

THE PAYLOAD INSPECTOR: A TOOL FOR THE VISUALIZATION OF CALIBRATION AND ALIGNMENT CONSTANTS STORED IN THE CMS CONDITION DATABASE

Audrius Mecionis (Vilnius University, CERN) - audrius.mecionis@cern.ch, **Salvatore Di Guida** (Università degli Studi Guglielmo Marconi), **Giovanni Franzoni** (CERN), **Marco Musich** (Université catholique de Louvain), **Gianluca Cerminara** (CERN), **Andreas Pfeiffer** (CERN), **Giacomo Govi** (Fermi National Accelerator Laboratory) **on behalf of the CMS collaboration**

ALIGNMENT AND CALIBRATION CONSTANTS

Non-event data also known as “Conditions” describe the evolving status and performance of the several detector components of CMS. They record a given “state” of the detector, and measure alignment and calibration constants with dedicated algorithms. Condition data are crucial for an optimal simulation and reconstruction of collision events. Based on the CMS data model, they are stored onto relational databases as a set of binary objects (BLOBs), serialised using boost libraries within the CMS offline software framework.

OVERVIEW OF THE PAYLOAD INSPECTOR

The Payload Inspector allows detector experts to inspect and monitor alignment and calibration constants stored in the CMS conditions database. The tool consists of two separate layers:

- Deserialisation layer: dedicated plugins in the CMS software framework (CMSSW) load payloads from the database and extract the relevant information for rendering the plots with the right display format.
- Visualisation layer: it allows users to generate interactive historical plots for monitoring conditions. It is integrated into the cmsDbBrowser [1] web-based application, which is the main entry point accessible for all CMS members to browse and manipulate conditions data and metadata.

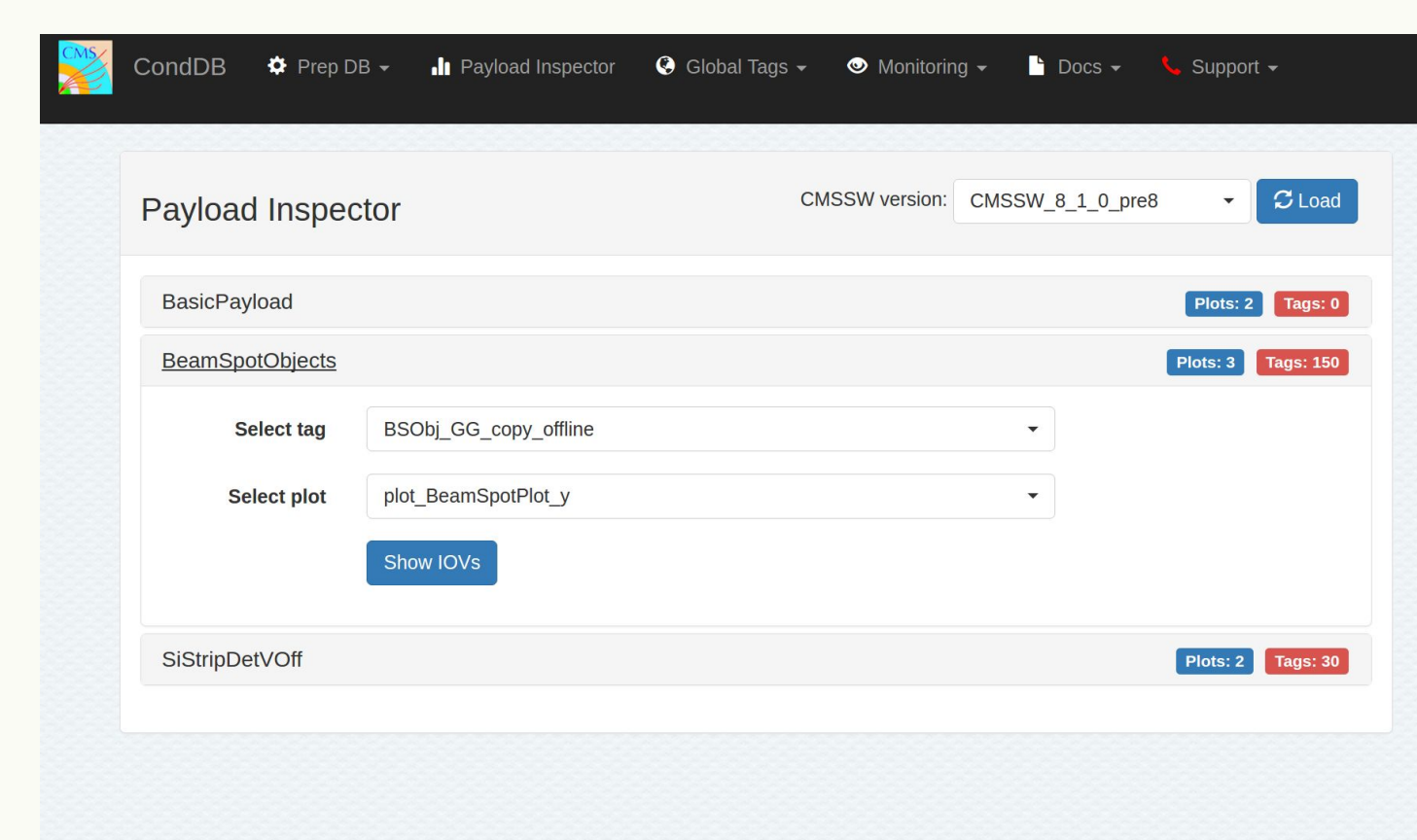


Payload Inspector user interface

Example of generated plot

[1] - A.Mecionis, S.Di Guida. 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. CHEP 2016.

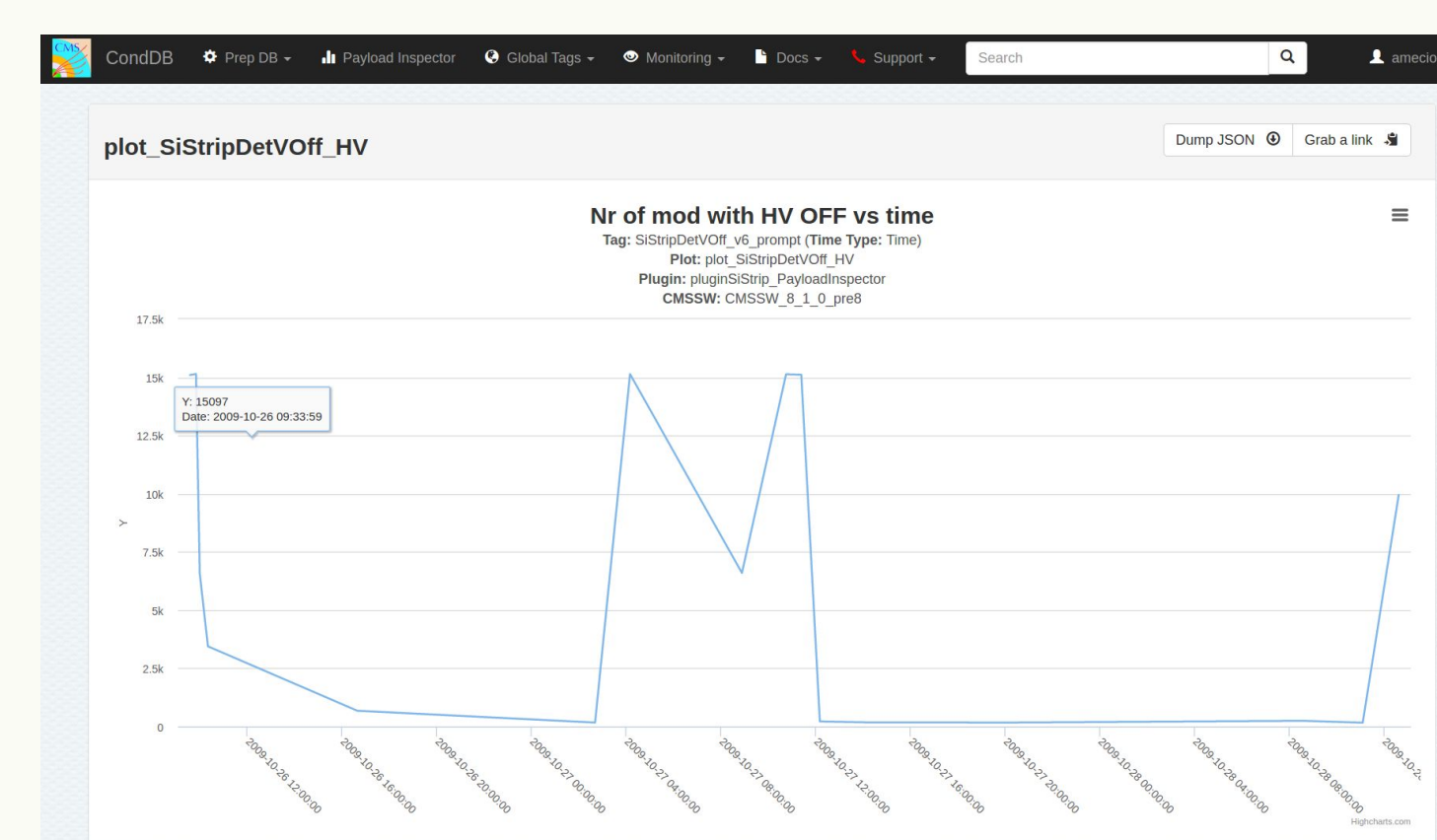
PAYLOAD INSPECTOR: HOW TO USE IT?



1. In order to use the tool a CMS user has to access the cmsDbBrowser and navigate to Payload Inspector page.
2. User have to select CMSSW version to be used to discover the plugins which consumes calibration constants and returns plot coordinates.

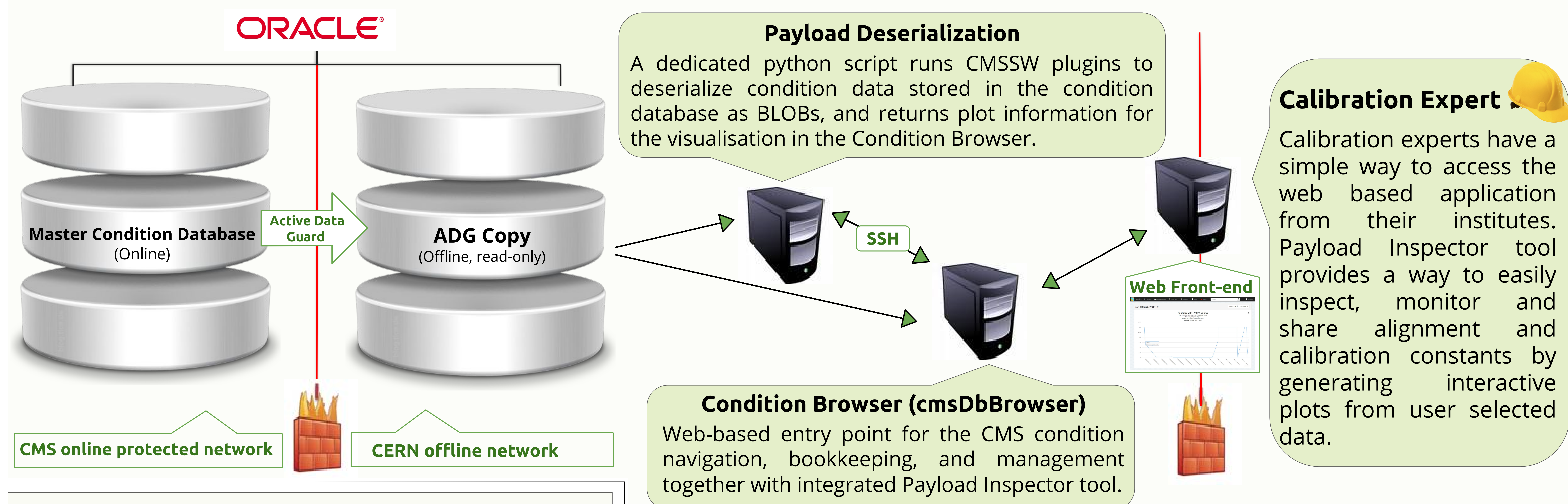
3. From a given set of condition classes, a user selects the plot type and the list of time-ordered payloads to be looked at.
4. From the list of payloads, a user marks the ones to be retrieved by the plugins running on the deserialisation layer.

Since	Insertion Time	Payload
211302	2014-04-06 06:43:37.880210	5458A475A8A04D0405130090C28520544
211303	2014-04-06 06:43:37.910223	6F2F1592A2A23B5B07766A409978920318A0
211371	2014-04-06 06:43:37.908821	4786C55615ac25c5058a9a281735ab9a480a5a
211390	2014-04-06 06:43:37.908734	8a80a170e4a0289e76c5c5f05c1011548f0a
211421	2014-04-06 06:43:37.940740	4752e2e38a0a08f98c25a5c35411a518a6f78c
211600	2014-04-06 06:43:37.960115	c6f5a232a4a00132a0c63c4b0a85802a0a
211537	2014-04-06 06:43:37.914236	5a51a4a051a1c4b08f0a0a0a0a0a0a0a0a0a
211538	2014-04-06 06:43:37.967740	0a0a38a0872f8a77c7a78a0a3c37f80a0a51a0



5. A new plot is generated from the selected payloads: it is interactive and shareable, and can be saved in pdf and png formats. A user can easily inspect and monitor selected calibration measurements and see the time evolution of the data.

PAYLOAD INSPECTOR ARCHITECTURE

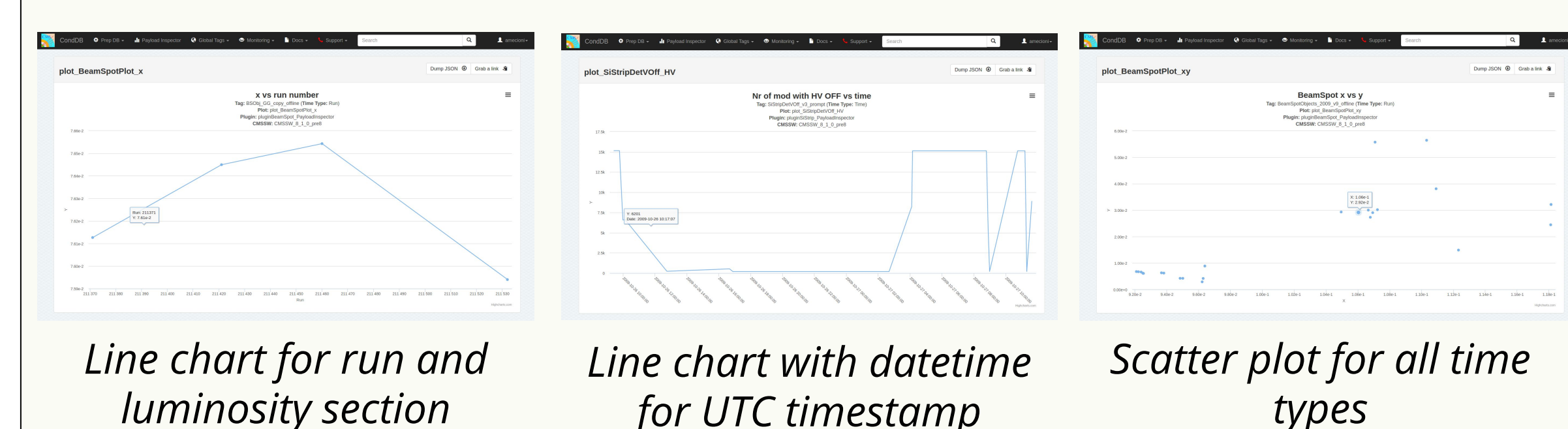


AVAILABLE PLOT TYPES

The interval of validity in which a given payload can be consumed for event processing can be expressed as:

- Run
- Luminosity section (23 s intervals)
- UTC timestamp

The representation of the condition data depends on the corresponding time validity:



DESIGN AND IMPLEMENTATION CHOICES OF THE PAYLOAD INSPECTOR

- The CMS software framework (CMSSW) plugins are developed by detector experts in C++.
- The Paramiko module, which provides a Python interface to SSH protocol, manages the communication between the two servers. JSON is used as a data-interchange format.
- The cmsDbBrowser application backend is developed in Python using the Flask web framework. The visualisation layer of the Payload Inspector is fully integrated into it.
- The Bootstrap CSS framework, together with the jQuery and Highcharts JavaScript libraries, is used for the frontend.

