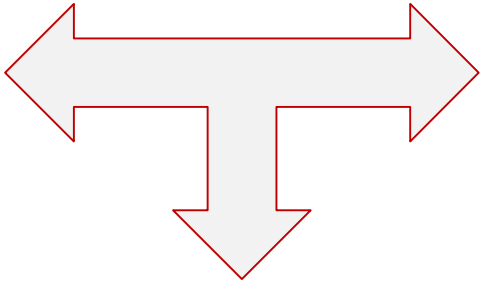
Internet Of Things enabler



Environmental services

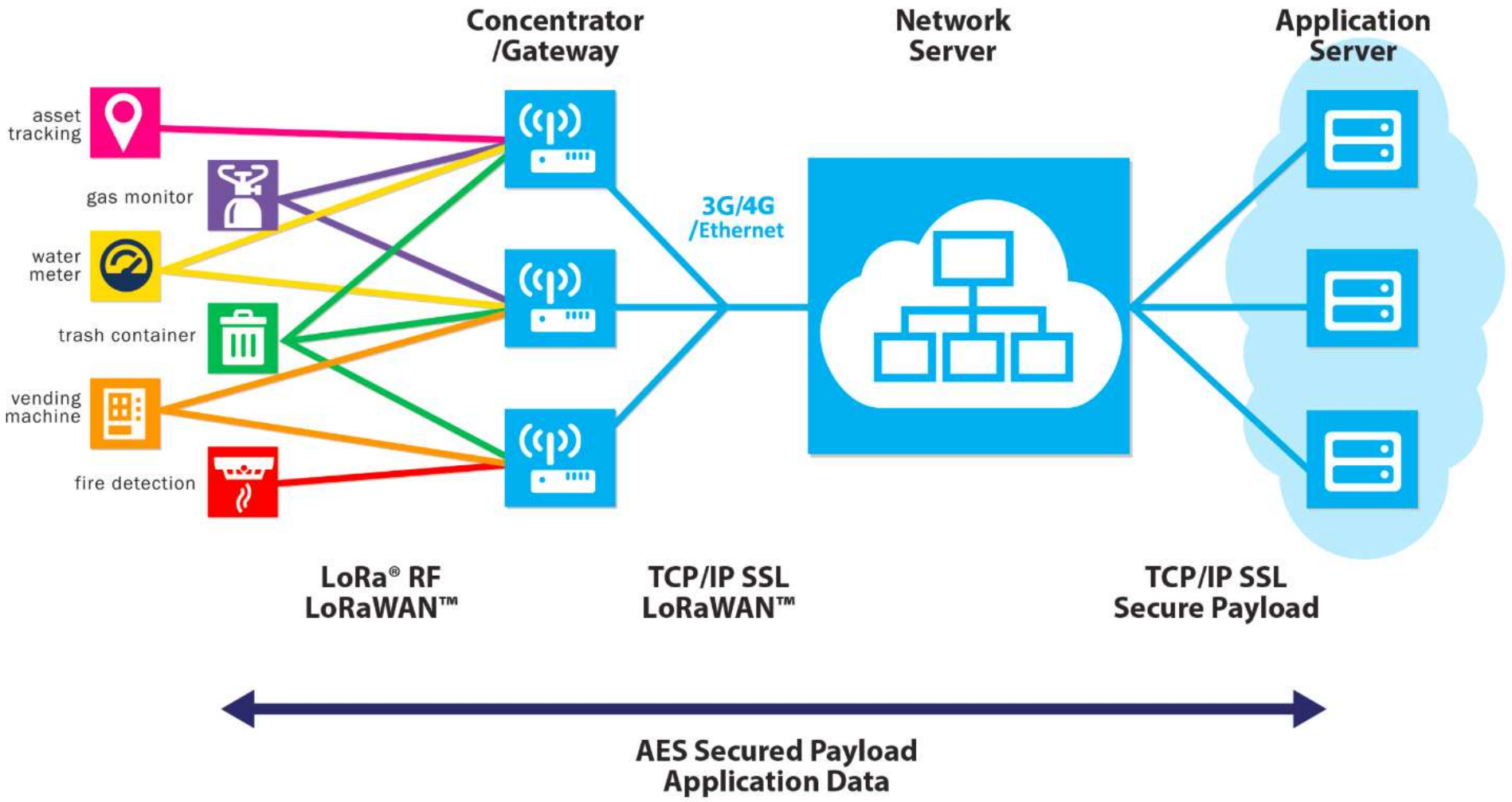
Proof Of Concept: LoRa-RFID-SAP integration for Waste & Recycling

Author: Marco Moschella
Date: November 2018

Contents - Proof Of Concept

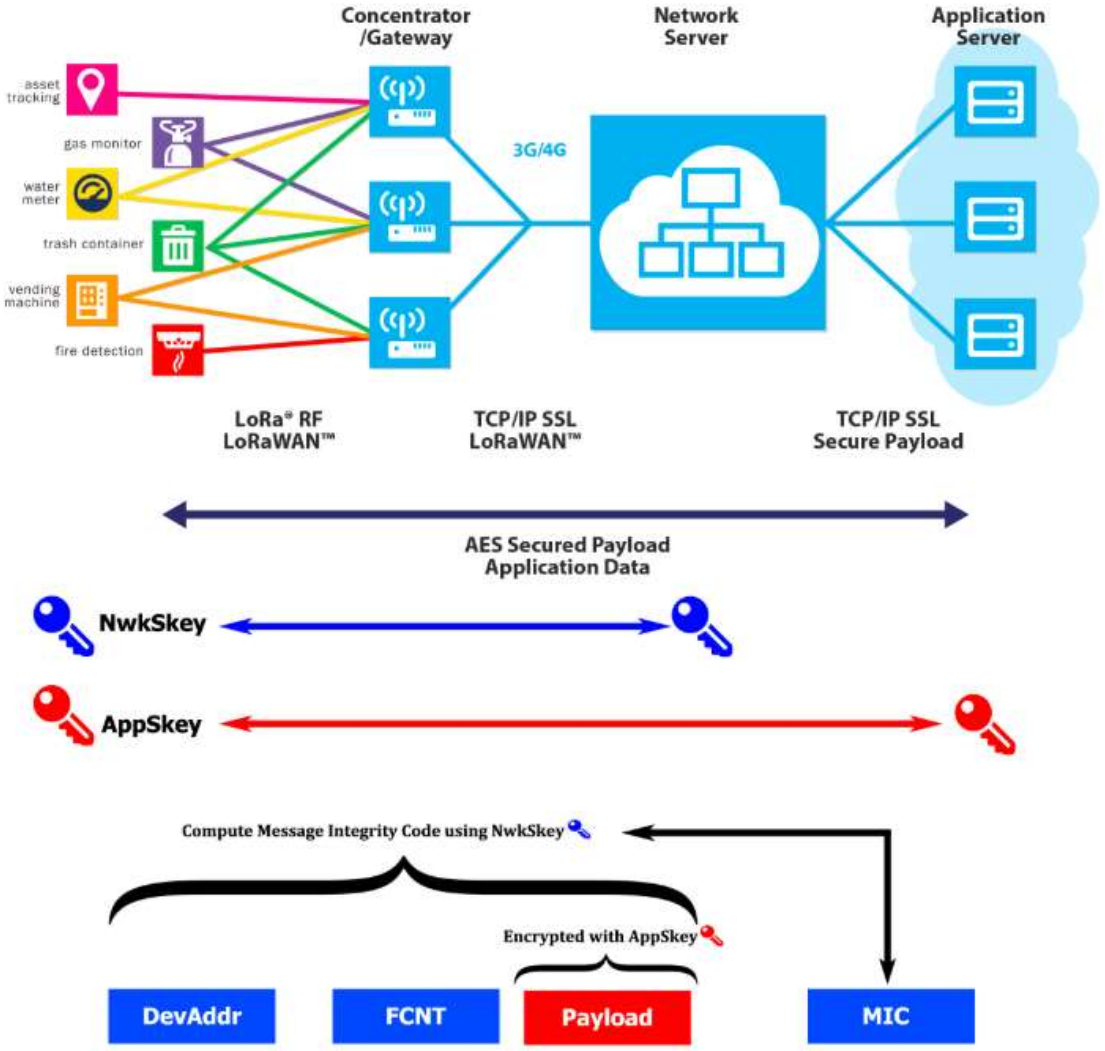
- LoRaWAN™
- POC Architecture
- LoRa-RFID Devices
- The Things Network
- Web Service SAP oDATA
- Node-Red sketch
- SAP backend data reception
- IBM IoT integration

LoRaWAN™ Network Structure



Specifiche Tecniche

LoRaWAN™ Network Security



LoRaWAN™ works with two symmetrical key types for comunication security, these are unique for each LoRa device.
The **NwkSkey** is used in order to ensure message integrity from the device to the Network server.

L'AppSkey is used for AES-128 end-to-end encryption from the device to the Application Server.

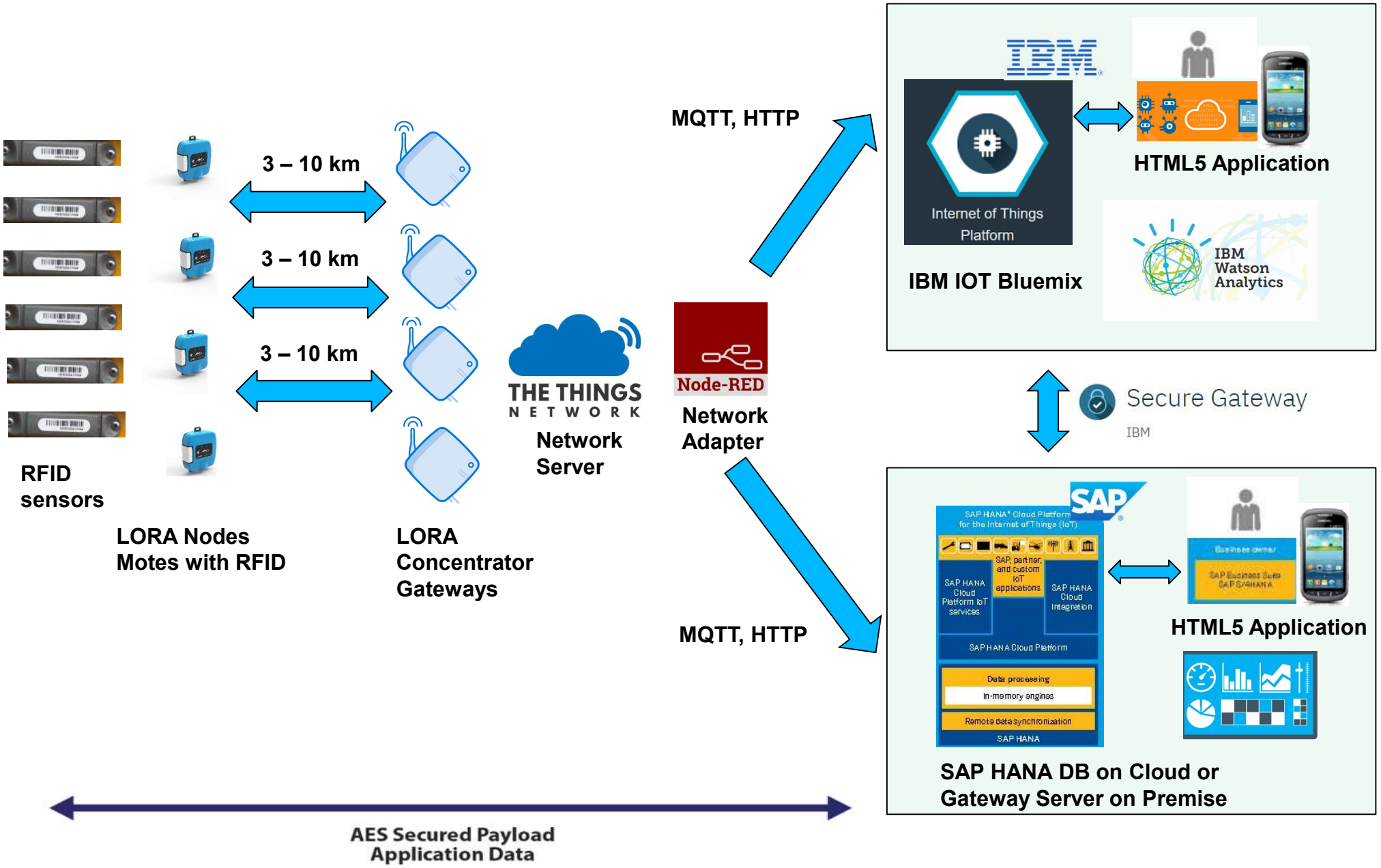
LoraWAN implementation for Waste Management

- Applicability:
  - Alerts comunication
  - Status comunication
- Potential applications:
  - GPS and tag reading
  - Massive transfer events sending
  - Configurations sending to sensors/bin caps
- Not applicable for:
  - Bin caps black list sending
  - Sensor firmware update (OTA)

Proof Of Concept

- LoRaWAN™
- POC Architecture
- LoRa-RFID Devices
- The Things Network
- Web Service SAP oDATA
- Node-Red sketch
- SAP backend data reception
- IBM IoT integration

POC Architecture – Sending data either do IBM Bluemix or to SAP HANA Database



Proof Of Concept

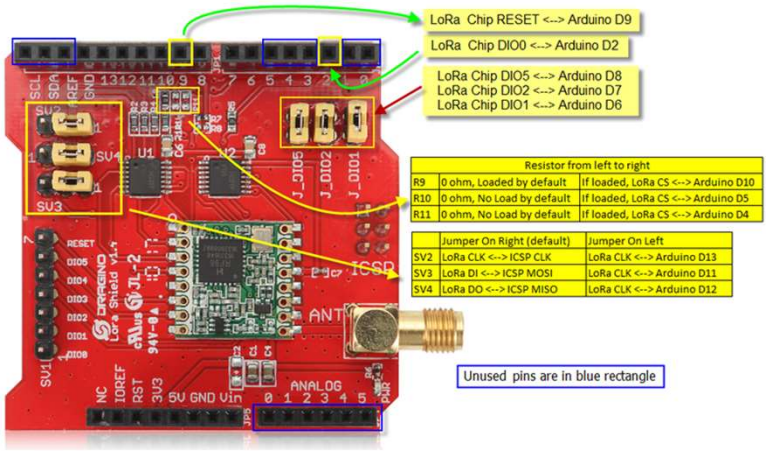
- LoRaWAN™
- POC Architecture
- LoRa-RFID Devices
- The Things Network
- Web Service SAP oDATA
- Node-Red sketch
- SAP backend data reception
- IBM IoT integration

RFID sensors used for the POC

TAG RFID

RFID reader
RC522

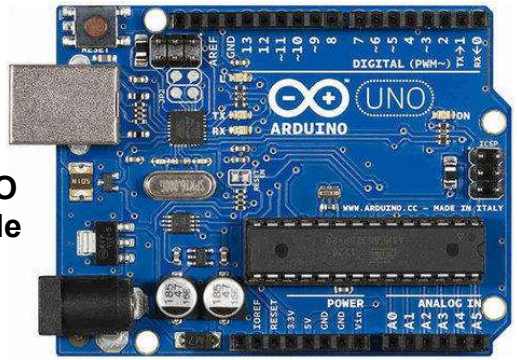
Pin Mapping For LoRa



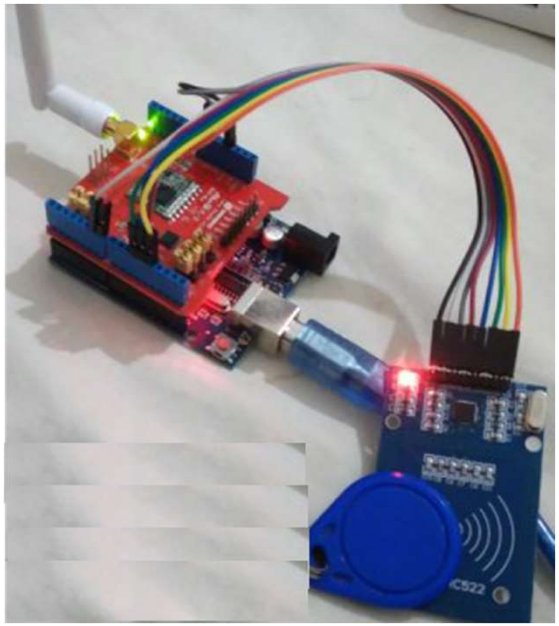
Resistor from left to right		
R9	0 ohm, Loaded by default	If loaded, LoRa CS <--> Arduino D10
R10	0 ohm, No Load by default	If loaded, LoRa CS <--> Arduino D5
R11	0 ohm, No Load by default	If loaded, LoRa CS <--> Arduino D4

	Jumper On Right (default)	Jumper On Left
SV2	LoRa CLK <--> ICSP CLK	LoRa CLK <--> Arduino D13
SV3	LoRa DI <--> ICSP MOSI	LoRa CLK <--> Arduino D11
SV4	LoRa DO <--> ICSP MISO	LoRa CLK <--> Arduino D12

Lora Shield with
SX1278 868Mhz
module

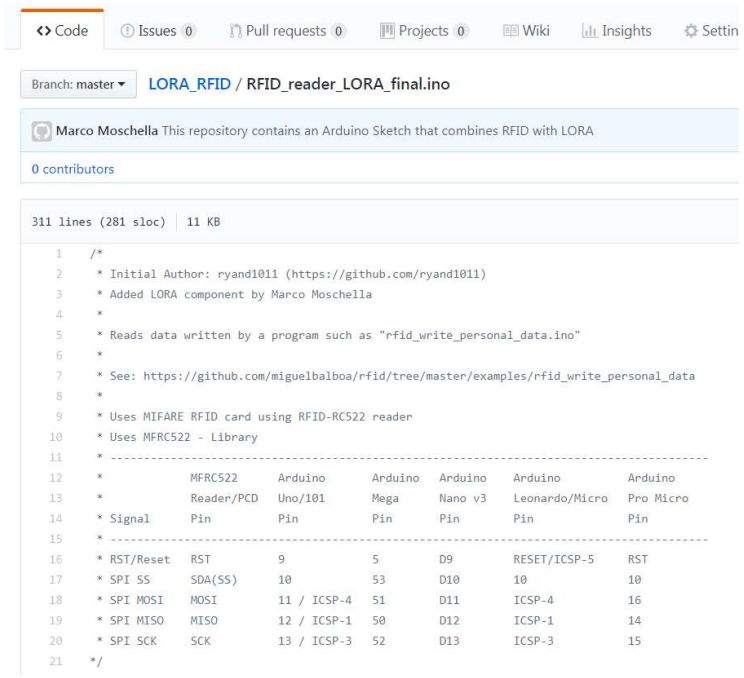


Arduino UNO
Basic module



LORA-RFID final reader
(Mote)

LORA + RFID control firmware



Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settin

Branch: master ▾ LORA_RFID / RFID_reader_LORA_final.ino

Marco Moschella This repository contains an Arduino Sketch that combines RFID with LORA

0 contributors

311 lines (281 sloc) | 11 KB

```
/*
 * Initial Author: ryand1011 (https://github.com/ryand1011)
 * Added LORA component by Marco Moschella
 *
 * Reads data written by a program such as "rfid_write_personal_data.ino"
 *
 * See: https://github.com/miguelbalboa/rfid/tree/master/examples/rfid_write_personal_data
 *
 * Uses MIFARE RFID card using RFID-RC522 reader
 * Uses MFRC522 - Library
 * -----------------------------------------------------------------------------------------
 *             MFRC522      Arduino       Arduino   Arduino    Arduino          Arduino
 *             Reader/PCD   Uno/101       Mega      Nano v3    Leonardo/Micro   Pro Micro
 * Signal      Pin          Pin           Pin       Pin        Pin              Pin
 * -----------------------------------------------------------------------------------------
 * RST/Reset   RST          9             5         D9         RESET/ICSP-5     RST
 * SPI SS      SDA(SS)      10            53        D10        10               10
 * SPI MOSI    MOSI         11 / ICSP-4   51        D11        ICSP-4           16
 * SPI MISO    MISO         12 / ICSP-1   50        D12        ICSP-1           14
 * SPI SCK     SCK          13 / ICSP-3   52        D13        ICSP-3           15
 */
```

The control firmware works with these libraries:

- RFID reader library
- LMIC radiofrequency library
- Serial port SPI control library

The following parameters are manually set in the firmware (Activation by Personalization, i.e. ABP):

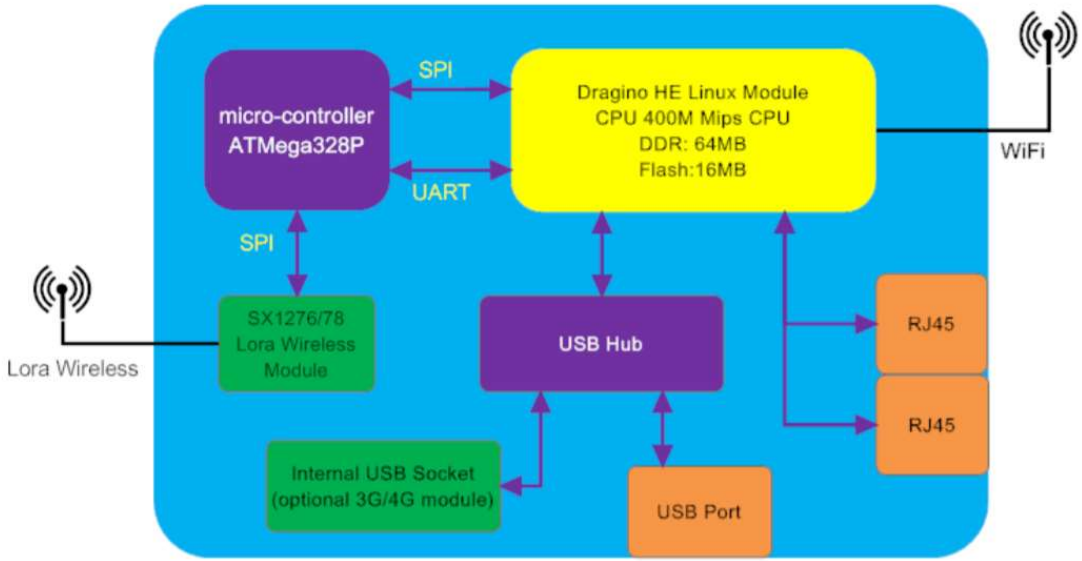
- NWKSKEY[16] = server key
- APPSKEY[16] = application key
- DEVADDR[8] = device ID

These keys are used in order to send the message to the Network Server by LORA Gateway, which is at long distance from the device and is connected to the network (internet).

https://github.com/landsat7/LORA_RFID/blob/master/RFID_reader_LORA_final.ino

Gateway type 1: LORA with DRAGINO LG01

LG01 System Overview:

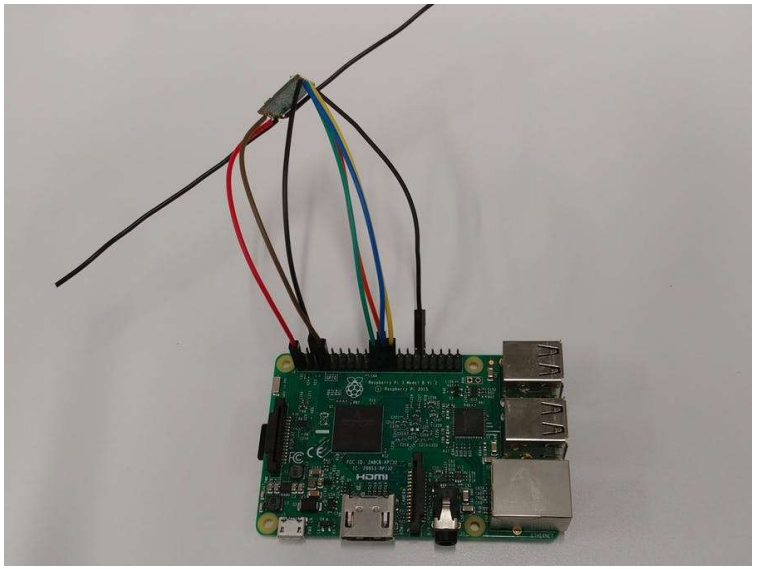


We used a DRAGINO LG01 Gateway that allows to:

- receive radiofrequency signlals from the sensors
- send them over the internet by a Wireless or RJ45 connection near the Gateway .

An ATMega328P processor is integrated in the Gateway, a SinglePacketForward firmaware is installed on this processor by a connection with Arduino Software.

Gateway type 2: LORA with Rasberry Pi model 3B and RFM95W transceiver

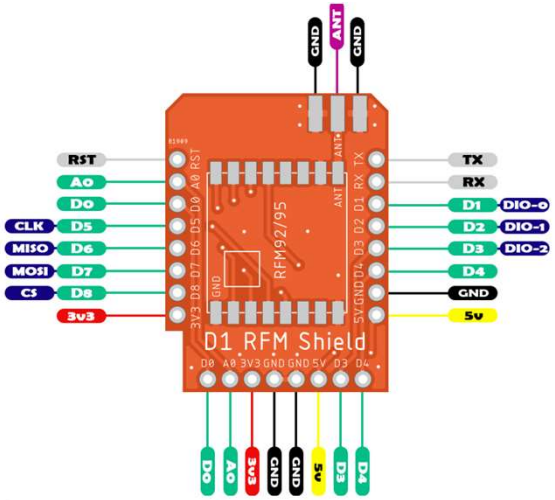


We used a Rasberry Pi Model 3B connected to a SX1278 SX1276 Lora Wireless Transceiver, 100mW RFM95W

Download the Single Packet Channel Forward software from the following website:
https://github.com/tftelkamp/single_chan_pkt_fwd
Then install the software using the «make» function and run it on the RaspBerry PI as follows.

```
pi@raspberrypi: ~/lorawan_gateway/single_chan_pkt_fwd
File  Edit  Tabs  Help
pi@raspberrypi:~/lorawan_gateway/single_chan_pkt_fwd $ ls
base64.c  base64.o  main.cpp  Makefile    single_chan_pkt_fwd
base64.h  LICENSE   main.o    README.md
pi@raspberrypi:~/lorawan_gateway/single_chan_pkt_fwd $ ./single_chan_pkt_fwd
SX1276 detected, starting.
Gateway ID: b8:27:eb:ff:ff:52:05:34
Listening at SF7 on 868.100000 Mhz.
------------------
stat update: {"stat":{"time":"2019-12-28 13:58:15 GMT","lati":41.91512,"long":12
.44185,"alti":100,"rxnb":0,"rxok":0,"rxfw":0,"ackr":0.0,"dwnb":0,"txnb":0,"pfrm"
:"Single Channel Gateway","mail":"marco.moschella@my-sap.it","desc":"RasPi3B wit
h RMF95"}}
```

Gateway type 3: LORA with Wemos D1 and RFM95W transceiver



We used a Wemos D1 connected to D1 RFM shield (that includes SX1276 Lora Wireless Transceiver, 100mW RFM95W).

Download the Single Packet Channel Forward software from the following website: https://diycon.nl/archives/1911

ESP-1ch-Gateway
-v5.0-master

diycon

Home Shop News My Account

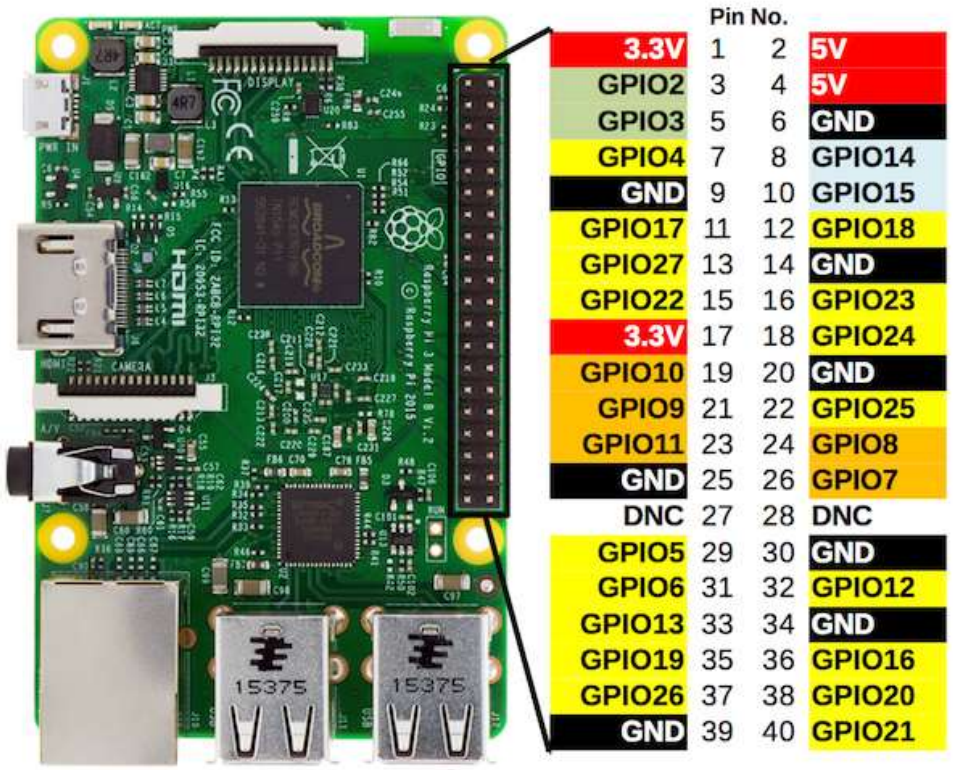
Costs

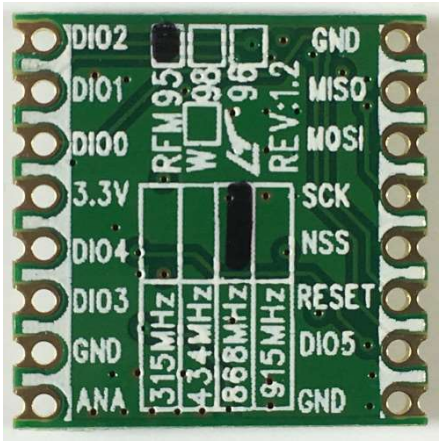
Thank you. Your order has been received.

Order details

PRODUCT	TOTAL
LoRa GateWay PCB 301 Shield Only for RFM92/RFM95 Wemos D1 × 4	€15,80
SUBTOTAL:	€15,80
SHIPPING:	€3,00 via Without Tracking
PAYMENT METHOD:	PayPal
TOTAL:	€18,80

Gateway LORA with Rasberry Pi model 3B and RFM95W transceiver – connection details





Pino Módulo LoRa	Pino Arduino
SCK	D13
MISO	D12
MOSI	D11
NSS	D10
(DI0)	D4
NRESET	D0

Rpi pin	Rpi descr.	RFM95W
1	3.3 V	3.3 V
6	GND	GND
7	GPIO4	DI00
11	GPIO17	RESET
19	GPIO10	MOSI
21	GPIO09	MISO
22	GPIO25	NSS
23	GPIO11	SCK
		ANA (antenna 8,6cm)
		GND (antenna 8,6cm)

Proof Of Concept

- LoRaWAN™
- POC Architecture
- LoRa-RFID Devices
- The Things Network
- Web Service SAP oDATA
- Node-Red sketch
- SAP backend data reception
- IBM IoT integration

The Things Network Manifest

Everything that carries power will be connected to Internet eventually.

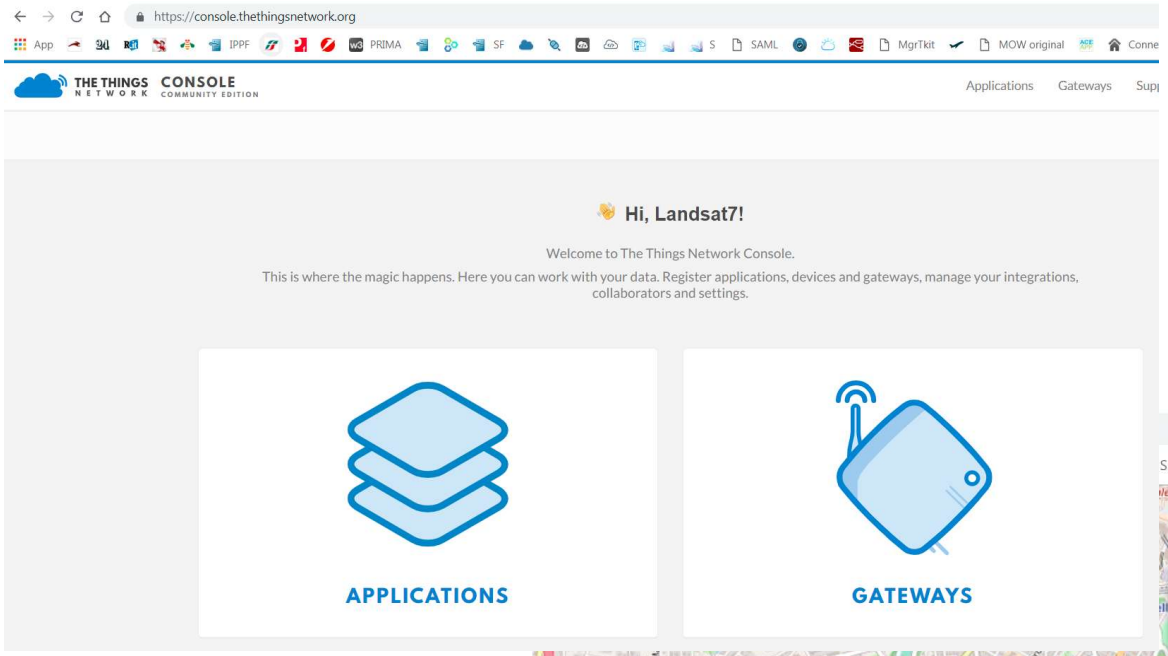
Controlling the network that makes this possible means controlling the world. ***We believe that this power should not be restricted to a few people, companies or nations. Instead this should be distributed over as many people as possible without the possibility to be taken away by anyone.*** We therefore founded "The Things Network".

The Things Network is an open source, free initiative with the following properties:

- It connects sensors and actuators, called "Things", with transceivers called "Things Gateways" to servers called "Things Access".
- The first connection is "Over The Air", the second is "Over The Net". The distributed implementation of these concepts is called "The Things Network".
- Anyone shall be free to set up "Things" and connect to "Things Gateways" that may or may not be their own.
- Anyone shall be free to set up "Things Gateways" and connect to "Things Access" that may or may not be their own. Their "Things Gateways" will give access to all "Things" in a net neutral manner, limited by the maximum available capacity alone.
- Anyone shall be free to set up "Things Access" and allow anonymous connections from the Internet. Their "Things Access" will give access to all "Things Gateways" in a net neutral manner, limited by the maximum available capacity alone. Furthermore their "Things Access" will allow connection of other "Things Access" servers for the distribution of data.
- The "Over The Air" and "Over The Net" networks shall be protocol agnostic, as long as these protocols are not proprietary, open source and free of rights.
- Anyone who perpetrates a "Things Access" or a "Things Gateway" will do so free of charge for all connecting devices and servers.
- Anyone making use of the network is allowed to do so for any reason or cause, possibly limited by local law, fully at own risk and realizing that services are provided "as is" and may be terminated for any reason at any moment. The use may be open for anybody, limited to customers, commercial, not-for-profit, or in any other fashion. "The Things Network" providers will not pose restrictions upon its users.

We invite you to sign this Manifesto, and uphold its principles to the best of your abilities.

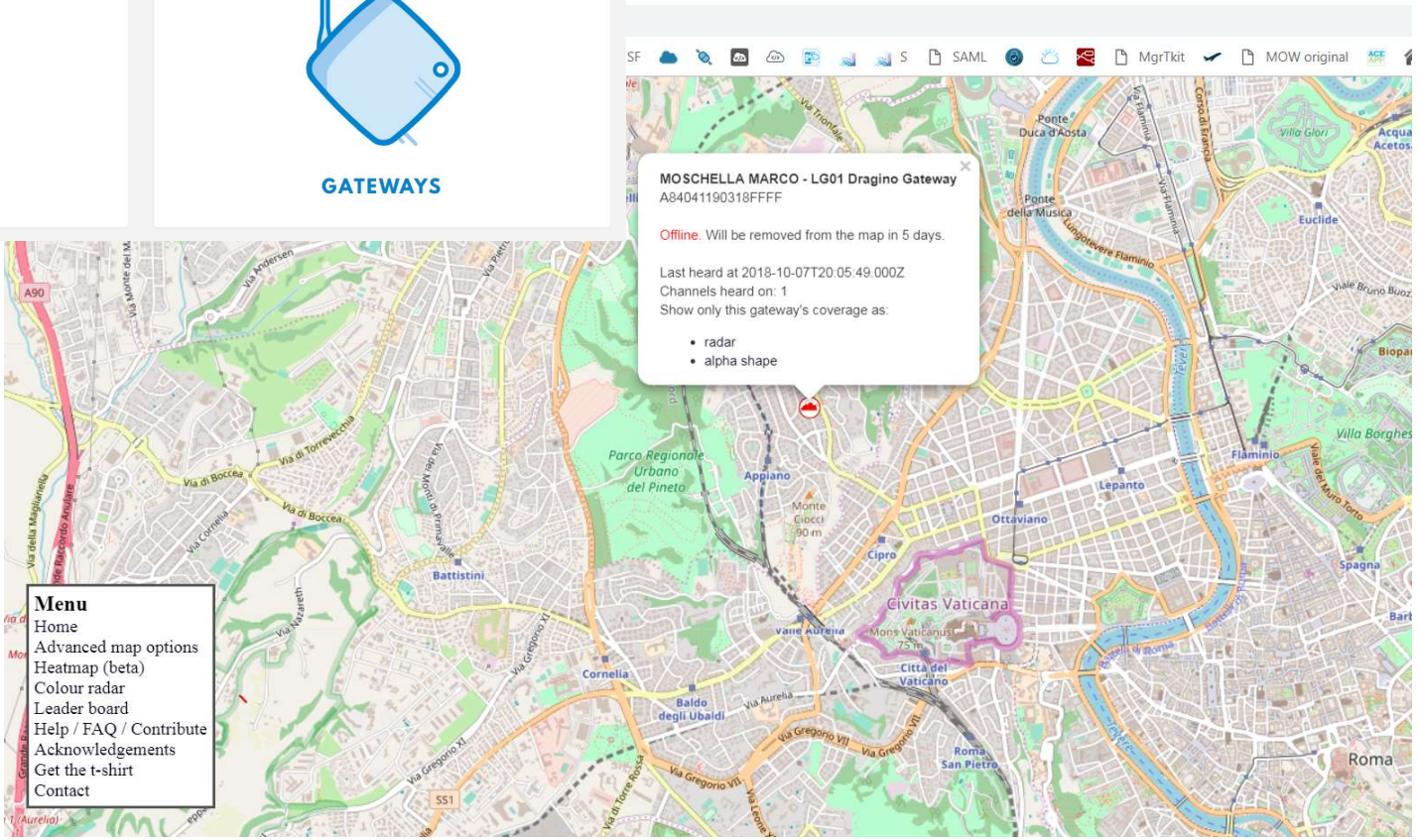
Signal reception on TheThingsNetwork



The LORA Gateway could be logged on TTN and it is shown on the map.

Data traffic addressed by the Gateway to every visible sensor is shown on TTN.

Each Device Address could be logged on TTN so that we can receive data traffic when data are sent to the nearby Gateway.



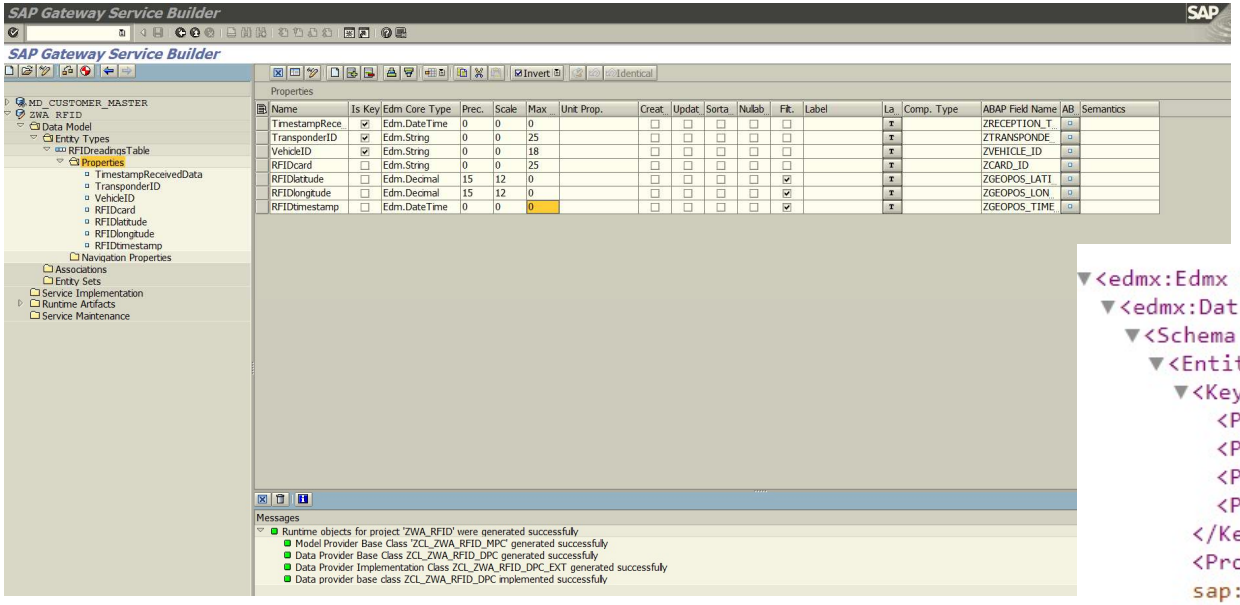
TTN payload decoding function for GPS node and RFID node

```
function Decoder(bytes, port) {
  // Decode an uplink message from a buffer
  // (array) of bytes to an object of fields.
  var decoded = {};
  // if (port === 1) decoded.led = bytes[0];
  decoded.latitude = ((bytes[0]<<16)>>>0) + ((bytes[1]<<8)>>>0) + bytes[2];
  decoded.latitude = (decoded.latitude / 16777215.0 * 180) - 90;
  decoded.longitude = ((bytes[3]<<16)>>>0) + ((bytes[4]<<8)>>>0) + bytes[5];
  decoded.longitude = (decoded.longitude / 16777215.0 * 360) - 180;
  var altValue = ((bytes[6]<<8)>>>0) + bytes[7];
  var sign = bytes[6] & (1 << 7);
  if(sign)
  {
    decoded.altitude = 0xFFFF0000 | altValue;
  }
  else
  {
    decoded.altitude = altValue;
  }
  decoded.hdop = bytes[8] / 10.0;
  return decoded;
}
```

```
function Decoder(bytes, port) {
  // Decode plain text; for testing only
  return {
      RFIDreading: String.fromCharCode.apply(null, bytes)
  };
}
```

oDATA SAP service Zwa_rfid_srv service definition

http://sapdemo/sap/opu/odata/sap/ZWA_RFID_SRV/$metadata




```
<edmx:Edmx xmlns:edmx="http://schemas.microsoft.com/ado/20
  <edmx:DataServices m:DataServiceVersion="2.0">
    <Schema xmlns="http://schemas.microsoft.com/ado/2008/09
      <EntityType Name="RFIDreadings" sap:content-version="
        <Key>
          <PropertyRef Name="TimestampReceivedData"/>
          <PropertyRef Name="TransponderID"/>
          <PropertyRef Name="VehicleID"/>
          <PropertyRef Name="RFIDcard"/>
        </Key>
        <Property Name="TimestampReceivedData" Type="Edm.Dat
        sap:filterable="false"/>
        <Property Name="TransponderID" Type="Edm.String" Nul
        sap:filterable="false"/>
        <Property Name="VehicleID" Type="Edm.String" Nullab
        sap:filterable="false"/>
        <Property Name="RFIDcard" Type="Edm.String" Nullable
        sap:filterable="false"/>
        <Property Name="RFIDlatitude" Type="Edm.Decimal" Nul
        <Property Name="RFIDlongitude" Type="Edm.Decimal" Nu
        <Property Name="RFIDtimestamp" Type="Edm.DateTime" N
      </EntityType>
      <EntityContainer Name="ZWA_RFID_SRV_Entities" m:IsDef
        <EntitySet Name="RFIDreadings" EntityType="ZWA_RFID_
        filter="true" sap:content-version="1"/>
      </EntityContainer>
      <atom:link xmlns:atom="http://www.w3.org/2005/Atom" r
      <atom:link xmlns:atom="http://www.w3.org/2005/Atom" r
    </Schema>
  </edmx:DataServices>
</edmx:Edmx>
```

Proof Of Concept

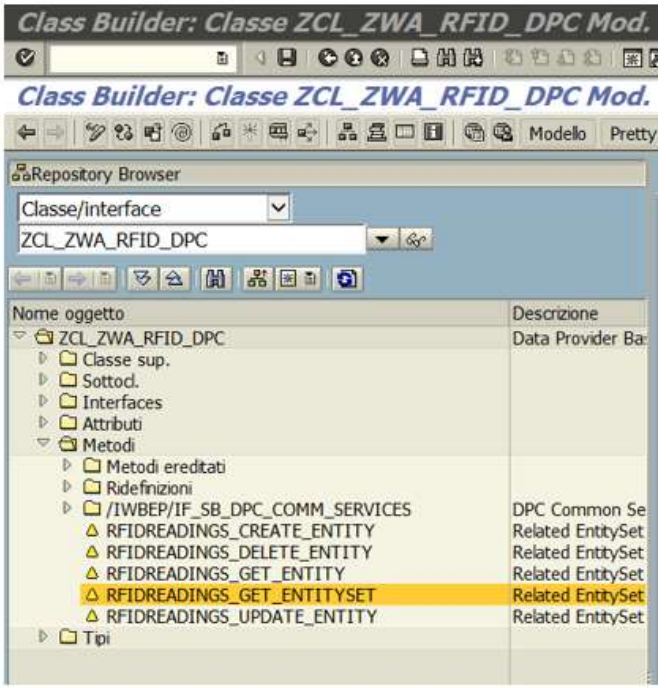
- LoRaWAN™
- POC Architecture
- LoRa-RFID Devices
- The Things Network
- Web Service SAP oDATA
- Node-Red sketch
- SAP backend data reception
- IBM IoT integration

oDATA SAP service GET & POST process and tables setting

Classe ZCL_ZWA_RFID_DPC

```
method RFIDREADINGS_CREATE_ENTITY.
*  RAISE EXCEPTION TYPE /iwbep/cx_mgw_not_impl_exc
*    EXPORTING
*      textid = /iwbep/cx_mgw_not_impl_exc=>method_not_implemented
*      method = 'RFIDREADINGS_CREATE_ENTITY'.
DATA: ls_zwa_rfid TYPE zwatb_rfid.
io_data_provider->read_entry_data( IMPORTING es_data = er_entity ).
MOVE-CORRESPONDING er_entity TO ls_zwa_rfid.
INSERT zwatb_rfid FROM ls_zwa_rfid.
endmethod.
```

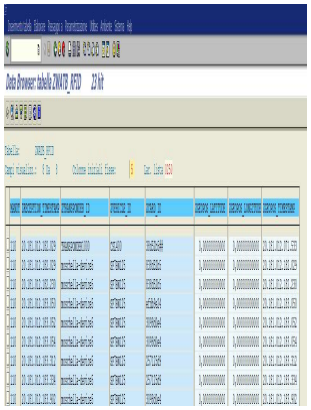
```
method RFIDREADINGS_GET_ENTITYSET.
*  RAISE EXCEPTION TYPE /iwbep/cx_mgw_not_impl_exc
*    EXPORTING
*      textid = /iwbep/cx_mgw_not_impl_exc=>method_not_implemented
*      method = 'RFIDREADINGS_GET_ENTITYSET'.
select * from zwatb_rfid INTO CORRESPONDING FIELDS OF TABLE et_entityset.
endmethod.
```



Only few lines of code are needed to implement a simple Web Service with oData !

```
method DATICATASTALIHSE_GET_ENTITY.
DATA : ls_key_tab LIKE LINE OF it_key_tab,
       ls_zwa_catastalih type zwa_catastalih,
       lv_vstelle TYPE zwa_catastalih-vstelle.

READ TABLE it_key_tab INTO ls_key_tab
WITH KEY name = 'Vstelle'. " Case sensitive
IF sy-subrc EQ 0.
  lv_vstelle = ls_key_tab-value.
ENDIF.
select single * from zwa_catastalih INTO CORRESPONDING FIELDS OF er_entity
     where vstelle eq lv_vstelle.
endmethod.
```

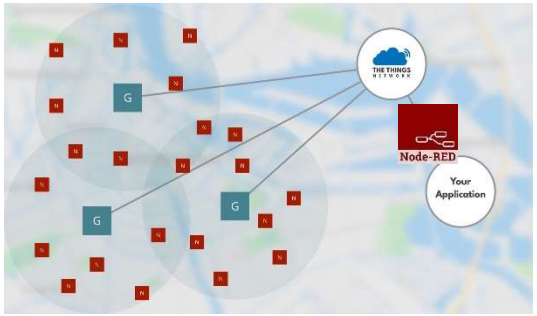


Data Table ZWATB_RFID

Proof Of Concept

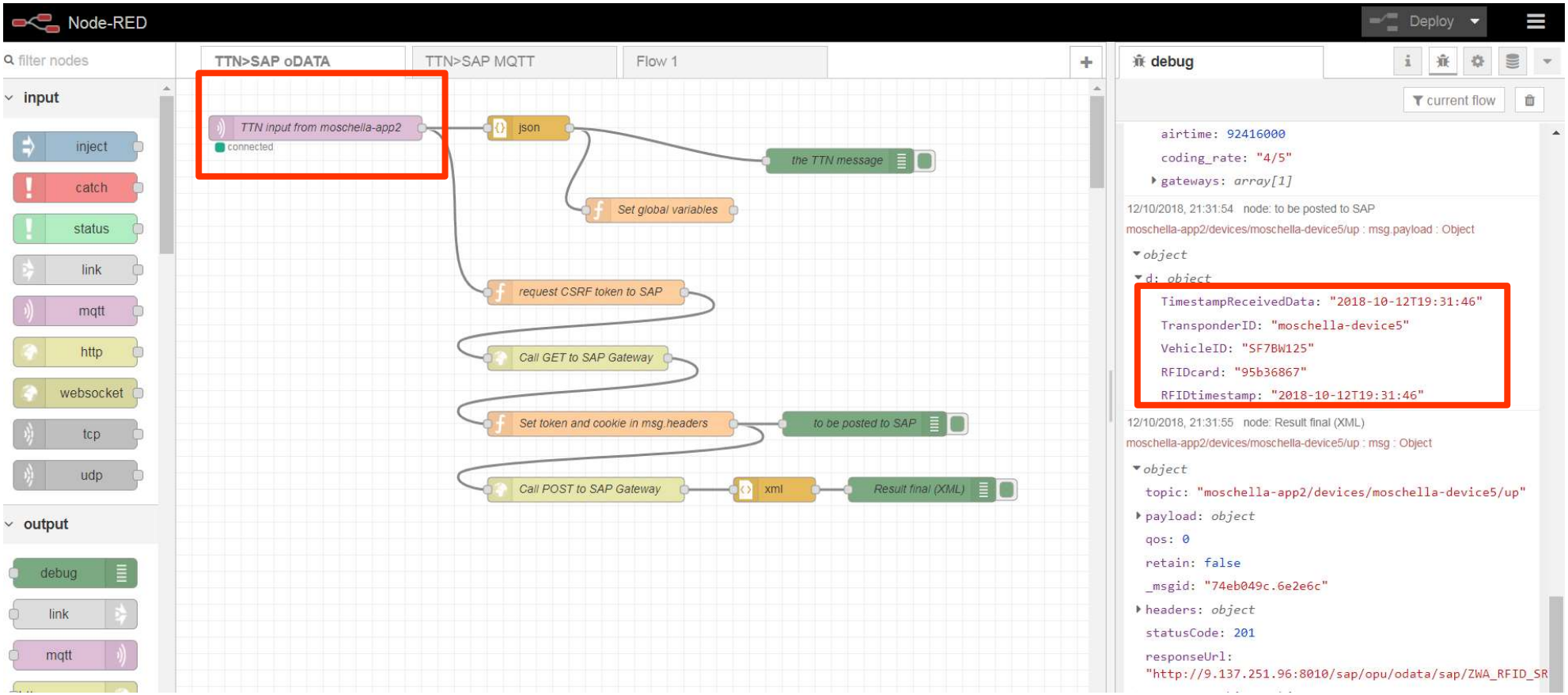
- LoRaWAN™
- POC Architecture
- LoRa-RFID Devices
- The Things Network
- Web Service SAP oDATA
- Node-Red sketch
- SAP backend data reception
- IBM IoT integration

Node-Red data mining from TTN Application Server and Get & Post call.



Node-Red adapter is easily configurable and enables:

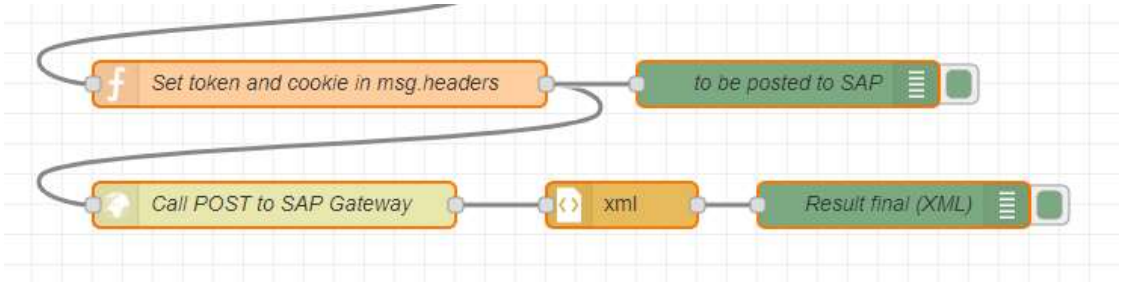
- instant check of sent and received data
- data conversion, when required.
- In order to post to SAP a CSRF and token exchange is needed upfront



Proof Of Concept

- LoRaWAN™
- POC Architecture
- LoRa-RFID Devices
- The Things Network
- Web Service SAP oDATA
- Node-Red sketch
- SAP backend data reception
- IBM IoT integration

oDATA SAP service execution results



12/10/2018, 21:31:54 node: to be posted to SAP

moschella-app2/devices/moschella-device5/up : msg.payload : Object


```
▾ object
▾ d: object
    TimestampReceivedData: "2018-10-12T19:31:46"
    TransponderID: "moschella-device5"
    VehicleID: "SF7BW125"
    RFIDcard: "95b36867"
    RFIDtimestamp: "2018-10-12T19:31:46"
```


Inserimento tabella Elaborare Passaggio a Parametrizzazione Utilities Ambiente Sistema Help

Data Browser: tabella ZWATB_RFID 23 hit

Tabella: ZWATB_RFID
Campi visualizz.: 8 Da 8 Colonne iniziali fisse: 5 Lar. lista 0250

MANDT	ZRECEPTION_TIMESTAMP	ZTRANSPONDER_ID	ZVEHICLE_ID	ZCARD_ID	ZGEOPOS_LATITUDE	ZGEOPOS_LONGITUDE	ZGEOPOS_TIMESTAMP
110	20.181.012.182.029	TRANSPONDER1000	CSL400	38d52b5##	0,000000000000	0,000000000000	20.181.012.071.533
110	20.181.012.182.029	moschella-device5	SF7BW125	838d52b5	0,000000000000	0,000000000000	20.181.012.182.029
110	20.181.012.182.230	moschella-device5	SF7BW125	838d52b5	0,000000000000	0,000000000000	20.181.012.182.230
110	20.181.012.183.050	moschella-device5	SF7BW125	efdb9e24	0,000000000000	0,000000000000	20.181.012.183.050
110	20.181.012.183.052	moschella-device5	SF7BW125	3098d0e4	0,000000000000	0,000000000000	20.181.012.183.052
110	20.181.012.183.054	moschella-device5	SF7BW125	3098d0e4	0,000000000000	0,000000000000	20.181.012.183.054
110	20.181.012.183.312	moschella-device5	SF7BW125	257125d9	0,000000000000	0,000000000000	20.181.012.183.312
110	20.181.012.183.334	moschella-device5	SF7BW125	257125d9	0,000000000000	0,000000000000	20.181.012.183.334
110	20.181.012.183.902	moschella-device5	SF7BW125	3098d0e4	0,000000000000	0,000000000000	20.181.012.183.902
110	20.181.012.183.952	moschella-device5	SF7BW125	257125d9	0,000000000000	0,000000000000	20.181.012.183.952
110	20.181.012.191.619	moschella-device5	SF7BW125	95b36867	0,000000000000	0,000000000000	20.181.012.191.619
110	20.181.012.191.625	moschella-device5	SF7BW125	95b36867	0,000000000000	0,000000000000	20.181.012.191.625
110	20.181.012.191.711	moschella-device5	SF7BW125	3098d0e4	0,000000000000	0,000000000000	20.181.012.191.711
110	20.181.012.191.814	moschella-device5	SF7BW125	95b36867	0,000000000000	0,000000000000	20.181.012.191.814
110	20.181.012.192.244	moschella-device5	SF7BW125	95b36867	0,000000000000	0,000000000000	20.181.012.192.244
110	20.181.012.192.302	moschella-device5	SF7BW125	3098d0e4	0,000000000000	0,000000000000	20.181.012.192.302
110	20.181.012.192.441	moschella-device5	SF7BW125	3098d0e4	0,000000000000	0,000000000000	20.181.012.192.441
110	20.181.012.192.446	moschella-device5	SF7BW125	95b36867	0,000000000000	0,000000000000	20.181.012.192.446
110	20.181.012.192.456	moschella-device5	SF7BW125	3098d0e4	0,000000000000	0,000000000000	20.181.012.192.456
110	20.181.012.192.504	moschella-device5	SF7BW125	95b36867	0,000000000000	0,000000000000	20.181.012.192.504
110	20.181.012.192.533	moschella-device5	SF7BW125	3098d0e4	0,000000000000	0,000000000000	20.181.012.192.533
110	20.181.012.192.547	moschella-device5	SF7BW125	95b36867	0,000000000000	0,000000000000	20.181.012.192.547
110	20.181.012.193.146	moschella-device5	SF7BW125	95b36867	0,000000000000	0,000000000000	20.181.012.193.146

Proof Of Concept

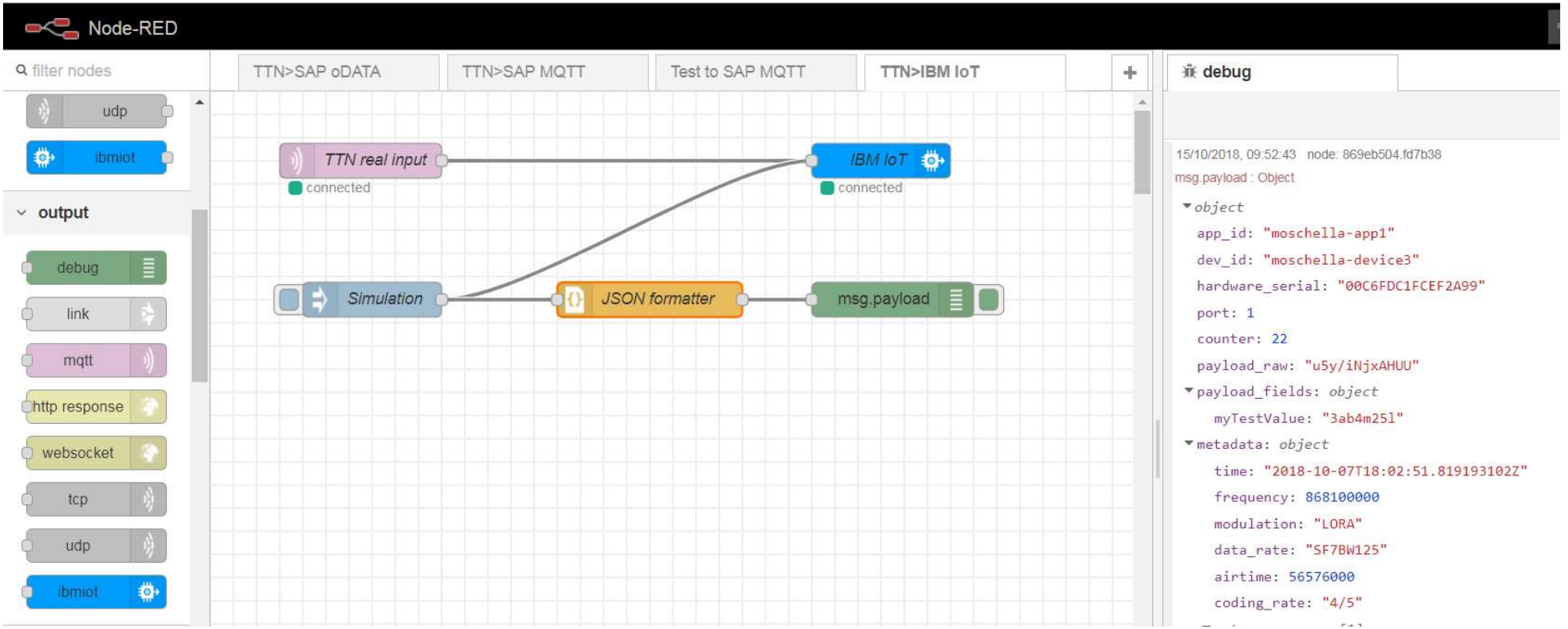
- LoRaWAN™
- POC Architecture
- LoRa-RFID Devices
- The Things Network
- Web Service SAP oDATA
- Node-Red sketch
- SAP backend data reception
- IBM IoT integration

API key generation for IBM IoT secured communication

Each sensor logged on IBM IoT has an authentication token so that we can monitor every login to the permitted applications on IBM IoT.

An API key will be generated to login on IBM IoT.

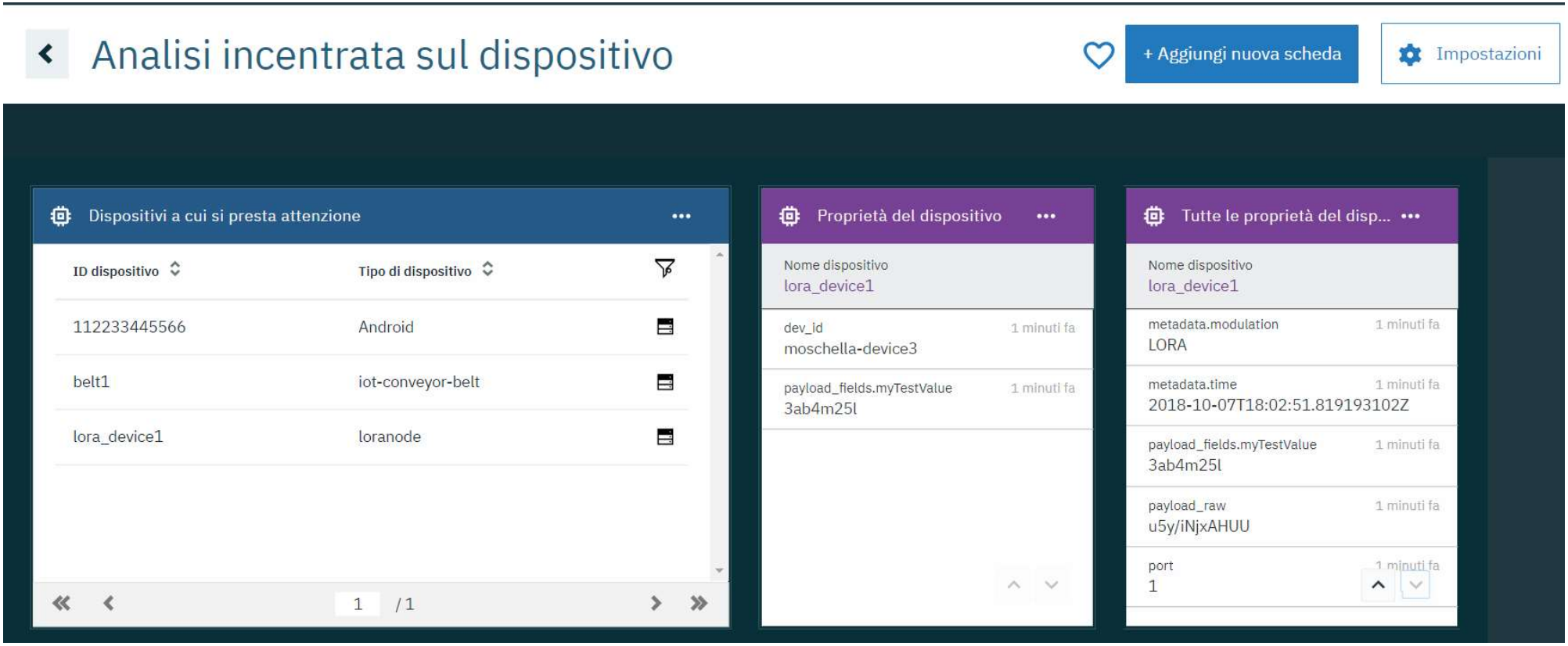
Node-red sketch for sensor data sending to IBM IOT Dashboard



IBM IoT nodes can be added to the node-red sketch.

It is easily possible to retrieve or send data from/to IBM IoT devices.

IBM IOT Device Analytics



Incoming data to IBM IoT are displayed immediately in the IoT Dashboard.

References

- https://www.resiot.io/it/cosa-e-lorawan/
- https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf
- https://www.thethingsnetwork.org/
- https://nodered.org/
- https://nayarweb.com/blog/2017/rfid-and-lora-on-arduino/
- http://blog.acorel.nl/2016/11/great-iot-opportunities-with-sap-and.html
- https://www.hackster.io/ChrisSamuelson/lora-raspberry-pi-single-channel-gateway-cheap-d57d36
- https://www.mobilefish.com/developer/lorawan/lorawan_quickguide_build_lora_node_rfm95_arduino_uno.html