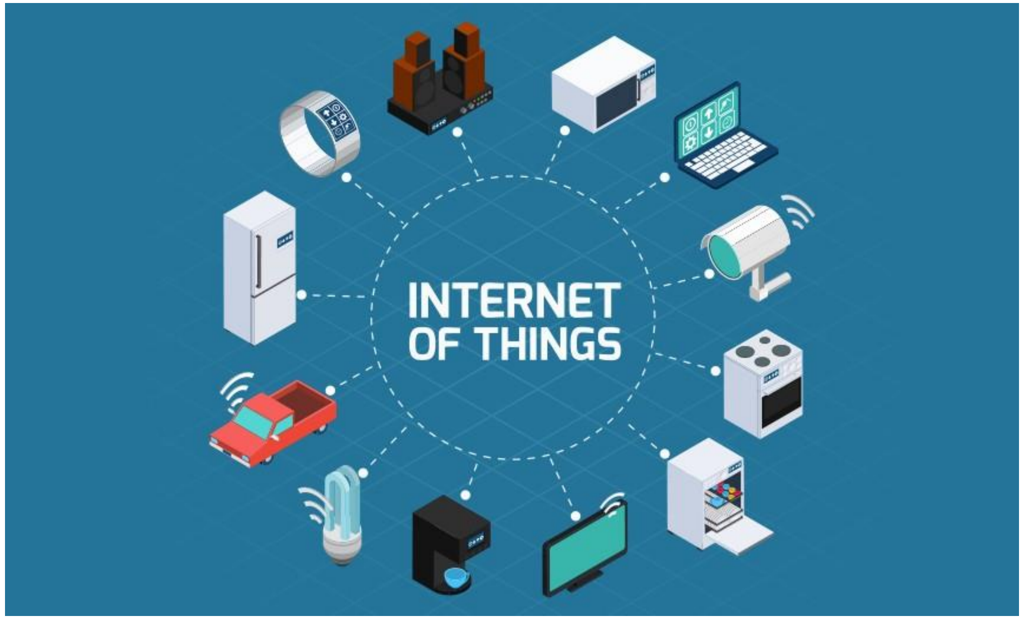
BI - Industrial IoT

NFE211

José ESLAVA – 15/01/2020

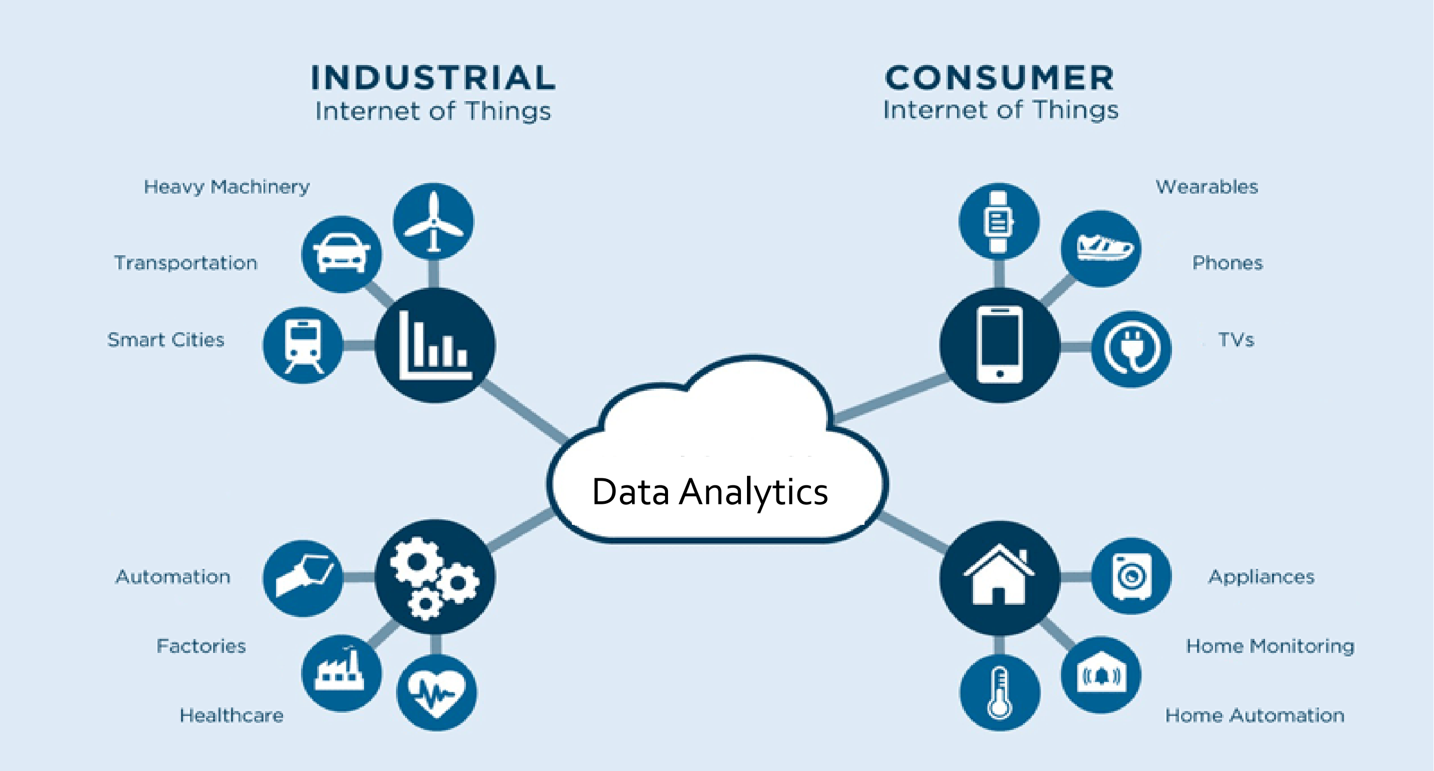
IoT

- Basic sensor data processing and transmit information.
- 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.



Industrial IoT

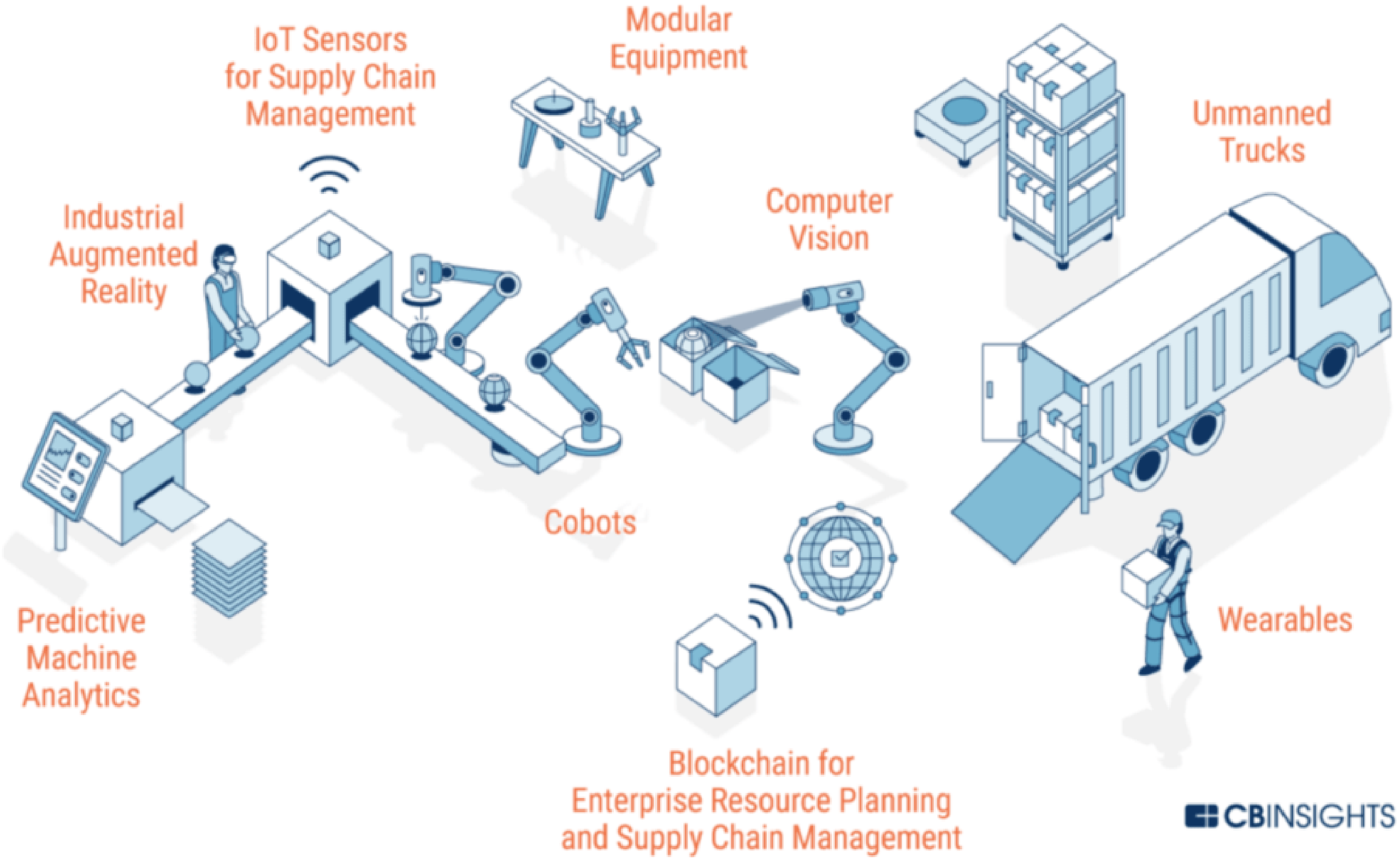
- Industrial environment robustness (Mechanical, EMC).
- Deterministic – Hard Real time → Low latency.
- Industrial production information → Data security.
- Availability → Autonomy
- Factory security constraints.



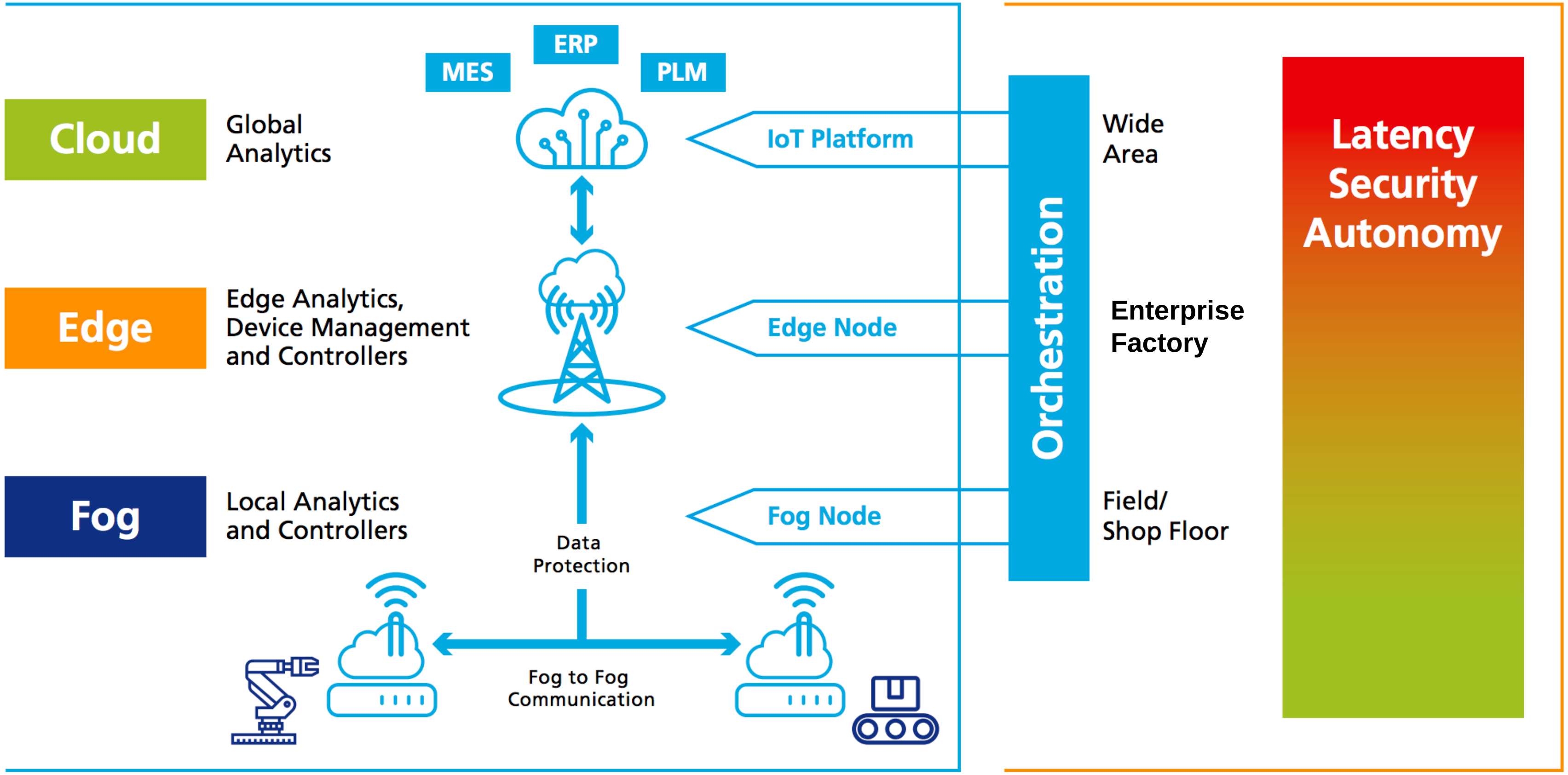
Industrial IoT
+
BI
Data analytics

Some use cases

- 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.


IoT Sensors for Supply Chain Management
Modular Equipment
Unmanned Trucks
Industrial Augmented Reality
Computer Vision
Cobots
Predictive Machine Analytics
Wearables
Blockchain for Enterprise Resource Planning and Supply Chain Management
CBINSIGHTS

BI - Industrial IoT Architecture

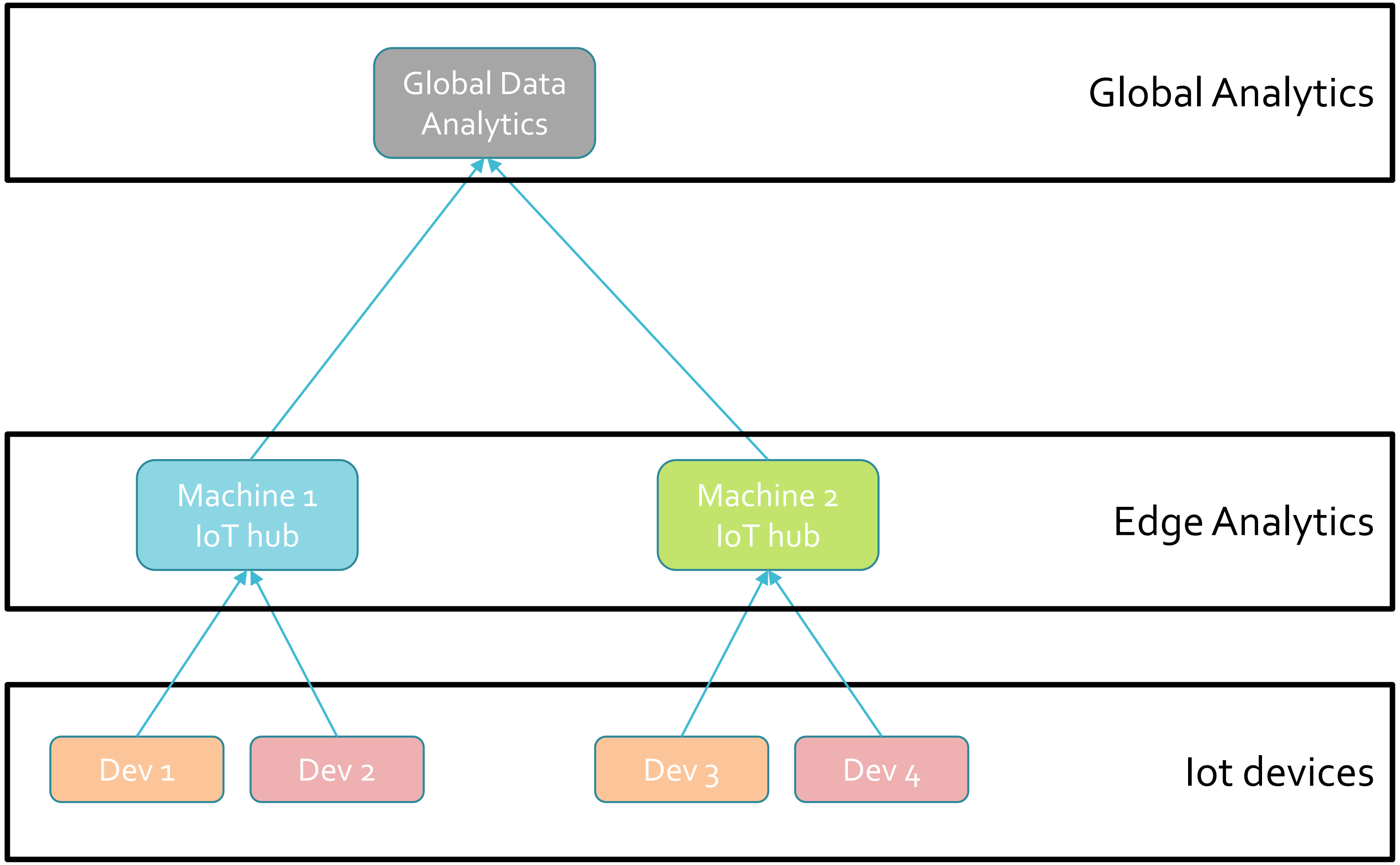


Latency – Security - Autonomy

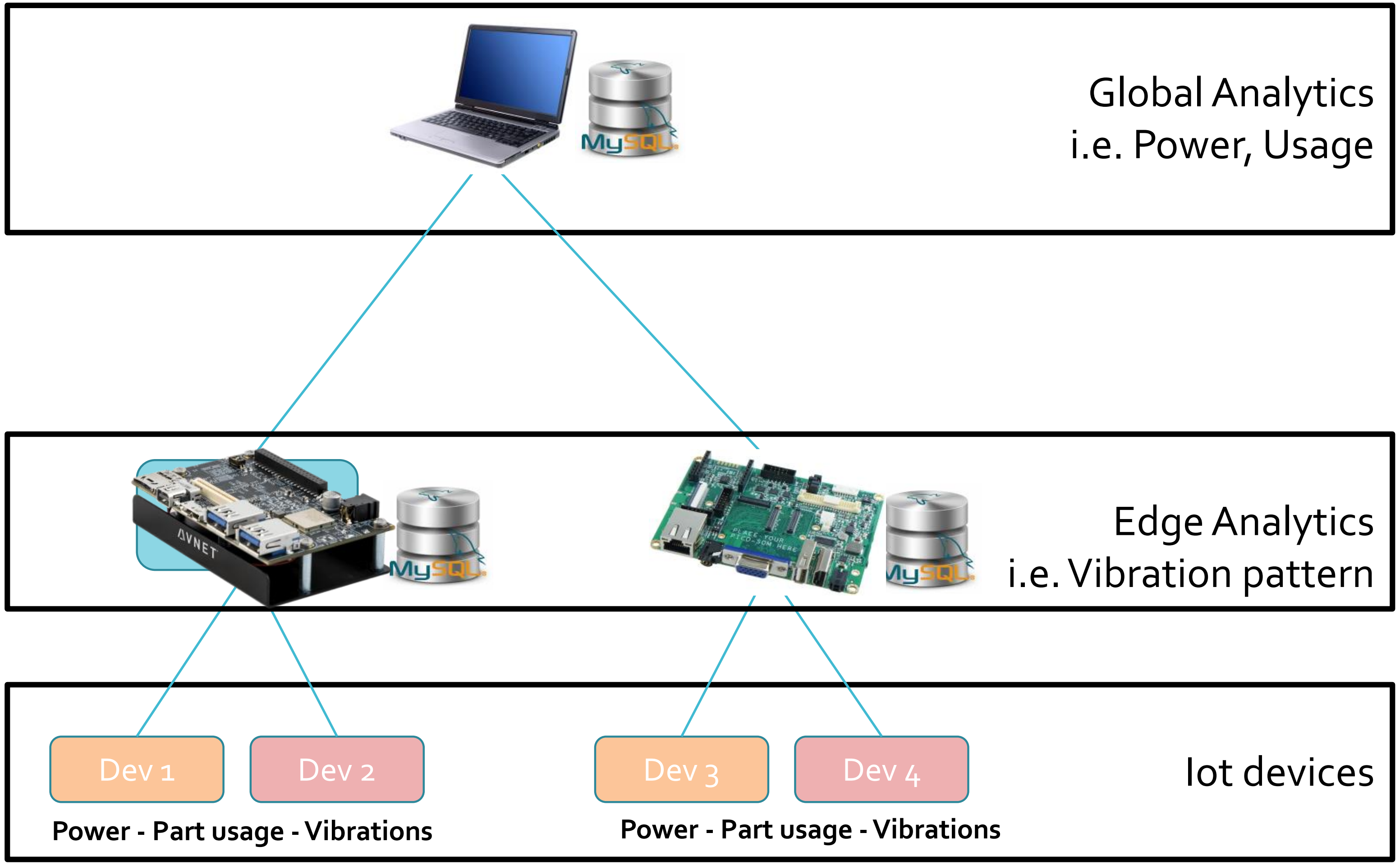
Demo use cases

- Power consumption monitoring.
- Part usage for predictive maintenance and planning of supplies order.
- Vibration pattern analyse to prevent/detect machine malfunctions.

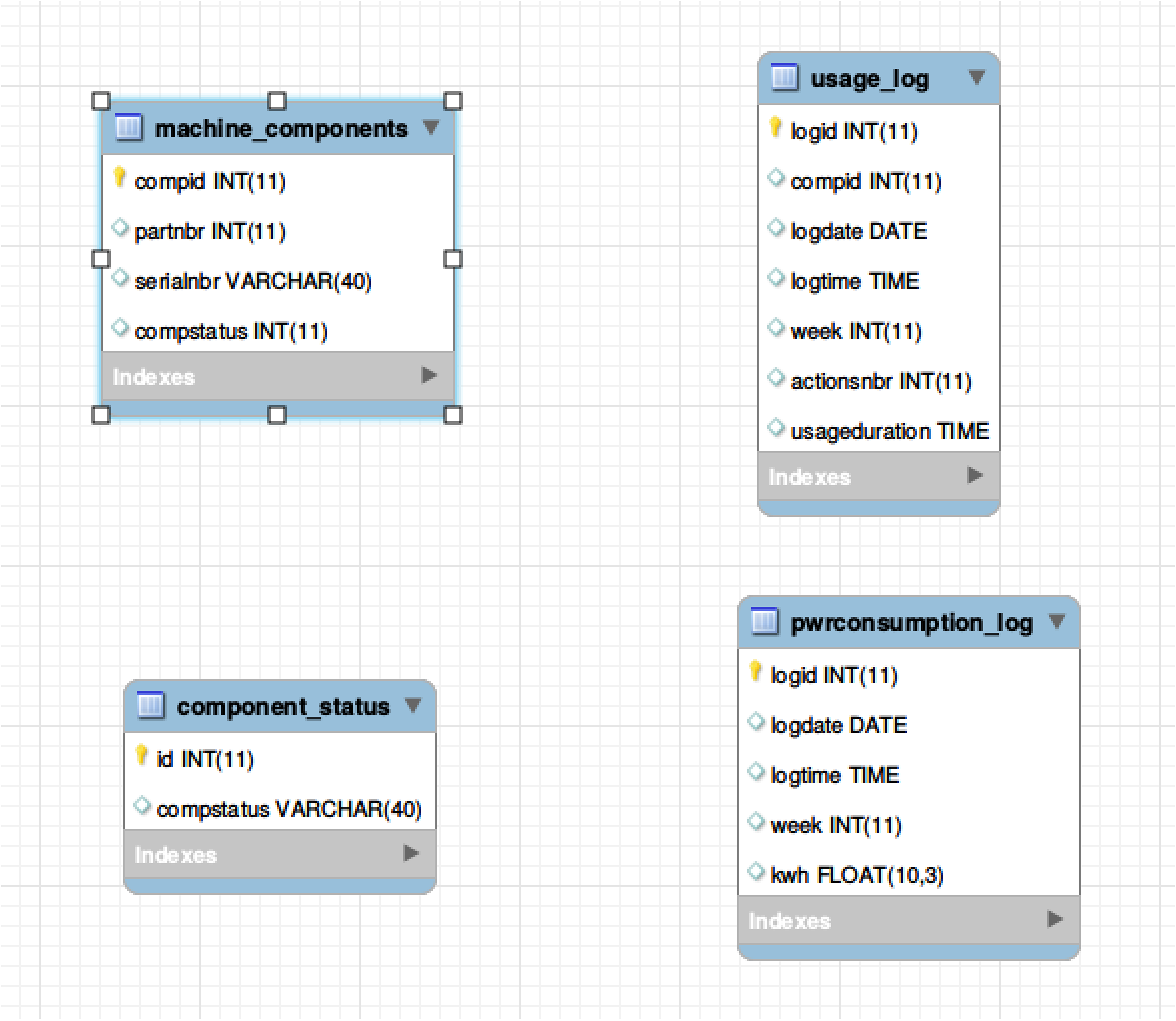
Demo Architecture



Demo Architecture



Machine1 DB model



machine_components

compid INT(11)

partnbr INT(11)

serialnbr VARCHAR(40)

compstatus INT(11)

usage_log

logid INT(11)

compid INT(11)

logdate DATE

logtime TIME

week INT(11)

actionsnbr INT(11)

usageduration TIME

component_status

id INT(11)

compstatus VARCHAR(40)

pwrconsumption_log

logid INT(11)

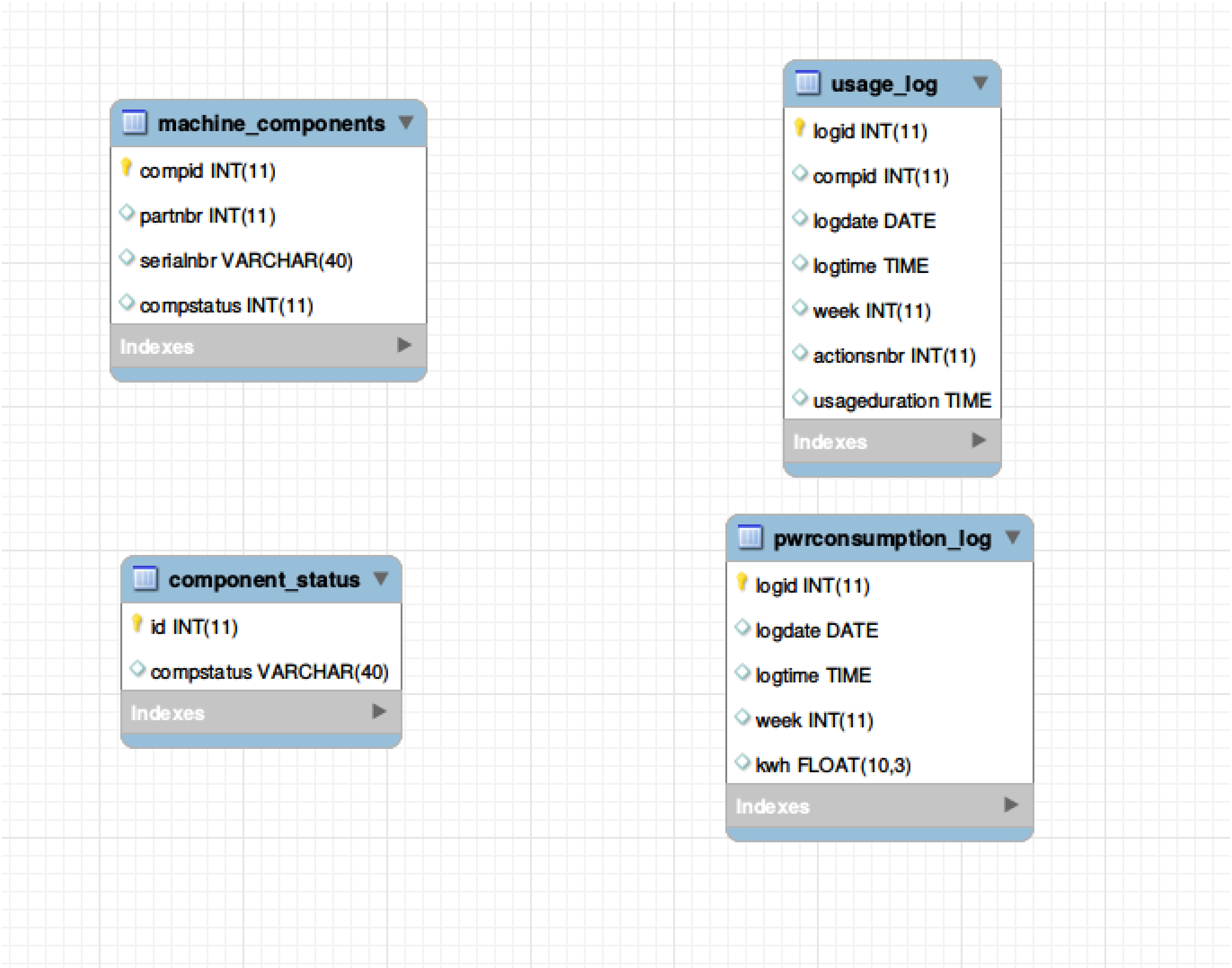
logdate DATE

logtime TIME

week INT(11)

kwh FLOAT(10,3)

Machine2 DB model



machine_components

compid INT(11)
partnbr INT(11)
serialnbr VARCHAR(40)
compstatus INT(11)

Indexes

usage_log

logid INT(11)
compid INT(11)
logdate DATE
logtime TIME
week INT(11)
actionsnbr INT(11)
usageduration TIME

Indexes

component_status

id INT(11)
compstatus VARCHAR(40)

Indexes

pwrconsumption_log

logid INT(11)
logdate DATE
logtime TIME
week INT(11)
kwh FLOAT(10,3)

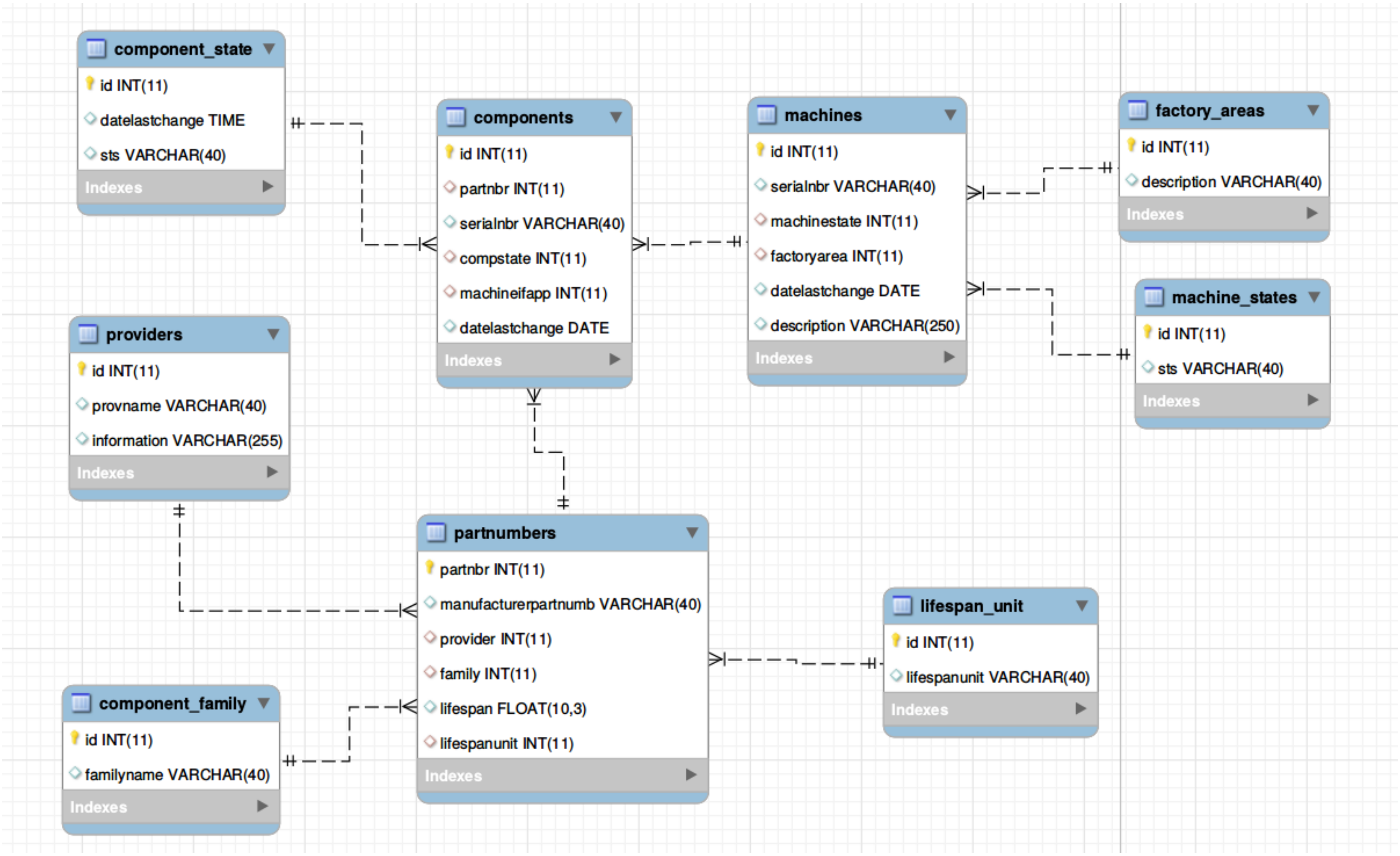
Indexes

DwHouse

- Databases:
  - Factory: table Factory_areas
  - Machine_inventory
  - Supplies: providers, part numbers, etc.
  - Components_inventory: Components and attributs.

- These tables have been inserted by using MySQL scripts.

DWHouse – suppliers, inventories …



component_state
id INT(11)
datelastchange TIME
sts VARCHAR(40)
Indexes

components
id INT(11)
partnbr INT(11)
serialnbr VARCHAR(40)
compstate INT(11)
machineifapp INT(11)
datelastchange DATE
Indexes

machines
id INT(11)
serialnbr VARCHAR(40)
machinestate INT(11)
factoryarea INT(11)
datelastchange DATE
description VARCHAR(250)
Indexes

factory_areas
id INT(11)
description VARCHAR(40)
Indexes

machine_states
id INT(11)
sts VARCHAR(40)
Indexes

providers
id INT(11)
provname VARCHAR(40)
information VARCHAR(255)
Indexes

partnumbers
partnbr INT(11)
manufacturerpartnumb VARCHAR(40)
provider INT(11)
family INT(11)
lifespan FLOAT(10,3)
lifespanunit INT(11)
Indexes

lifespan_unit
id INT(11)
lifespanunit VARCHAR(40)
Indexes

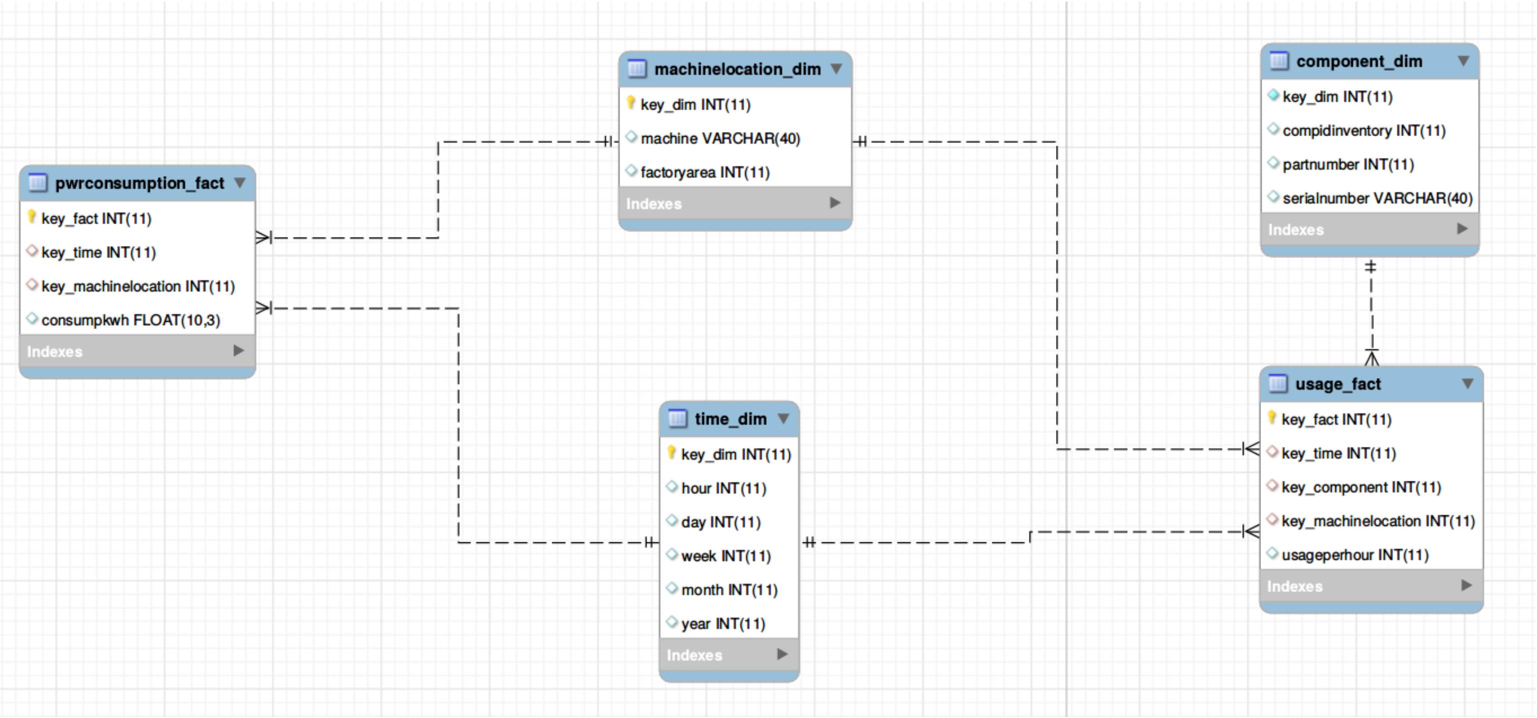
component_family
id INT(11)
familyname VARCHAR(40)
Indexes

Dwhouse – multidimensional model

- Facts:
  - Part usage
  - Power consumption
- Dimensions:
  - Time
  - Machine
  - Component

- These table are filled periodically by retrieving data from Machines logs with Java routines in ETL Talend.

DWHouse Multidim. Model



pwrconsumption_fact
- key_fact INT(11)
- key_time INT(11)
- key_machinelocation INT(11)
- consumpkwh FLOAT(10,3)
- Indexes

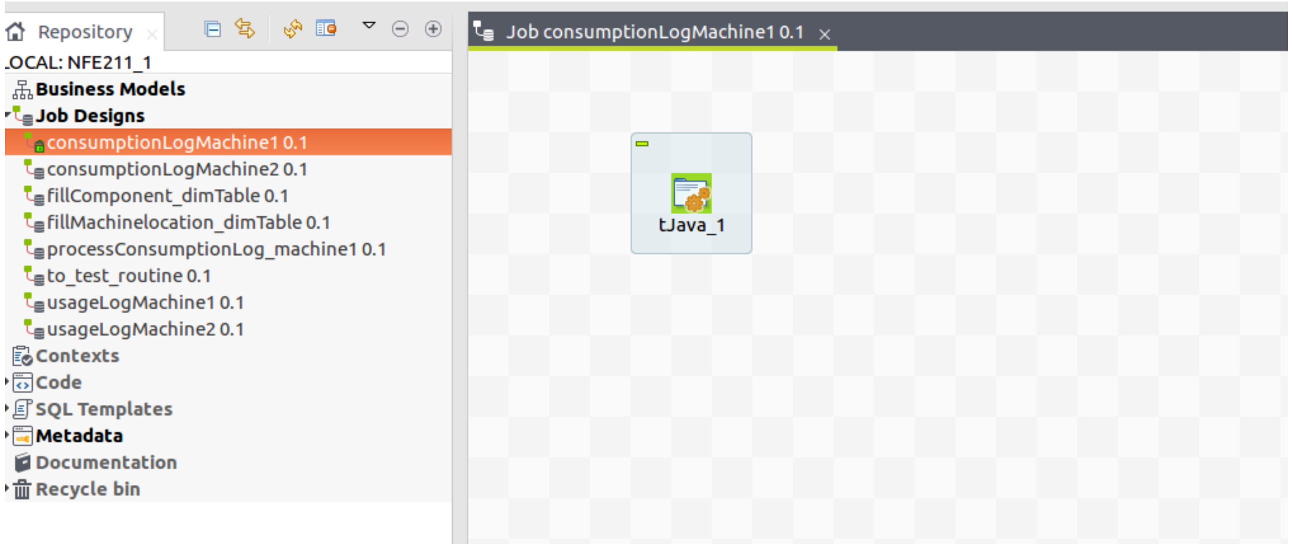
machinelocation_dim
- key_dim INT(11)
- machine VARCHAR(40)
- factoryarea INT(11)
- Indexes

component_dim
- key_dim INT(11)
- compidinventory INT(11)
- partnumber INT(11)
- serialnumber VARCHAR(40)
- Indexes

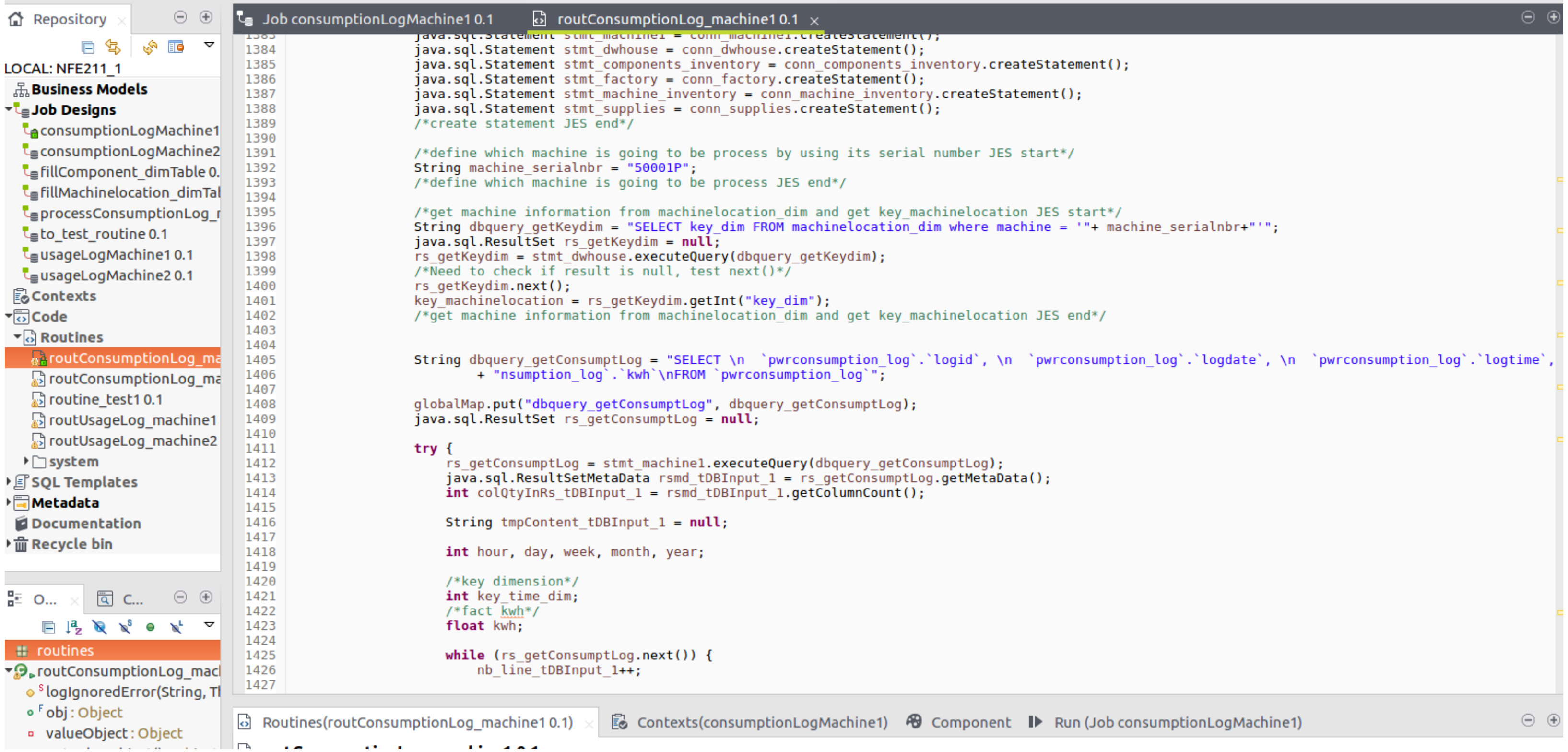
time_dim
- key_dim INT(11)
- hour INT(11)
- day INT(11)
- week INT(11)
- month INT(11)
- year INT(11)
- Indexes

usage_fact
- key_fact INT(11)
- key_time INT(11)
- key_component INT(11)
- key_machinelocation INT(11)
- usageperhour INT(11)
- Indexes

Jobs in Talend



Repository
LOCAL: NFE211_1
Business Models
Job Designs
consumptionLogMachine1 0.1
consumptionLogMachine2 0.1
fillComponent_dimTable 0.1
fillMachinelocation_dimTable 0.1
processConsumptionLog_machine1 0.1
to_test_routine 0.1
usageLogMachine1 0.1
usageLogMachine2 0.1
Contexts
Code
SQL Templates
Metadata
Documentation
Recycle bin
Job consumptionLogMachine1 0.1
tJava_1



```
java.sql.Statement stmt_machine1 = conn_machine1.createStatement();
java.sql.Statement stmt_dwhouse = conn_dwhouse.createStatement();
java.sql.Statement stmt_components_inventory = conn_components_inventory.createStatement();
java.sql.Statement stmt_factory = conn_factory.createStatement();
java.sql.Statement stmt_machine_inventory = conn_machine_inventory.createStatement();
java.sql.Statement stmt_supplies = conn_supplies.createStatement();
/*create statement JES end*/

/*define which machine is going to be process by using its serial number JES start*/
String machine_serialnbr = "50001P";
/*define which machine is going to be process JES end*/

/*get machine information from machinelocation_dim and get key_machinelocation JES start*/
String dbquery_getKeydim = "SELECT key_dim FROM machinelocation_dim where machine = '"+ machine_serialnbr+"'";
java.sql.ResultSet rs_getKeydim = null;
rs_getKeydim = stmt_dwhouse.executeQuery(dbquery_getKeydim);
/*Need to check if result is null, test next()*/
rs_getKeydim.next();
key_machinelocation = rs_getKeydim.getInt("key_dim");
/*get machine information from machinelocation_dim and get key_machinelocation JES end*/


String dbquery_getConsumptLog = "SELECT \n  `pwrconsumption_log`.`logid`, \n  `pwrconsumption_log`.`logdate`, \n  `pwrconsumption_log`.`logtime`,
        + "nsumption_log`.`kwh`\nFROM `pwrconsumption_log`";

globalMap.put("dbquery_getConsumptLog", dbquery_getConsumptLog);
java.sql.ResultSet rs_getConsumptLog = null;

try {
    rs_getConsumptLog = stmt_machine1.executeQuery(dbquery_getConsumptLog);
    java.sql.ResultSetMetaData rsmd_tDBInput_1 = rs_getConsumptLog.getMetaData();
    int colQtyInRs_tDBInput_1 = rsmd_tDBInput_1.getColumnCount();

    String tmpContent_tDBInput_1 = null;

    int hour, day, week, month, year;

    /*key dimension*/
    int key_time_dim;
    /*fact kwh*/
    float kwh;

    while (rs_getConsumptLog.next()) {
        nb_line_tDBInput_1++;

```

Questions ???