



# Architecture

© Hurence

Version v1.0, 07.06.2020: First book

# Table of content

Description of the different modules .....	2
gateway.....	2
grafana-historian-datasource .....	2
historian-modele .....	2
integration-tests .....	2
loader .....	3
logisland-timeseries.....	3
timeseries.....	3

This section describes the architecture of the historian project.

# Description of the different modules

Here are the different modules of this project and their role (see the sections below for more details)

- **gateway** : The gateway is our rest server which provides our rest API.
- **grafana-historian-datasource** : It is a git sub-module which is our Grafana plugin to expose a dedicated datasource . We will use it for viewing on historian timeseries. The datasource serves as a junction between Grafana and the gateway (REST API).
- **historian-modele** : Contains objects in common between the different modules.
- **integration-tests** : Contains utility classes for testing integration.
- **loader** : Contains code that allows you to recompact chunks already contained in the historian.
- **logisland-timeseries** : This module extends the "timeseries" module with logisland specificities.
- **timeseries** : A library that allows you to manipulate timeseries and compress them.

## gateway

The gateway (or historian server) is implemented with vertx. This server offers a REST API that grafana-historian-datasource uses to trace the graphs in Grafana. This implies that these two modules are closely linked, if we modify the gateway API this will have an impact on the Grafana datasource and vice versa.

## grafana-historian-datasource

The datasource is implemented using javascript, the corresponding github project is available [here] (<https://github.com/Hurence/grafana-historian-datasource>), we recommend developing this plugin directly in your own project. Once you have pushed your modifications in the datasource project, you can update the submodule in this project with the following command:

```
git submodule update --remote
```

This will automatically update the project with the last commit of the submodule project.

## historian-modele

It is a module which is used to share classes. It is used by the gateway (the historian server) but also by other modules. For example the loader module. This avoids having to embed the dependency on the gateway (which contains many things).

## integration-tests

This module contains utilities for doing integration tests. It typically contains junit5 extensions for Spark and for Solr. The extension for Solr uses the libraries [testcontainers]

(<https://www.testcontainers.org/>), to form a Solr cluster with docker compose. This module also contains in its resources the necessary configurations to create the different collections used by our historian.

## loader

The loader contains batch jobs. For the time being, this module contains only one job which is used to recompact the chunks of the historian with different configurable sizes (the compactor). For example during real-time injection into the data historian, small chunks are injected. However, to optimize performance, we need large chunks. In addition, depending on the type of requests made and the frequency of points for a given metric, being able to play on the size of the chunks, and to vary it according to the use case, is very important.

For example if we make requests over long periods, we need big chunks to gain performance. Indeed we will not even need to decompress these chunks, using their precalculated aggregation will be enough. Besides, this recompaction job could also be used as a basis to allow us in the future to add new pre-calculated aggregations if necessary, or to add any information that seems necessary to us.

The compactor uses Spark and Solr.

## logisland-timeseries

This module contains logisland processors using the timeseries library. These processors handle timeseries. Some are used to compact data in the compactor. Ultimately, it would be good to delete this module and only use the tools present in the timeseries module. Indeed it does not seem necessary to embed logisland business logics in the historian.

## timeseries

It is a library that allows you to manipulate metrics in general. We can for example compress them and calculate pre-aggregations.