

Sentilo :

The structure of the Sentilo platform consists of several components that provide the services.

In order to operate the platform, it uses the following components:

- **Publication/subscription server** for communication with sensors and actuators and also, to be able to subscribe the actuators to alerts.

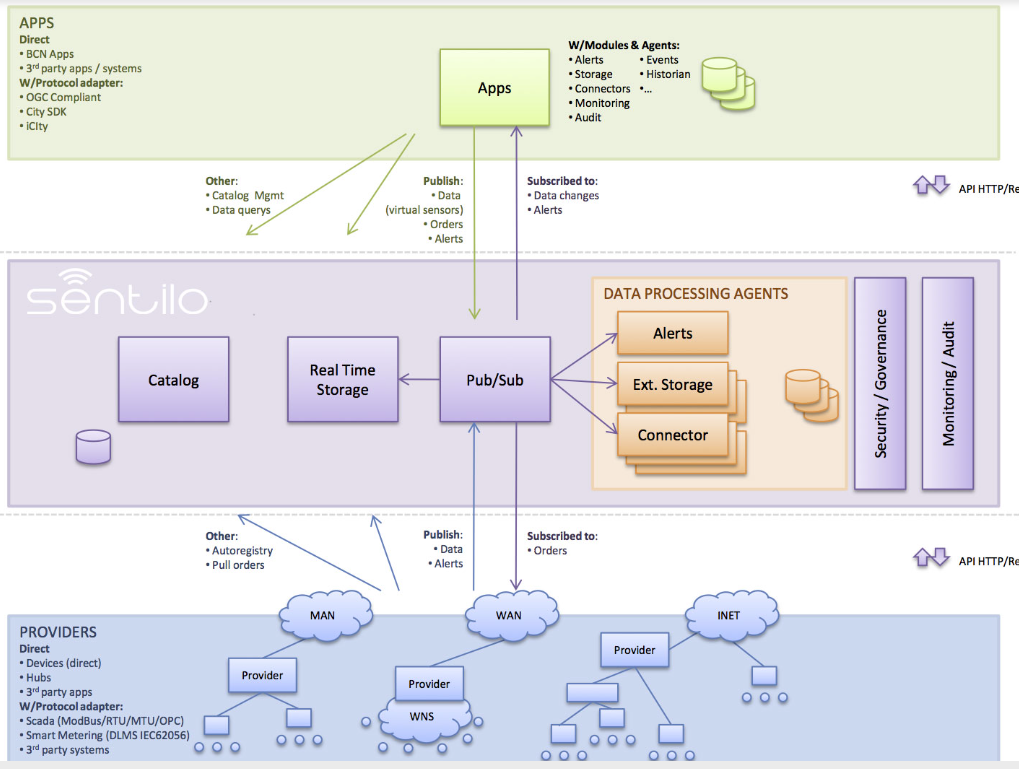
It is based on Redis, a non-relational database engine in memory, based on storage in dictionary-type tables and designed to act as a data structure server.

The subscription allows you to listen to a channel and the publication allows you to send messages through the channel. The format used by the platform to send the messages will be **JSON** (JavaScript Object Notation).

-  **NoSQL Database**(Not Only Structured Query Language)**MongoDB**, to have a more efficient and elastic database.  
MongoDB is an open source software for the creation and management of document-oriented, scalable and high-performance databases.

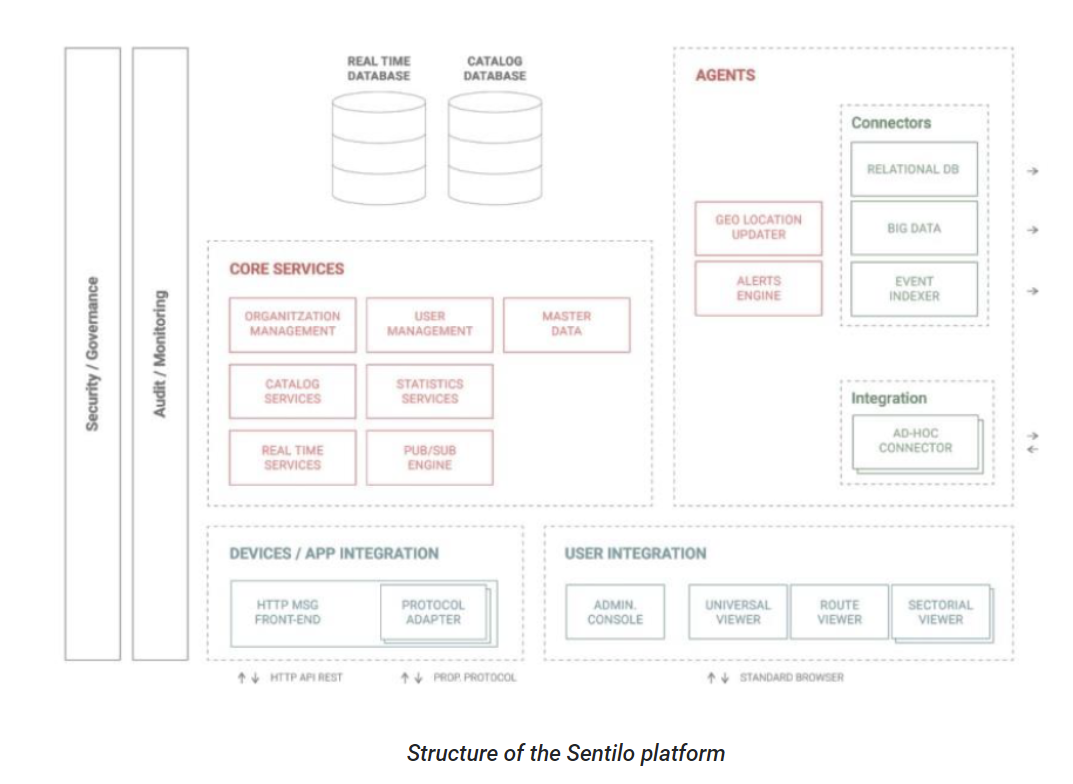
-  **MySQL database manager**, relational database management system, which uses the SQL language. The platform is used to be able to export the data of the events to external relational databases.

-     **Tomcat Web Server** to host the Web Catalog application. It provides an environment for Java coding to be executed in cooperation with a stand-alone web server that allows a high level of traffic and high availability.



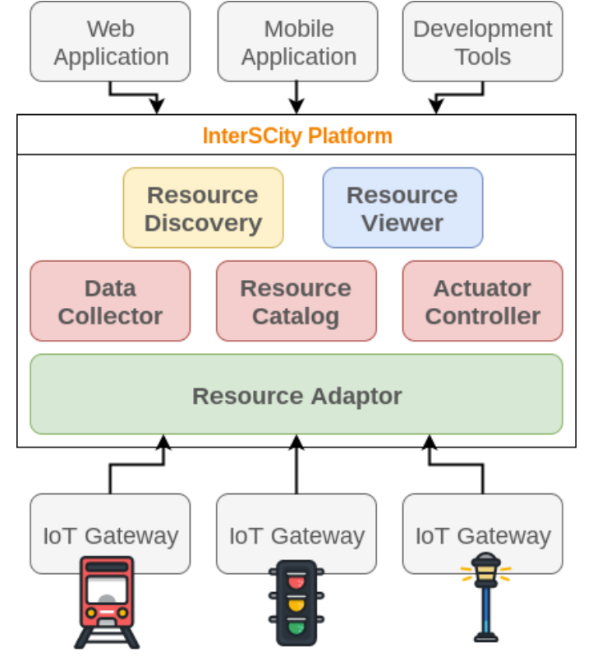
Sentilo includes:

* A front-end for message processing, with a simple REST API
* A administration console for configure the systrem and manage the catalog
* A memory database, aimed to acomplish high performance rates
* A non-SQL database, in order to get a more flexible and scalable system
* A universal viewer, provided as a public demo what can be used as a start point for specific business visualizers.
* A basic statistics module that records and display basic platform performance indicators.
* An extensible component architecture, to enlarge the platform funcionality without modifying the core system. Sentilo starts with an initial set of agents: one for exporting data to relational databases and another to process internal alarms based on basic rules.

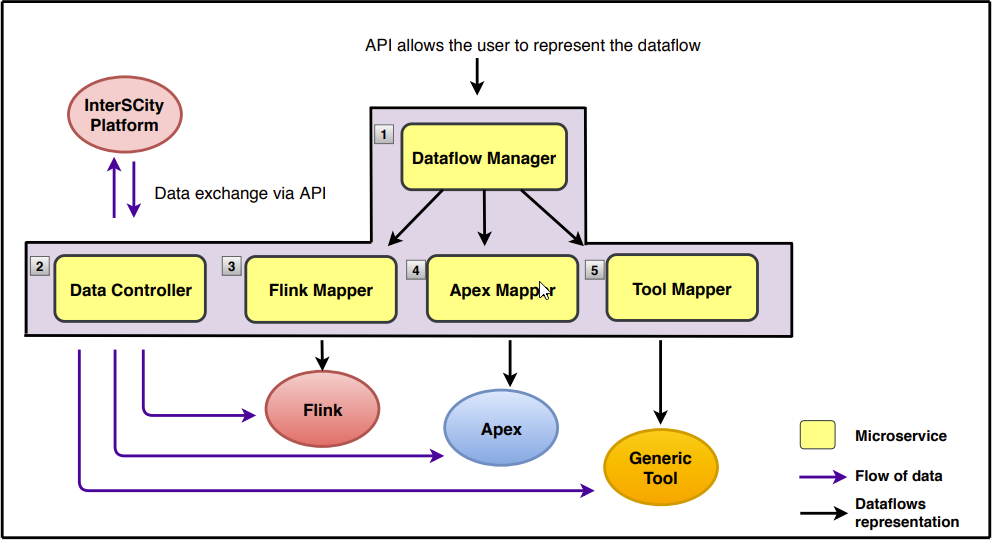


https://www.industrialshields.com/blog/arduino-industrial-1/post/acquisition-and-management-of-data-using-sentilo-100

**InterScity :**



Microservice Architecture  :

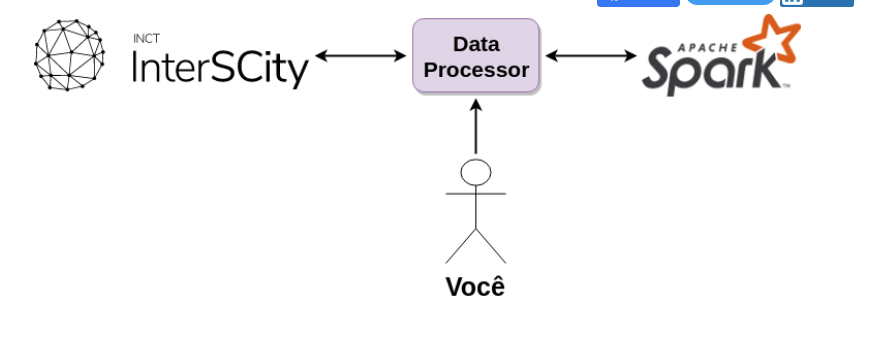


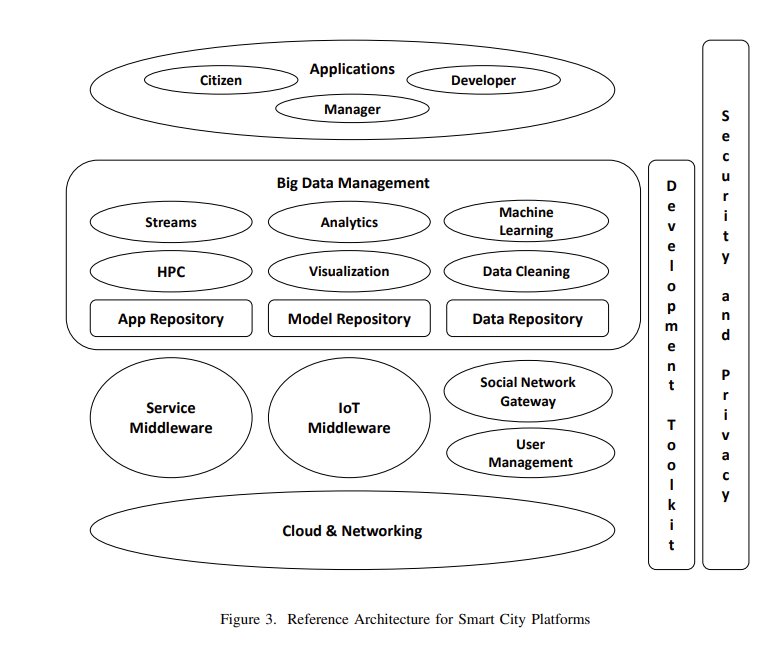
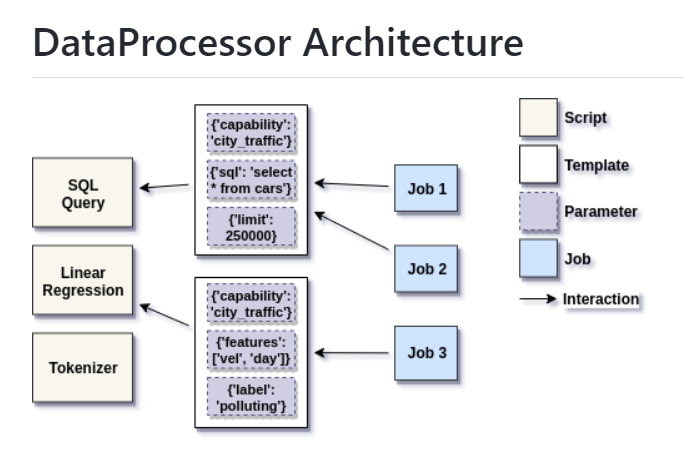
Requirements:

* Java 8 (or higher)
* Maven 3.2.5
* Hadoop 2.8.x
* Apex Core 3.7
* Apex Malhar 3.8
* Apache Flink 1.5.3

utilise

-Apache Spark





SmartMe:

* TOO(L) smart is based on (and evolves) the #SmartMe IoT Platform
* #SmartME was born as a crowdfunding project for the construction of an infrastructure of smart services within the city of Messina.
* The basic technological requirements are based on the "open source" paradigm therefore, "open" solutions have been adopted for software, hardware and data.
* Stack4Things is the framework used for IoT devices management. It constitutes an evolution of OpenStack.
* The reference hardware adopted is based on the Arancino board, which integrates the Raspberry PI compute module and the Arduino control module in a single device.