Business Intelligence for Industrial IoT

NFE211

José ESLAVA

15 – 01 – 2020

Contents

1. Preliminary concepts

1.1. What is IoT

The **Internet of Things** (**IoT**) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

1.2. IoT constraints

- Basic sensor data processing and information transfer.
- Limited processing power.
- Memory footprint constraints.
- Low power consumption.
- Can be isolated or with limited network availability or low bandwidth.
- Small physical size –Mechanical integration.

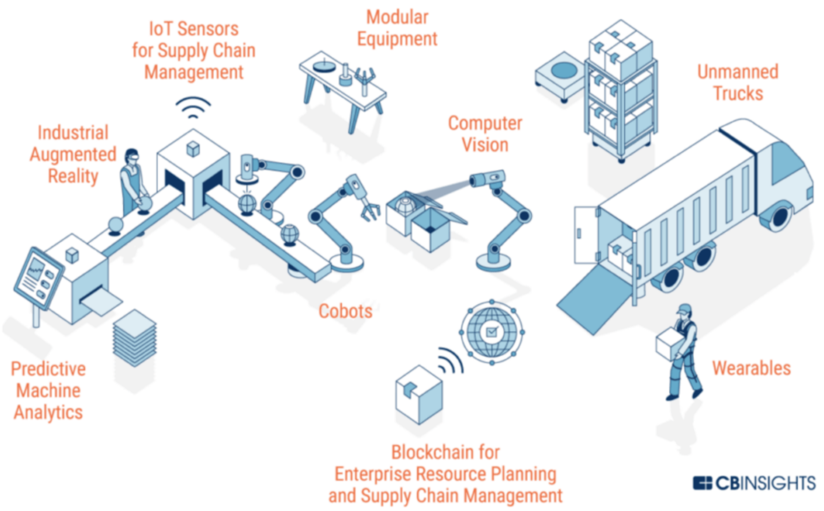
1.3. Industrial IoT

The application of IoT in the Industry implies other constraints:

- Industrial environment robustness (Mechanical, EMC).
- Deterministic –Hard Real time → Low latency.
- Industrial production information → Data security.
- Availability → Autonomy/Independent.
- Factory security constraints.

1.4. Examples of use cases of BI for Industrial IoT

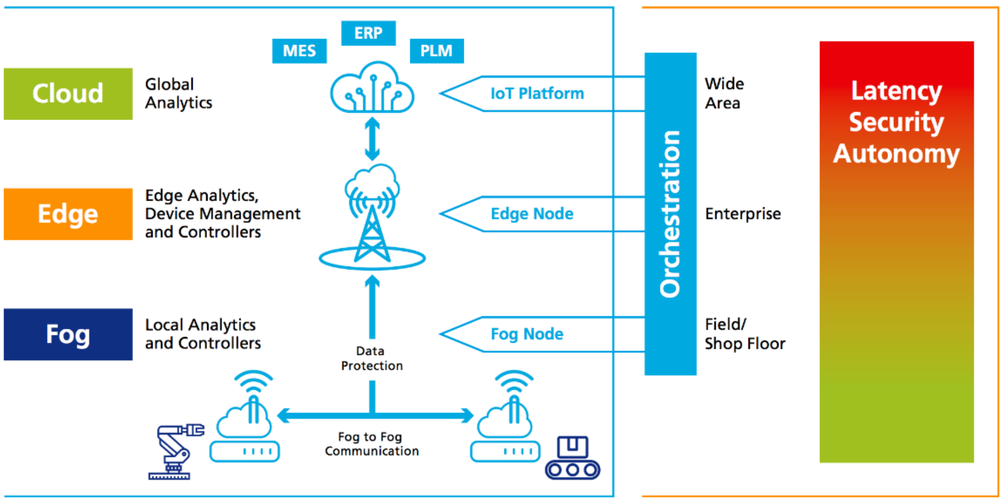
- Vibration patterns analyse for machine anomaly detection -Predictive maintenance.
- Analyse power consumption monitoring to optimize consumption.
- Real time diagnostics.
- Analyse production flow against client orders to optimize mounting chain layout.
- Analyse parts usage to design supplies order plan.



1.5. BI for Industrial IoT Architecture

The classical architecture where IoT devices push data directly to the Cloud has been adapted for industrial application due to Latency constraints.

Industrial Applications can need real time and deterministic processing. This processing can be too complicated to be implemented on a low calculation power device and processing results can be needed with latencies of less than 1ms. In these cases, the processing algorithms are placed between the device and the Cloud. See the following picture about device – edge and cloud levels:



2. Decision-making chain for Industrial IoT

To illustrate a decision-making chain for Industrial IoT, I choose as example an Industrial Machine provider for manufacturing factories, I will call this company "IMP". Its client the company "CMF" wanted to monitor the following facts:

- Fact1: Industrial Machine power consumption.
- Fact 2: Electro-mechanical parts usage to order/replace or maintain used parts.
- Fact 3: Real-Time monitoring of machine vibrations to predict or detect malfunctions.

2.1. Fact1: Industrial Machine power consumption.

To monitor Power consumption, the following actions at the **device level** were done:

- Installation of current and voltage sensors
- Low power Microcontrollers to do basic pre-processing of current and voltage measurements
- Transmit this information through Zigbee, Bluetooth, Wifi or Ethernet protocol to the edge level.

2.2. Fact2: Electro-mechanical parts usage to order/replace or maintain used parts.

To monitor parts usage, the following actions at the **device level** were done:

- Software program on the Microcontroller to count the number of activations and the usage time.
- Transmit this information through Zigbee, Bluetooth, Wifi or Ethernet protocol to the edge level.

2.3. Fact 3: Real-Time monitoring of machine vibrations to predict or detect malfunctions.

To monitor vibration pattern, the following actions at the **device level** were done:

- Vibration sensor connected and pre-processed by the Microcontroller.
- Transmit this information through Zigbee, Bluetooth, Wifi or Ethernet protocol to the edge level.

2.4. At the Edge level.

At the edge level, we put IoT hub to concentrate data coming from the IoT devices. For consumption and part usage monitoring I only log the information to MySQL Tables on the IoT hub database server.

For Vibration pattern analysis, there was a low latency constraint, so it should be implemented on IoT hub microprocessor. The algorithm has not yet specified but some candidates are:

- Sound analysis to output a spectrogram in frequency domain, then process this spectrogram with convolutional neural networks.
- Self-organizing maps, to describe the current state and predict the possible one.

For the IoT hub, I choose embedded Linux as operating system and two development boards with ARM Cortex-A processors.

I used Yocto build system to build the Linux distribution adapted to this Microprocessor and generate the file system with the needed packages as MySQL server.

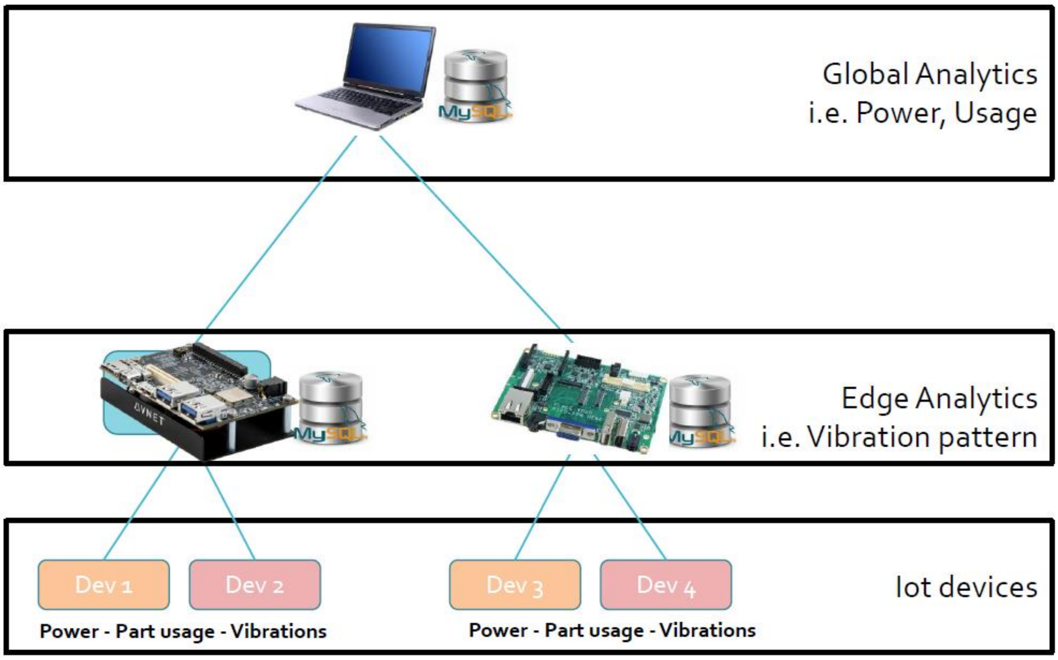
2.5. At the cloud level

To simulate the cloud level I used my computer with MySQL server. I put the following databases on this server:

- Factory: table Factory_areas
- Machine_inventory
- Supplies: providers, part numbers, etc.
- Components_inventory: Components and attributes
- Multidimensional model with 2 facts: part usage and power consumption.

2.6. System integration

I put two IoT hubs and my computer on a wifi network to interconnect the system:



2.7. Data for Logs

As I did not have data logs, I had to generate it artificially, I did some scripts MySQL to create log tables and data on IoT hubs please see the following scripts:

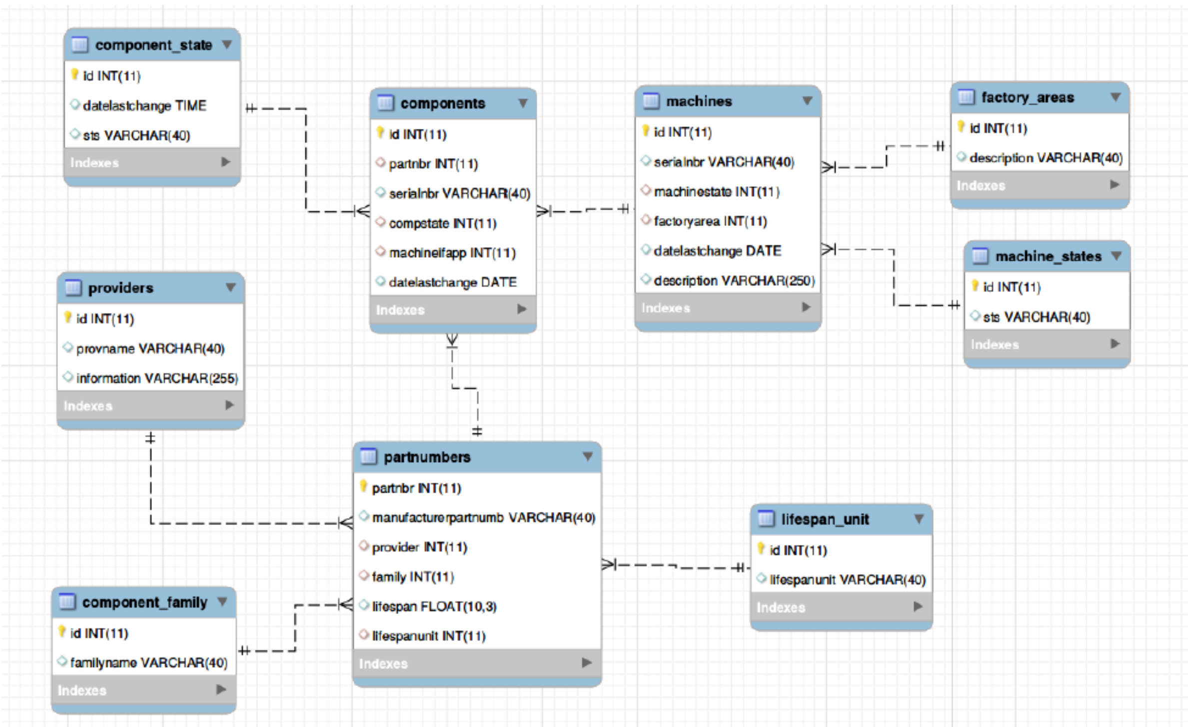
- Machine1.sql
- Machine1_logW1, Machine1_logW2, Machine1_logW3, Machine1_logW4.
- Machine2.sql
- Machine2_logW1, Machine2_logW2, Machine2_logW3, Machine2_logW4.

2.8. Dataware house implementation

I implemented the dataware house on my PC (the cloud) by using MySQL scripts with the following scripts:

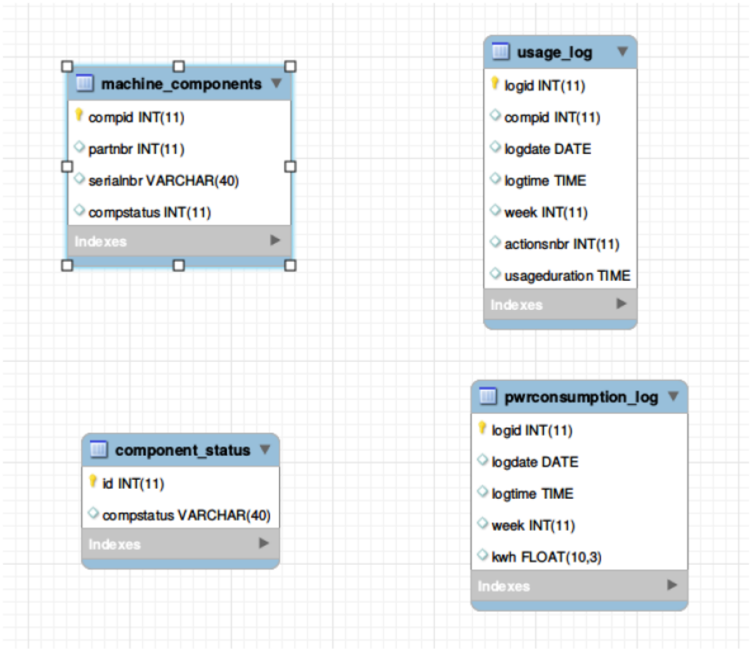
- central_server.sql
- central_server_dwhouse_common.sql
- central_server_dwhouse_consumption.sql
- central_server_dwhouse_fillTimedim.sql
- central_server_dwhouse_usage.sql
- central_server_factory.sql
- central_server_machine_inventory.sql

I generated the databases model with MySQL workbench:



2.9. Machine Logs

Every IoT hub is associated to one machine, so It contains the consumption and part usage logs for its machine, the logs are stored on MySQL databases, the model is the following:



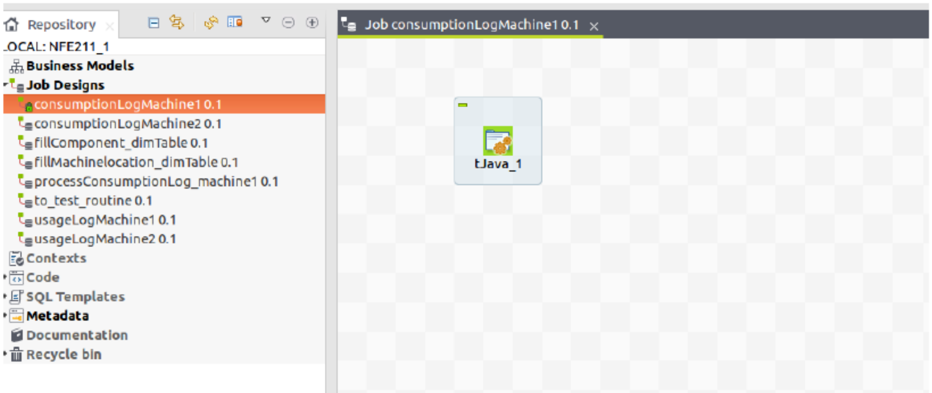
2.10. Periodical jobs in Talend

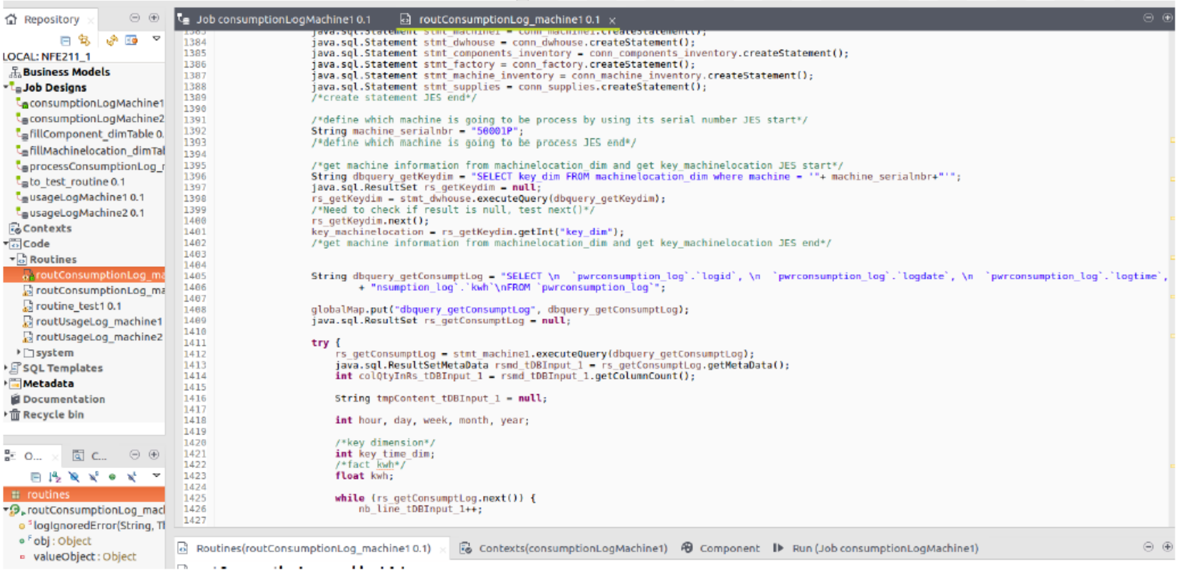
I did Talend jobs to extract the logs from IoT hubs and store the data in the dataware house and fill the **multidimensional model**.

Initially I tried to use Talend objects but I did not find exactly what I needed maybe with more time I could have found something. Finally I took some code generated by Talend for database connection/disconnection and query and I modified it.

I put the modified code into Talend routines and I called these routines from a talend job by using the component TJava.

I have included on the package the Talend routines and jobs. See compressed folder NFE211_1.zip it contains the whole Talend project.





2.11. Multidimensional model

Finally, after executing the Talend jobs, the multidimensional has the following architecture