS Condition Database Architecture



CMS online protected network

CERN offline network

## cmsDbBrowser user interface



## Overview of CMS Conditions Browser

The Conditions involved in a typical production workflow are usually grouped into up to 300 Tags. This implies the existence of a large number of Tags and Global Tags in the database. A web based application called cmsDbBrowser was created in order to:

- provide an intuitive and easy way to inspect, navigate and search the existing conditions metadata by any CMS member. The result of the query is presented in a categorized and structured way and displays Tags, Global Tags and Payloads in data tables;
- manage the condition metadata by handling update requests by experts and by creating Global Tags by experts and managers.

**Physicist**

Physicists can easily navigate and browse all the conditions related data. Also it is possible to draw various plots of deserialized Payload data.

**Calibration Expert**

Calibration experts have a simple way to request a record updates for a specific production workflows as well as create a special Global Tag type called a Candidate to test their requested Payload changes in the CMSSW.

**Condition Manager**

Condition managers have the ability to accept or reject requests done by the calibration experts, manage all the condition metadata and to compose a complete calibration scenarios (Global Tags) to be consumed in a production workflows.

## Design and Implementation choices of CMS Conditions Browser

- The backend of cmsDbBrowser is implemented in Python programming language using the Flask web framework.
- As an Object Relational Mapper SQLAlchemy is used which handles all the database transactions.
- For the frontend the Bootstrap CSS framework is used together with the jQuery and Highcharts JavaScript libraries.