

# An integrated system for data quality and conditions assessment for the ATLAS Tile Calorimeter

## Tile-in-One

Juraj Smiesko<sup>1,\*</sup>, on behalf of the ATLAS Tile Calorimeter System

<sup>1</sup>Comenius University in Bratislava

**Abstract.** An integrated system for data quality and conditions assessment for the ATLAS Tile Calorimeter is known amongst the ATLAS Tile Calorimeter as the Tile-in-One. It is a platform for combining all of the ATLAS Tile Calorimeter offline data quality tools in one unified web interface. It achieves this by using simple main web server to serve as central hub and group of small web applications called plugins, which provide the data quality assessment tools. Every plugin runs in its own virtual machine in order to prevent interference between the plugins and also to increase stability of the platform.

## 1 ATLAS Tile Calorimeter

ATLAS [1] is one of the two large general-purpose detectors at the Large Hadron Collider (LHC) at CERN. The LHC started its operation in 2008 and since then it has gone through several gradual upgrades of beam energy and luminosity. Currently, in 2018, it provides collisions at center of mass energy of 13 TeV with instantaneous luminosity reaching  $2.06 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$  [2].

The ATLAS detector investigates a wide range of physics, from the Higgs boson and other Standard Model studies to the searches of extra dimensions and particles that could make up 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. Moving away from the interaction point, the sub-systems are arranged in the order: Inner Detector, Solenoid Magnet, Electromagnetic Calorimeter, Hadronic Calorimeter, Toroid Magnet and Muon Spectrometer. Apart from the barrel, there are also two additional wheel shaped end-caps belonging to the Muon spectrometer. These end-caps are perpendicular to the beam pipe and they are situated 6 m from each side of the barrel. The ATLAS detector also weighs 7000 tons and is situated in a cavern 100 m underground. The computer generated image of the detector could be seen in Figure 1.

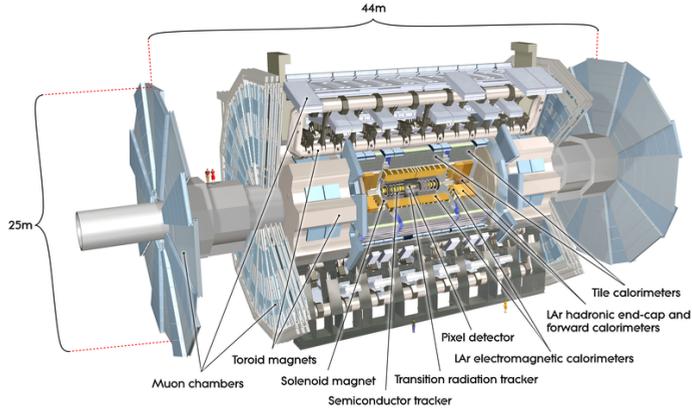
The Tile Calorimeter (TileCal) [3], shown in Figure 2, is located in the central section of the ATLAS hadronic calorimeter. It detects hadrons, jets and taus, while also contributing to the jet energy and missing  $E_T$  reconstruction, as well as assisting the spectrometer in the identification and reconstruction of muons.

The TileCal is a sampling calorimeter using plastic scintillating tiles as the active medium and steel plates as the absorber. It covers the pseudorapidity<sup>1</sup> range up to  $|\eta| < 1.7$  with one

---

\*e-mail: [juraj.smiesko@cern.ch](mailto:juraj.smiesko@cern.ch)

<sup>1</sup>Pseudorapidity is defined as  $\eta = -\ln \tan(\theta/2)$ , where  $\theta$  is the angle from the beam axis.



**Figure 1.** ATLAS detector, the figure shows placement of TileCal within the detector. The TileCal is shown in gray color.

central Long Barrel (LB) and two Extended Barrels (EB). Each barrel is segmented in  $\phi$  in 64 modules. Radially each module is divided into three layers which are approximately 1.5, 4.1 and 1.8  $\lambda$  (nuclear interaction length for protons) thick in case of LB and 1.5, 2.6 and 3.3  $\lambda$  in case of EB. The total number of cells is 5182, while the number of channels is  $\sim 10000$ , as most cells are read by two photomultipliers (PMTs).

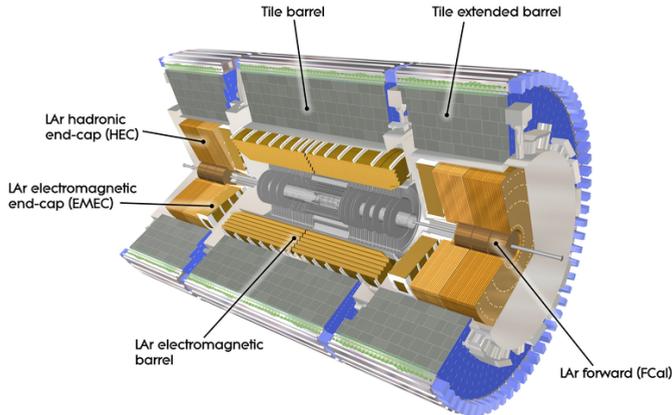
The scintillators are read out by wavelength shifting fibers and photomultiplier tubes. The analogue signals from the PMTs are digitized by sampling the PMT signals every 25 ns. For each interesting event selected by the ATLAS trigger, the Tile Calorimeter reads out seven consecutive samples, thus covering a time window of 175 ns. The seven consecutive samples are combined by the electronics in a remote counting room, at a rate of up to 100 kHz, to compute the pulse amplitudes and their position in time. These quantities are required for energy reconstruction.

## 2 TileCal Software

To support basic TileCal data monitoring and maintenance activities several tools were developed during commissioning and maintained by different groups. However, these tools made use of distinctive technologies, data sources that require different forms of data recovery and documentation was not well consolidated. Most of these tools were developed independently, without following the same guidelines, sometimes lacking basic software infrastructure and implemented without a global perspective of the calorimeter. Therefore, in many cases, they overlap in the objectives and resources with each other. Besides, in order to perform a single task, the collaborator must navigate through different tools, databases and software, which is time consuming and susceptible to mistakes.

Thus, the use, maintenance and enhancement of existing functionalities becomes time consuming and costly work. Tile-in-One (TiO) aims to integrate different Tile tools, sharing the same infrastructure and accessing common services, such as: access to different databases, user authentication, commonly used libraries, and software infrastructure to allow collaborators to integrate their own tools.

It is therefore necessary to have an infrastructure which allows the implementation of any functionality without having to duplicate the effort while being possible to integrate with an



**Figure 2.** The ATLAS Tile Calorimeter together with sub-detectors it encloses.

overall view of the detector status. The TiO web platform consists of a unique development standard and documentation for web systems and tools used by the TileCal collaboration. It aims to provide information on TileCal features without demanding from the user knowledge where and how to get the data needed. The TiO platform is a visual display of the most important information needed to achieve one or more collaborator's objectives, consolidated and arranged on a single screen so the information can be monitored at a glance.

### 3 Current Approach

The history of the current Tile-in-One spans several years and shows signs typical of projects with lack of manpower. In addition, the design expected a much larger number of plugins and users to operate with. We acknowledge that, the approach could offer advantages in organizing of the large amount of data, which TileCal data quality works with.

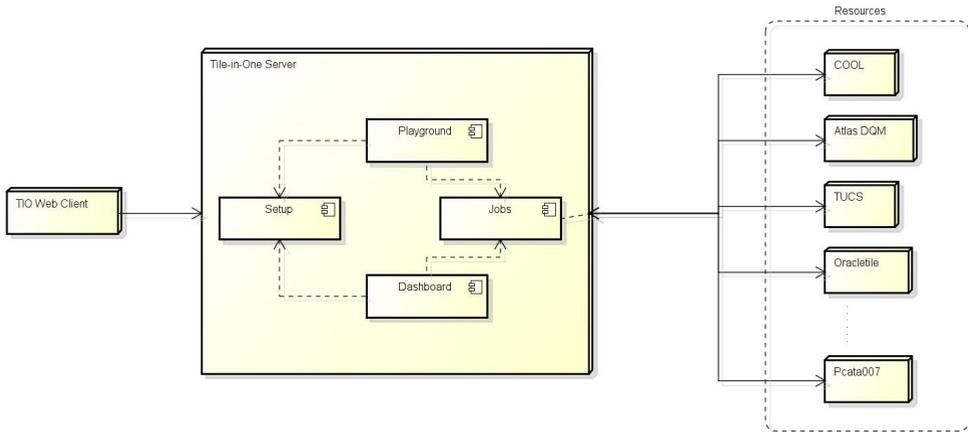
The operation of the currently used Tile-in-One could be described as follows. The infrastructure of TiO is composed of three different nodes:

- The TiO Web Client, a user web based application running on an ordinary workstation.
- The TiO Server, which supplies broker functionality between the client demands and the services availability.
- The Resources which provide services from third-party servers (TileCal and ATLAS servers outside TiO).

Between the nodes, two types of objects are passed: jobs and plugins. The job wraps a request made by the client (this is when a resource is accessed). The plugin is a wrapper for a successful job and its output. The plugins are in the end presented to the client. Figure 3 visualizes the infrastructure of the TiO architecture.

During the operation of the current Tile-in-One we encountered several problems, which are summarized in the following bullet points.

- Slow communication between main server and slaves.
- Transferring large amount of files between main server and slaves.



**Figure 3.** The diagram of the current Tile-in-One architecture.

- No possibility of multi-user usage (a plugin can not be used by more than one client at a time).
- Unnecessary caching of jobs.
- Need for micromanaging every aspect of the platform. Also, adding a non-crucial feature requires changes to the whole architecture.
- All system parts which deal with job brokerage are custom-made.
- The plugins are not maintained by their creators (as of now 9 out of 16 are not working).

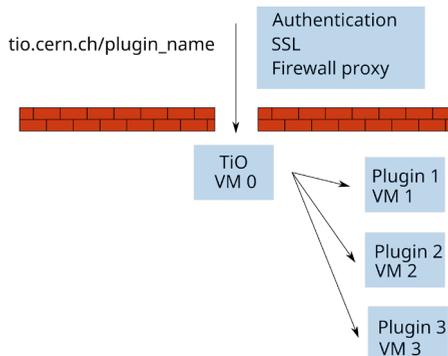
In the end we could say that, we are trying to solve the problem of visualizing a large amount of data in different forms by one monolithic system which has to have support for every type of input and output.

## 4 Starting Over

The idea of Tile-in-One is still deemed to be valuable for the TileCal community, that is why the project is being rebooted. This time, however, it will function just as a simple bridge or gatekeeper between the client and small web applications called plugins. The job of a plugin will be to make all the steps necessary for creating response to the client request. The diagram of the platform can be seen in Figure 4 and the properties of the system are described in following points:

- The main server is just a bridge (reverse proxy), its purpose is to provide authentication and authorization for the plugins behind it.
- A plugin is a small independent web application.
- Due to requirements on different data sources and avoiding interface between plugins every plugin is enclosed in its own Virtual Machine (VM).
- Plugins are based on one of the three templates, since the plugin developers mainly will not be skilled in web development.
- All the source code for the plugin will be stored in a Git repository. CERN runs it's instance of GitLab server, which is centrally managed.

- There will be a person or a group responsible for the maintenance of the plugin.
- Platform developers will have full access to all the VMs in order to be able to take over the maintaining of the machine.



**Figure 4.** The diagram of the new Tile-in-One platform.

Plugin development requires from the plugin developer to design, program and operate the whole machine with a web server on top. It is expected that the majority of the plugin developers will not be sufficiently skilled in web development, therefore the platform will provide setup scripts to setup a VM and three plugin templates for the web application. The platform will provide the following templates:

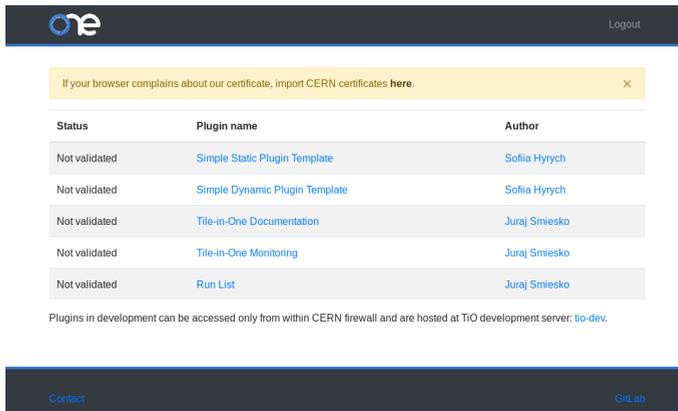
- Simple Static Plugin Template:
  - Dynamic elements only on the user side (JavaScript).
  - Server could regularly run cron jobs to update data files (e.g. csv or json).
  - Ideal for simple status pages of a subsystem.
  - Implemented with HTML, CSS, JavaScript.
- Simple Dynamic Plugin Template:
  - All the above functionality.
  - Can process parameters on the server site.
  - Ideal for simple plugins with several parameters.
  - Implemented with Python and Bottle.
- Complex Dynamic Plugin Template:
  - All the above functionality.
  - Can store web page data in database.
  - Ideal for complex plugins, which require user management.
  - Implemented with Python and Django.

The new implementation of TiO platform will rely on several open source projects as well as services already used at CERN and was designed to use as much existing software and services as possible. In the future we would like to transfer tasks related to the creation and development of plugins into CERN-GitLab’s CI/CD environment. Table 1 lists the technologies used in the new TiO.

**Table 1.** Rundown of technologies used in the new Tile-in-One.

Provided by platform	Technology	Status
Secure connection	CERN CA [4]	Works
Firewall proxy	OpenResty [5]	Works
Routing of requests	OpenResty [5]	Works
Authentication	oauth2_proxy [6]	Works
User management		Not implemented
Common place for tile related web applications	CERN GitLab [7]	Works
Web application templates	Bottle [8], Django [9]	Partially works
Monitoring	Monitorix [10]	Works
VM images (install scripts)	OpenStack [11]	Partially works

Since the CERN computing rules do not allow us to show unfinished plugins outside CERN network we operate two main servers. One shows the finished plugins to all authenticated clients, the other one shows unfinished plugins only to authenticated clients inside CERN network. An example screenshot of their user interface is shown in Figure 5.

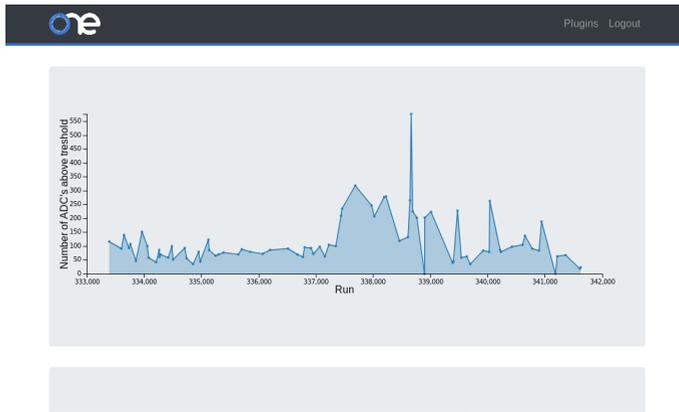


**Figure 5.** Screenshot of the user interface of the main Tile-in-One server. The figure shows main server for finished plugins.

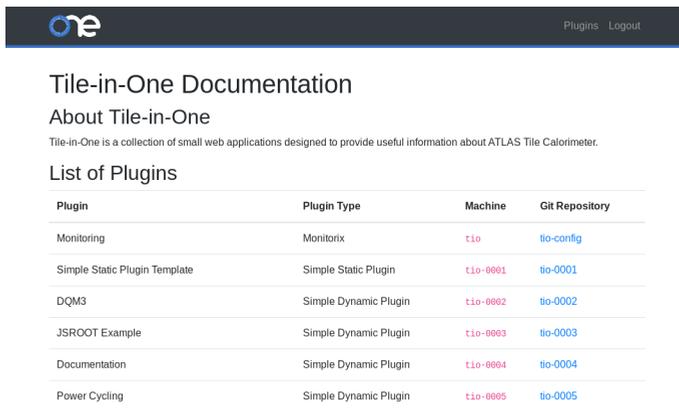
## 5 First Plugins

Apart from two plugin templates, Simple Static Plugin Template (example screenshot shown in Figure 6) and Simple Dynamic Plugin Template, which run as demo plugins for the plugin developers we started with development of two classes of plugins. Plugins which complement the Tile-in-One platform and regular plugins for TileCal data quality assessment.

For now, we have two plugins, which complement the new Tile-in-One platform. One is for monitoring of the VMs on which plugins and main servers run. This plugin is based on Monitorix [10]. The second one is for hosting of the platform documentation. It's screenshot is shown in Figure 7.



**Figure 6.** Screenshot of the Simple Static Plugin Template for new Tile-in-One platform.

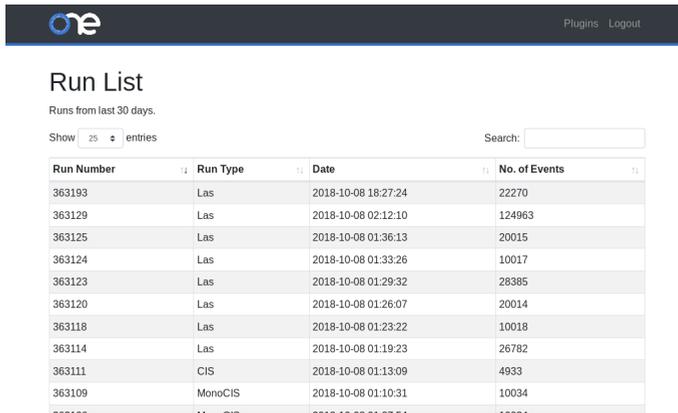


**Figure 7.** Screenshot of a TiO platform complementing plugin, which serves documentation.

First plugins intended to serve the TileCal community are being developed, this can be seen on Figure 5 and one of the plugins is ready to be used by the TileCal community, see Figure 8. For now it simply shows basic information about runs taken in the last 30 days, in future we plan to expand its functionality.

## 6 Conclusion

We recreated a common web space for the ATLAS Tile Calorimeter related offline data quality tools. After realizing that manpower resources would not be sufficient in keeping the current platform afloat we decided to design a much lighter and more stable platform with emphasis on using already developed and common software. We hope that this will lead to an easier transfer of knowledge about inner workings of the platform, quicker and easier development of the plugins, and also easier integration of the currently working ATLAS Tile Calorimeter web tools into the Tile-in-One platform.



Run List

Runs from last 30 days.

Show 25 entries Search:

Run Number	Run Type	Date	No. of Events
363193	Las	2018-10-08 18:27:24	22270
363129	Las	2018-10-08 02:12:10	124963
363125	Las	2018-10-08 01:36:13	20015
363124	Las	2018-10-08 01:33:26	10017
363123	Las	2018-10-08 01:29:32	28385
363120	Las	2018-10-08 01:26:07	20014
363118	Las	2018-10-08 01:23:22	10018
363114	Las	2018-10-08 01:19:23	26782
363111	CIS	2018-10-08 01:13:09	4933
363109	MonoCIS	2018-10-08 01:10:31	10034

**Figure 8.** Screenshot of a regular plugin in new Tile-in-One platform called Run List.

## References

- [1] ATLAS Collaboration, JINST 3, *The ATLAS Experiment at the CERN Large Hadron Collider*, S08003 (2008)
- [2] C. Pralavorio, *Record luminosity: well done LHC*, available from <https://home.cern/about/updates/2017/11/record-luminosity-well-done-lhc/> [accessed 2018-Oct-08]
- [3] ATLAS Collaboration, *ATLAS tile calorimeter: Technical Design Report*, CERN-LHCC-96-042 (1996)
- [4] CERN Certification Authority, available from <https://ca.cern.ch/ca/>, [accessed 2018-Oct-08]
- [5] OpenResty Inc., “OpenResty” [software], version 1.13.6.2, 2018. Available from <https://openresty.org/en/> [accessed 2018-Oct-08]
- [6] Bitly, Inc., “oauth2\_proxy” [software], commit a94b0a8b25e553f7333f7b84aeb89d9d18ec259b, available from [https://github.com/bitly/oauth2\\_proxy/](https://github.com/bitly/oauth2_proxy/) [accessed 2018-Oct-08]
- [7] GitLab, “GitLab” [software], version 11.2.4-ee, 2018. Available from <https://about.gitlab.com/> [accessed 2018-Oct-08]
- [8] M. Hellkamp, “Bottle” [software], version 0.12, 2013. Available from <https://bottlepy.org/docs/dev/> [accessed 2018-Oct-08]
- [9] Django Software Foundation, “Django” [software], available from <https://www.djangoproject.com/> [accessed 2018-Oct-08]
- [10] J. Sanfeliu, “Monitorix” [software], version 3.10.1, 2018. Available from <https://www.monitorix.org/> [accessed 2018-Oct-08]
- [11] The OpenStack Foundation, “OpenStack” [software], available from <https://www.openstack.org/> [accessed 2018-Oct-08]