Tile-in-One: an integrated system for data quality and condition assessment for the ATLAS Tile Calorimeter

Yuri Smirnov[1,*] and *Juraj* Smieško[2,**]

[1]Northern Illinois University, USA
[2]Slovak Academy of Sciences, SK

Abstract. The Tile Calorimeter (TileCal) is a crucial part of the ATLAS detector, which jointly with other calorimeters reconstructs hadrons, jets, tau-particles, missing transverse energy and assists in muon identification. It consists of alternating steel absorber layers and active scintillating tiles and covers the region $|\eta| < 1.7$. The TileCal is regularly monitored by several systems, which were developed mainly during the commissioning of the detector in order to meet distinct requirements. Any problem is reported and immediately investigated, which results in data quality efficiency very close to 100%, as achieved over the last few years. Although the TileCal tools are maintained, the underlying technologies are becoming gradually outdated.

The Tile-in-One web platform strives to integrate all data quality and condition assessment TileCal tools into one common system. This system is implemented as a web application with the main machine being the gateway for so-called plugins. These are standalone small web application hosted on a single virtual machine. The plugins are separated into virtual machines due to the requirement of different data sources and to avoid interference in order to increase the stability of the platform. The main server is responsible for the authentication and authorization of the users, as well as the management of the plugins. Currently the platform consists of 13 plugins in various stages of development. The implementation details of the Tile-in-One web system and selected plugins are presented in this document.

1 ATLAS Tile Calorimeter

The ATLAS (A Toroidal LHC ApparatuS) detector [1] is one of the large general-purpose detectors at the Large Hadron Collider (LHC) [2]. The LHC started its operation in 2008 and since then it has gone through several gradual upgrades of the beam energy and luminosity. Last year marked the end of the second round of operation at the LHC, called run II, and the total recorded luminosity in pp collisions at a center-of-mass energy of 13 TeV since 2015 is 156 fb^{-1} [3].

*e-mail: iouri.smirnov@cern.ch
**e-mail: juraj.smiesko@cern.ch

The main purpose of the ATLAS detector is to investigate a wide range of physics, among them the study of the Higgs boson and other Standard Model phenomena, as well as searches for extra dimensions and particles that could make up the dark matter. The detector is a 44 m long barrel with a diameter of 25 m, where sub-detectors and magnets are organized in cylindrical layers around the beam pipe. One of the sub-detectors is the Tile Calorimeter (TileCal) [4], which is part of the ATLAS Calorimeter System, see Figure 1.

The TileCal constitutes the outermost layer of the ATLAS Calorimeter System. It detects hadrons, jets and taus, while also contributing to the jet energy and missing transverse energy reconstruction, as well as assisting the spectrometer in the identification and reconstruction of muons. The calorimeter central (barrel) part covers the pseudorapidity region $|\eta| < 1.0$, and its two extended barrels cover the range $0.8 < |\eta| < 1.7$. The TileCal is a sampling calorimeter using plastic scintillating tiles as the active medium and steel plates as the absorber. The total number of calorimeter cells is 5182, while the number of channels is around 10000, as most cells are read by two photomultiplier tubes.

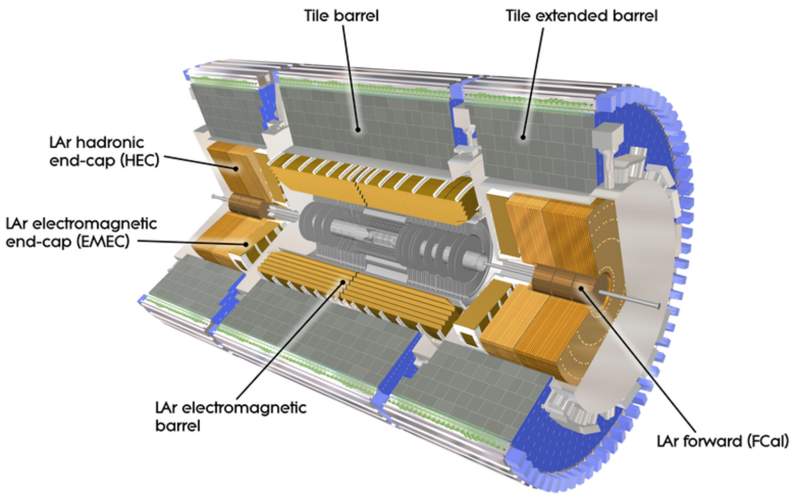

Tile barrel
Tile extended barrel
LAr hadronic end-cap (HEC)
LAr electromagnetic end-cap (EMEC)
LAr electromagnetic barrel
LAr forward (FCal)

Figure 1. The ATLAS Calorimeter System with the TileCal shown in gray.

2 Tile Calorimeter Software

Over the years, several web tools were developed mainly during the commissioning of the TileCal and first LHC run. Since the development was done by different groups to support different TileCal data monitoring and maintenance activities, those tools make use of distinctive technologies and the data sources require different forms of data recovery. Usually, collaborators have to browse among several tools (web pages) in order to perform a given task. Additionally, the documentation is not well consolidated. In the end, the use, maintenance and enhancement of existing functionalities become time consuming and a costly work.

The Tile-in-One (TiO) web platform aims to integrate different TileCal web tools into one common platform, which will share the computing infrastructure and access to common data and services inside or outside the TileCal Collaboration, e.g. access to different databases, user authentication, commonly used libraries. This is done to not repeat functionality and to encourage collaborators to integrate their tools into the TiO.

3 Platform Architecture

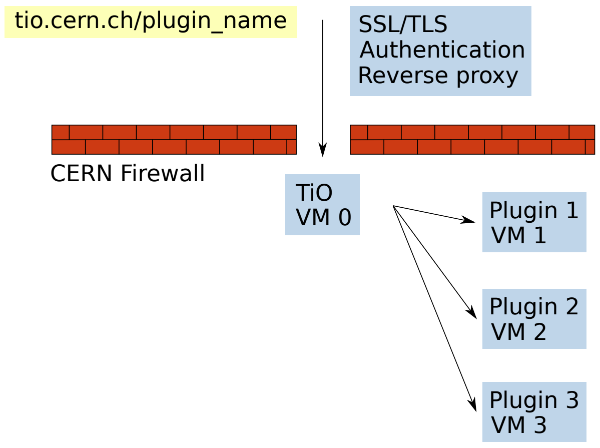


Figure 2. The architecture of the TiO web platform.

The TiO web platform is designed with flexibility and ease of maintenance in mind. The architecture of the platform is shown in Figure 2. Flexibility is a key requirement, because there is a large collection of data sources to be integrated, and the ease of maintenance is required by the fact that the platform should be mainly developed by students, who spend only about a year developing a specific feature and then leave the project.

The design of the TiO platform is based around one main server, which is in charge of the secure connection to the platform, authentication of users and routing of the user requests and their responses. Behind this server, machines which host small web applications, called plugins, receive the user's requests, process them and return the outcome back to the user. The plugins usually implement only one self contained Data Quality (DQ) tool. The key elements of the TiO design are the following:

- The main server is just a bridge between plugins and users.
- Every plugin is an independent web application, which implements one DQ tool.
- Plugins are isolated from each other and run in separate Virtual Machines (VMs).
- Plugins can be based on a template provided by the platform developers.
- The source code is version controlled with Git.
- There is a person or a group responsible for the maintenance of the plugin.

In order not to interrupt the operation of the main server and also to expose unfinished plugins outside of the CERN network the platform developers run a copy of the main server inside of CERN's firewall to which all unfinished plugins are connected. Once the plugin development is finished the connection is transferred from this development server to the main server.

Since the majority of the plugin developers are not skilled in web development the platform developers provide several templates i.e. fully functional example plugins, which can be used as a starting point in the plugin development. The plugin templates are running as any other plugin and are regularly updated. So far the platform provides two templates:

- Simple Static Plugin Template:
  - Dynamic elements on the user side.
  - Cron jobs run on the server to update data files.
  - Implemented with HTML, CSS, JavaScript.
- Simple Dynamic Plugin Template:
  - All of the Static Plugin capabilities.
  - Parameters are processed on the server side.
  - Implemented with Python and Bottle.

An example screenshot of the Simple Dynamic Plugin Template is shown in Figure 3. The users can select the range of data-taking or calibration runs, which will be fetched from the plugin server and then they can further manipulate the interactive plot that was received.

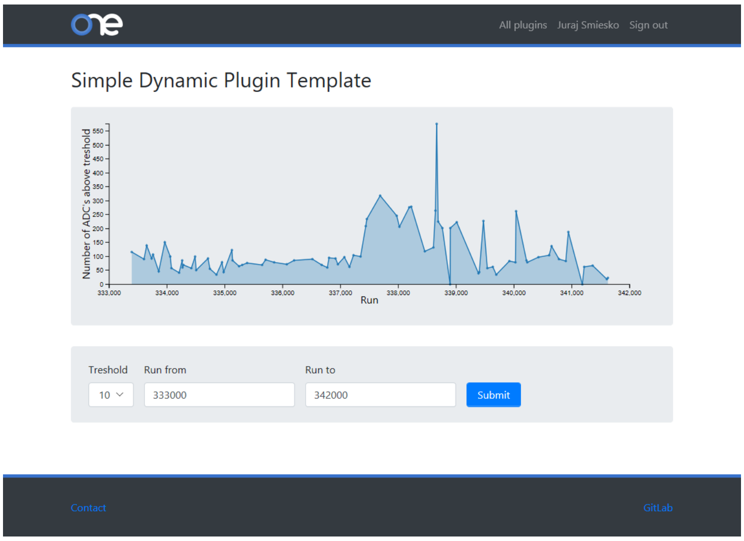
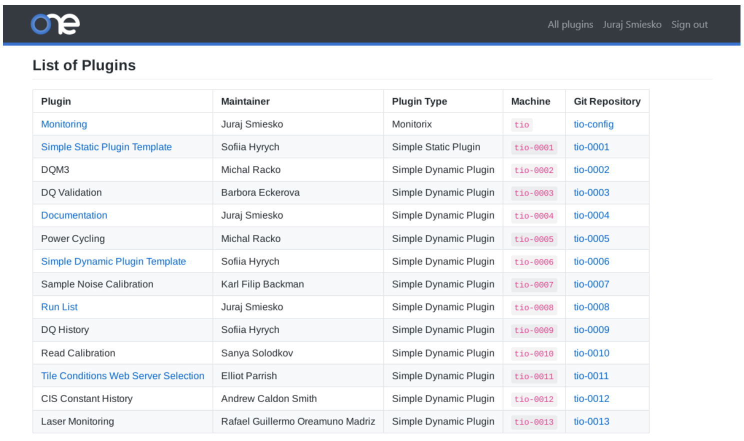


Figure 3. A screenshot of the Simple Dynamic Plugin Template.

4 Notable Plugins

Apart from ordinary plugins, there are several plugins that complement the TiO platform itself. One of these plugins hosts the platform documentation, which is based on a Simple Dynamic Plugin Template and allows users to write documentation in Markdown [6]. Its screenshot in Figure 4 shows all currently maintained/developed plugins. Another plugin, which complements the platform, is used for monitoring the VMs on which other plugins and the main server run. This plugin is based on the Monitorix [7] tool, and provides detailed information on the health of the VMs and the network.

Because the TiO platform is web based it allows for relatively easy creation of interactive plots/lists. Among plugins developed so far is the so-called Run List, which shows information about runs taken in the last 30 days. The plugin which shows how many times there was a power cycle over a defined period of time is named Power Cycling. Then, there are plugins



List of Plugins

Plugin	Maintainer	Plugin Type	Machine	Git Repository
Monitoring	Juraj Smiesko	Monitorix	tio	tio-config
Simple Static Plugin Template	Sofiia Hyrych	Simple Static Plugin	tio-0001	tio-0001
DQM3	Michal Racko	Simple Dynamic Plugin	tio-0002	tio-0002
DQ Validation	Barbora Eckerova	Simple Dynamic Plugin	tio-0003	tio-0003
Documentation	Juraj Smiesko	Simple Dynamic Plugin	tio-0004	tio-0004
Power Cycling	Michal Racko	Simple Dynamic Plugin	tio-0005	tio-0005
Simple Dynamic Plugin Template	Sofiia Hyrych	Simple Dynamic Plugin	tio-0006	tio-0006
Sample Noise Calibration	Karl Filip Backman	Simple Dynamic Plugin	tio-0007	tio-0007
Run List	Juraj Smiesko	Simple Dynamic Plugin	tio-0008	tio-0008
DQ History	Sofiia Hyrych	Simple Dynamic Plugin	tio-0009	tio-0009
Read Calibration	Sanya Solodkov	Simple Dynamic Plugin	tio-0010	tio-0010
Tile Conditions Web Server Selection	Elliot Parrish	Simple Dynamic Plugin	tio-0011	tio-0011
CIS Constant History	Andrew Caldon Smith	Simple Dynamic Plugin	tio-0012	tio-0012
Laser Monitoring	Rafael Guillermo Oreamuno Madriz	Simple Dynamic Plugin	tio-0013	tio-0013

Figure 4. A screenshot of the documentation plugin, which shows all TiO plugins.

which show drifts in the Laser and Charge Injection System calibration constants over a set period of time (plugins Laser Monitoring, shown in Figure 5, and CIS Constant History [5]). Finally, a plugin will serve as a replacement for the current system which compares data quality histograms from two different runs, called DQ Validation.

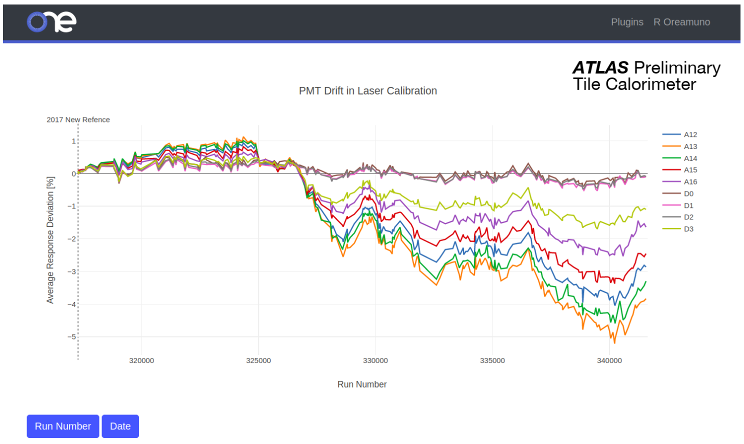


Figure 5. A screenshot of the TiO Laser Monitoring plugin. The plugin shows drifts in the Laser calibration constants over a set period of time for selected cells of Tile Calorimeter [5].

5 Tools Employed

There are several requirements on the technologies/software used to build the TiO web platform, where by far the largest hurdle is to connect two different worlds. On one hand there is the world of the detector, which requires robust and stable software, and on the other hand there is the ever-changing world of the web.

In the end the key requirements are that the technology/software should be widely adopted, stable over a long period of time, open source and easily replaceable. The technologies employed in the current implementation of the platform are summarized in the Table 1.

Table 1. Rundown of technologies employed by the TiO web platform.

Service/Feature	Technology/Implementation
Secure connection	CERN CA [8]
Reverse proxy	Nginx [9]
Authentication	CERN OAuth2 & oauth2_proxy [10]
User management	oauth2_proxy & TiO plugin
Source code hosting	CERN GitLab [11]
Plugin templates	Static sites, Python & Bottle [12]
Monitoring	Monitorix [7]
Virtual Machine	CERN OpenStack [13]

6 Conclusion

A common web space for the ATLAS Tile Calorimeter offline data quality tools is presented. The platform developers aim at a resilient and easy to extend system, which does not require day to day attention. The platform should provide easy transfer of knowledge between collaboration members about the internal structure of the platform, quicker and easier development of the new plugins, and also easier integration of the currently working ATLAS Tile Calorimeter web tools into the Tile-in-One platform.

References

[1] ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, JINST **3**, S08003 (2008)

[2] L. Evans and P. Bryant (Editors), *LHC Machine*, JINST **3**, S08001 (2008)

[3] ATLAS Collaboration, *Atlas luminosity public plots*, https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LuminosityPublicResultsRun2 [last accessed 2019-10-03]

[4] ATLAS Collaboration, *ATLAS tile calorimeter: Technical design report*, CERN/LHCC/96-42 (1996), https://cds.cern.ch/record/331062

[5] M. Marjanovic [ATLAS], *ATLAS Tile calorimeter calibration and monitoring systems*, IEEE Trans. Nucl. Sci. **66** (2019) no.7, 1228-1235, http://cds.cern.ch/record/2629424/

[6] J. Gruber and A. Swartz, *Markdown* (2004), https://daringfireball.net/projects/markdown/ [last accessed 2019-09-30]

[7] J. Sanfeliu, *Monitorix* (2018), https://www.monitorix.org/ [last accessed 2018-10-08]

[8] Cern certification authority, https://ca.cern.ch/ca/ [last accessed 2018-10-08]

[9] Nginx, Inc., *Nginx* (2004), https://nginx.org [last accessed 2020-01-30]

[10] Bitly, Inc., *oauth2_proxy* (2014), https://github.com/bitly/oauth2_proxy/ [last accessed 2020-01-30]

[11] GitLab Inc., *GitLab* (2018), https://about.gitlab.com/ [last accessed 2018-10-08]

[12] M. Hellkamp, *Bottle* (2013), https://bottlepy.org/docs/dev/ [last accessed 2018-10-08]

[13] The OpenStack Foundation, *Openstack*, https://www.openstack.org/ [last accessed 2018-10-08]