100% OPEN SOURCE

EU GDPR COMPLIANT

PEN Test Passed

SNAP4CITY on EUROPEAN OPEN SCIENCE CLOUD MARKETPLACE

SNAP4 Appliances and Dockers Installations

SNAP4CITY

KM4CITY



Be smart in a SNAP!

LIVING LAB

Smart IOT Applications & IOT Networks



12 November 2020, Course
https://www.snap4city.org/577

SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT CITIES

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO | DISIT

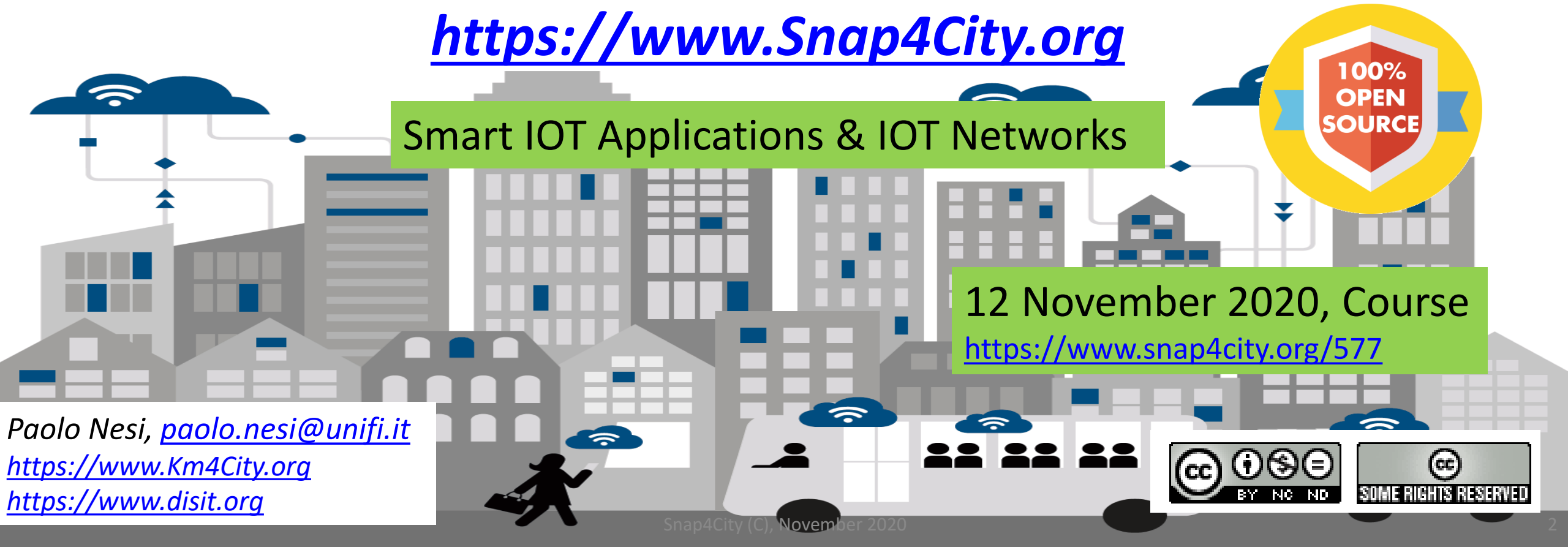
UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

SNAP4CITY

Powered by KM4CITY

scalable Smart aNalytic APplication builder for sentient Cities: for Living Lab and co-working with Stakeholders

https://www.Snap4City.org



Smart IOT Applications & IOT Networks

100% OPEN SOURCE

12 November 2020, Course
https://www.snap4city.org/577

Paolo Nesi, paolo.nesi@unifi.it
https://www.Km4City.org
https://www.disit.org

CC BY NC ND SOME RIGHTS RESERVED

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

SNAP4CITY
www.snap4city.org

KM4CITY

Powered by FIWARE

SNAP4 Appliances and Dockers Installations

FREE TRIAL

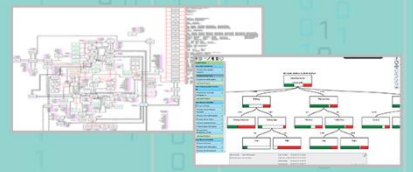
PEN Test Passed

EU GDPR COMPLIANT

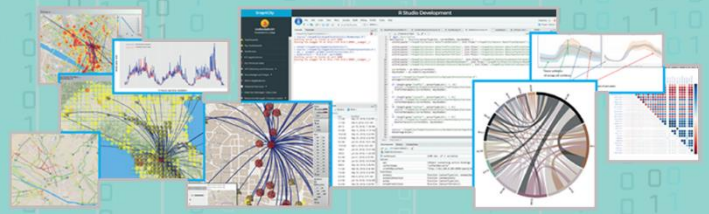
100% OPEN SOURCE



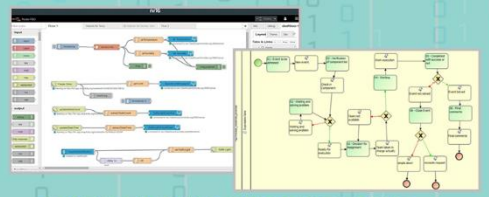
DASHBOARDS AND APPS - CONTROL ROOMS - DECISION SUPPORT SYSTEMS - WHAT-IF ANALYSIS



EXPERT SYSTEM
KNOWLEDGE BASE
STORAGE



BIG DATA ANALYTICS
ARTIFICIAL INTELLIGENCE
BUSINESS INTELLIGENCE
MACHINE LEARNING

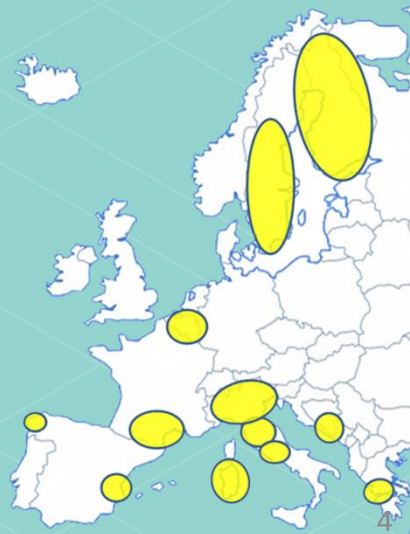


DATA FLOWS, WORKFLOWS
MICROSERVICES
MANAGEMENT

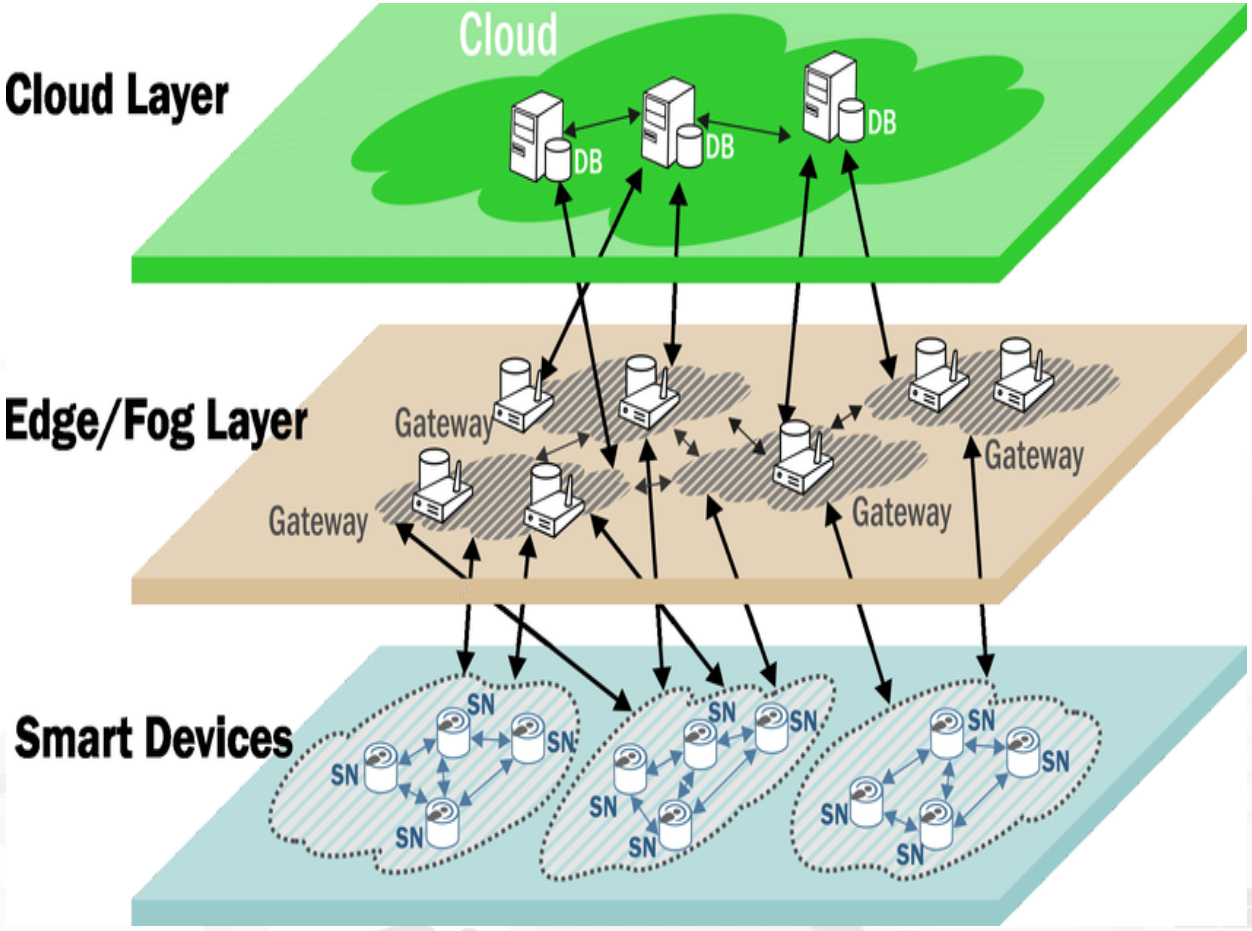


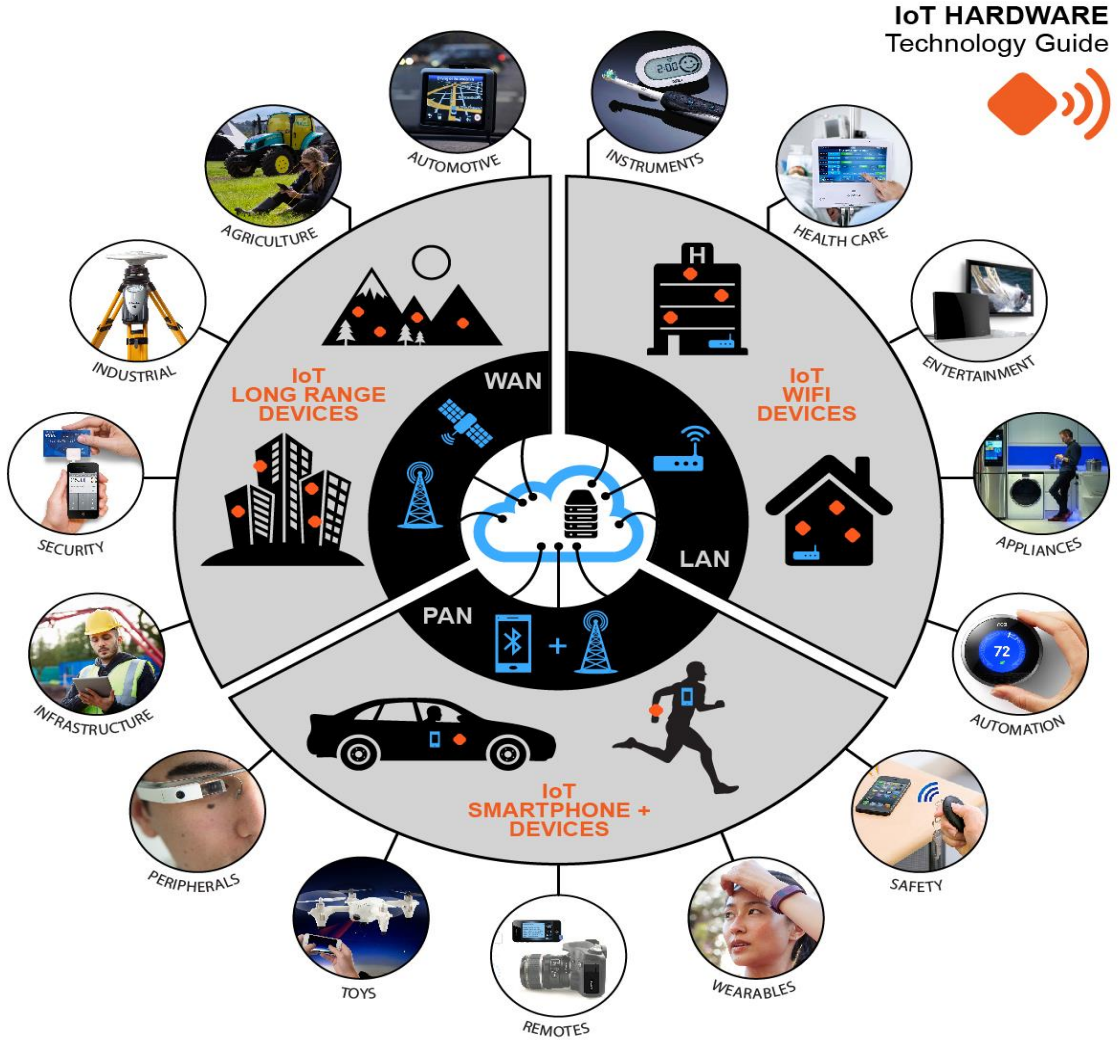
METHODOLOGIES
COURSES AND COMMUNITY
LIVING LABS
DEVELOPMENT TOOLS





Cloud vs Fog/Edge Computing





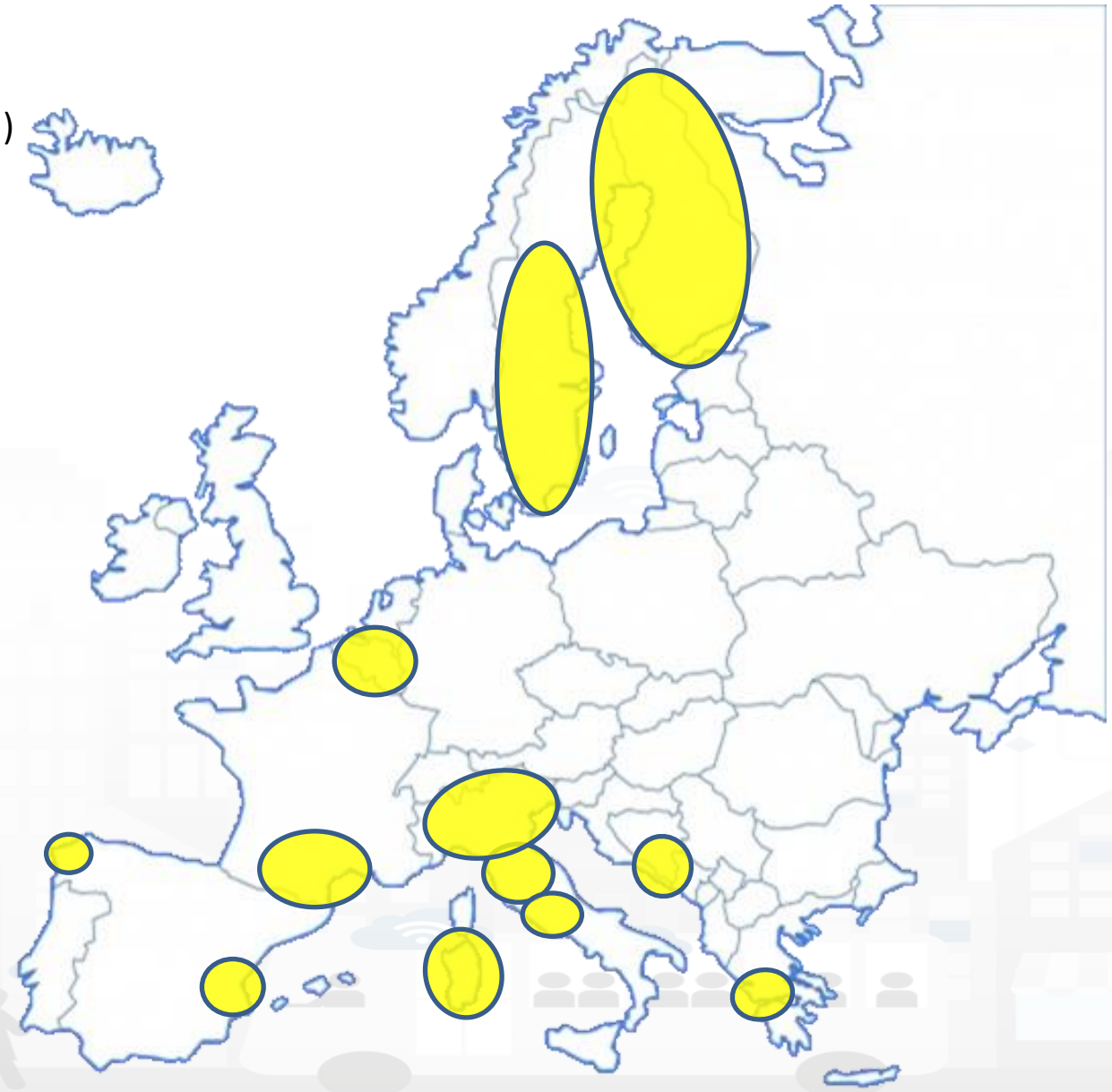
Snap4City/Industry structure

- The **Snap4xxxx** solution is released in Open Source, VM and Docker with fully support of MultiTenant/multiple-Organizations
  - Each Organization may be configured for a separate environment with a set of Maps, Menus, Users, Data, Dashboards, IOT Apps, MicroApplications, Custom Widgets, Models, resources, open data, etc.
- **Https://www.Snap4City.ORG** is the main instance of Snap4xxxx solution managed by DISIT Lab. The main documentation is located and updated on Snap4City.org, GitHUB, dockerHub and Node-Red Library. Snap4City.org is where the last tools are tested and news published.
  - Organizations on Snap4City.org have been created with contracts as for *Platform as a Service*, for testing and for providing *SmartCity as a Service* as well as *Industry 4.0 as a Service*

Coverage 2020

Main Organizations/areas

- Antwerp area (Be)
- Capelon (Sweden: Västerås, Eskilstuna, Karlstad)
- DISIT demo (multiple)
- Dubrovnik, Croatia
- Firenze area (I)
- Garda Lake area (I)
- Helsinki area (Fin)
- Livorno area (I)
- Lonato del Garda (I)
- Modena (I)
- Mostar, Bosnia-Herzegovina
- Pisa area (I)
- Pont du Gard, Occitanie (Fr)
- Roma (I)
- Santiago de Compostela (S)
- Sardegna Region (I)
- SmartBed (multiple)
- Toscana Region (I), SM
- Valencia (S)
- Venezia area (I)
- WestGreece area (Gr)



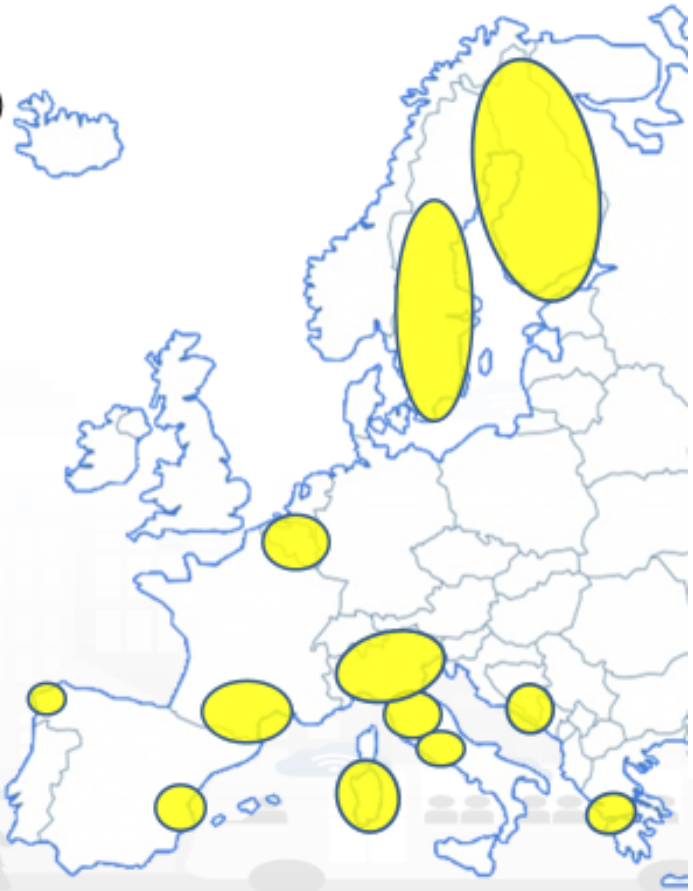
Snap4City/Industry Community

- Most of Organizations on Snap4City.org also correspond to companies or institutions that have an installation of Snap4City tools on their Premise,
  - such as: Pisa, SmartGarda Lake, Snap4, ALTAIR, etc.
- This double way allows them to:
  - test the news,
  - share experiences with other groups,
  - get visibility,
  - work in the collaborative environment, and
  - be better supported by Snap4City.org and DISIT Lab personnel.
- Each instance of Snap4xxxx solution ***can decide to join the federation*** of Smart City API to exploit shared data.
  - This allows to exploit regional data for city installations applications (web, mobile, dashboards, etc.) without reloading them for example.

Main Organizations/areas

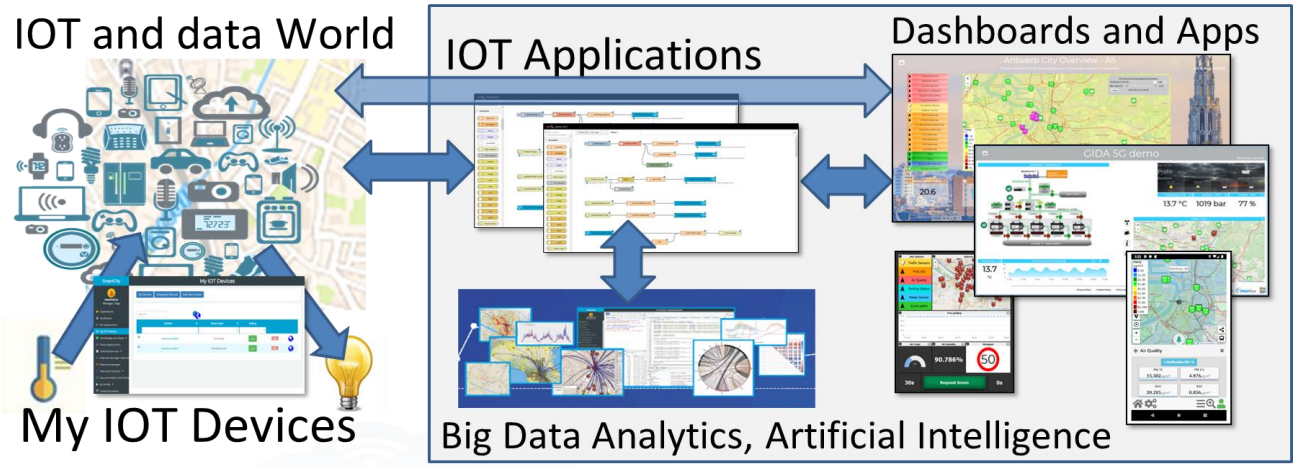
- Antwerp area (Be)
- Capelon (Sweden: Västerås, Eskilstuna, Karlstad)
- DISIT demo (multiple)
- Dubrovnik, Croatia
- Firenze area (I)
- Garda Lake area (I)
- Helsinki area (Fin)
- Livorno area (I)
- Lonato del Garda (I)
- Modena (I)
- Mostar, Bosnia-Herzegovina
- Pisa area (I)
- Pont du Gard, Occitanie (Fr)
- Roma (I)
- Santiago de Compostela (S)
- Sardegna Region (I)
- SmartBed (multiple)
- Toscana Region (I), SM
- Valencia (S)
- Venezia area (I)
- WestGreece area (Gr)

Snap4City (C), October 2020



Free Trial

- Register on WWW.snap4city.org
  - Subscribe on **DISIT Organization**
- **You can:**
  - Access on basic Tools
  - Access to a large volume of Data
  - Create Dashboards
  - Create IOT Applications
  - Connect your IOT Devices
  - Exploit Tutorials and Demonstrations



IF you need to go more in deep you can ask us to pass at the next Role becoming full AreaManager with full rights of development, also for Data Analytics, machine learning, etc.

https://www.snap4city.org/577

On Line Training Material (free of charge)

	1st part (*)	2nd part (*)	3rd part (*)	4th part (*)	5th part (*)	6th part (*)	7th part (*)
what	General	Dashboards	IOT App, IOT Network	Data Analytics	Data Ingestion processes	System and Deploy Install	Smart City API: Web & Mob. App
PDF							
Inter active							
Video1							
Video2							
Video3							
Video4				none		none	none
duration	2:55	3:16	3:41	2:00	2:48	2:35	1:47

General Overview of the full Course

- ***1st part:*** *General Overview*
- ***2nd part:*** *Dashboards Creation and Management*
- ***3rd part:*** *IOT Applications development, IOT Devices, IOT Networks*
- ***4th part:*** *Data Analytics, in R Studio, in Python, how to Exploit and Manage Data Analytics in IOT Applications*
- ***5th part:*** *Data Ingestion, Data Warehouse, Data Gate, IOT Device Data ingestion, IOT App for Data Ingestion, etc.*
- ***6th part:*** *Snap4City Architecture, How To Install and Manage Snap4City*
- ***7th part****: Smart city API (internal and external) Web and Mobile App development tool kit*

A number of the training sections include exercitations

Updated versions on: https://www.snap4city.org/577

See also courses in ITALIANO: https://www.snap4city.org/485

this part Agenda

GO
- **IOT App Smartening Dashboards and Solutions**
  - IOT App Smartening Devices and Dashboards, Smart Parking, IOT App vs Smart City Solutions
  - IOT App Smart Industry 4.0 Snap4Industry, ModBus Integration
  - IOT App vs Smart Home Snap4Home, Moving IOT Devices / Sensors, Tracking Devices

GO
- **Managing IOT Applications and Containers all**
  - Remote control of IOT Applications on IOT edge devices, and Containers all

GO
- **Creating IOT Applications with Node-RED**
  - Node-RED Hello World

GO
- **IOT App = Node-RED + Snap4City**
  - IOT App = Node-RED + Snap4City solution, demo, Synoptics, NGSI
  - IOT App self training, IOT App examples, IOT App and Dashboard Integration, dynamic widgets, IOT App vs Synoptics in real time data driven

GO
- **Data Analytics and web Scraping**
  - IOT App vs Data Analytics: R-Studio, Python; IOT App and web scraping,

GO
- **IOT App = Node-RED + Snap4City examples**
  - Demo + exercises, Data processing with IOT Applications

GO
- **Integration of External Services into IOT Applications**
  - Integration/Automating with: Ticketing Systems Workflow, Twitter Vigilance for social media analysis, CKAN open data portals
  - Automated production of MicroServices for IOT app from External REST CALL APIs

GO
- **IOT Network Management and Control**
  - IOT Networks on Snap4City, Data Ingestion Strategy, IOT Broker Registration, IOT Brokers with Service paths, IOT Brokers with MultiTenant
  - IOT Directory, IOT Device vs Time Series, IOT Device Model, IOT Device Management, IOT Device Registration, exploiting model

GO
- **Complete examples**
  - Time Series, automating security, IOT device moving sensors
  - Snap4City Self training sources

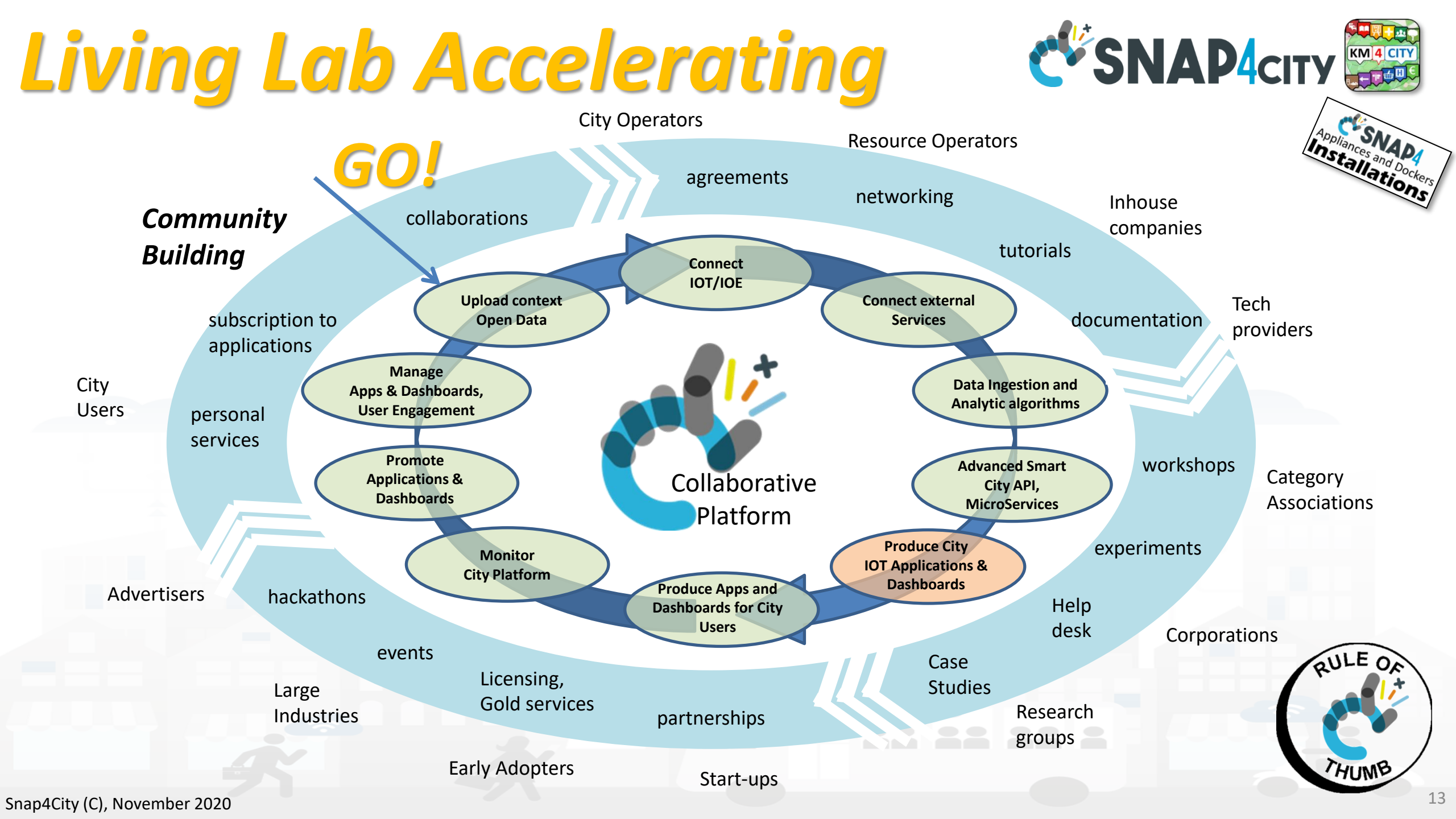
GO
- **IOT Devices hardware-software integration**
  - Open and Proprietary devices, Open HW and Open SW, IOT Devices, IOT Gateway, IOT Edge (Arduino, Raspberry, etc.)
  - (IBE CNR, Libelium, SIGFOX, Lora..), IOT Tracking devices

GO
- **IOT end-2-end Secure Stack, IOT ←→ Dashboards**

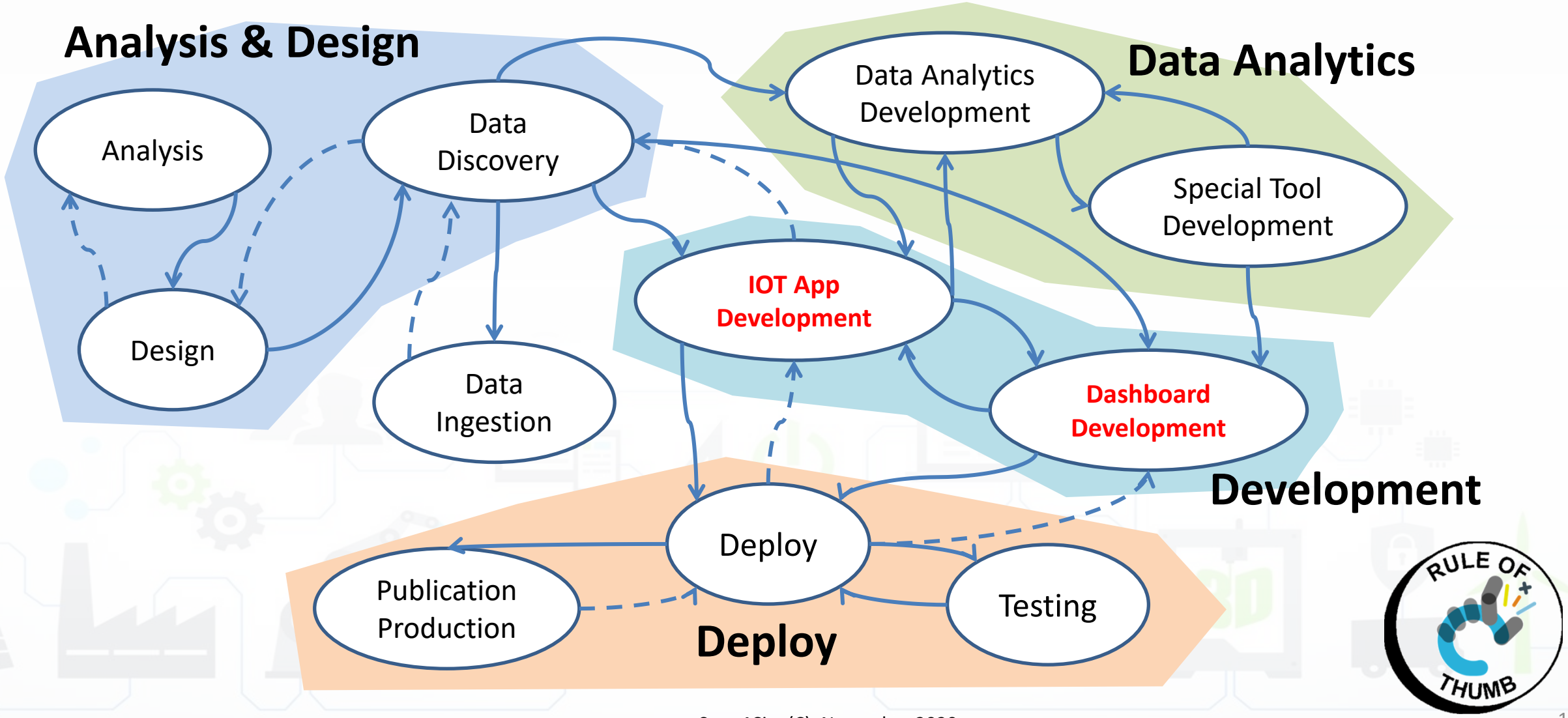
GO
- **Comparison with other Platforms and Fi-Ware**
  - FiWare vs Snap4City

GO
- **Acknowledgments**

Living Lab Accelerating



Development Life Cycle
Smart City Services



Levels of Difficulty

- Easy.  non programmer level
- Moderate.  Some JavaScript rudiment coding
- Good.  JavaScript programming
- Golden.  Programming in R Studio
- Professional.  Exploiting Smart City API
- Excellent.  Developing Full IOT Applications, Dashboard and Mobile Apps

TOP

IOT App for Smartening Solutions

FROM CITY DASHBOARD TO APPLICATIONS

DATA GATHERING AND CITY DATA KNOWLEDGE MANAGEMENT

IOT/IOE DEVICES AND NETWORKS

FORGING & MANAGING OPEN AND FLEXIBLE WEB AND MOBILE APPS

IOT APPLICATIONS, THE LOGIC AND THE SMARTNESS

SMART CITY API, MICROSERVICES, SNAP4CITY API

SNAP4CITY LIVING LAB FOR COLLABORATIVE WORK

DATA ANALYTICS, BUSINESS INTELLIGENCE, WHAT-IF AND SIMULATION

SNAP4CITY FOR BEGINNERS

SNAP4CITY ARCHITECTURE AND ECOSYSTEM. OPENED TO DEVELOPERS

DECISION SUPPORT SYSTEM AND CITY RESILIENCE

TWITTER VIGILANCE: SOCIAL MEDIA ANALYSIS

HOW TO ADOPT SNAP4CITY, AND OUR ROADMAP

SNAP4CITY AND KM4CITY PROJECTS

SNAP4CITY THE VIEW OF THE ADMINISTRATORS

SNAP4CITY

KM4CITY

URBAN PLATFORM: SMART CITY IOT AS A SERVICE AND ON PREMISE

100% OPEN SOURCE

APPLIANCES CONTAINERS



- LOCAL GOVERN
- STAKEHOLDERS
- CITY USERS
- IN-HOUSE
- ENERGY OPERATORS
- MOBILITY OPERATORS
- COMMERCIAL OPERATORS
- SECURITY OPERATORS
- INDUSTRIES
- RESEARCHERS
- START-UPS
- ASSOCIATIONS

- GDPR
- SECURITY
- PRIVACY
- ASSESSMENT
- AUDITING
- PENTESTED

- OPEN IOT DEVICES
- IOT EDGE
- IOT GATEWAY
- PAX COUNTERS
- IOT BUTTONS

- TEST CASES, SCENARIOUS, VIDEOS, HACKATHONS
- OPEN SOURCES, COMMUNITY OF CITIES
- TRAINING TUTORIALS, COMMUNITY MANAGEMENT

IOT APPLICATIONS - INSTANT APPS



DATA DRIVEN APPLICATIONS • REAL TIME PROCESSING • BATCH PROCESSING • ANY PROTOCOL & FORMAT

DASHBOARDS & APPLICATIONS



CONTROL ROOM • SITUATION ROOM • OPERATOR DASHBORDS • BUSINESS INTELLIGENCE • WHAT-IF ANALYSIS • DECISION SUPPORT • SIMULATIONS • RISK ANALYSIS • RESILIENCE ANALYSIS

MOBILE & WEB APPLICATIONS



DEVELOPMENT KIT • SUGGESTIONS • MOBILE APPS • MONITORING PANELS • PLATFORM UTILITIES • READY TO USE SMART APPLICATIONS

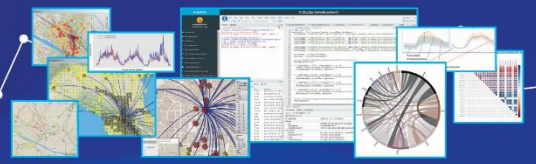
MICROSERVICES & ADVANCED SMART CITY API

LIVING LAB - DEV TOOLS - COWORKING

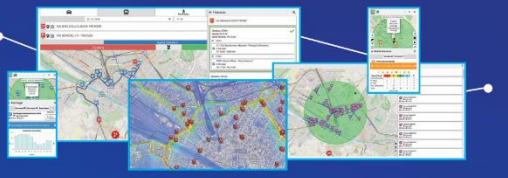


IOT DIRECTORY • SERVICE MAP • RESOURCE MANAGER • DATA GATE • R STUDIO • ETL

BIG DATA - DATA ANALYTICS



DATA ANALYTICS TOOLS - MICRO-APPLICATIONS



PREDICTIONS • ANOMALY DETECTION • WHAT-IF ANALYSIS • TRAFFIC FLOW RECONSTRUCTION • ORIGIN-DESTINATION MATRICES • SOCIAL MEDIA ANALYSIS • OFFER VS DEMAND ANALYSIS • ENVIRONMENTAL DATA ANALYSIS AND PREDICTIONS • REAL TIME HEATMAPS • ROUTING • ALERTING • EARLY WARNING • PERSONAL AND VIRTUAL ASSISSTANTS • SMART SOLUTIONS • SMART SHARING • PARTECIPATORY

KM4CITY DATA AGGREGAT KNOWLEDGE BASE - EXPERT SYSTEM OF THE CITY – BIG DATA STORE

IOT MNG - DATA MNG - DATA INSPECTOR – PROCESS MNG – USER ENGAGEMENT – GDPR MNG ...

GIS	CITY UTILITIES	OPEN DATA	LEGACY & EXTERNAL SERVICES	PERSONAL DATA	IOT / IOE	BROKERS	KPI	INDUSTRY 4.0	SOCIAL MEDIA



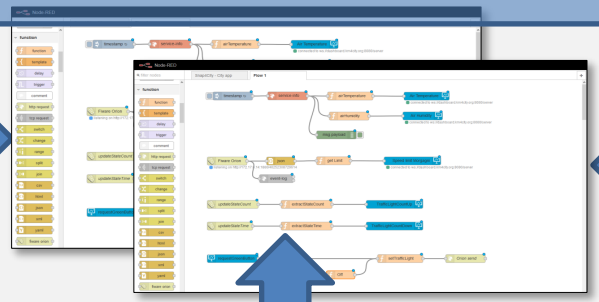
Snap4City: Builder of Sentient Cities Solutions

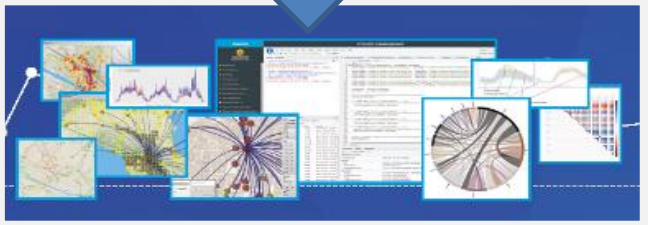
Dashboards with data driven IOT Applications enforcing intelligence

IOT and data World

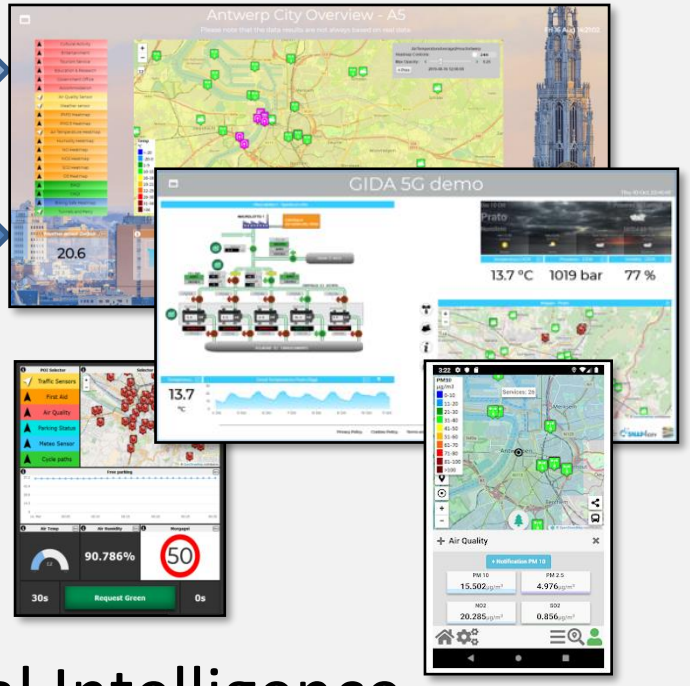


IOT Applications



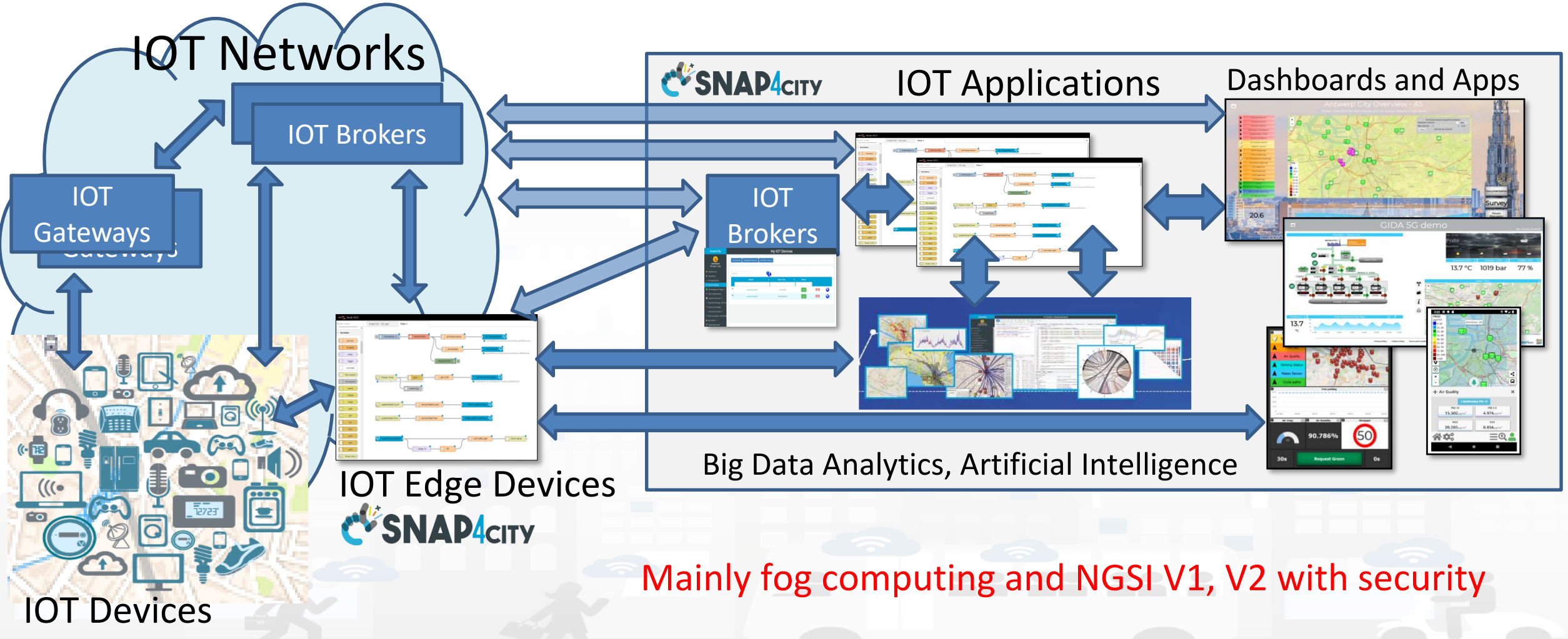


Dashboards and Apps



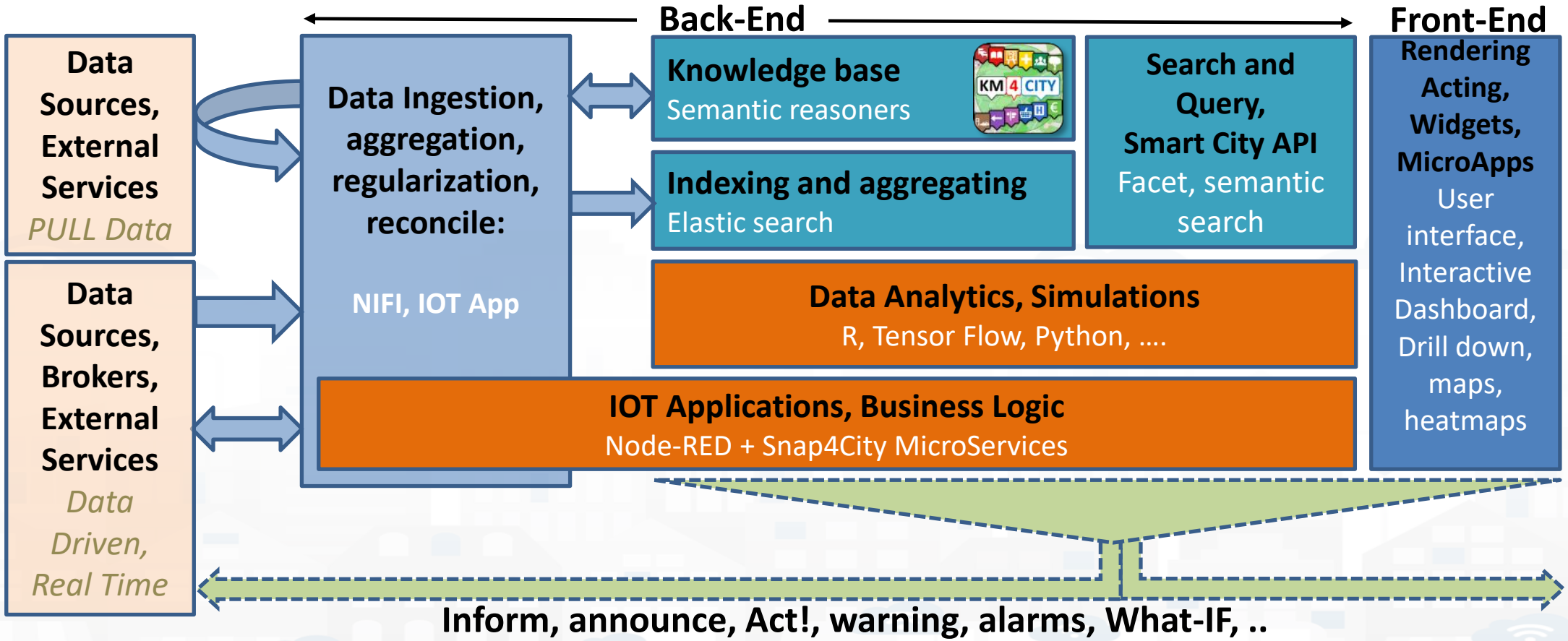
Big Data Analytics, Artificial Intelligence

Snap4City Services also on IOT Edge!!!



Mainly fog computing and NGSI V1, V2 with security

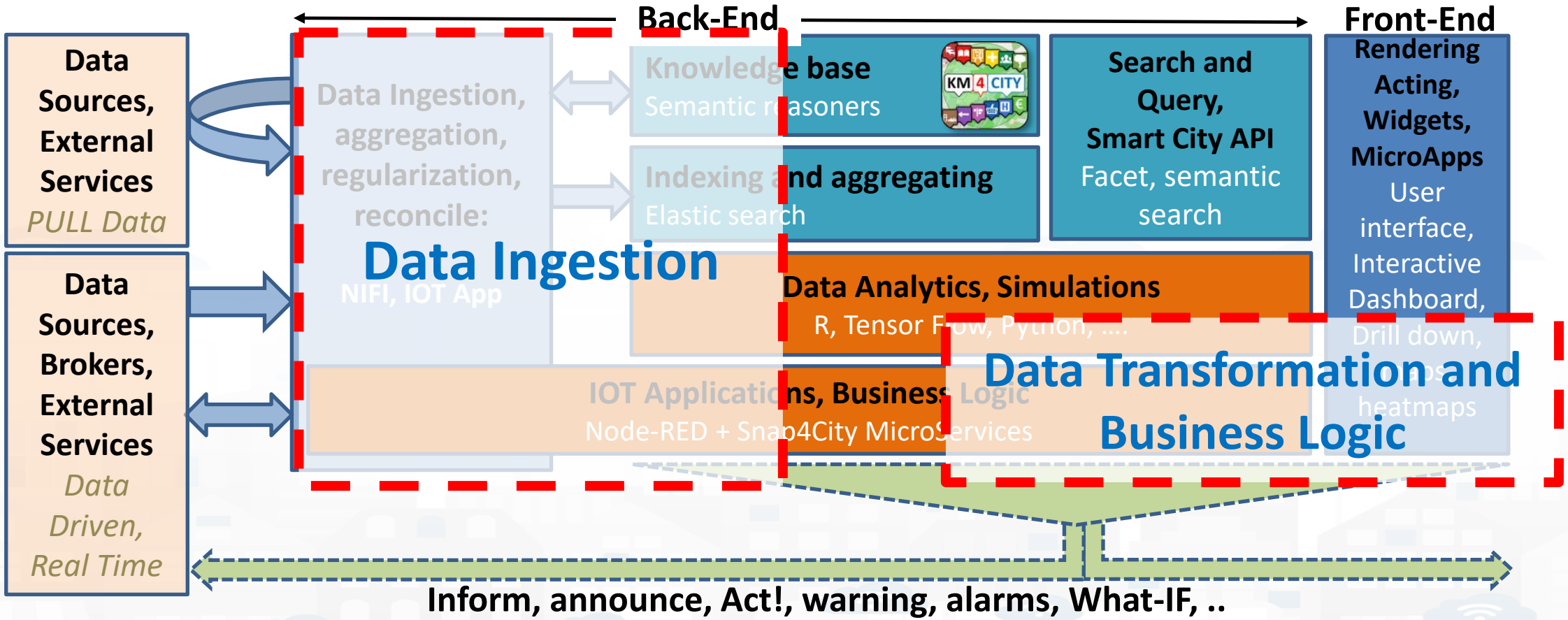
Snap4City Architecture vs Data Ingestion



The usage of IOT Applications

- IOT Applications = Node-RED + Snap4City Libraries
- Used for:
  - **Data Ingestion, Transformation, Extract, Load, and Adaptation** (format and protocol), See Part 5 of the Course
  - **IOT Edge Devices logic**, for implementing logic on IOT Edge, including **IOT Device control** (see on Part 3 of the course, this part)
  - **Business Logic** control of Dashboards, via Web Sockets secure
    - see Part 2 of the Course
  - **Control and schedule of Data Analytic**, and Machine Learning (see part 4 of the Course)
  - **Firing and condition identification and alerting**.

Snap4City Architecture vs Data Ingestion

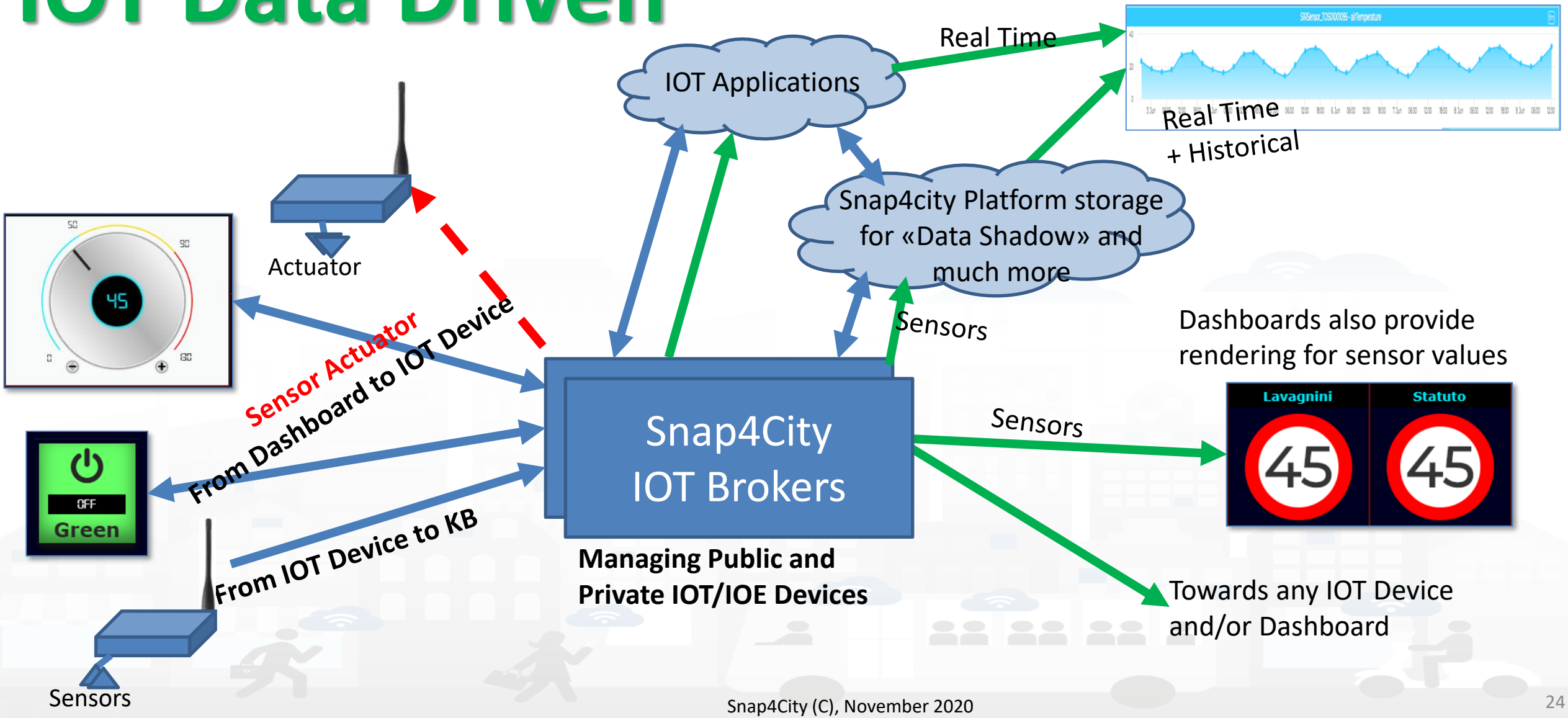


Standards and Interoperability

Compliant with: AMQP, COAP, MQTT, OneM2M, HTTP, HTTPS, TLS, Rest Call, SMTP, TCP, UDP, NGSI, LoRa, LoRaWan, TheThingsNetwork, SigFOX, DATEX II, SOAP, WSDL, Twitter, FaceBook, Telegram, SMS, OLAP, MySQL, Mongo, HBASE, SOLR, SPARQL, EMAIL, FTP, FTPS, WebSocket, WebSocket Secure, ModBUS, OPC, GML, RS485, RS232, WFS, WMS, ODBC, JDBC, Elastic Search, Phoenix, XML, JSON, CSV, db, GeoJSON, Enfuser FMI, Android, Raspberry Pi, Local File System, ESP32, Libelium, IBIMET/IBE, OBD2, SVG, XLS, XLSX, TXT, HTML, CSS, KNX, Enocean, Zigbee, DALI, ISEMC, Alexa, Sonoff, HUE Philips, Tplink, BACnet, TALQ, etc.

https://www.snap4city.org/65

IOT Data Driven



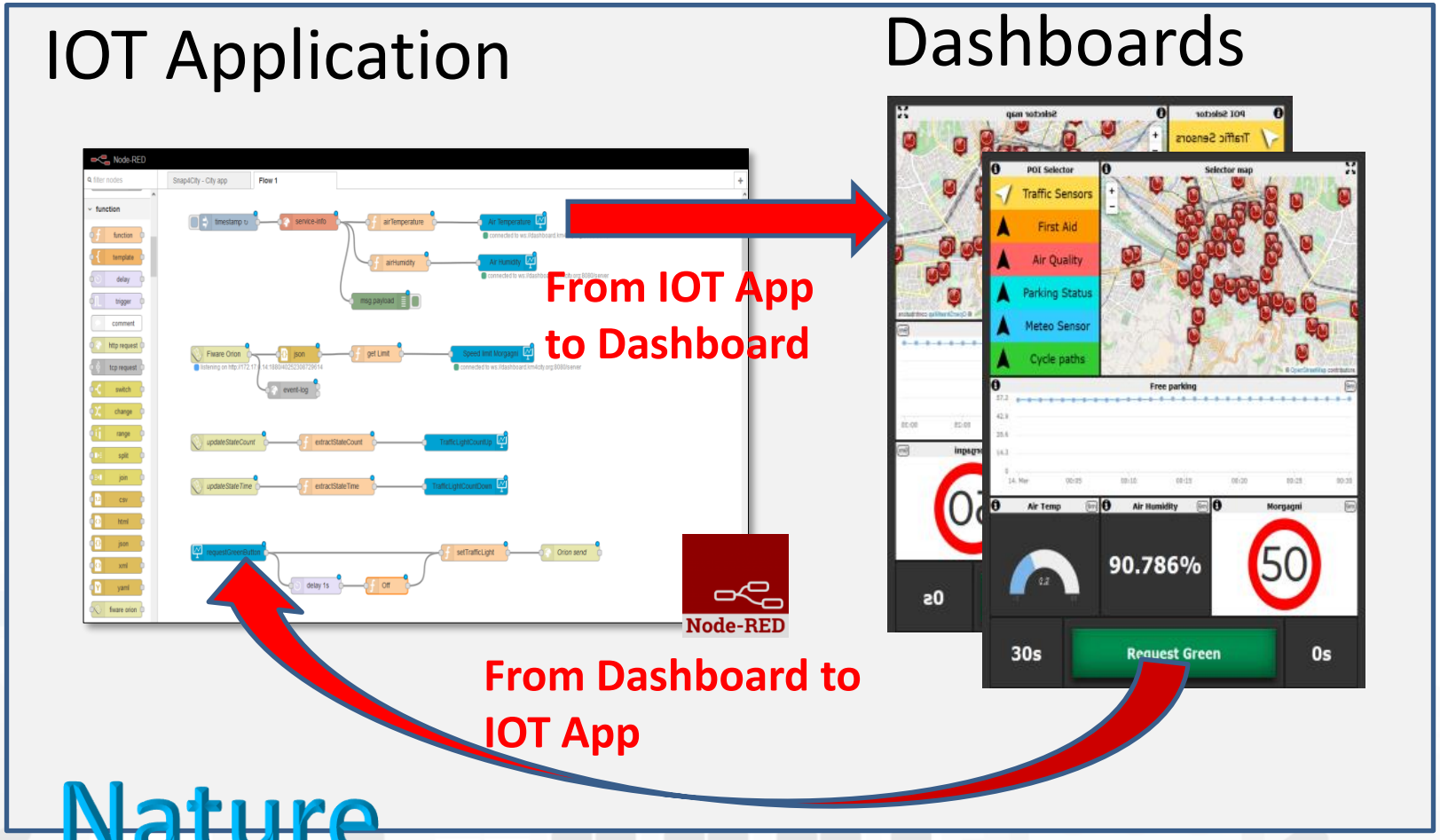
IOT App Smartening Dashboards



HLT: Sensors-Actuators

High Level Types

- **Complex Event**
- **Dashboard-IOT App**
- **External Service**
- **Heatmap**
- **KPI** (Key Performance Indicator)
- **MicroApplication**
- My Personal Data
- MyKPI
- MyPOI
- **POI** (Point of Interest)
- **Sensor**
- **Sensor Actuator**
- **Special Widget**
- Wfs (GIS)



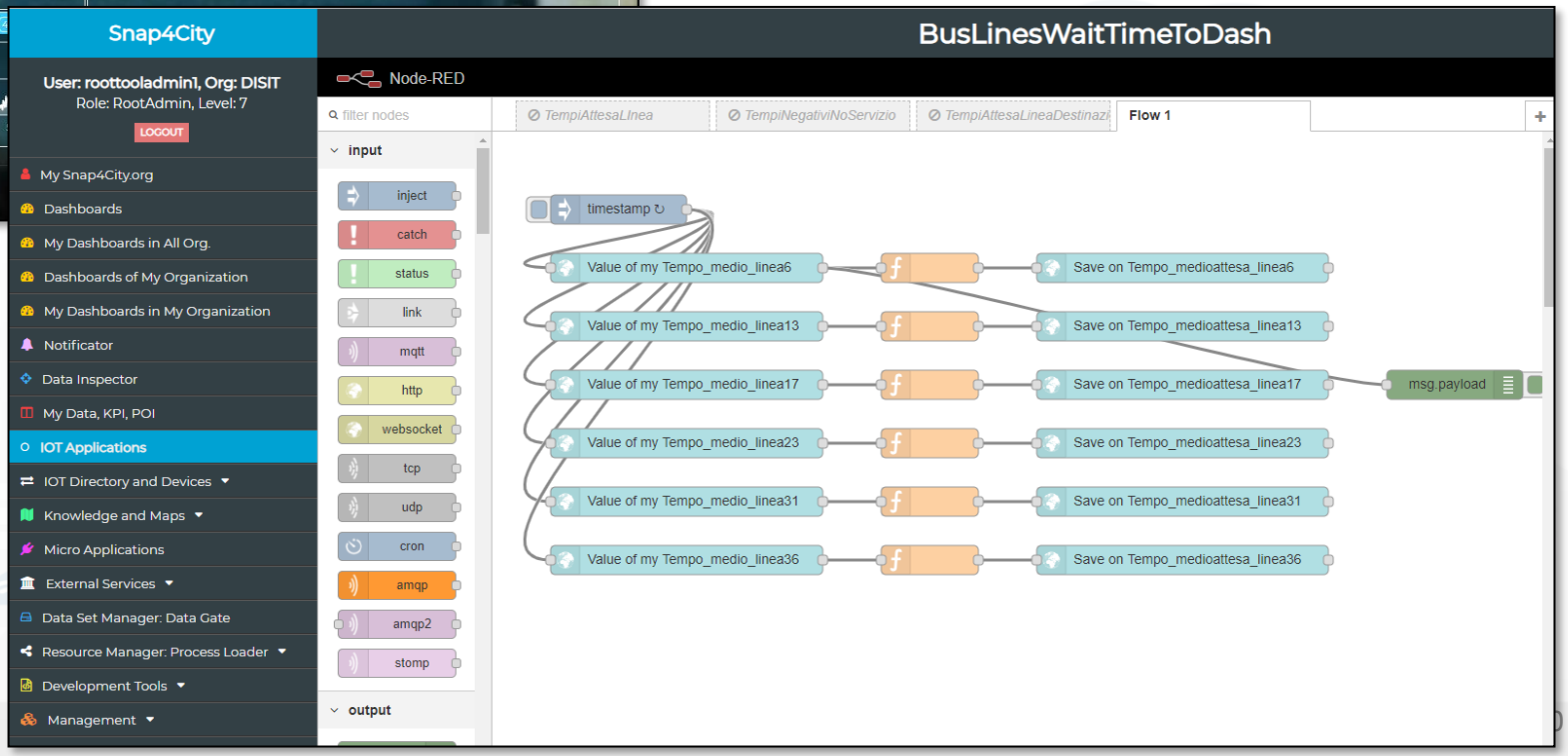
Florence







Estimation of the mean waiting time at bus stops



Antwerp



IOT App Smartening Devices and Dashboards



PaxCounter devices







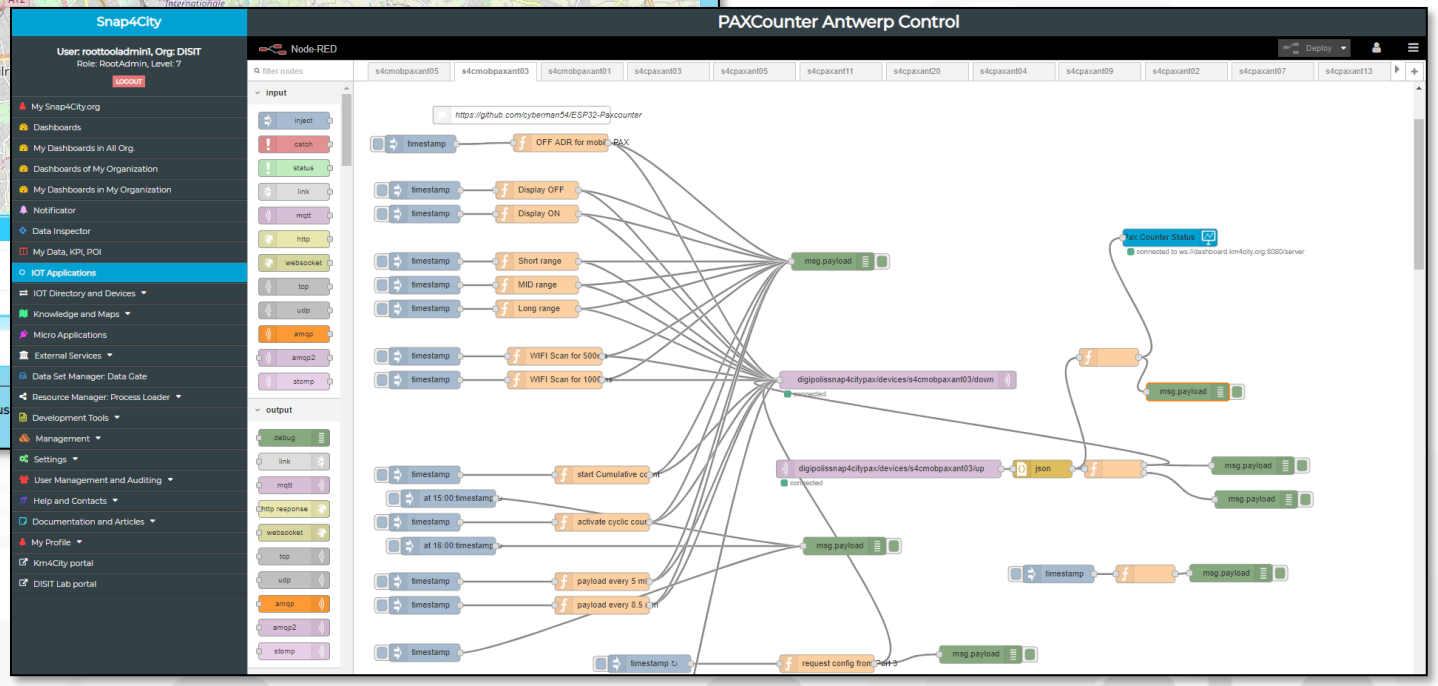
- **Fix PaxCounter LoraWan**
  - sniffing on: Wi-Fi, Bluetooth
  - Sending data via LoraWan
- **Mobile PaxCounter LoraWan**
  - sniffing on: Wi-Fi, Bluetooth
  - Sending data via LoraWan
- **Fix PaxCounter, multiple out**
  - Sending data via LoraWan and Wi-Fi
  - sniffing on: Wi-Fi, Bluetooth

LoRa® Wi Fi Bluetooth™

https://www.snap4city.org/456

Programmable PAX counting





Antwerp

IOT App Smart Parking



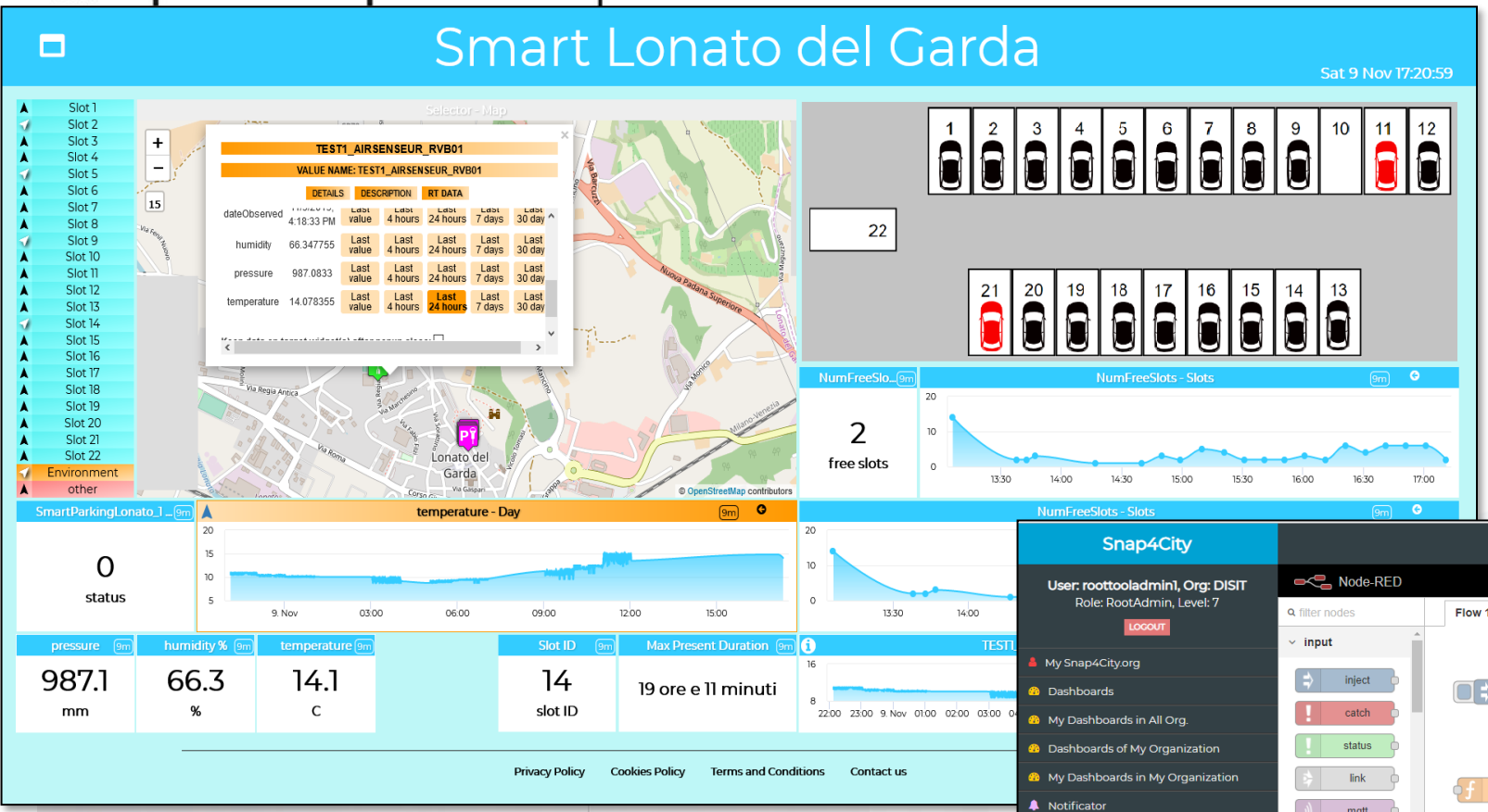


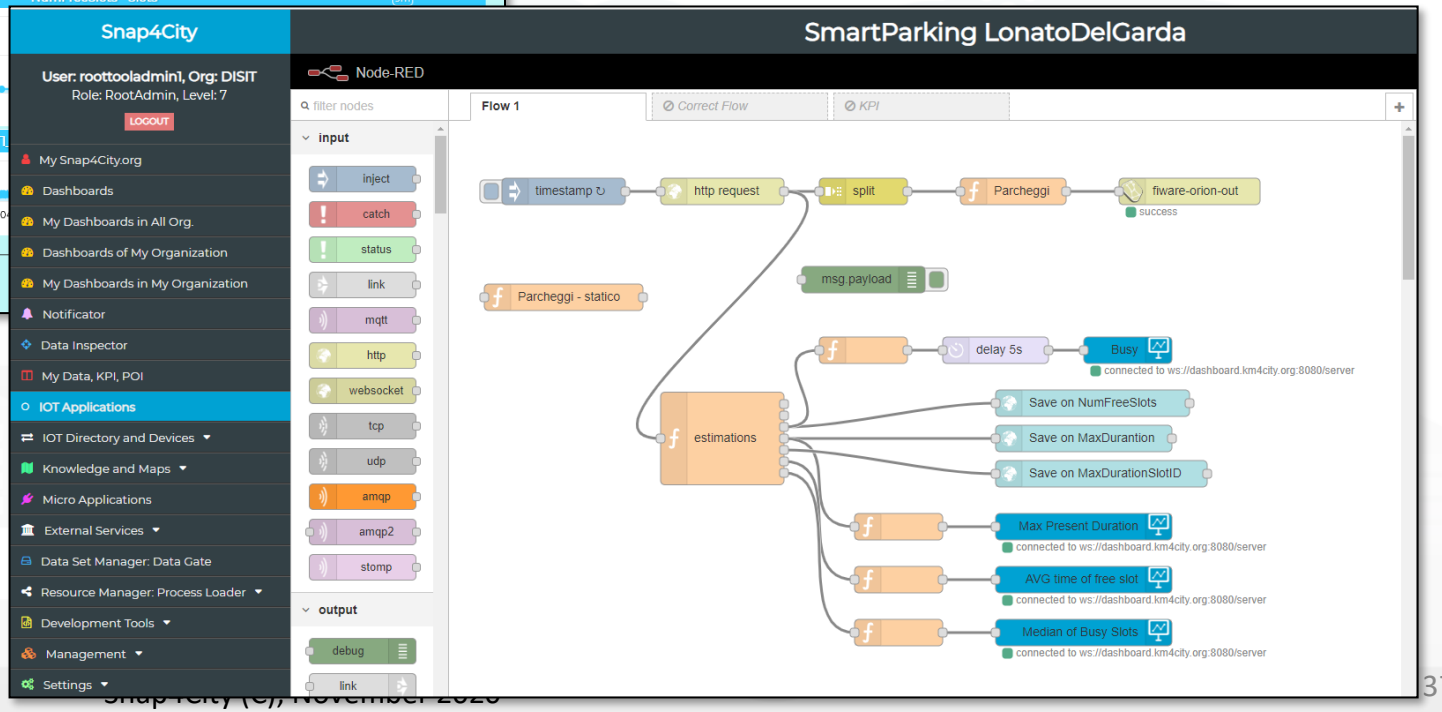
Lonato del Garda

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

SNAP4CITY | KM4CITY

Smart Parking Monitoring







IOT App Smart Industry 4.0
Snap4Industry

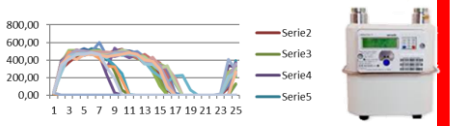


UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

https://www.snap4city.org/369

SNAP4CITY KM4CITY

External Services



Fleet management



IoT Devices/Edge

IoT Broker



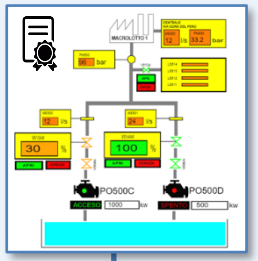
Internet

SECURE





Admin





DCS

IoT Broker



PLC

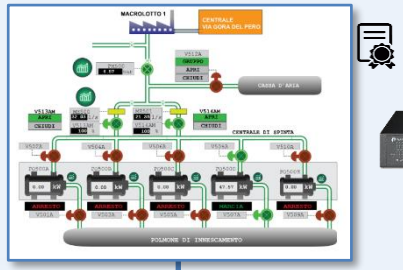






IoT Devices/Edge

Industry Plant1.....



SCADA



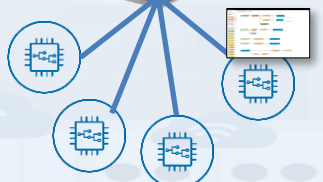


PLC/ RTU



IoT Broker



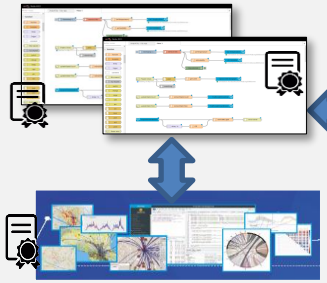


IoT Devices/Edge

Industry Plant2.....

IOT Applications

Dashboards and Apps



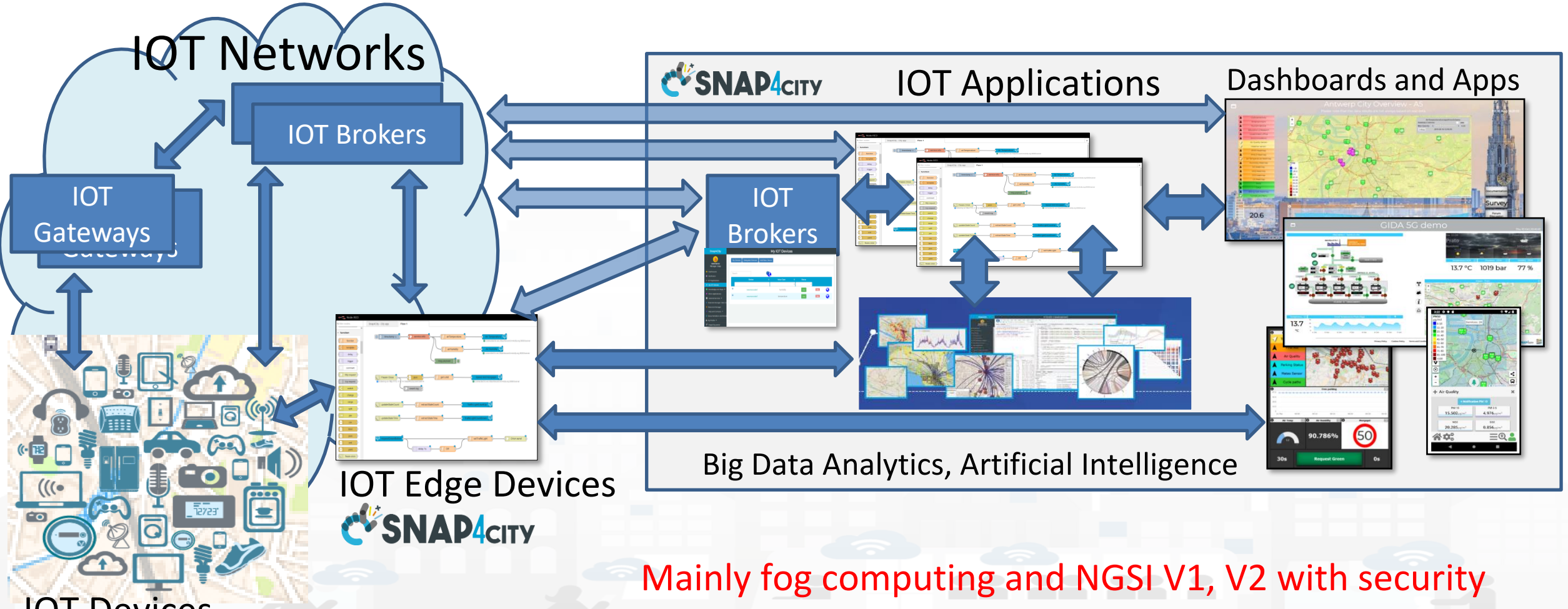


Big Data Analytics, Artificial Intelligence

Control and Supervision on Multiple Supply Chains

Industry 4.0 as a Service

Snap4City Services also on IOT Edge!!!



Mainly fog computing and NGSI V1, V2 with security



Prato
Smart City vs Industry 4.0

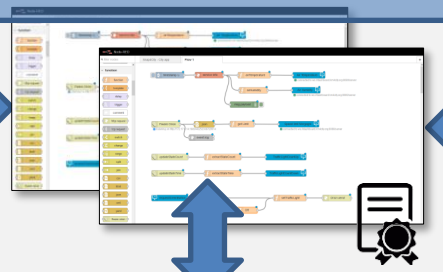
GIDA set up

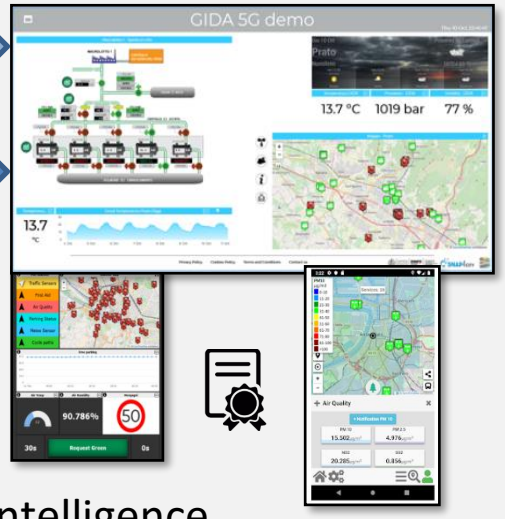
Smart City data from many sources



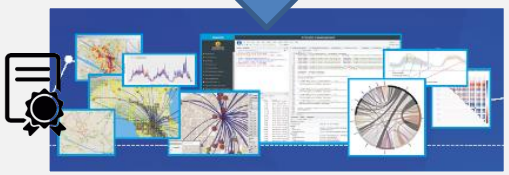
IOT Applications

Dashboards and Apps

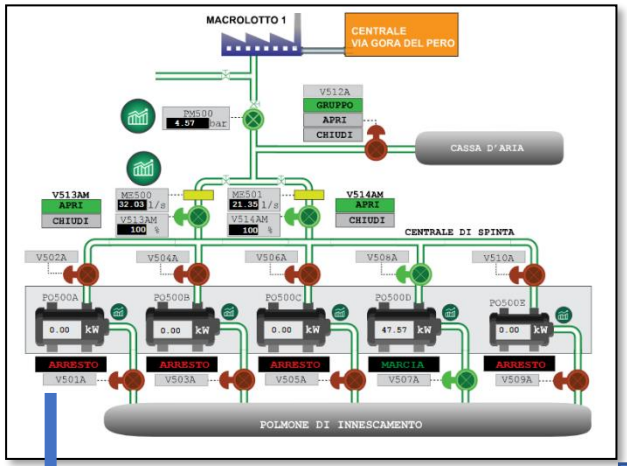




IOT Data Shadow Snap4City



Big Data Analytics, Artificial Intelligence



ModBus to Snap4City Gateway Edge

5G network devices

Telemonitoring Telecontrol

https://www.snap4city.org/369

https://www.snap4city.org/369

Measuring any kind of sensors values

Controlling Energy Power

Measuring
Energy Consumption

Any kind of notification channel

IOT Edge:
Node-RED
+
Snap4City

Local Control

Modbus

SCADA

OPC UA

DCS

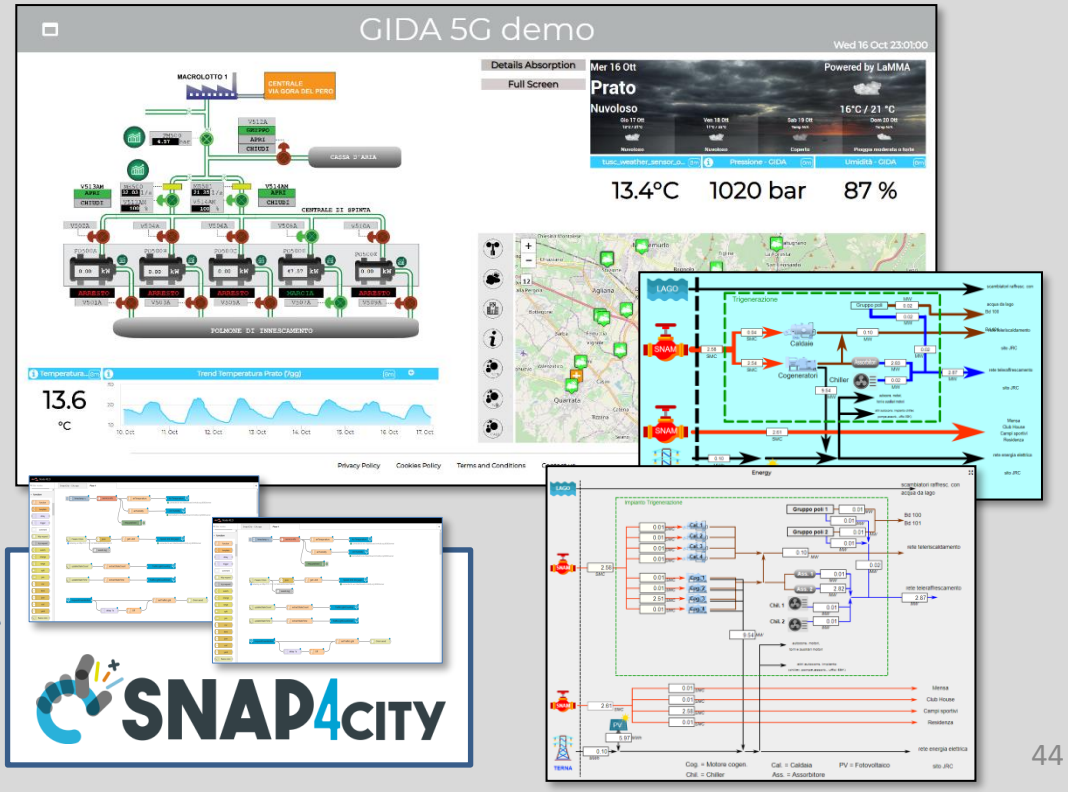
PLC

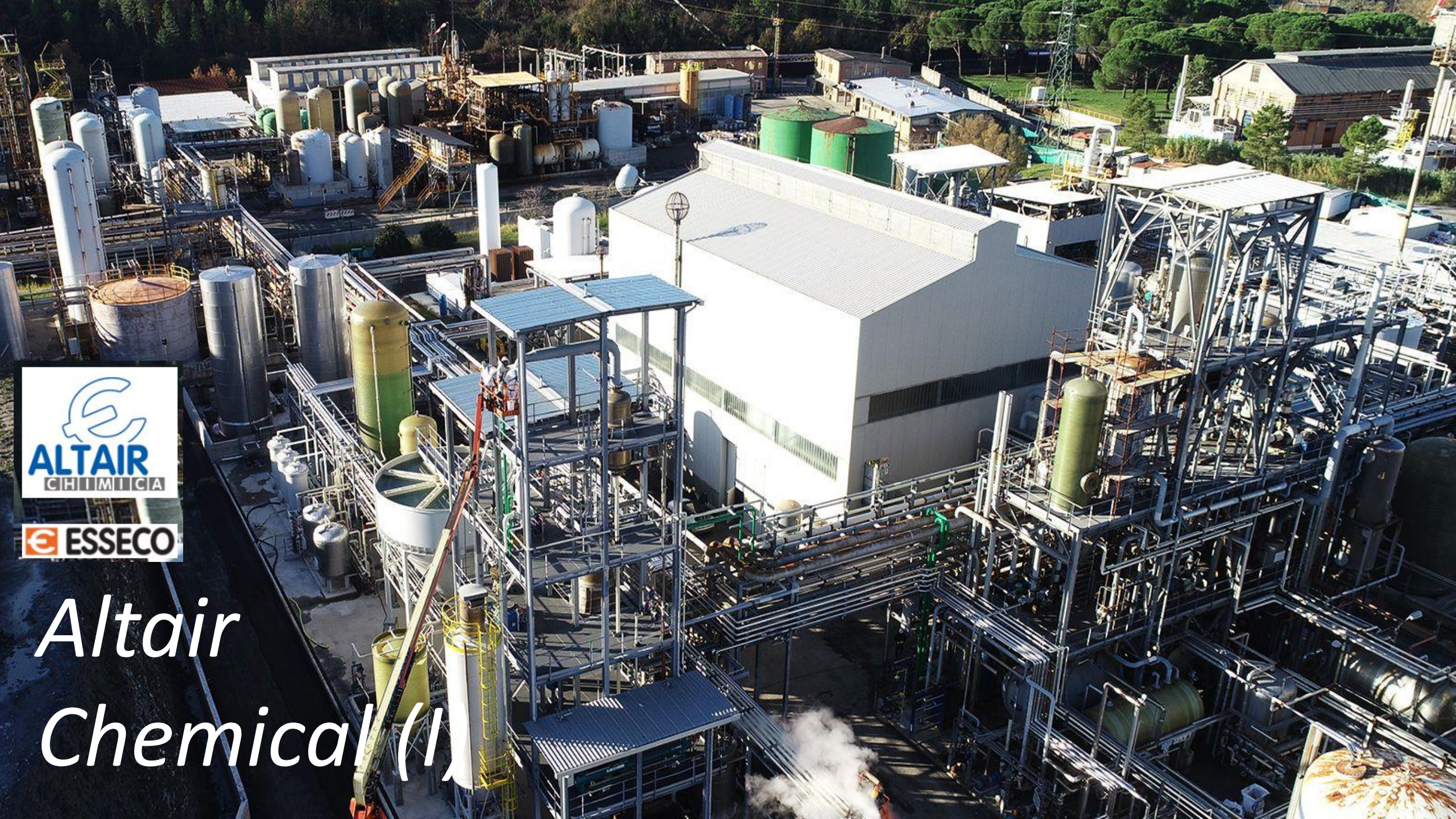
ODBC

Administrative Servers

Alexa: Voice Commands

Contextual (smart city/home) data, Data Analytics
Historical Data, Remote Control, Mobile App





Altair Chemical (I)

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

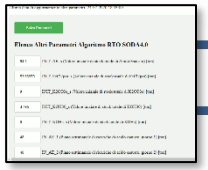
SNAP4INDUSTRY | KM4CITY



SNAP4ALTAIR

DCS

OPC UA



Parameters

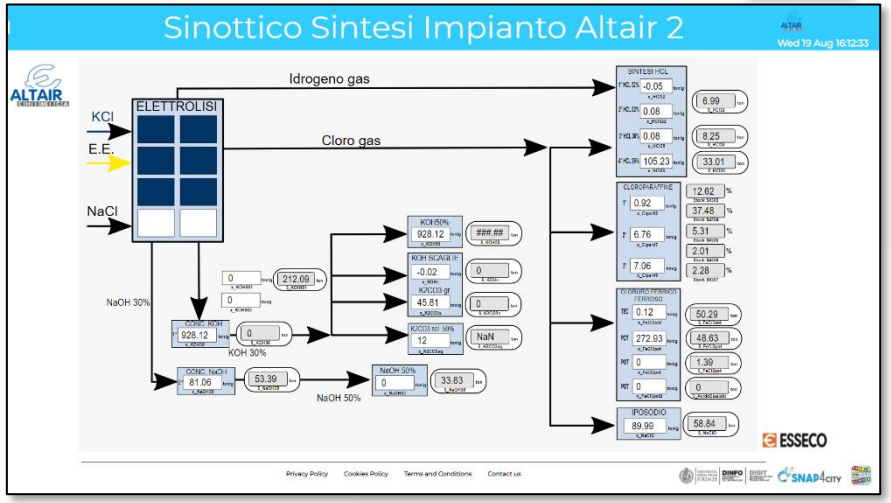


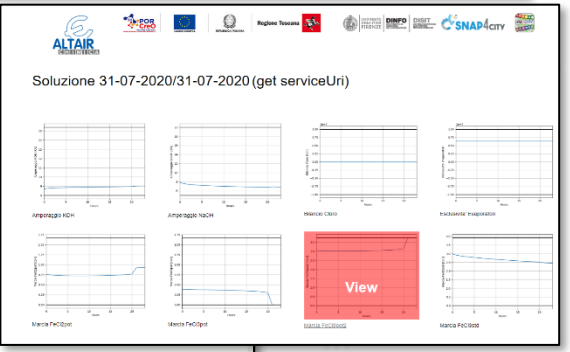
AS400 (orders)

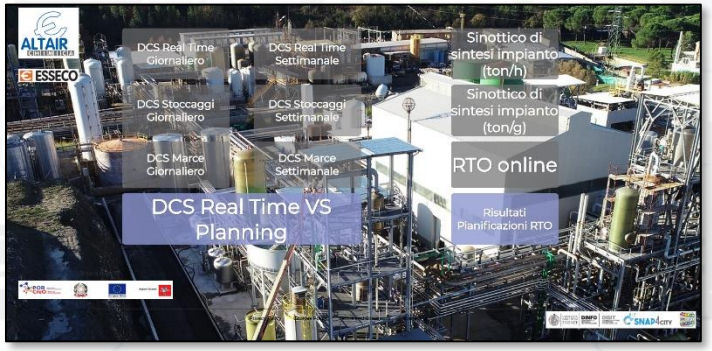
PUN (energy)

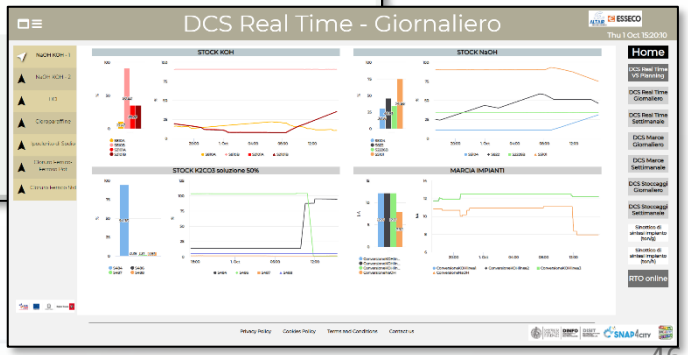
Snap4Altair *OnLine*

Monitoring
Planning





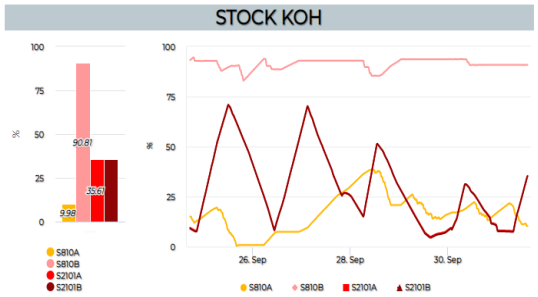


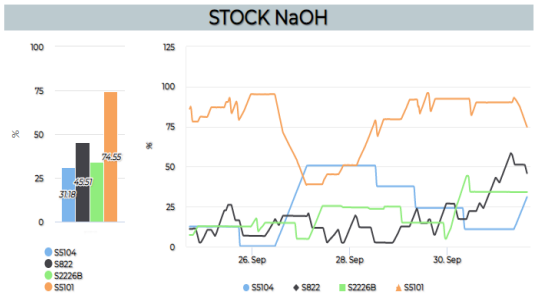


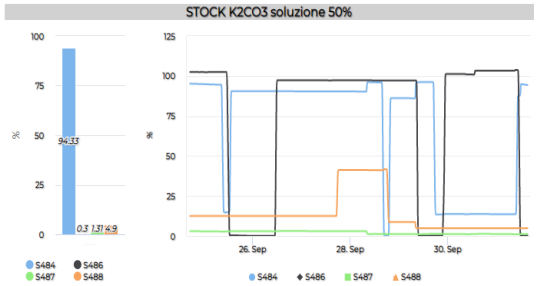
DCS Real Time - Settimanale

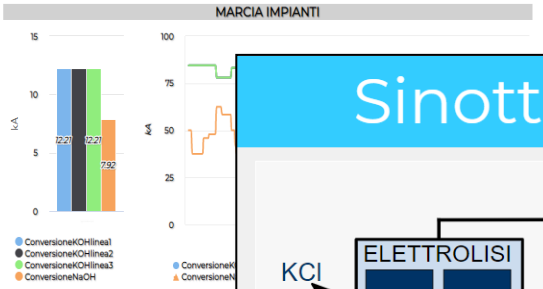
Thu 1 Oct 15:25:11

NaOH KOH - 1
NaOH KOH - 2
HCl
Cloroparaffine
Ipoclorito di Sodio
Cloruro Ferrico-Ferroso Pot
Cloruro Ferrico Std









Home
DCS Real Time VS Planning
DCS Real Time Giornaliero
DCS Real Time Settimanale
DCS Marce Giornaliero
DCS Marce Settimanale
DCS Stoccaggi Giornaliero

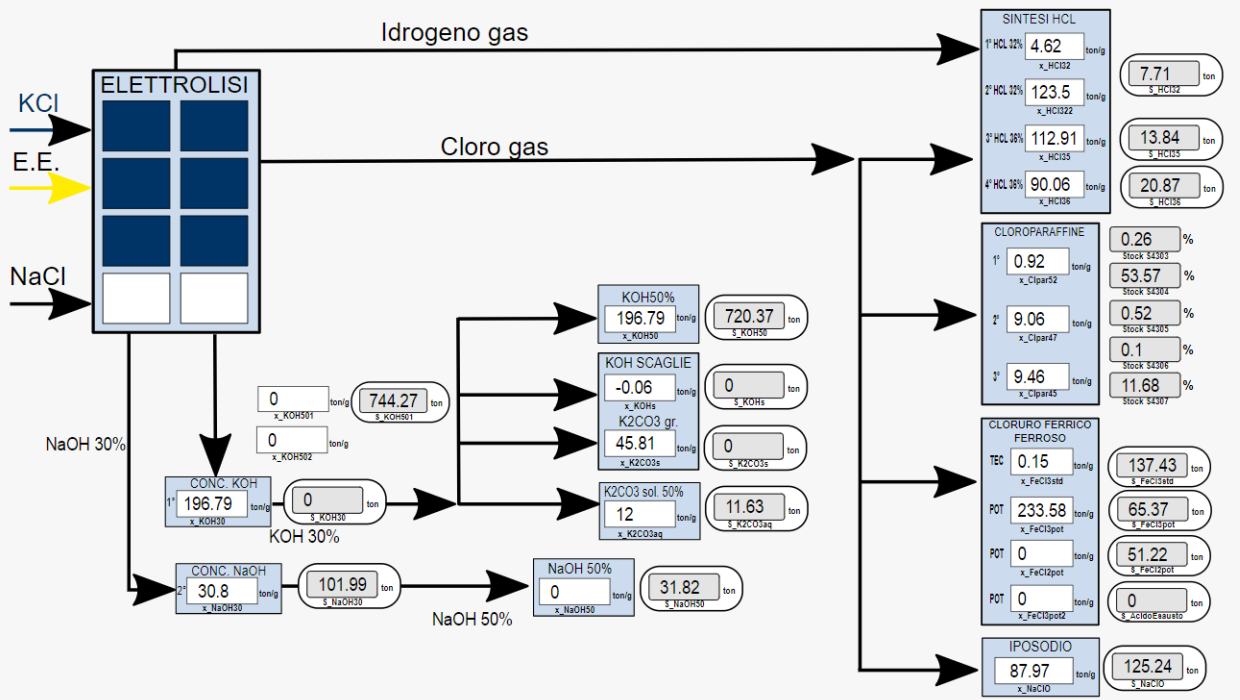
ALTAIR CHIMICA
ESSECO

SNAP4INDUSTRY
KM4CITY

SNAP4ALTAIR

Sinottico Sintesi Impianto Altair 2

Thu 1 Oct 15:32:19



Home
DCS Real Time VS Planning
DCS Real Time Giornaliero
DCS Real Time Settimanale
DCS Marce Giornaliero
DCS Marce Settimanale
DCS Stoccaggi Giornaliero
DCS Stoccaggi Settimanale
Sinottico di sintesi impianto (ton/g)
Sinottico di sintesi impianto (ton/h)
RTO online

Privacy Policy | Cookies Policy | Terms and Conditions | Contact us

RTO online

Home RTO online

Parametri (TabPar)	DCS (OPC-UA)	Amministrazione (AS400)	Pianificazione Consolidata (AS400)	Energia (PUN)	Altri Parametri
2020-09-25 18:47:36	2020-10-01 09:33:01	2020-10-01 09:32:54	2020-10-01 09:32:54	2020-10-01 23:00:00	2020-07-24 18:43:00
2020-09-25 18:47:36	2020-09-30 17:20:02	2020-09-30 17:20:50	2020-09-30 17:20:50	2020-09-30 23:00:00	2020-07-24 18:43:00
2020-09-25 18:47:36	2020-09-30 16:27:02	2020-09-30 16:24:57	2020-09-30 16:24:57	2020-09-30 23:00:00	2020-07-24 18:43:00
2020-09-25 18:47:36	2020-09-30 14:56:02	2020-09-30 14:54:11	2020-09-30 14:54:11	2020-09-30 23:00:00	2020-07-24 18:43:00
2020-09-25 18:47:36	2020-09-30 13:43:02	2020-09-30 13:43:47	2020-09-30 13:43:47	2020-09-30 23:00:00	2020-07-24 18:43:00
2020-09-25 18:47:36	2020-09-29 19:03:01	2020-09-29 19:03:27	2020-09-29 19:03:27	2020-09-30 23:00:00	2020-07-24 18:43:00
2020-09-25 18:47:36	2020-09-28 18:30:02	2020-09-28 18:30:13	2020-09-28 18:30:13	2020-09-29 23:00:00	2020-07-24 18:43:00
2020-09-25 18:47:36	2020-09-28 17:57:02	2020-09-28 17:57:14	2020-09-28 17:57:14	2020-09-29 23:00:00	2020-07-24 18:43:00
2020-09-25 18:47:36	2020-09-28 15:50:02	2020-09-28 15:50:21	2020-09-28 15:50:21	2020-09-28 23:00:00	2020-07-24 18:43:00
2020-09-25 18:47:36	2020-09-25 18:47:02	2020-09-25 18:46:02	2020-09-25 18:46:02	2020-09-26 23:00:00	2020-07-24 18:43:00

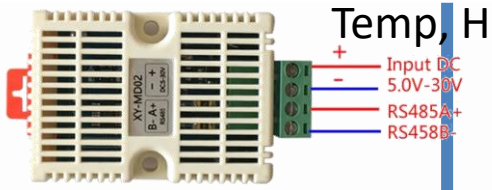
<< 1 2 3 4 5 6 7 8 >>

Sinottico di sintesi impianto (ton/g)

IOT App Smart Industry 4.0
ModBus Integration



Devices



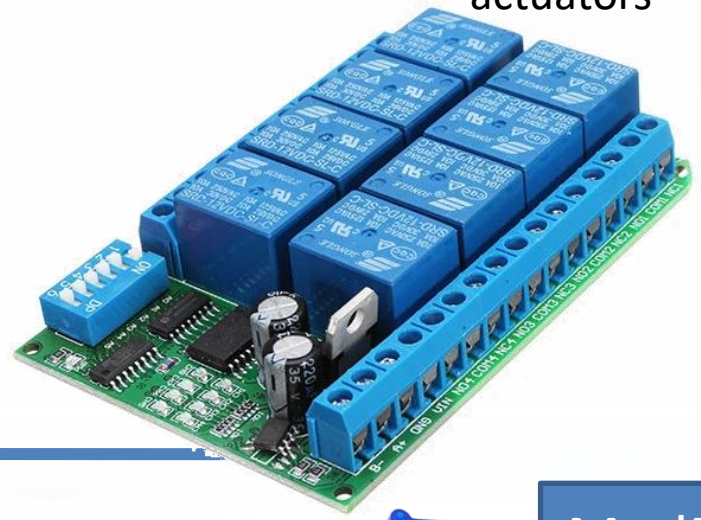
Temp, Humidity



Energy Metering

Modbus

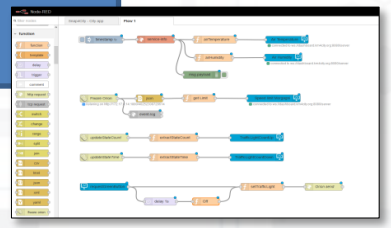
8 Relais, actuators

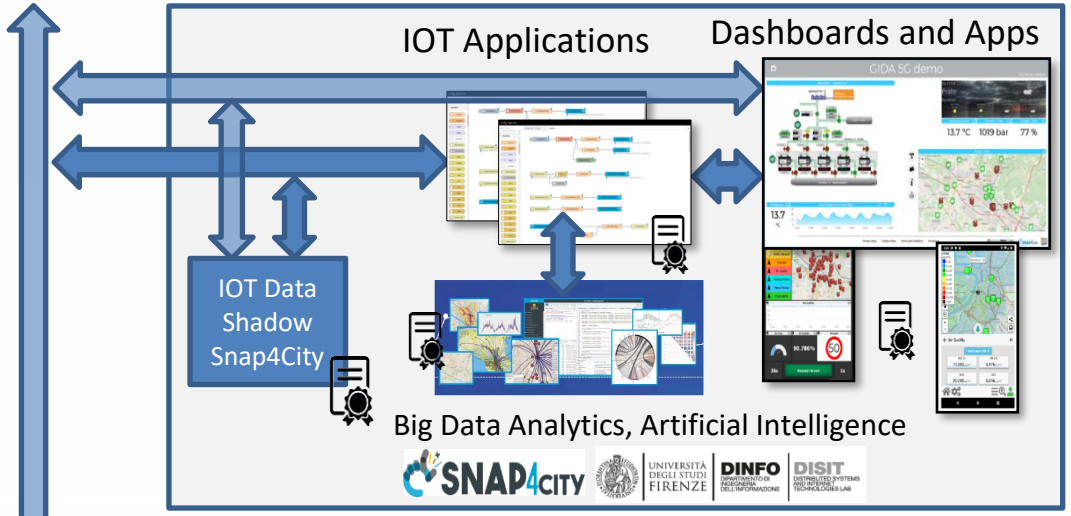




USB -- ModBus

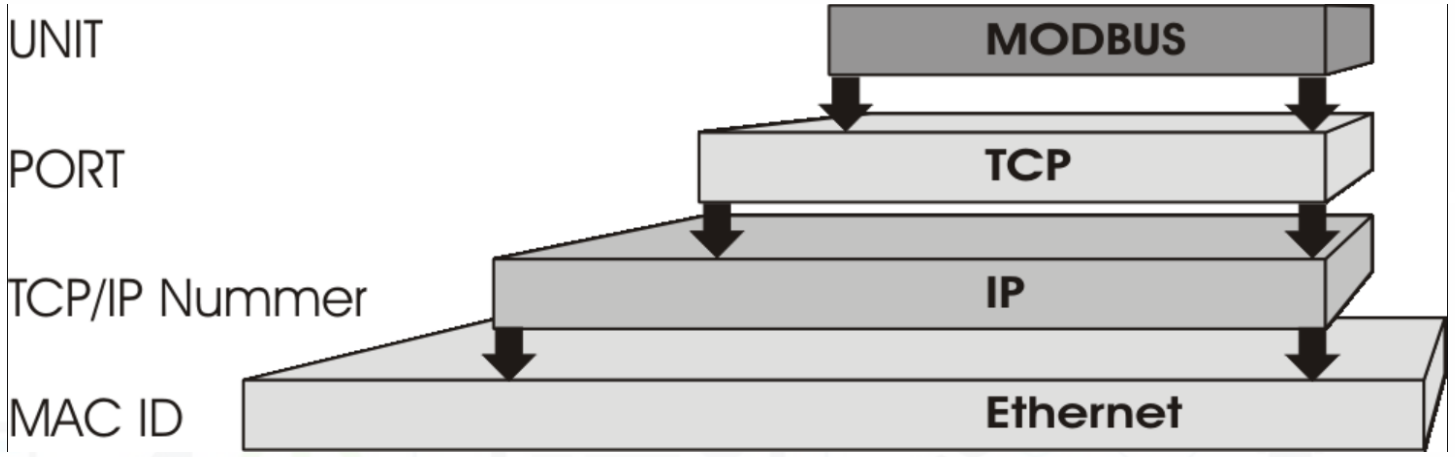
ModBus to Snap4City Gateway Edge





- A large range of devices: sensors and actuators
- Over serial as RS485 and/or IP

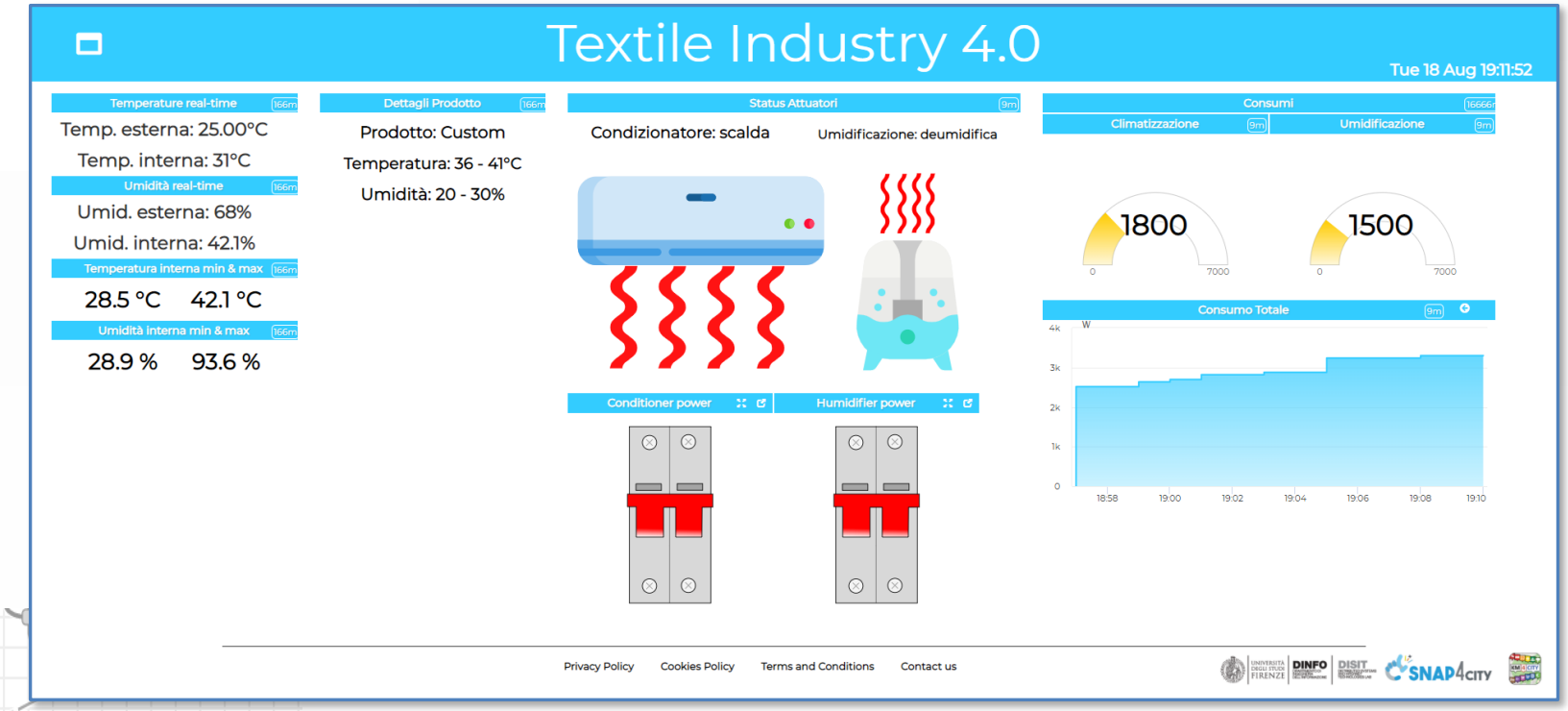
- modbus-response
- modbus-read
- modbus-getter
- modbus-flex-getter
- modbus-write
- modbus-flex-write
- modbus-server
- modbus-flex-server
- modbus-queue-info
- modbus-flex-connector
- modbus-io-config
- modbus-response-filter

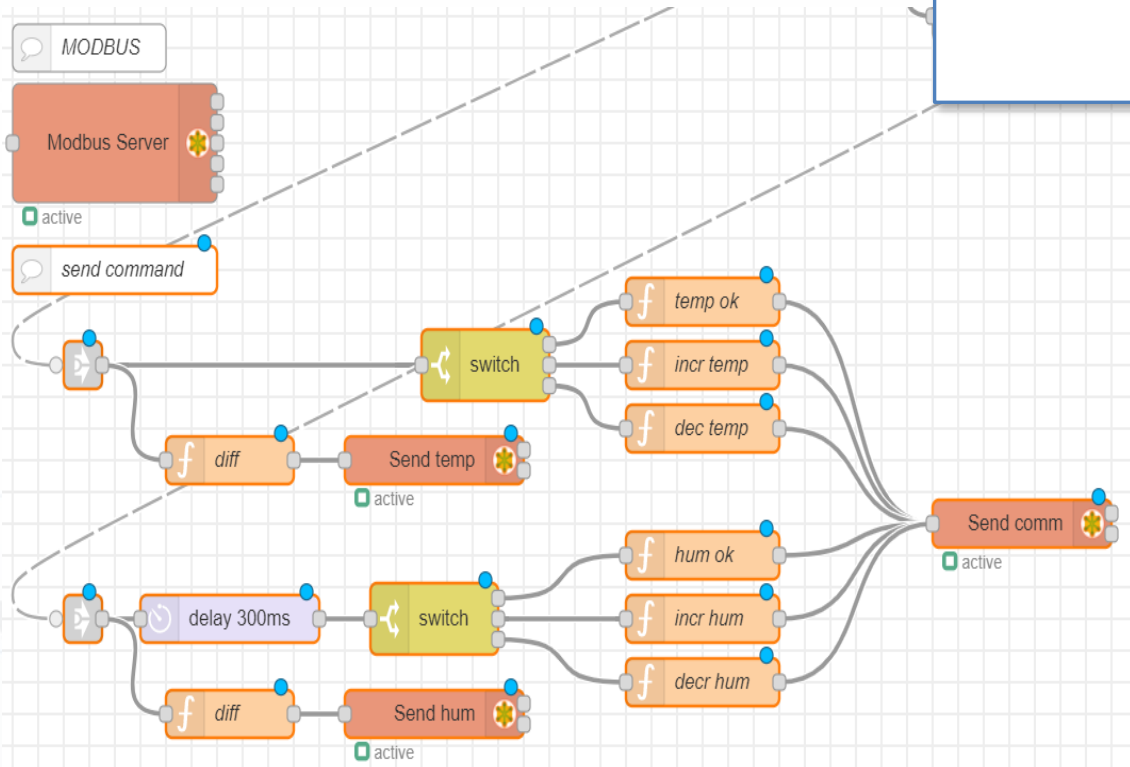


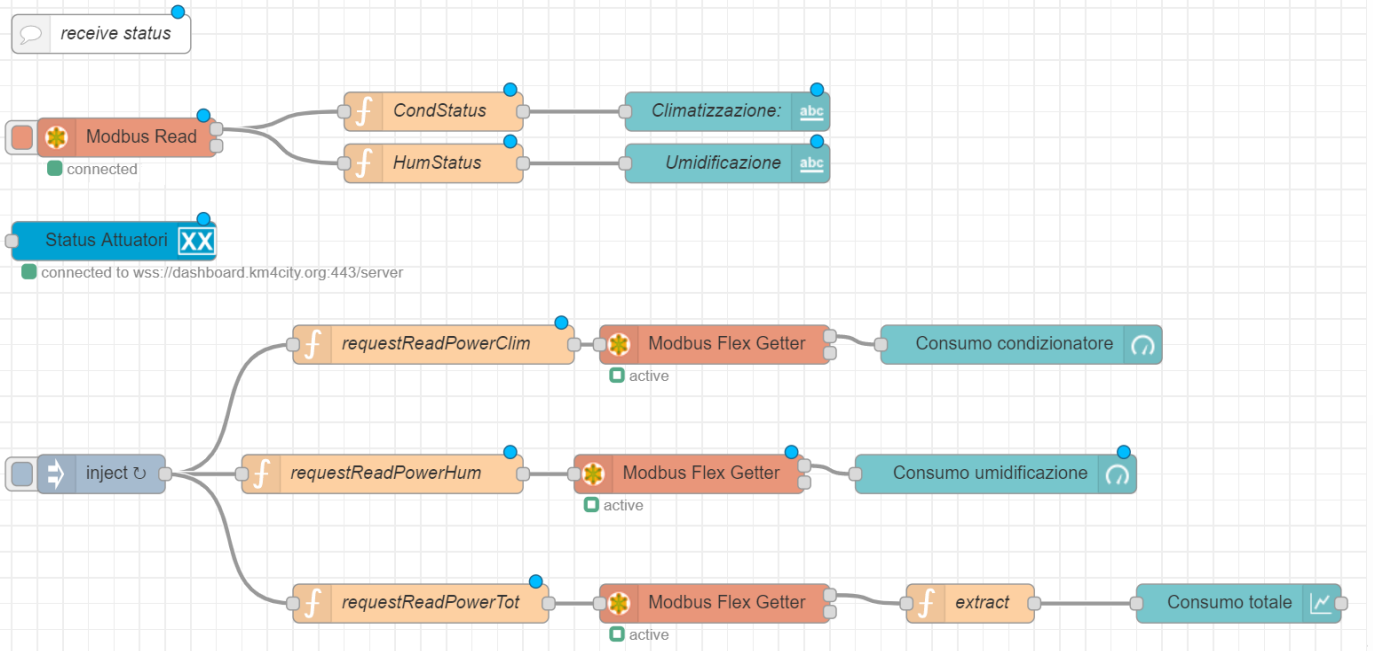
- ..

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

Modbus







IOT App vs Smart Home
Snap4Home





Prato
Smart City vs Smart Home Estra

Snap4Home

Sonoff: Controlling Energy Power



Philips Hue: Controlling Lights





Hue: Motion Control / Alarm



Measuring
Energy Consumption



TP Link: Controlling / Measuring Energy Plugs

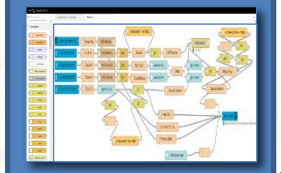


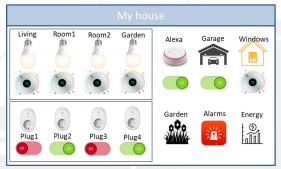
Alexa: Voice Control



IOT Edge:

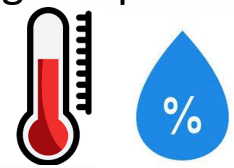
Raspberry pi: Node-RED + Snap4City





Local Control

Measuring Temperature and Humidity



Garage Door

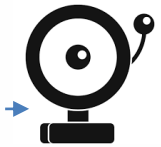


Window
Roller Shutters



Controlling Motors

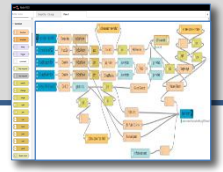




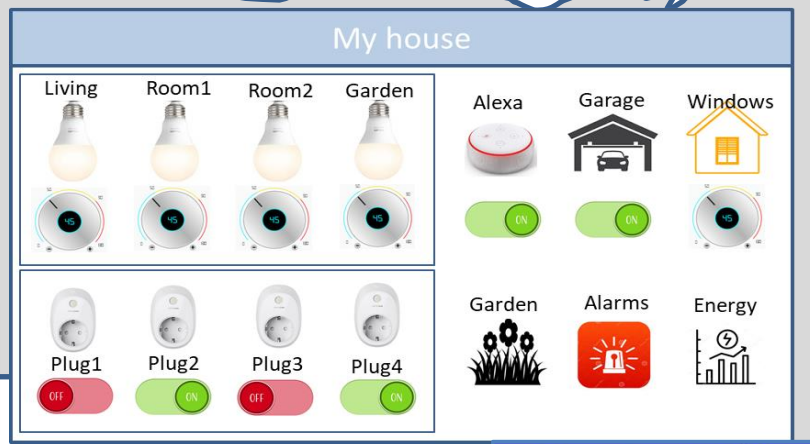
Alarm sound
and light

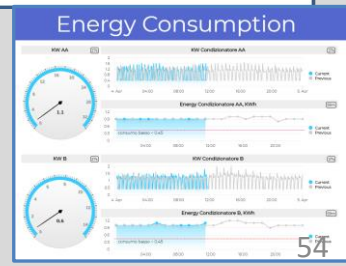
Controlling
Irrigators





SNAP4CITY

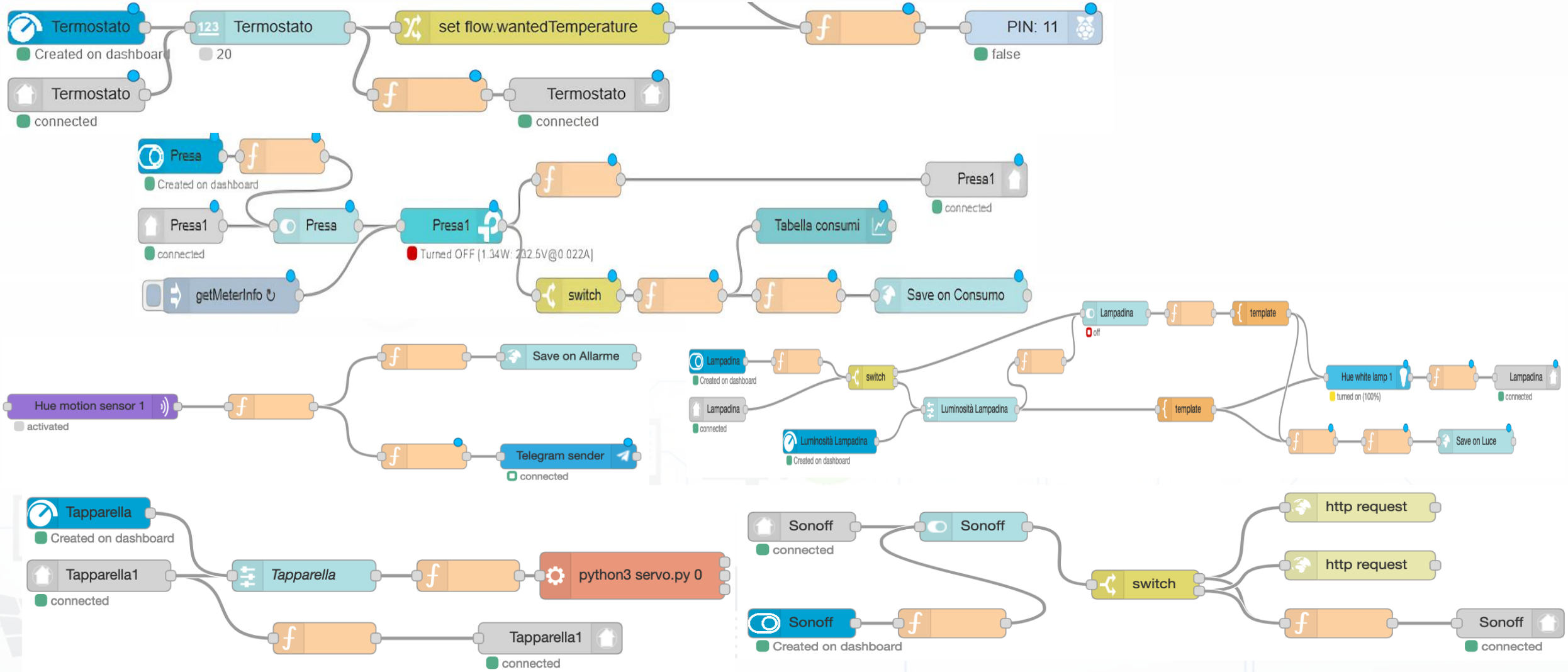




Environmental Contextual data from the city
Historical Data, Remote Control, Mobile App

https://www.snap4city.org/620

Example: IOT App on Snap4Home



Snap4Home

Hue Hub



Motion Control / Alarm



TP Link plugs: meter

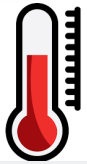


Alexa: Voice Control











IOT Edge:

Raspberry pi: Node-RED + Snap4City MicroService Library

5G gateway

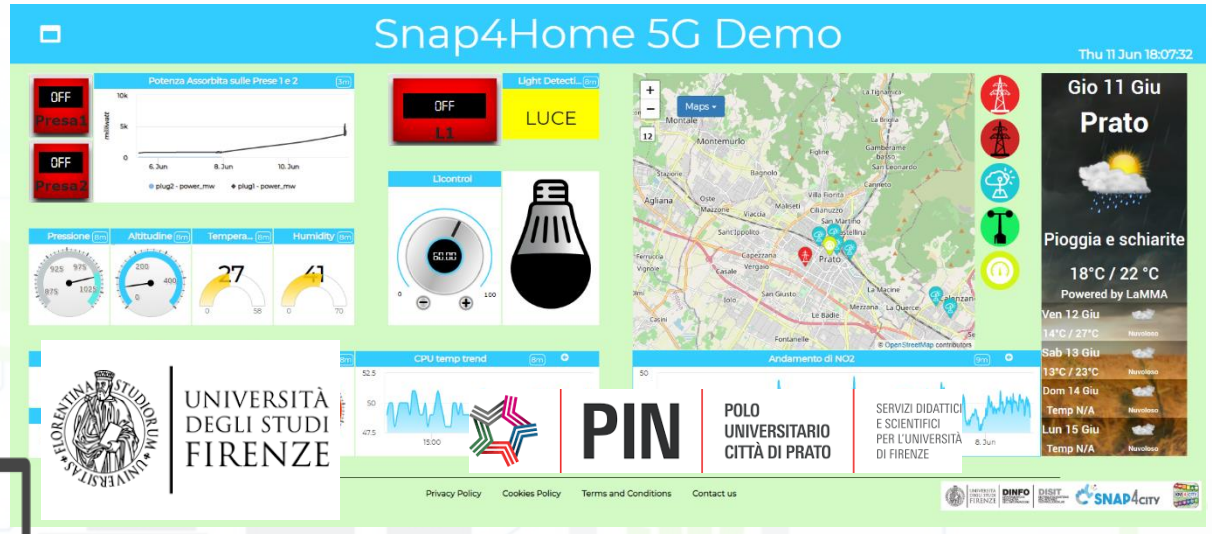
FIWARE Orion Broker

Dashboards

Advanced Smart City API

Environmental Contextual data from the city. Historical Data, Remote Control, Mobile App





open fiber WIND 3 ZTE estra

https://www.snap4city.org/369

Snap4Home

Philips Hue: Controlling Lights



Hue Hub

Hue: Motion Control / Alarm



TP Link: Controlling / Measuring Energy Plugs



Alexa: Voice Control



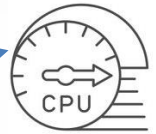


IOT Edge:

Raspberry pi: Node-RED + Snap4City

Measuring: Temperature, Humidity, light in the room



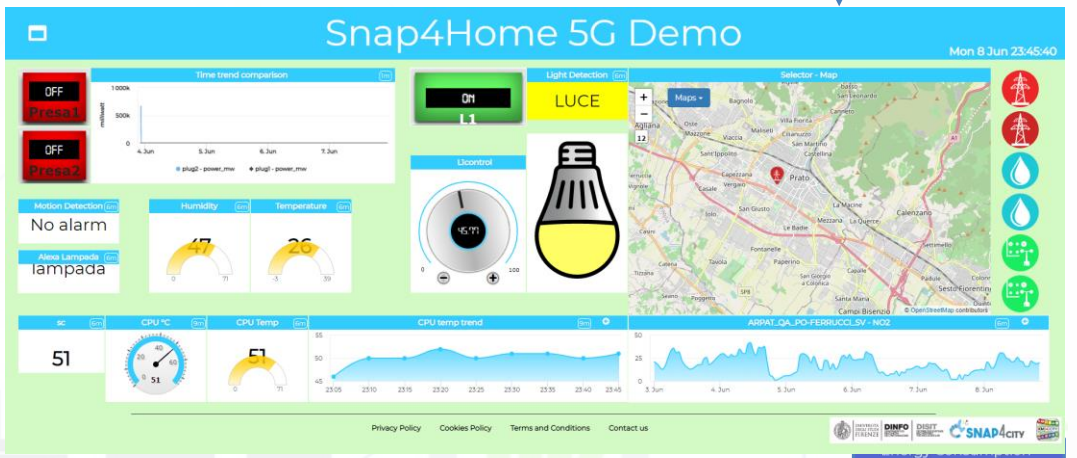


Monitoring: CPU clock, status

5G gateway

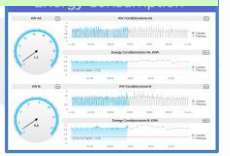
Internet

SNAP4CITY





Environmental Contextual data from the city.
Historical Data, Remote Control, Mobile App



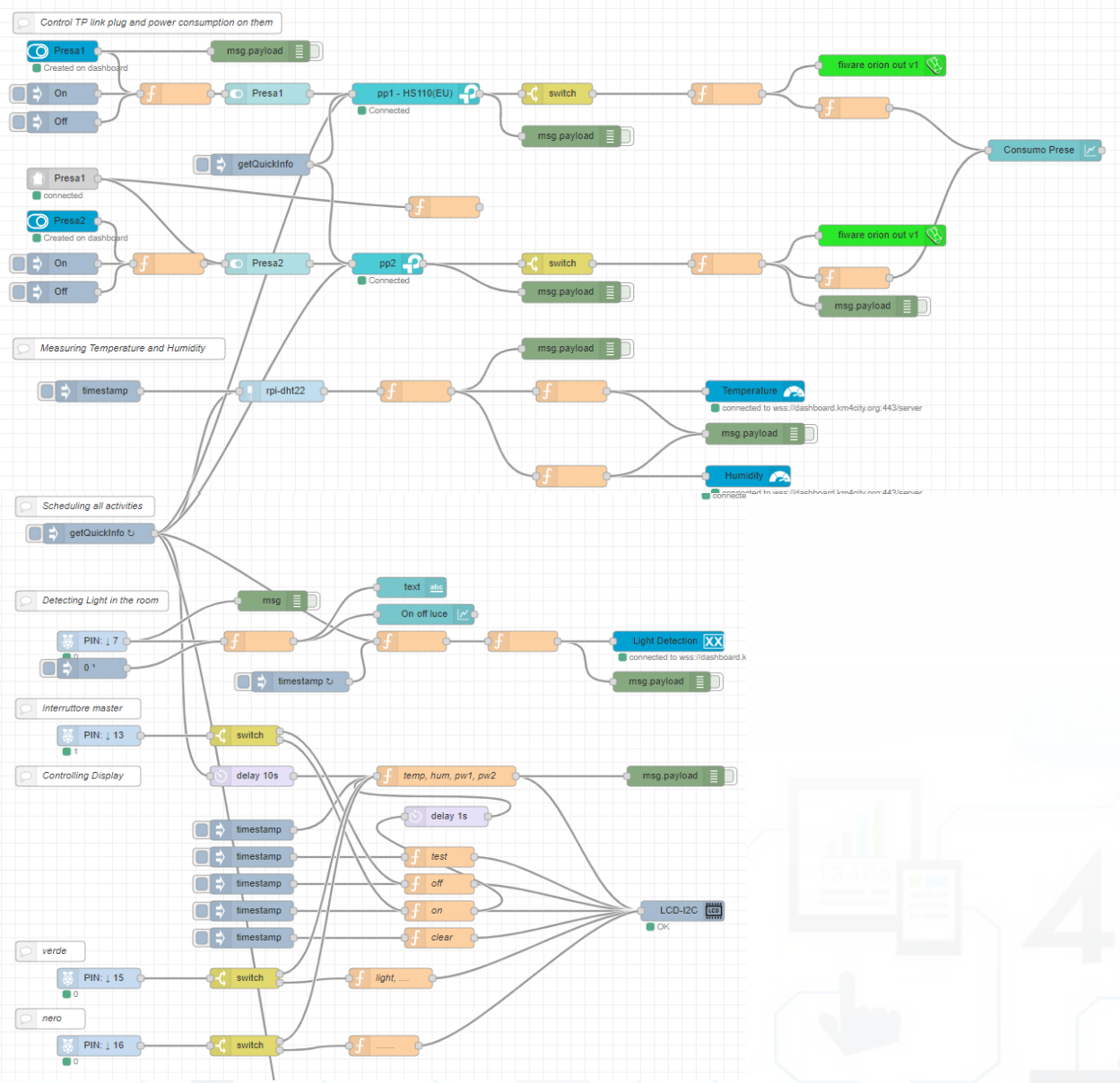
Snap4Home 5G Demo

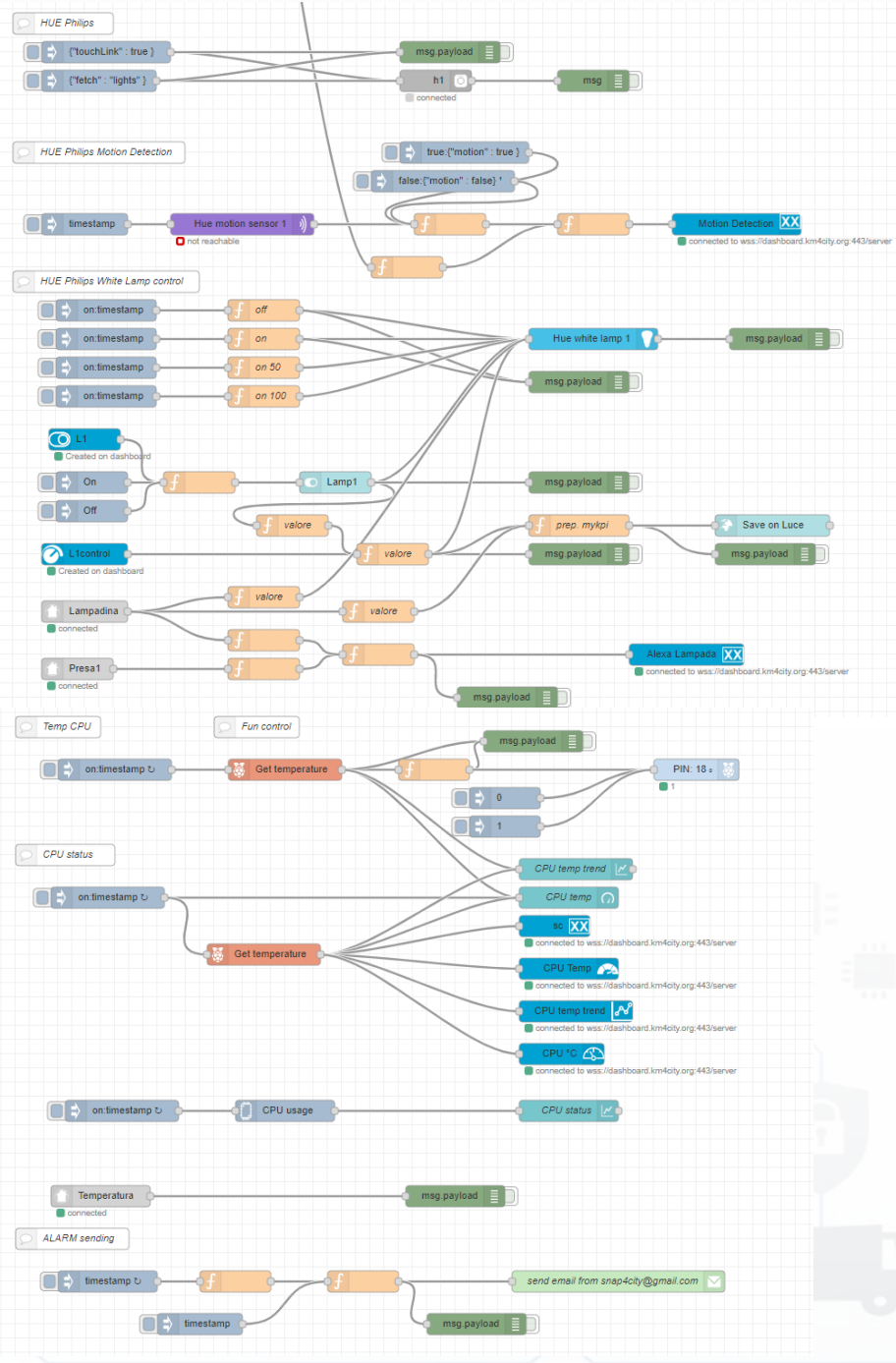
Thu 11 Jun 18:07:32



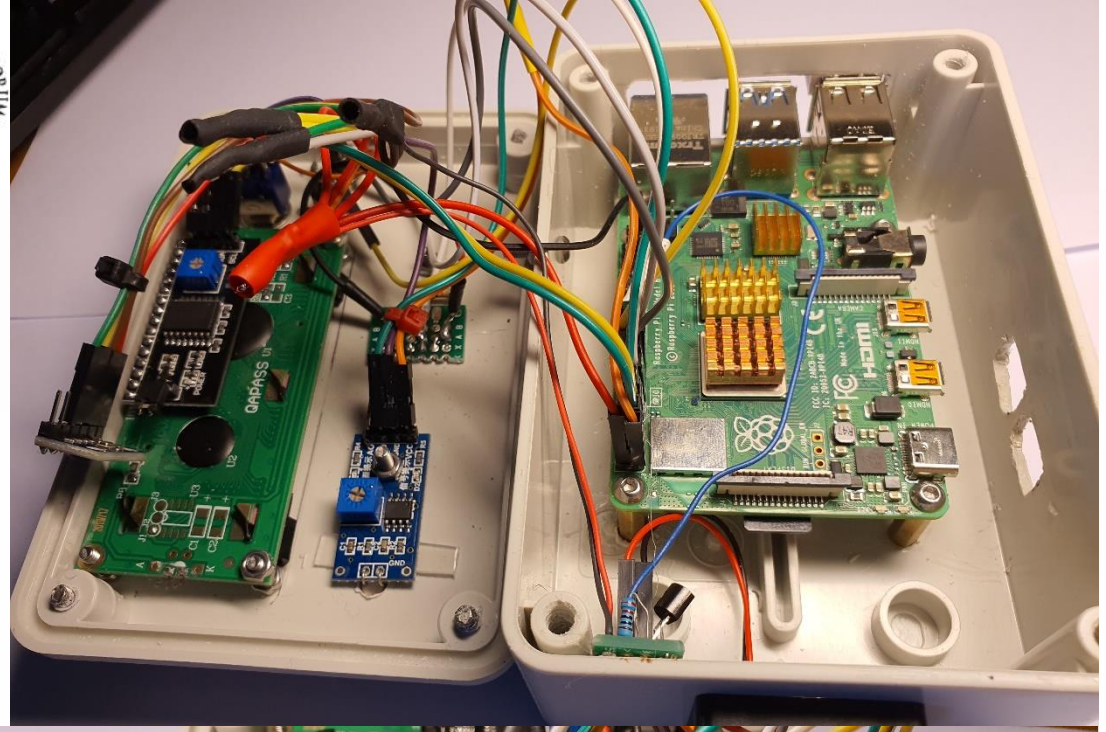
Privacy Policy Cookies Policy Terms and Conditions Contact us

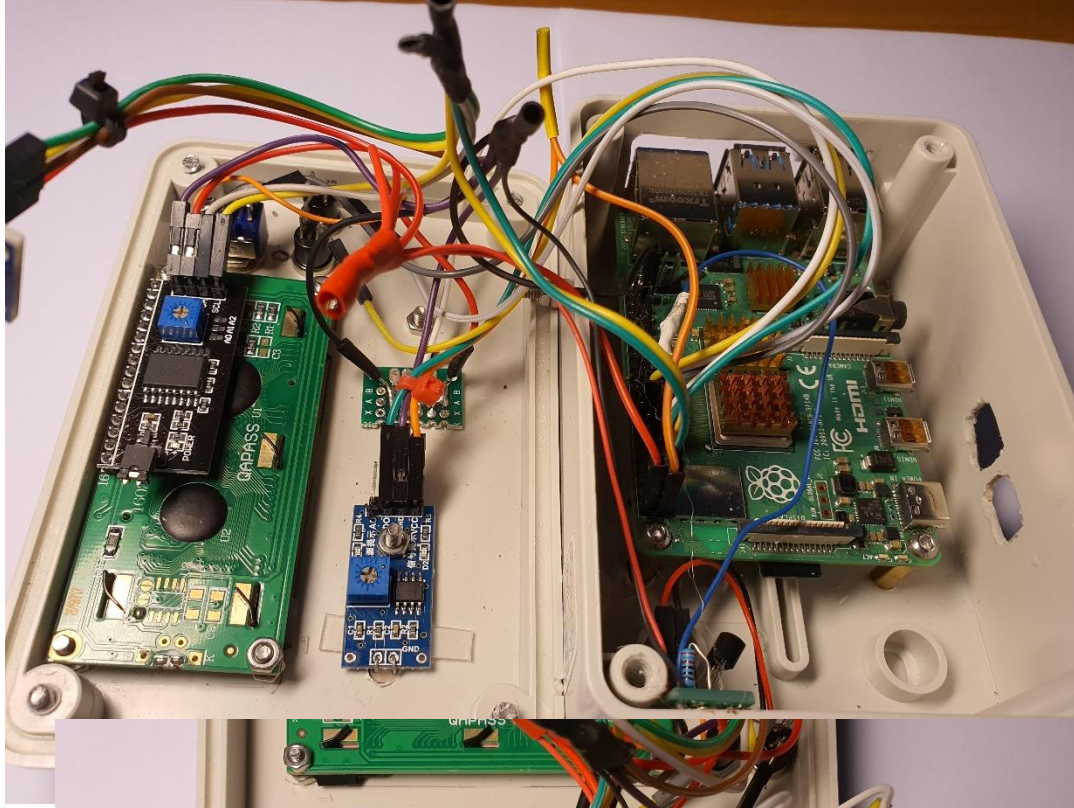
open fiber WIND 3 ZTE estra

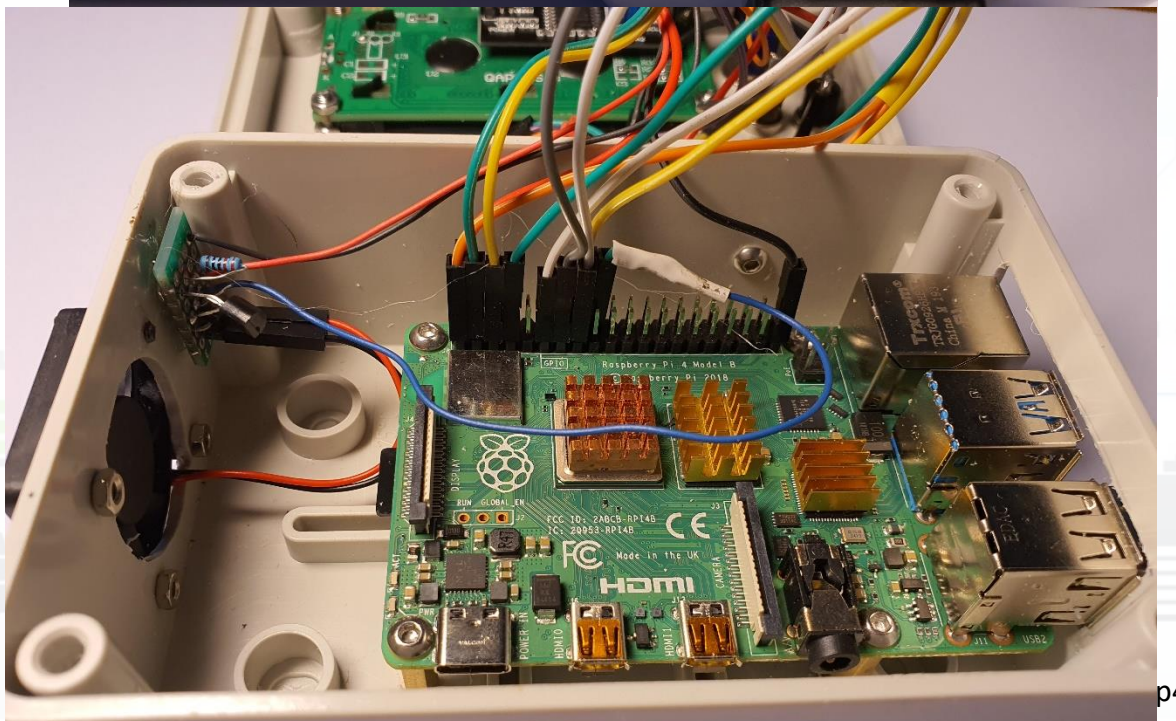

Control TP link plug and power consumption on them
Presa1
Created on dashboard
msg.payload
On
Off
Presa1
pp1 - HS110(EU)
Connected
switch
fiware orion out v1
msg.payload
Consumo Prese
getQuickInfo
Presa1
connected
Presa2
Created on dashboard
On
Off
Presa2
pp2
Connected
switch
fiware orion out v1
msg.payload
Measuring Temperature and Humidity
timestamp
rpi-dht22
msg.payload
Temperature
connected to wss://dashboard.km4city.org:443/server
msg.payload
Humidity
Scheduling all activities
getQuickInfo
Detecting Light in the room
msg
text
On off luce
PIN: ↓ 7
0
timestamp
Light Detection
connected to wss://dashboard.k
msg.payload
Interruttore master
PIN: ↓ 13
1
switch
Controlling Display
delay 10s
temp, hum, pw1, pw2
msg.payload
delay 1s
timestamp
timestamp
timestamp
timestamp
timestamp
test
off
on
clear
LCD-I2C
OK
verde
PIN: ↓ 15
0
switch
light, ...
nero
PIN: ↓ 16
0
switch
.....

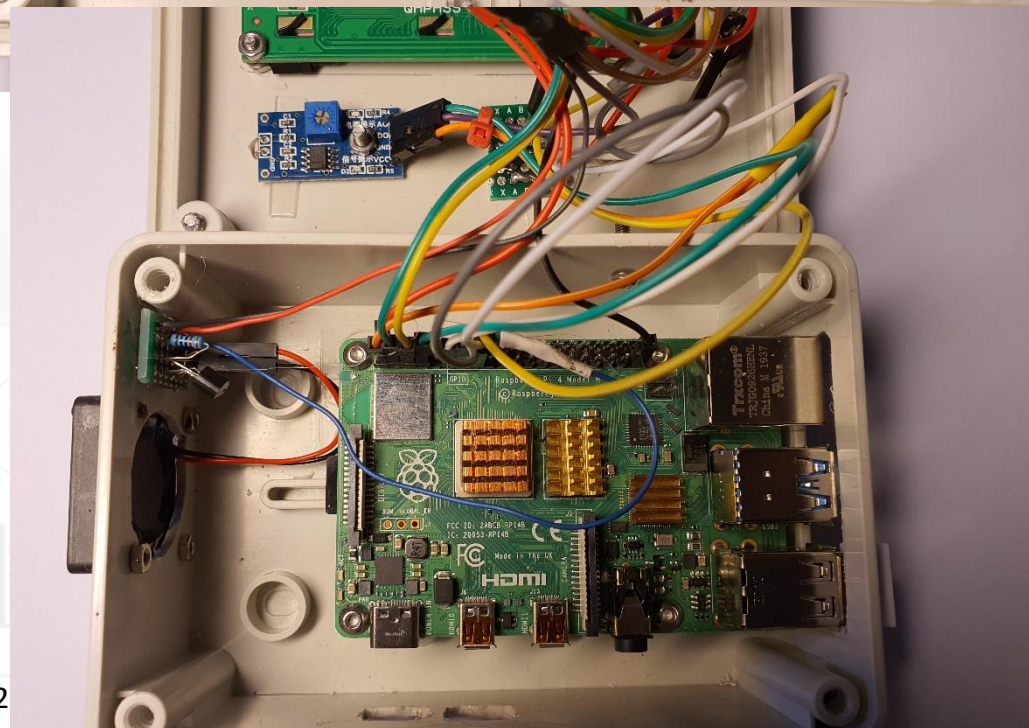

HUE Philips
{"touchLink" : true }
{"fetch" : "lights" }
msg.payload
h1
connected
msg
HUE Philips Motion Detection
true:{"motion" : true }
false:{"motion" : false} '
timestamp
Hue motion sensor 1
not reachable
Motion Detection
connected to wss://dashboard.km4city.org:443/server
HUE Philips White Lamp control
on:timestamp
on:timestamp
on:timestamp
on:timestamp
off
on
on 50
on 100
Hue white lamp 1
msg.payload
msg.payload
L1
Created on dashboard
On
Off
Lamp1
msg.payload
valore
prep. mykpi
Save on Luce
msg.payload
L1control
Created on dashboard
valore
valore
msg.payload
Lampadina
connected
valore
valore
Presa1
connected
Alexa Lampada
connected to wss://dashboard.km4city.org:443/server
msg.payload
Temp CPU
Fun control
on:timestamp
Get temperature
msg.payload
PIN: 18
1
0
1
CPU status
on:timestamp
Get temperature
CPU temp trend
CPU temp
sc
connected to wss://dashboard.km4city.org:443/server
CPU Temp
connected to wss://dashboard.km4city.org:443/server
CPU temp trend
connected to wss://dashboard.km4city.org:443/server
CPU °C
connected to wss://dashboard.km4city.org:443/server
on:timestamp
CPU usage
CPU status
Temperatura
connected
msg.payload
ALARM sending
timestamp
timestamp
send email from snap4city@gmail.com
msg.payload


T:27'C; Hum:43%
P1:NaNW;P2:NaNW
SNAP4CITY
PHILIPS
tp-link


QAPASS


QAPASS


Raspberry Pi 4 Model B
FCC ID: 2ABCB-RPI4B
IC: 20953-RPI4B
Made in the UK
HDMI

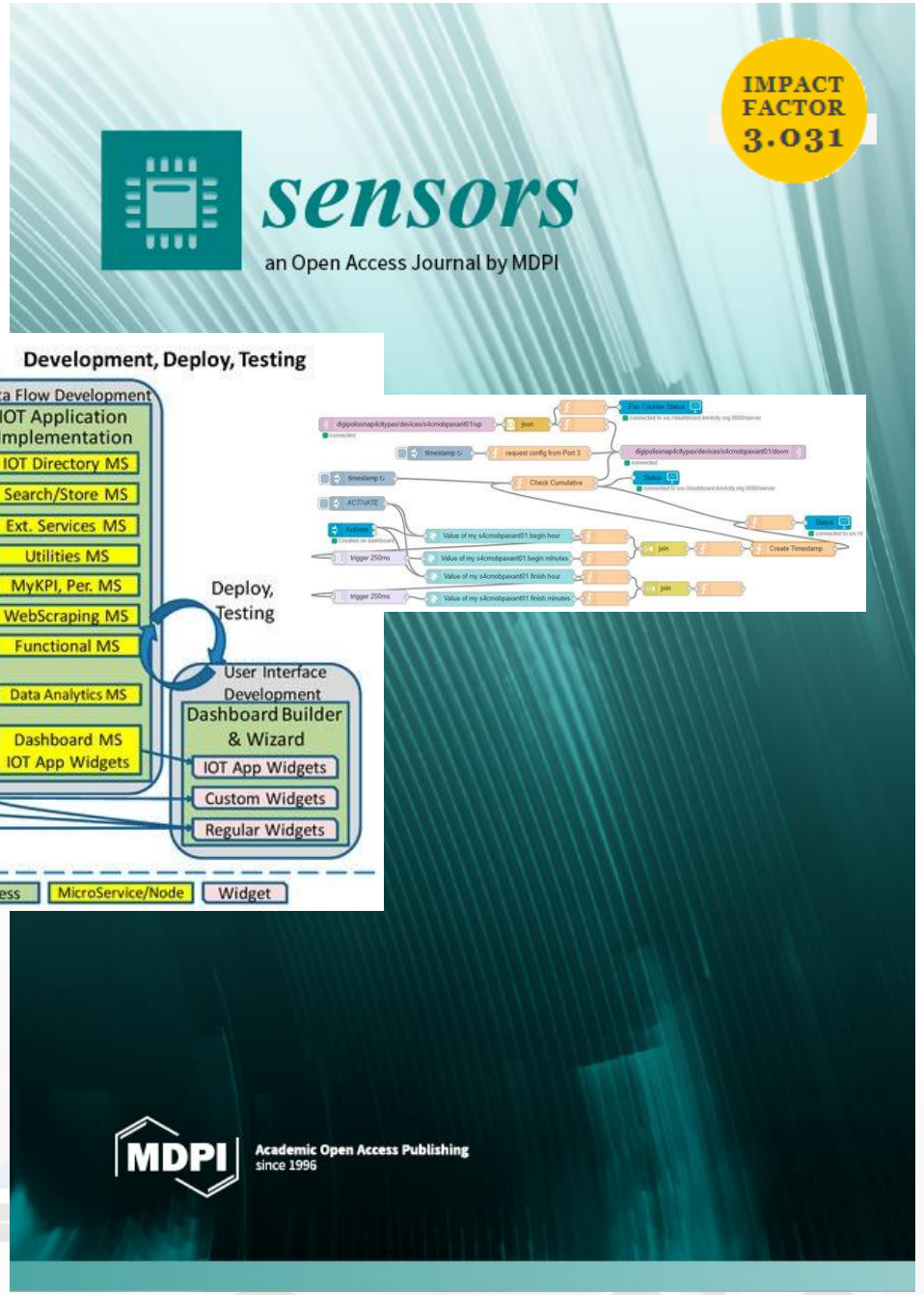

QAPASS
HDMI

IOT App vs Smart City Solutions



MicroServices Suite for Smart City

- Badii, C.; Bellini, P.; Difino, A.; Nesi, P.; Pantaleo, G.; Paolucci, M. MicroServices Suite for Smart City Applications.
- *Sensors* **2019**, *19*, 4798.
- https://www.mdpi.com/1424-8220/19/21/4798/pdf

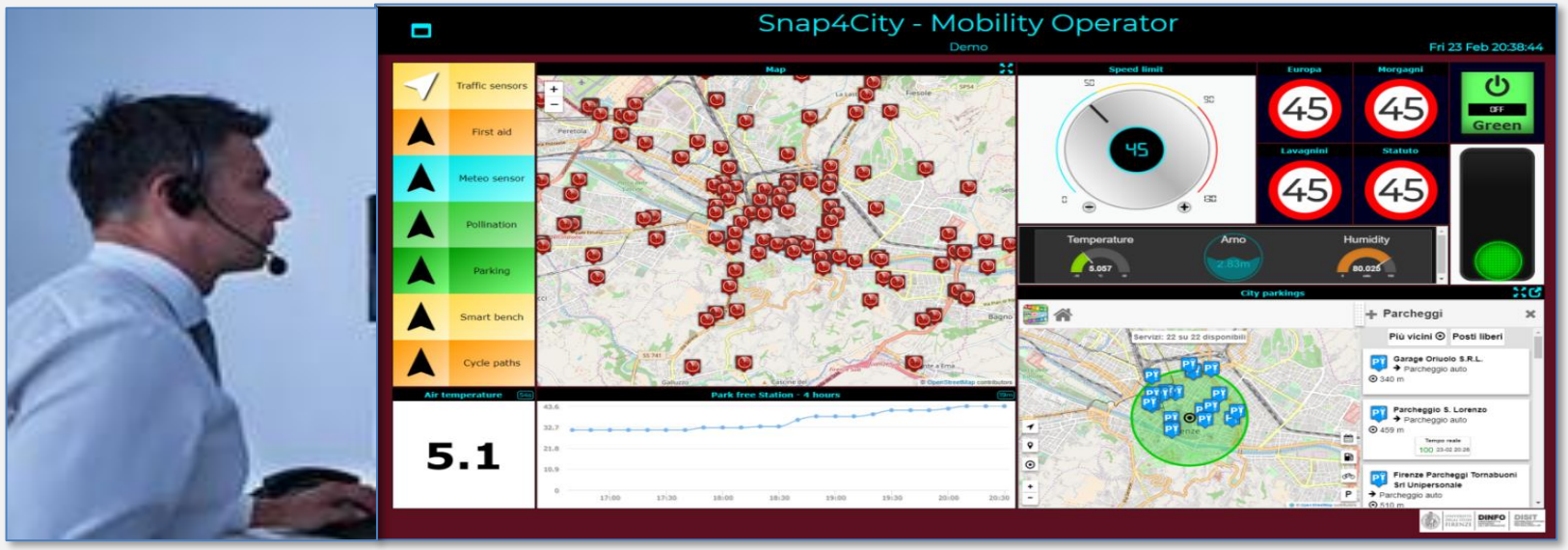


City Dashboard + IOT App

Control Room Operator

Would like to:

- **Monitor** traffic flow, Environment, Car parking, Cycling, First aid, temp., ..
- **Act and** monitor Dynamic Plates
- **Act and** monitor red lights

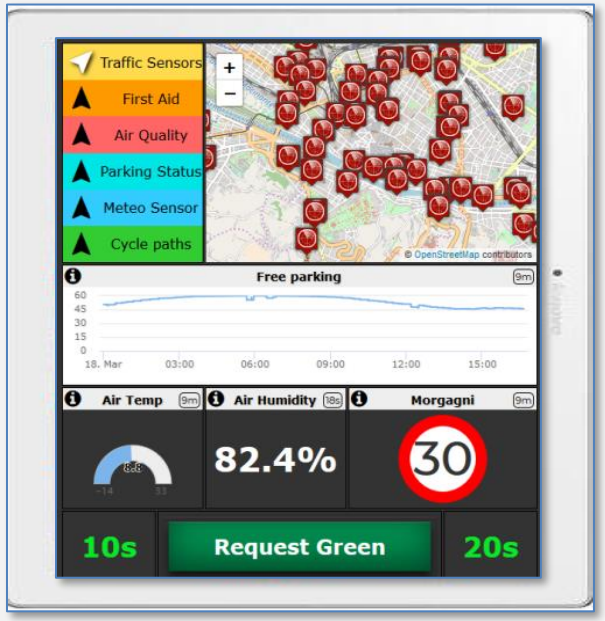


Driver, Policeman

Would like to:

- Monitor traffic, Parking, traffic events, speed limit, …
- **Act and** monitor red lights





Dashboards with city data and your data/actuators

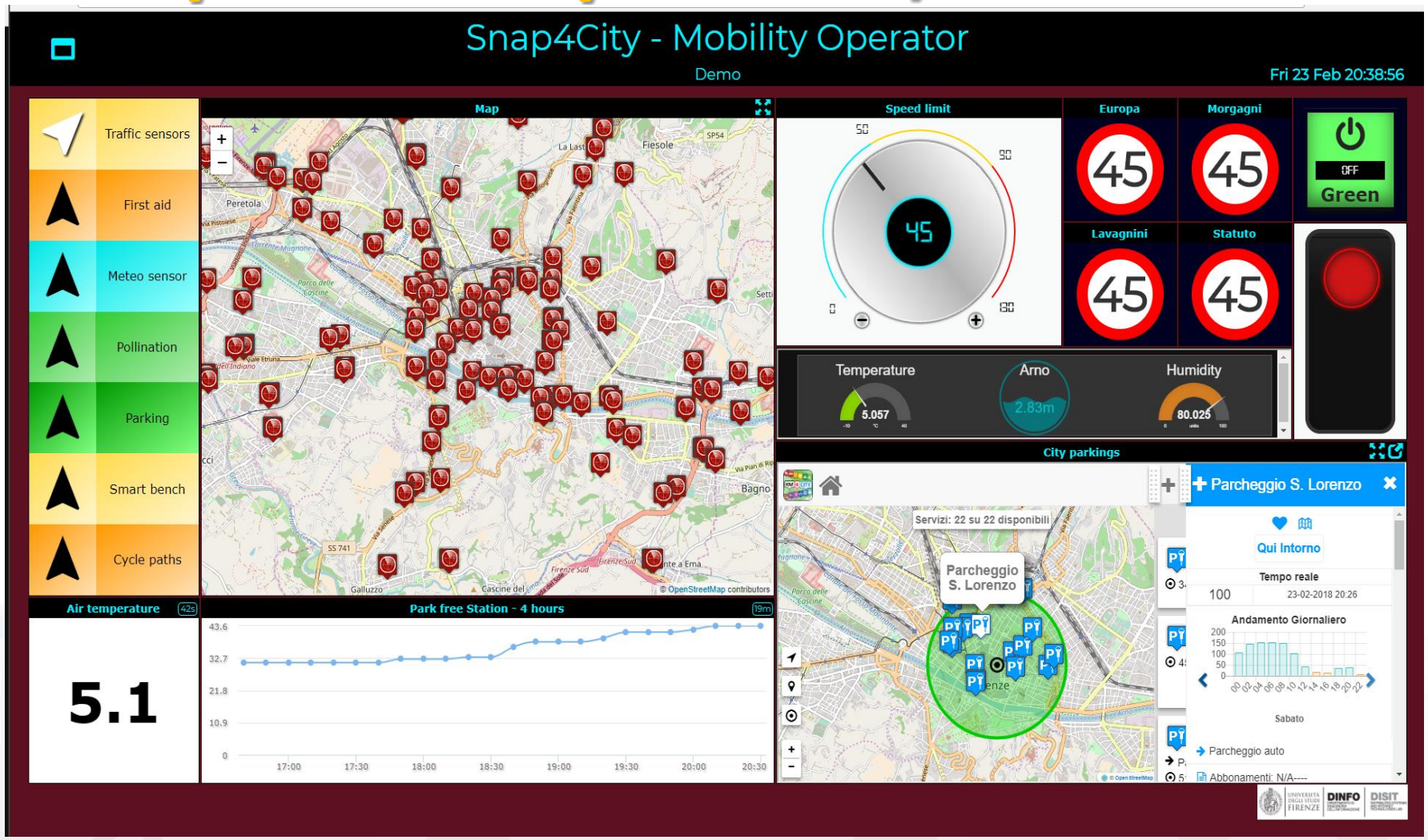
Sensors:

- Values
- Status

Actuators:

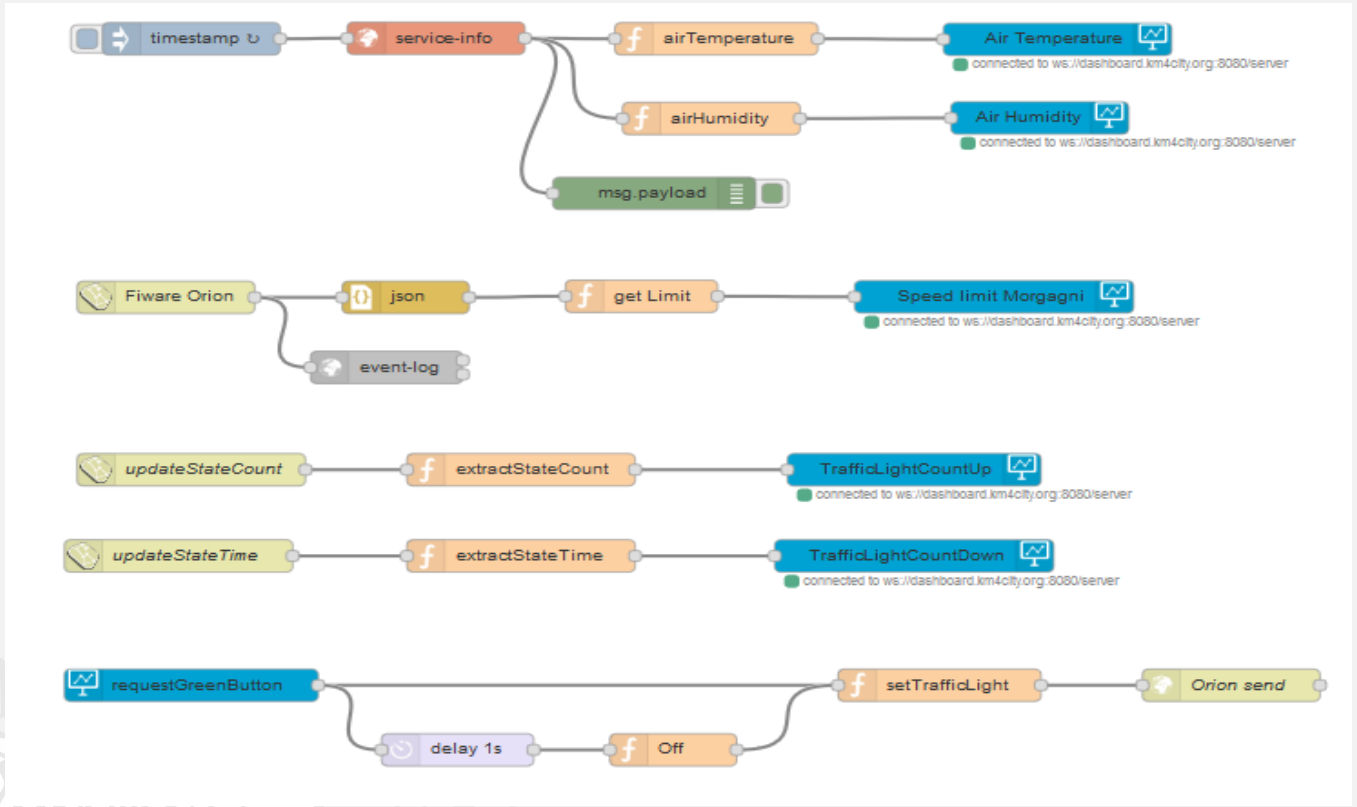
- Buttons
- Dimers
- Etc.

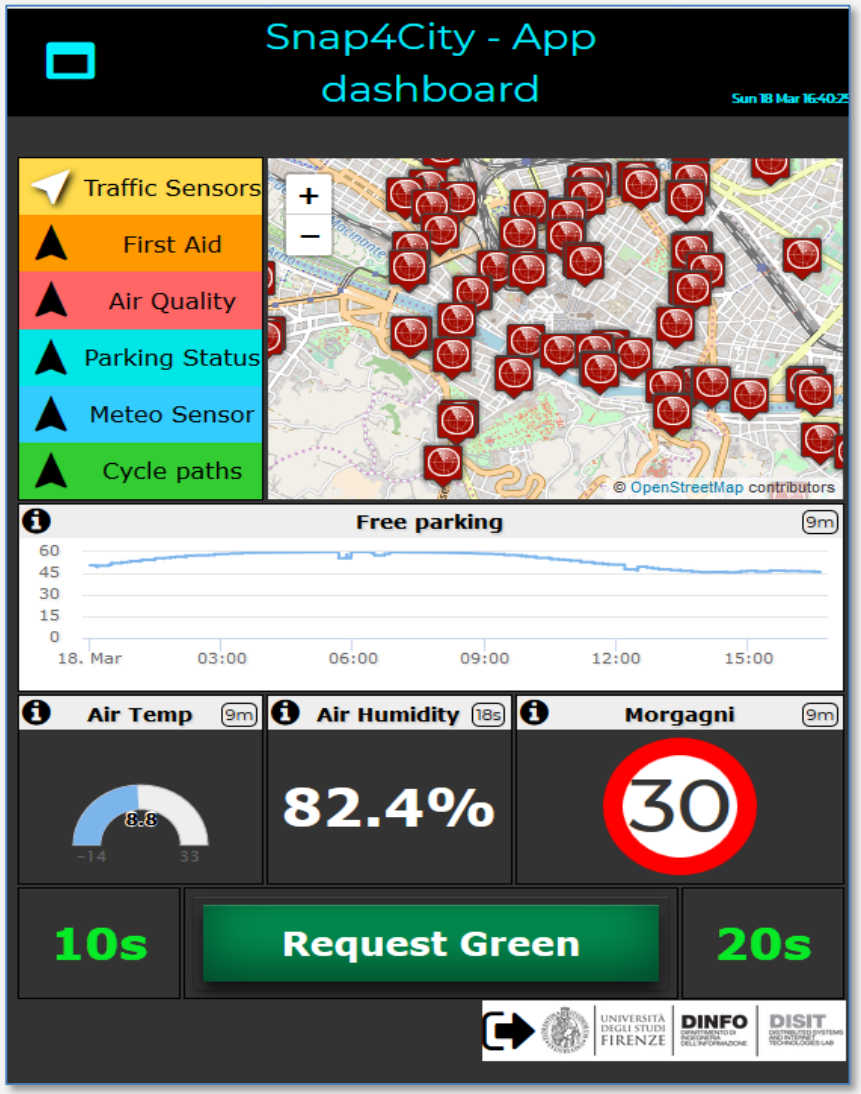
Virtual Sensors and Actuators

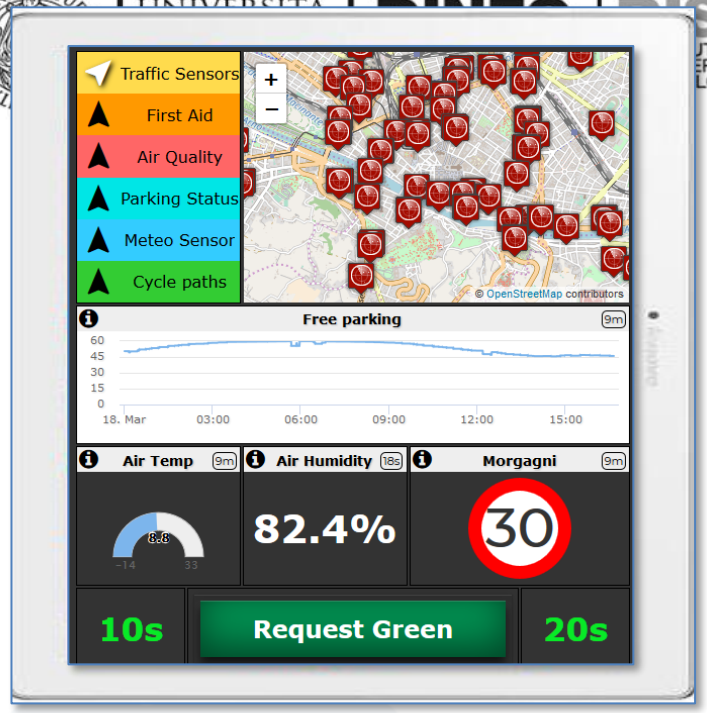


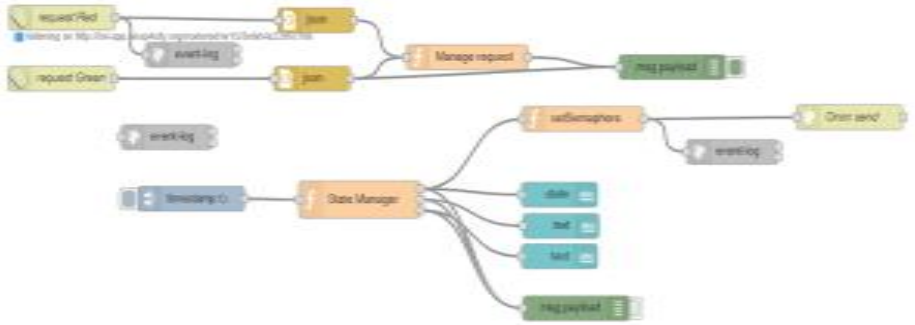
IOT Application with City Dashboard simple development








Traffic Sensors
First Aid
Air Quality
Parking Status
Meteo Sensor
Cycle paths
© OpenStreetMap contributors
Free parking
9m
60
45
30
15
0
18. Mar
03:00
06:00
09:00
12:00
15:00
Air Temp
9m
Air Humidity
18s
Morgagni
9m
82.4%
30
10s
Request Green
20s







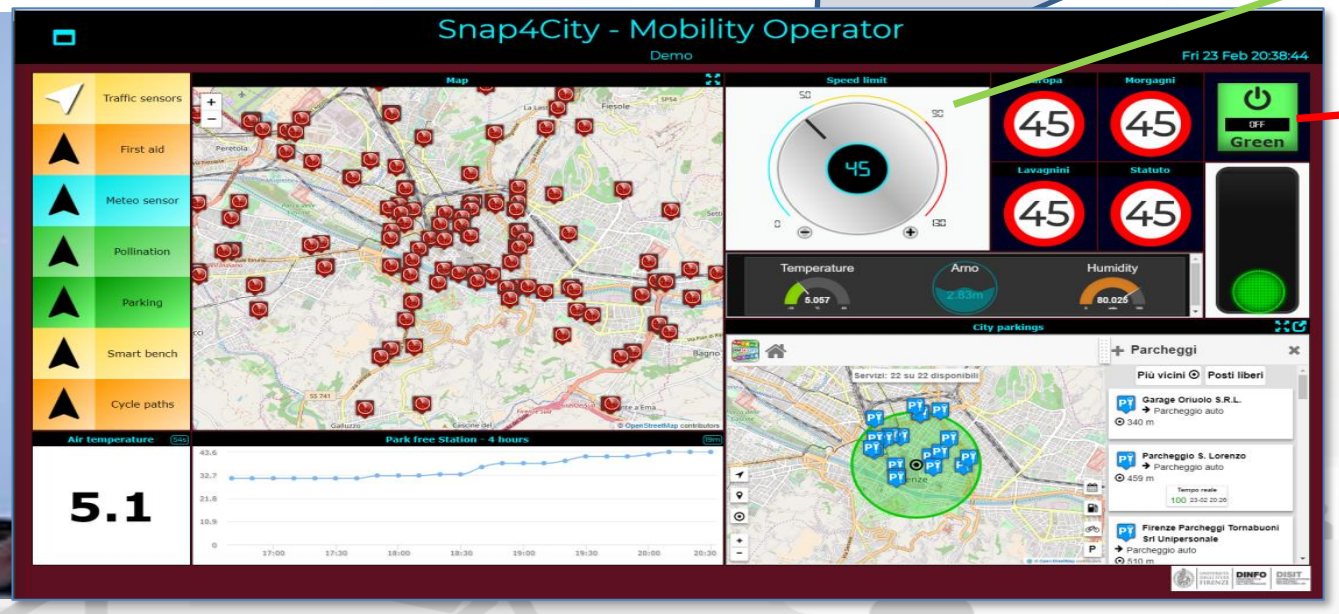

30

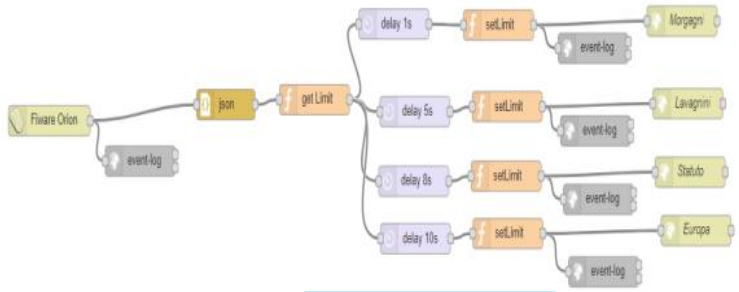

30


30


30




Snap4City - Mobility Operator
Demo
Fri 23 Feb 20:38:44
Traffic sensors
First aid
Meteo sensor
Pollination
Parking
Smart bench
Cycle paths
Map
Speed limit
45
Europa
45
Morgagni
45
Lavagnini
45
Statuto
45
OFF
Green
Temperature
6.057
Arno
2.83m
Humidity
80.025
City parkings
Parcheggi
Più vicini
Posti liberi
Garage Oriuolo S.R.L.
Parcheggio auto
340 m
Parcheggio S. Lorenzo
Parcheggio auto
459 m
Firenze Parcheggi Tornabuoni Srl Unipersonale
Parcheggio auto
Air temperature
5.1
Park free Station - 4 hours

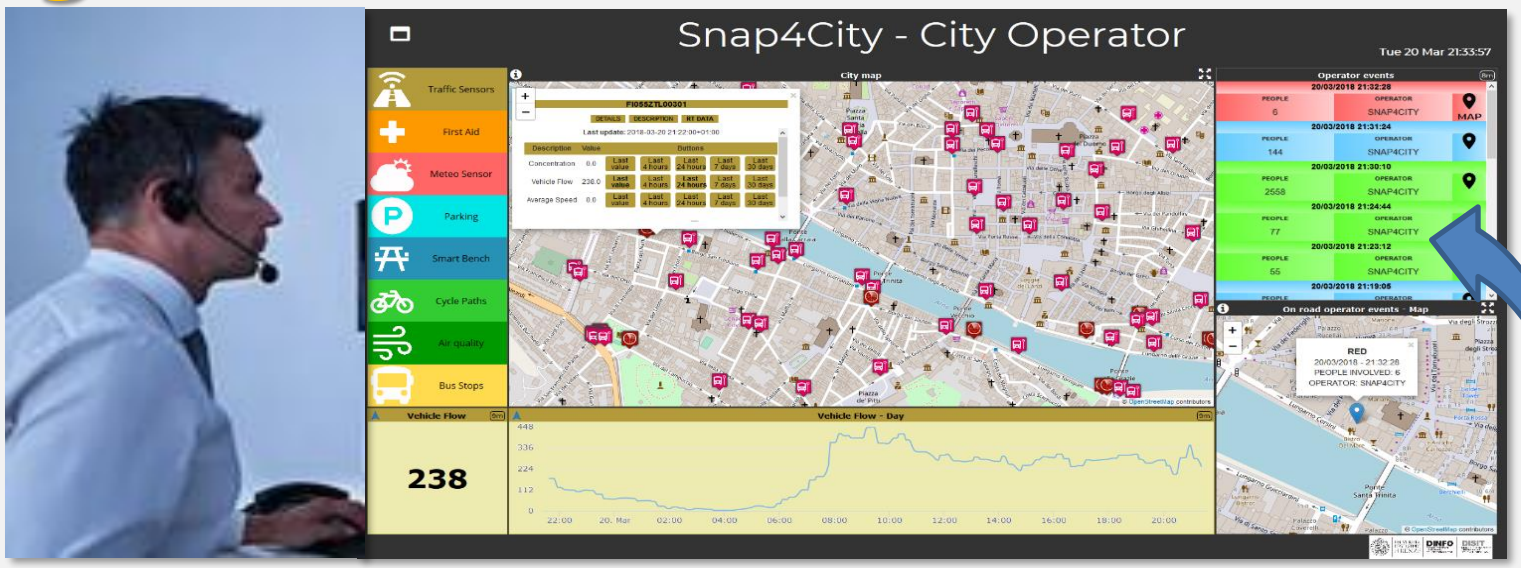

Fiware Orion
event-log
json
get Limit
delay 1s
delay 5s
delay 8s
delay 10s
setLimit
setLimit
setLimit
setLimit
event-log
event-log
event-log
event-log
Morgagni
Lavagnini
Statuto
Europa

Reporting Critical Events

Control Room Operator

Would like to:

- **Monitor** events vs services in the city and receive critical event notifications from on the road operators.
- **Assess contextual condition,** services status



On the road operator

Would like to:

- Monitor data of traffic, Parking, environment, speed limit, services,
- **Send critical event notifications via coded description**

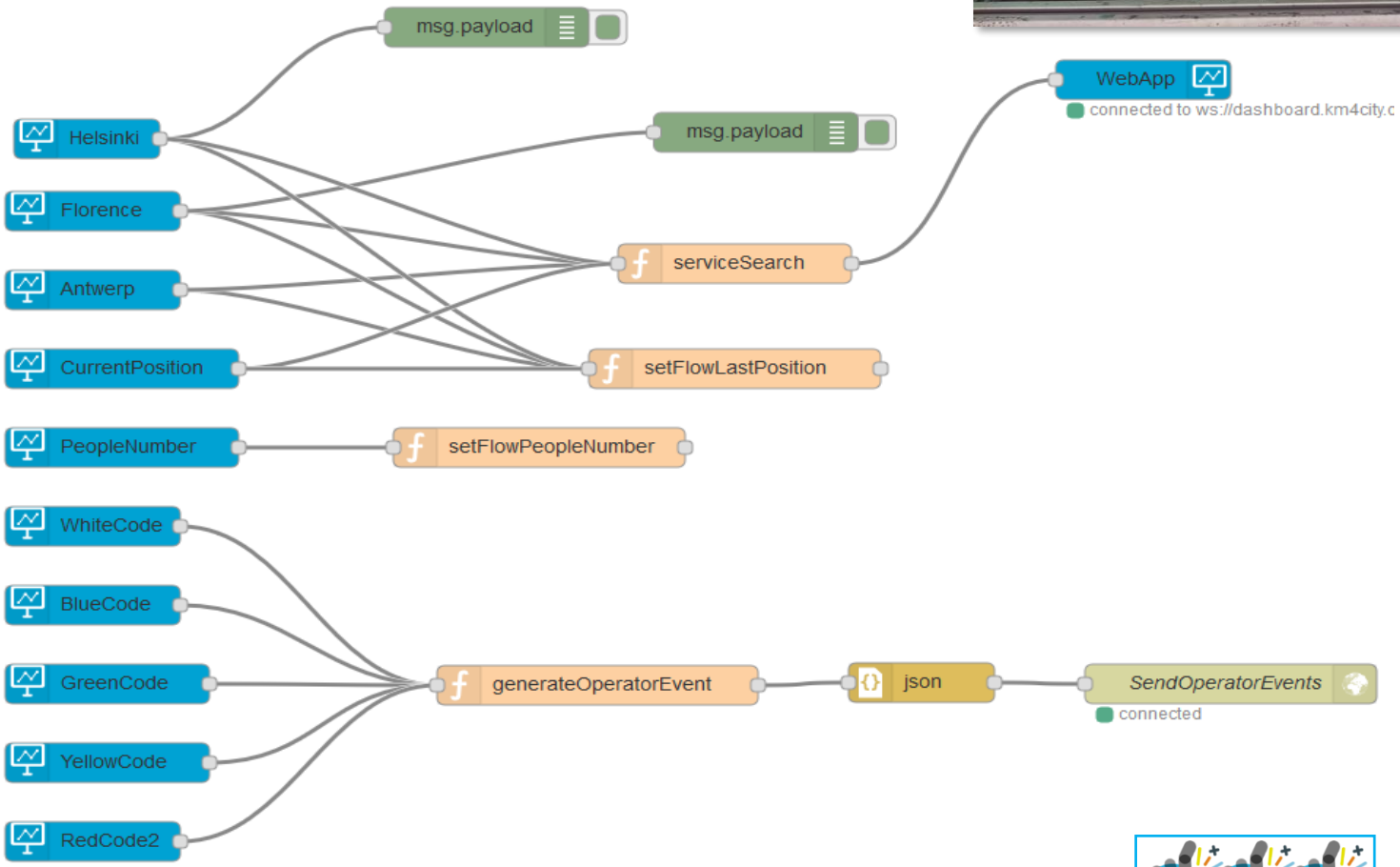




IOT Application with City Dashboard simple development







Other Cases

- *Telecontrol for water depuration plant*
- *Traffic Flow computing in Florence*
- *Snap4Home: Casa Domotica*
- *Snap4Industry: industry control*
- COVID 2019
- Computing of public transport quality
- Managing Heatmap production
- Managing Smart Parking
- Managing Smart Bike Sharing
- Telecontrol of Energy Plant



COVID 2019

Andamenti Nazionali e Regionali infezione COVID-19

Sulla base dei dati della protezione civile, elaborazioni DISITLab

Fri 3 Apr 23:11:01

per evidenziare gli andamenti di vostro interesse: eliminare le curve che non interessano selezionandole in legenda.
Alcuni dati in passato non sono pervenuti alla protezione civile

1648 nuovi_att_positivi

75528 Attualmente_positivi

14681 deceduti

19758 dimessi_guariti

LOG view

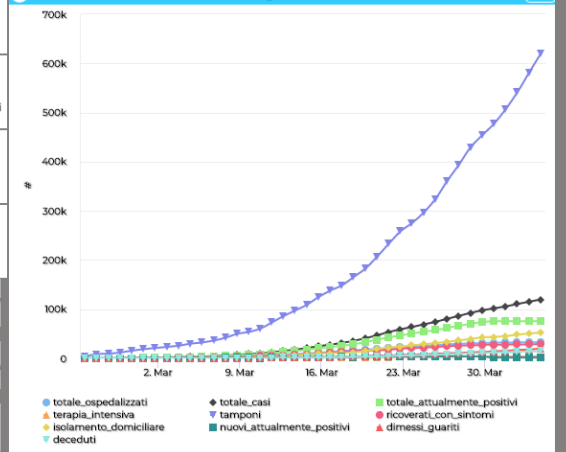
Mobile

Staked

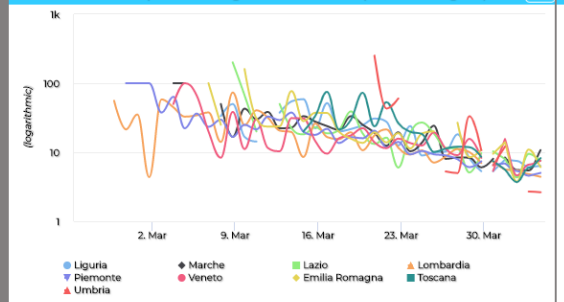
Regione Toscana

Regione LOG

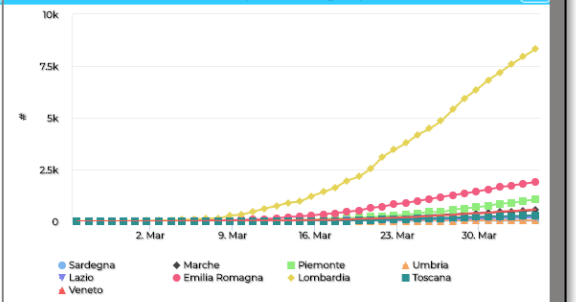
Andamento Contagio Coronavirus in Italia



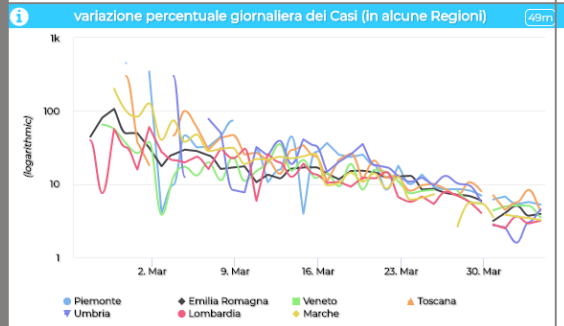
variazione percentuale giornaliera Decessi (In alcune Regioni)



Decessi (In alcune regioni)



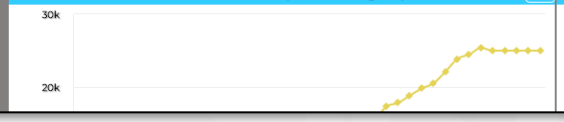
variazione percentuale giornaliera dei Casi (In alcune Regioni)

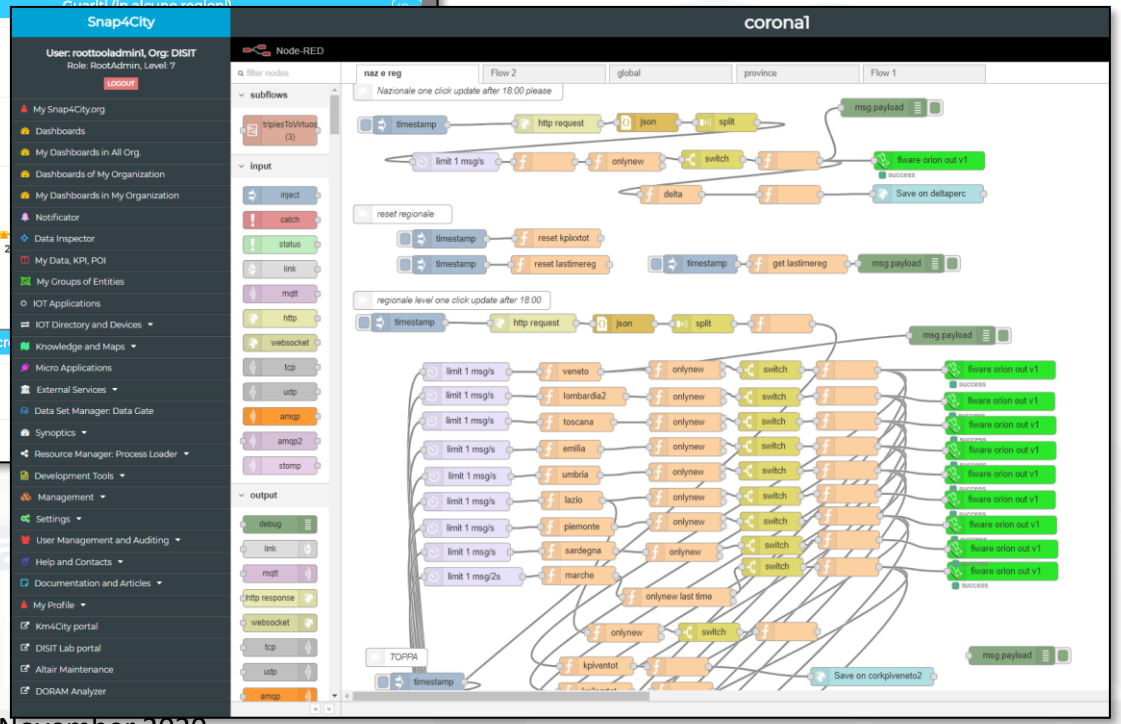


variazione percentuale giornaliera dei Casi (Nazionale)



Totale_Attuali Casi (in alcune regioni)



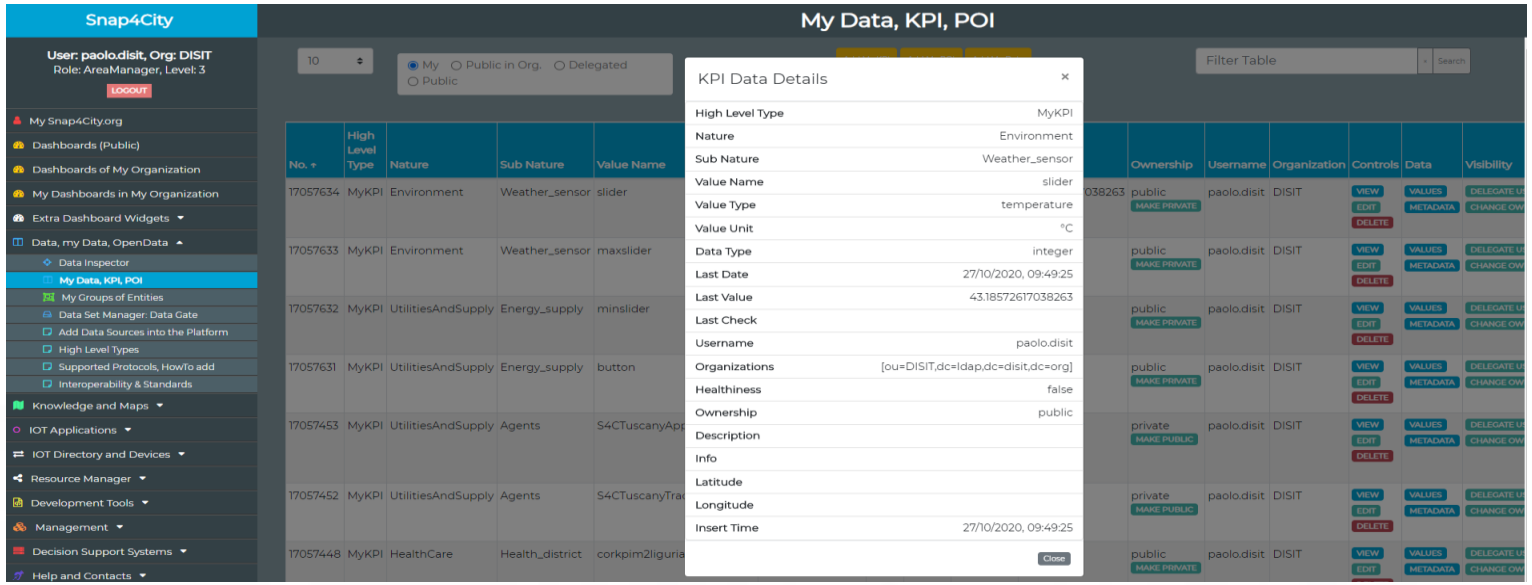


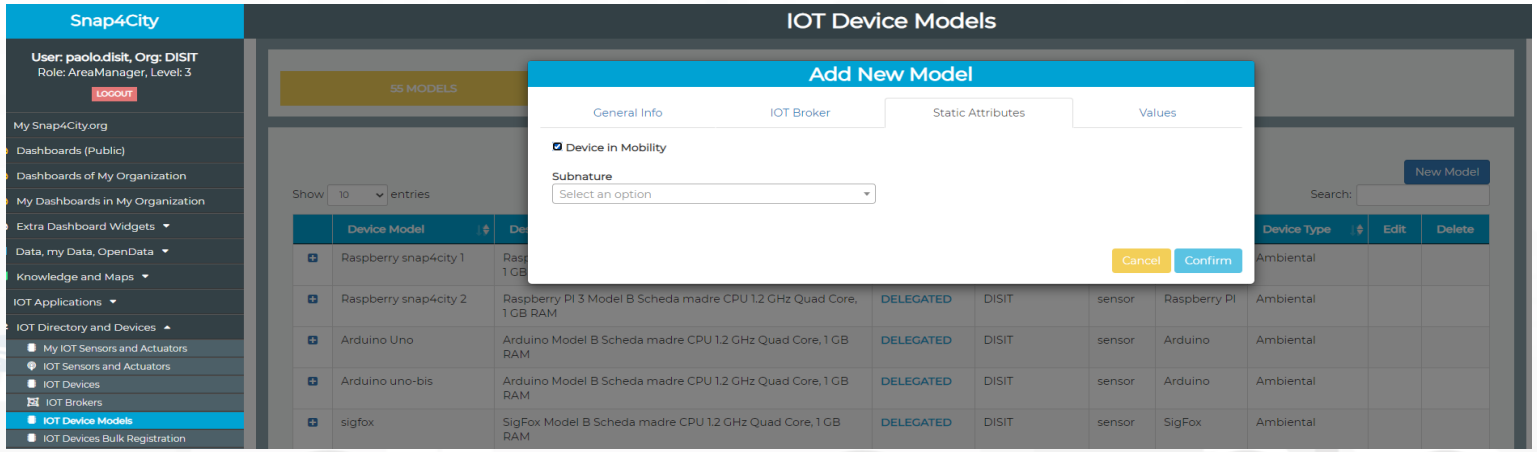
Moving IOT Devices / Sensors, Tracking Devices



Working with Sensor Data from Moving Devices

- Moving data can be collected by using:
  - **MyKPI**: in which each MyKPI has a ValueName, Unit, Type, etc.. And also GPS location
  - **IOT Device in Mobility**: which generates a new HLT SensorMobile which is partially developed so far

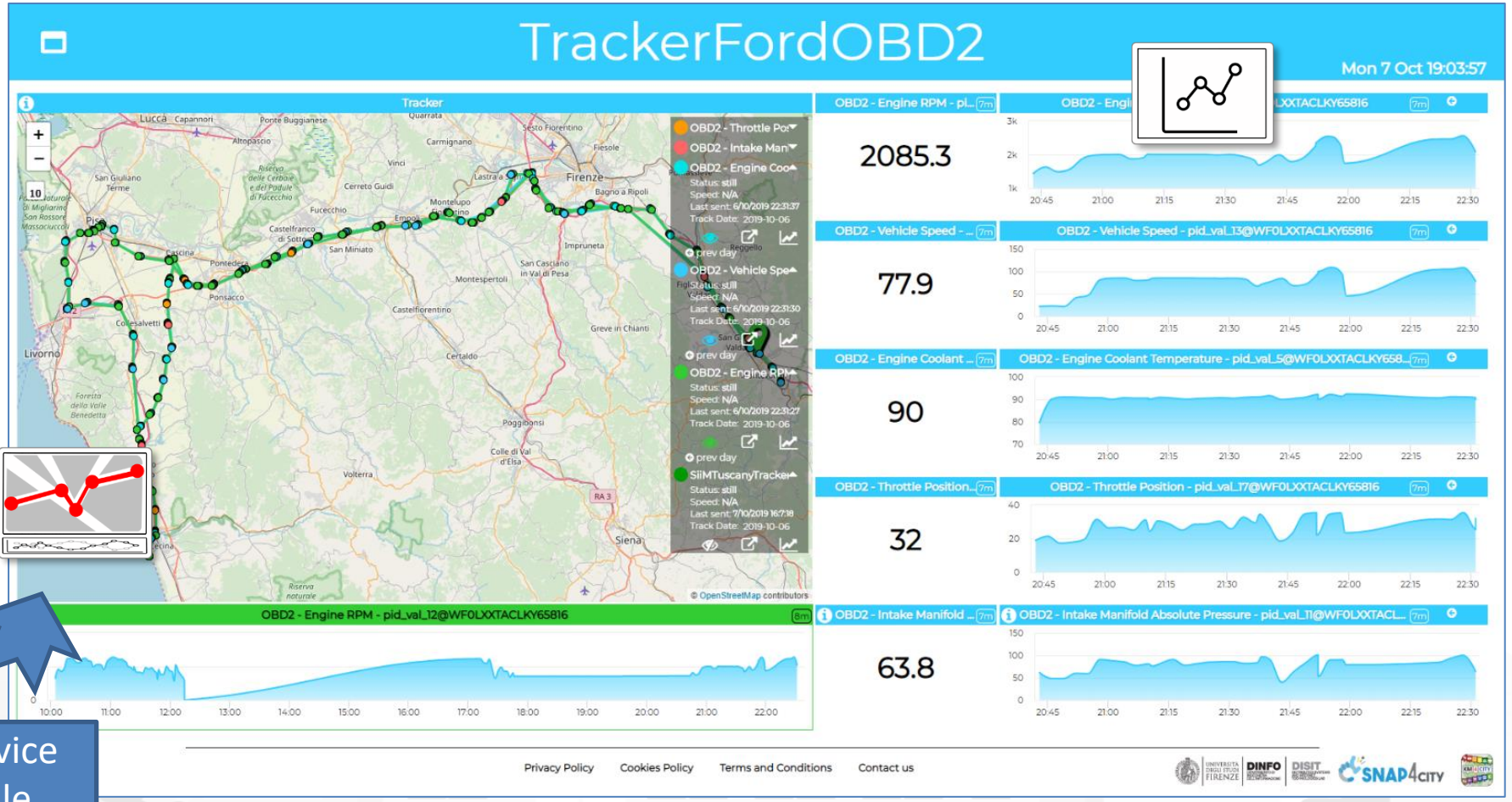




MyKPI: Tracking of Devices and Mobiles

- Real Time Trajectories for
  - Mobile Phone
  - Moving IOT Devices
  - OBU, Vehicular Kits
  - Multiple tracks
  - Day by day
- Micro Application





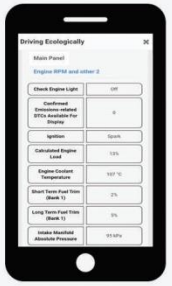
IOE – Vehicle Monitoring

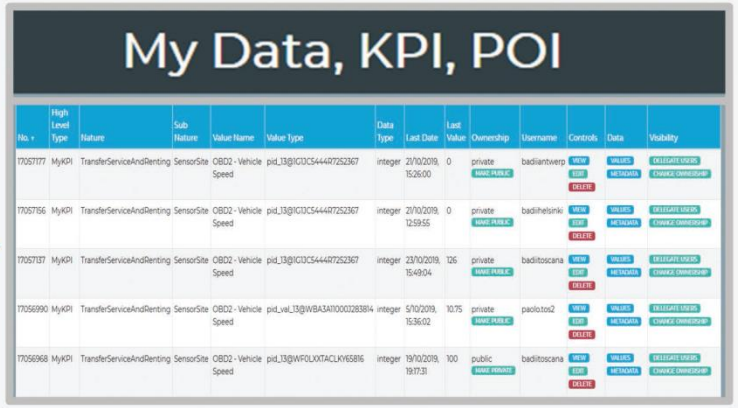


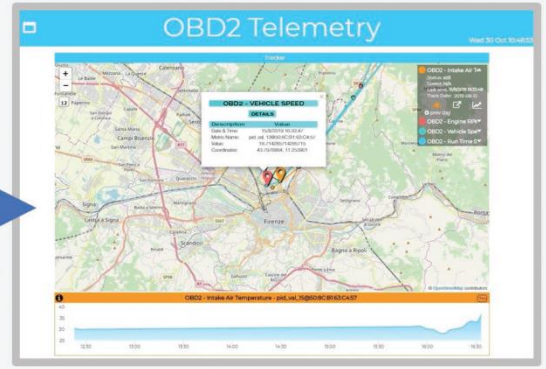
CANBUS sniffer



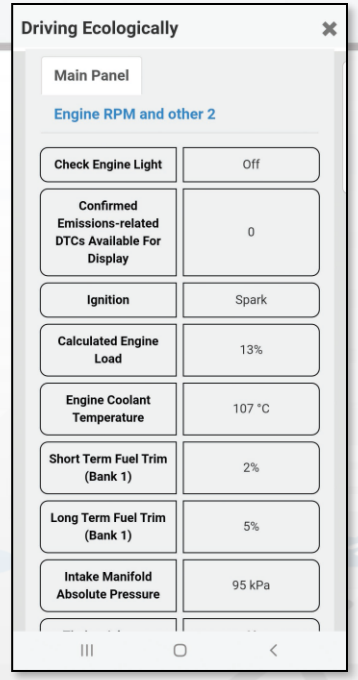
Bluetooth

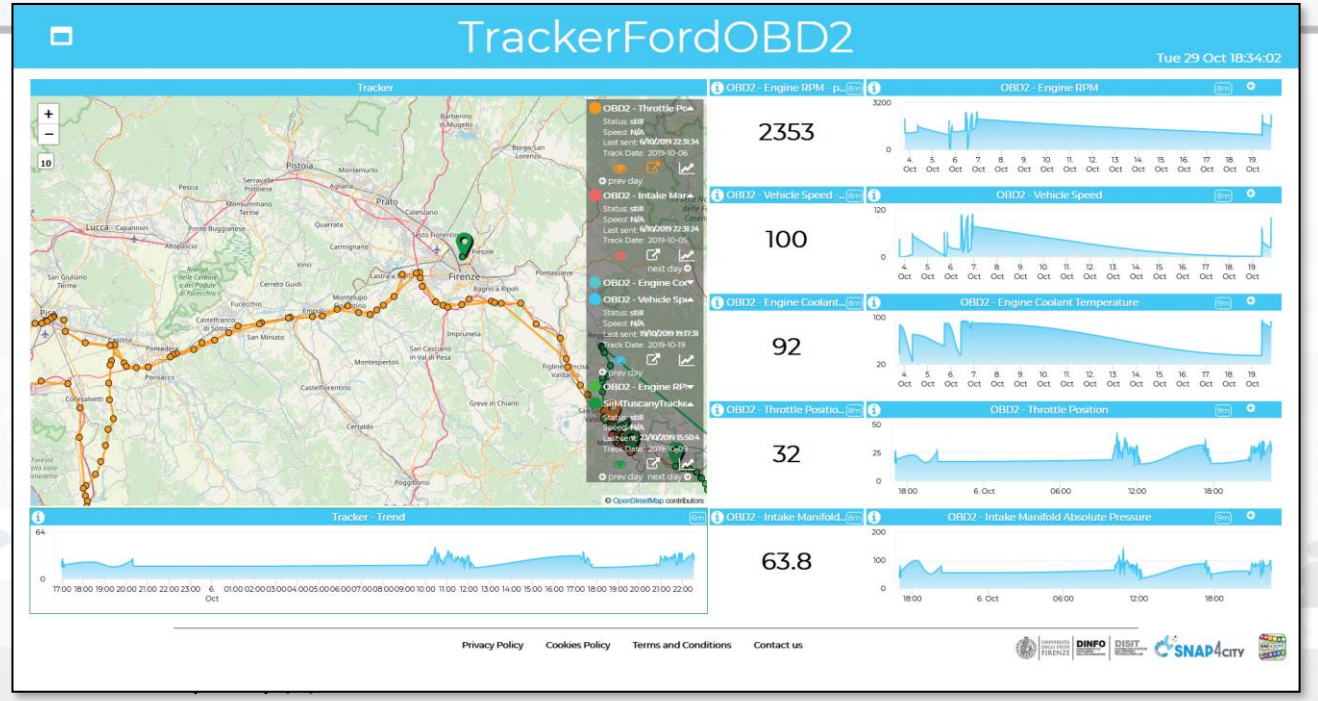






Tuscany in a Snap Mobile App on Android





PaxCounter devices







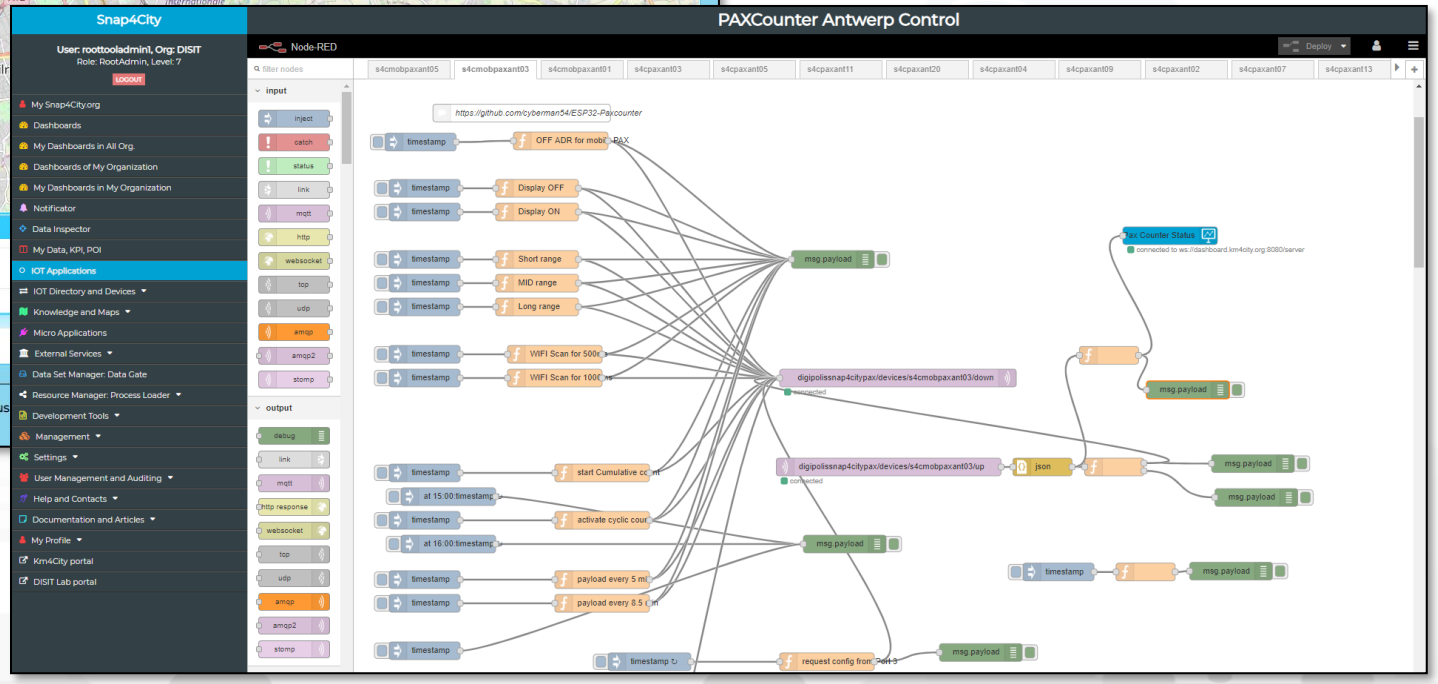
- **Fix PaxCounter LoraWan**
  - sniffing on: Wi-Fi, Bluetooth
  - Sending data via LoraWan
- **Mobile PaxCounter LoraWan**
  - sniffing on: Wi-Fi, Bluetooth
  - Sending data via LoraWan
- **Fix PaxCounter, multiple out**
  - Sending data via LoraWan and Wi-Fi
  - sniffing on: Wi-Fi, Bluetooth

LoRa® Wi Fi Bluetooth™

https://www.snap4city.org/456

Programmable PAX counting

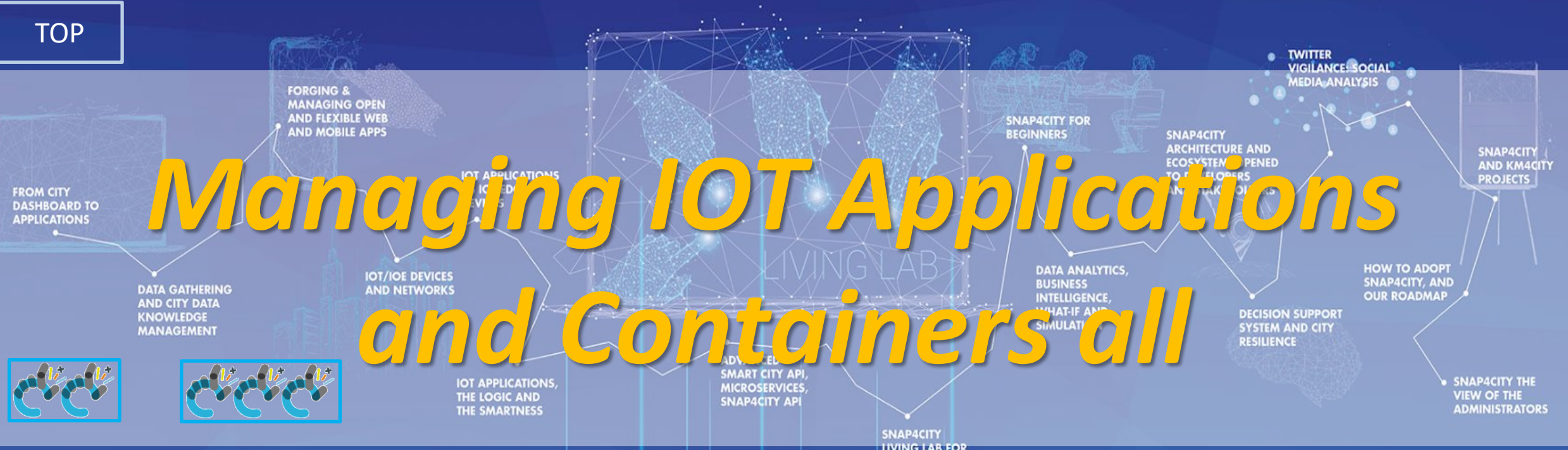




Antwerp

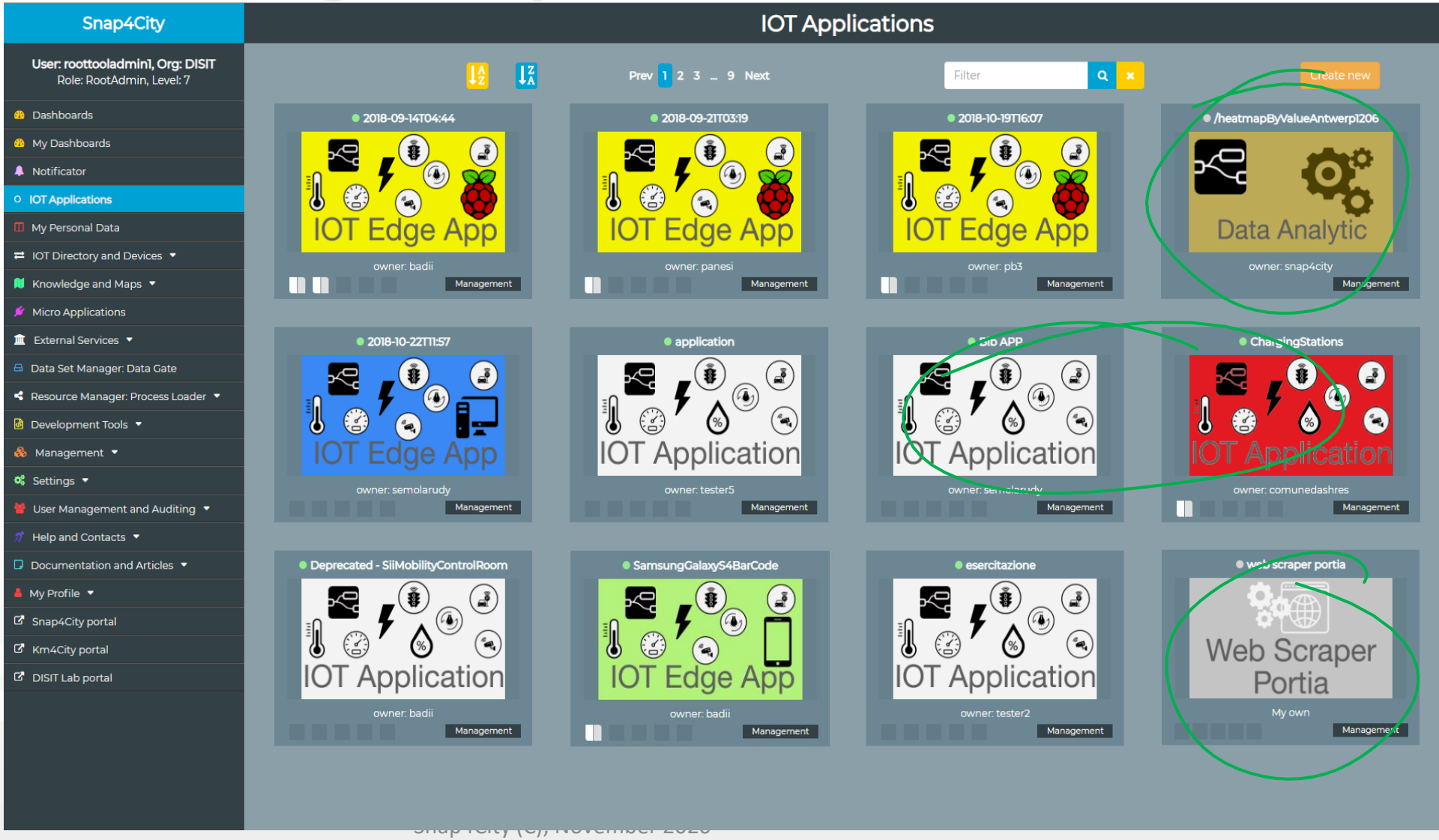
TOP

Managing IOT Applications and Containers all



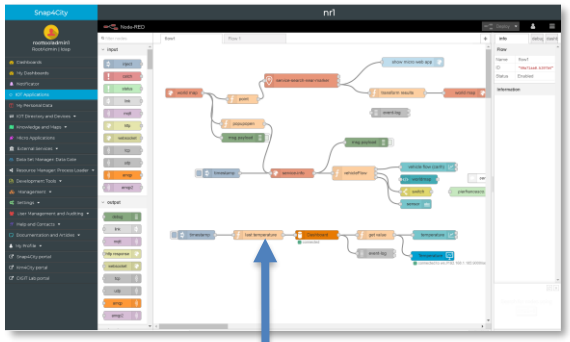
IOT Application Listing, they can be

- Basic (white)
- Advanced (red)
- IOT Edge
  - Raspberry Pi
  - Android
  - Win/Linux
- Data Analytic (Plumber)
- Web Scraper (Portia)

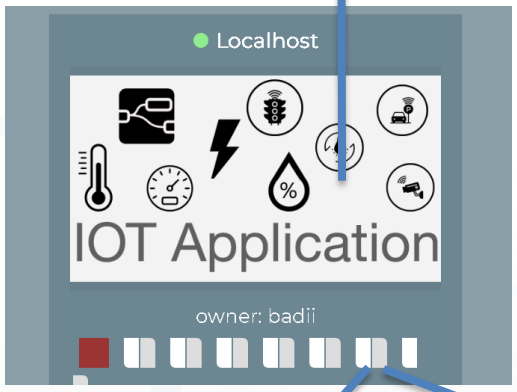


IOT Applications Listing

- Basic / Advanced
- On IOT Edge Raspberry Pi
- On IOT Edge Android
- On IOT Edge Win/Linux



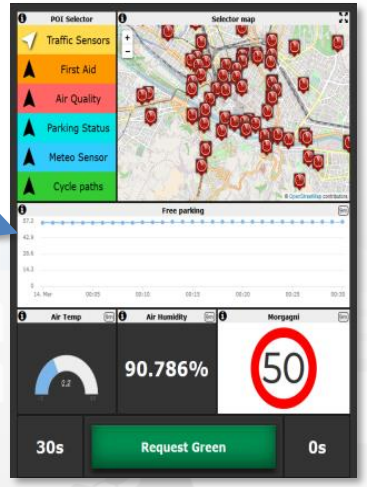
EDIT IOT APP

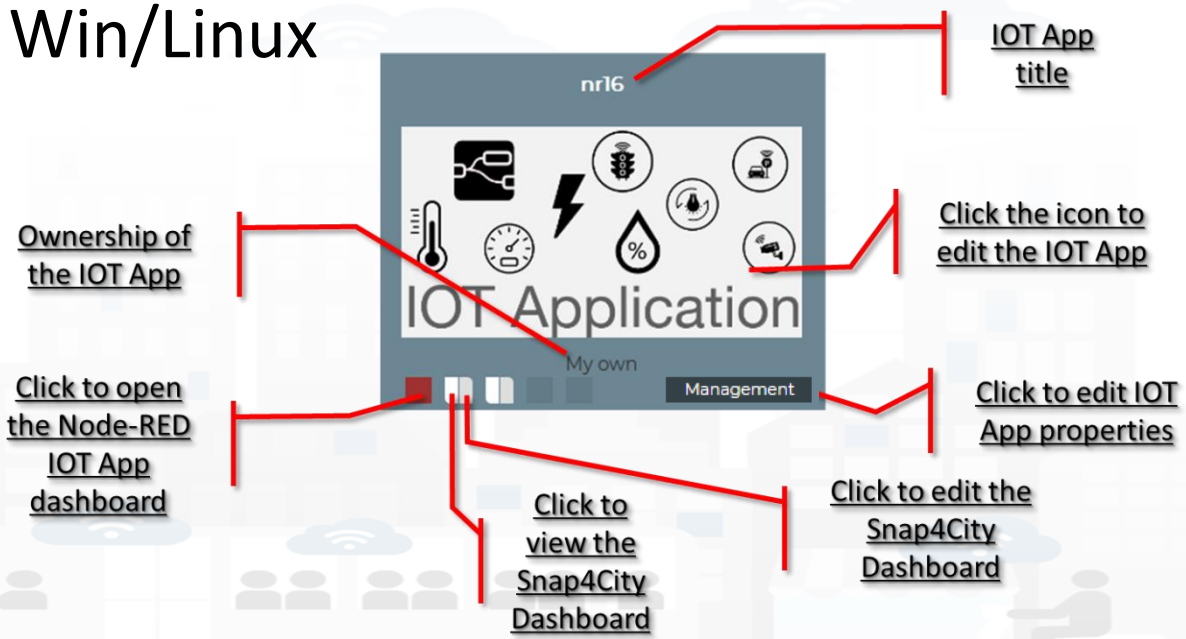


VIEW

EDIT

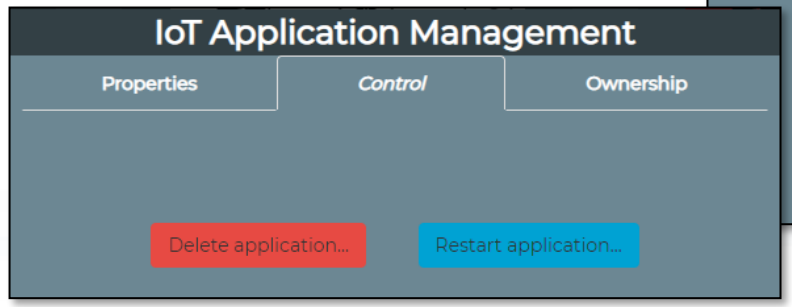


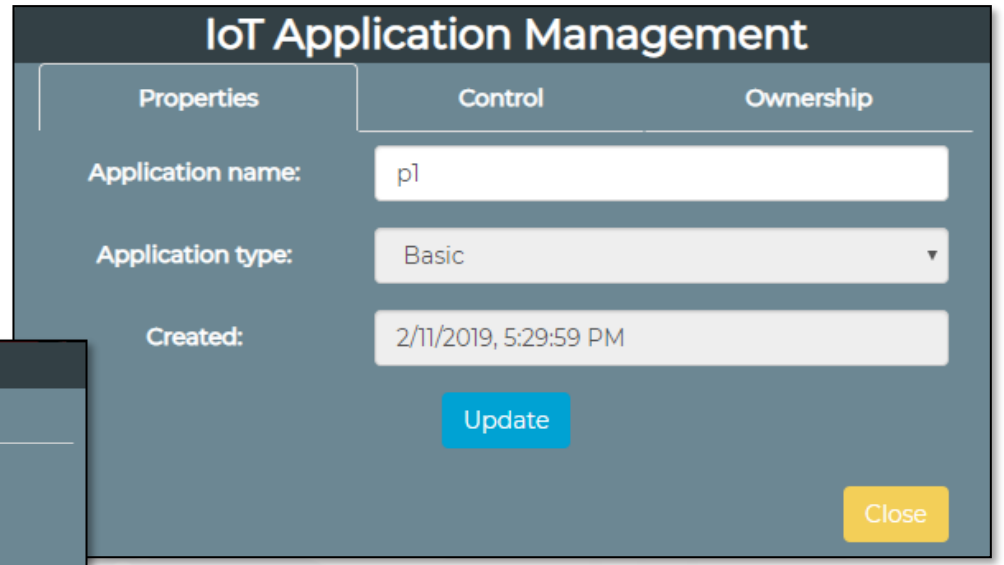


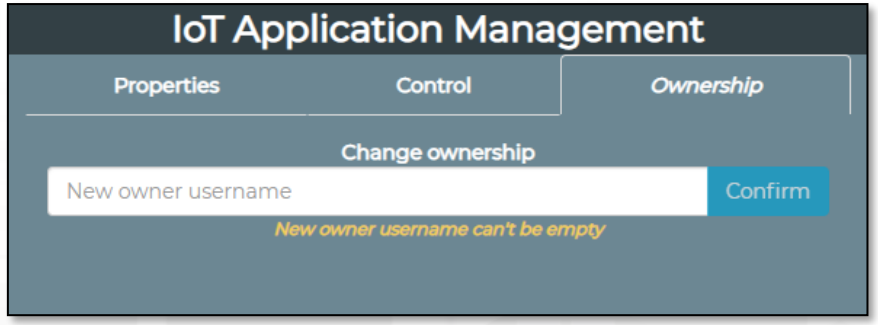


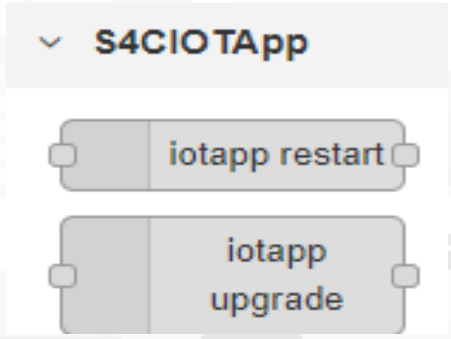
IOT Application Self Control

- **Properties**
  - Name, Type, Creation date
- **Control**
  - Restart Container
  - Delete IOT App
- **Change of ownership**
  - Pass to another Snap4City User
- **From inside the IOT App**
  - Restart
  - Update Snap4City Library







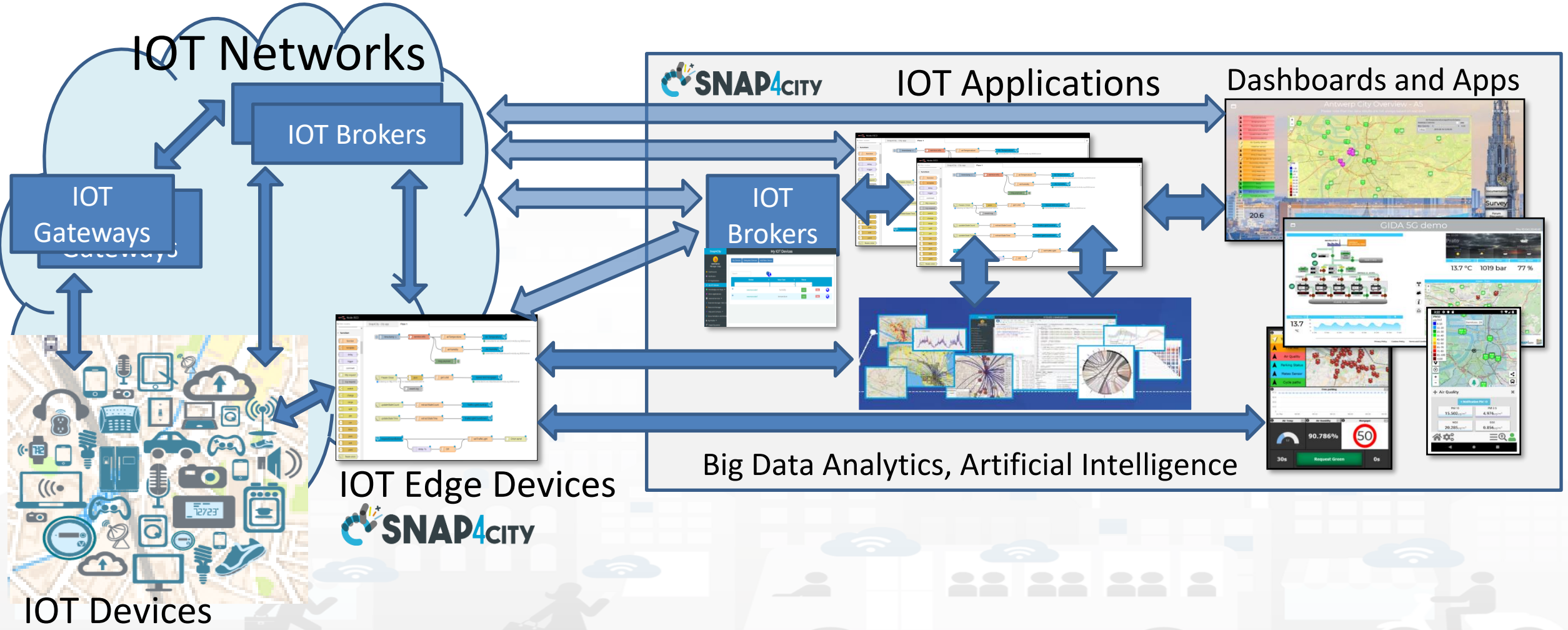


Automating restart and update

Remote Control of IOT Applications on IOT Edge Devices



Snap4City Services also on IOT Edge!!!



Why it is useful

- **Need to transform** data from your local devices to Snap4City devices or data storage or dashboards,

- **Need to play the role of multiprotocol HUB** as in:
  - **Snap4Home**: Scenario: Snap4Home, how to exploit Snap4City solution on home automation
  - **Snap4Industry**:
    - **Scenario: High Level Control of Industrial Plant**
    - **Snap4Industry: Snap4City for Industry 4.0**
    - **Scenario: 5G Enabled Water Cleaning Control**

IOT Edge Device for Snap4City

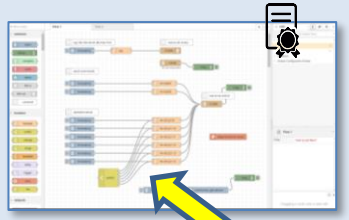
- **Computer based solutions with *Node-RED + Snap4City Library***
- **Node-red supports:**
  - Raspberry pi, Linux based, Windows based, android with Termux, and also on a several servers. https://nodered.org/docs/getting-started/
- **Snap4City Library:**
  - From Node-RED: "manage palette" in the main menu'.
  - From Node-RED library: https://flows.nodered.org/search?term=snap4city
- ***You can add any kind of protocol and interface to IOT Edge Device***

Remote Control/Management of IOT Edge

SNAP4CITY

edge-tunnel-to-cloud

IOT Edge Device

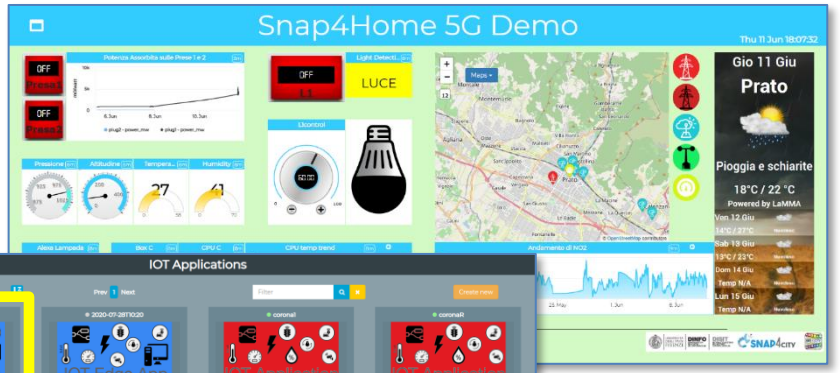


FireWall Gateway

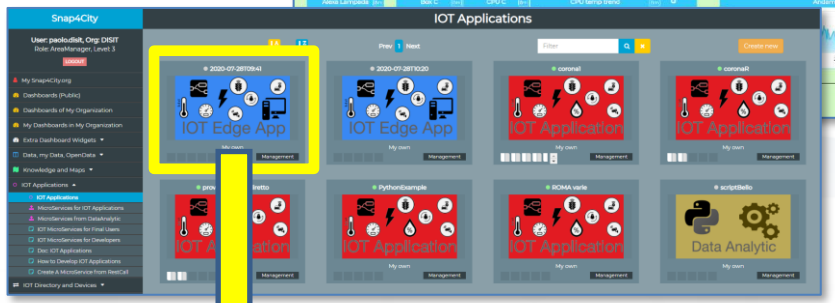
SNAP4CITY On Cloud

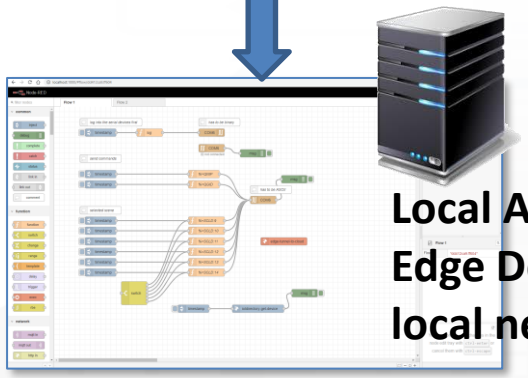
Internet

Intranet



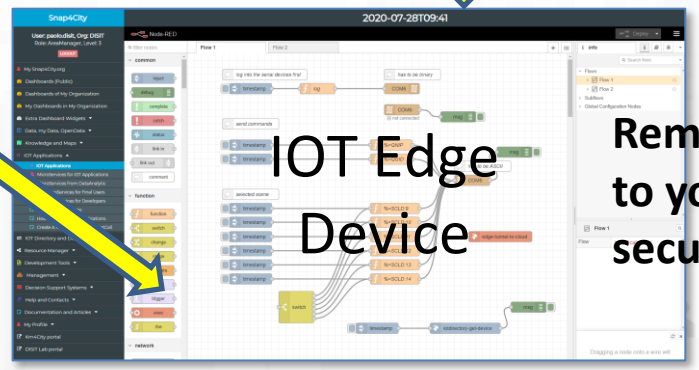
Click on it





Local Access on IOT Edge Device on your local network

On premise, on your local plant/home



IOT Edge Device

Remote Access/Control/program to your IOT Applications via secure connection

Advantages of IOT Edge remote control/program

- You do not need to be/move in the local network to access at Your IOT Edge Devices for programming or maintenance, SINCE With Snap4City:
  - ***You can update the logic flow of your IOT Edge Devices from remote,***
  - ***You can perform remote maintenance of your IOT Edge Devices and programms without moving from your office***



AND

- You can access to the IOT edge from remote by using a secure connection
- You can activate the remote control feature singularly for each IOT Edge Device
- You do not need to reconfigure your Firewall or Gateway in most of the cases

IOT Edge Device

Other Local Connections

IOT Brokers

IOT Brokers

HTTPS

HTTPS

WiFi

Modbus

OPC UA

IOT Edge:
Node-RED + Snap4City

edge-tunnel-to-cloud

Search and Query, Smart City API
Facet, semantic search

Knowledge base
Semantic reasoners

Indexing and aggregating
Elastic search

WSs

HTTPS

Firewall
Gateway
Toward internet

IOT Devices

Remote Access/Control/program to your IOT Applications via secure connection

HOW To install IOT Edge Remote Control feature

- **The installation is very simple**

1. install Snap4City basic library
2. Drag and drop block from S4CUtility

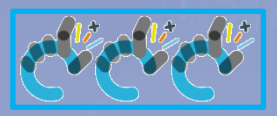


3. Configure the block with your credentials
4. Deploy of the IOT App
5. Go in the list of Your IOT Applications on Snap4City.org or other cloud or on premise installations
6. Identify the IOT Edge IOT App and click on it to open the view on the IOT Applications flows

TOP

Creating IOT Applications with Node-RED

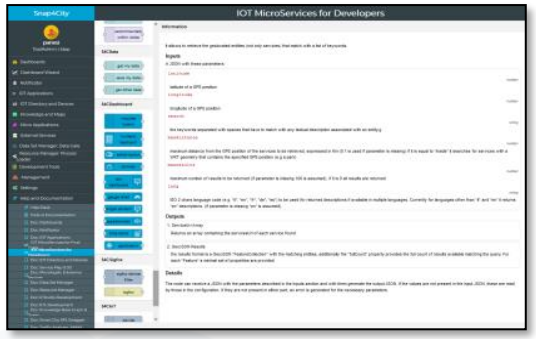




IOT Applications Development

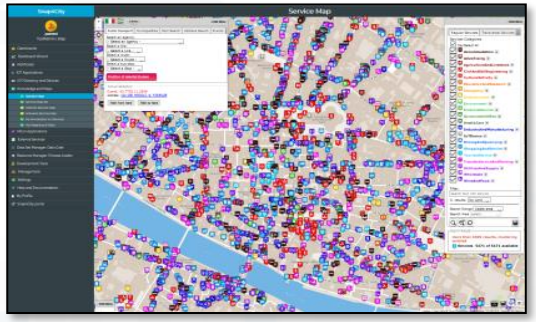
IOT Discovering

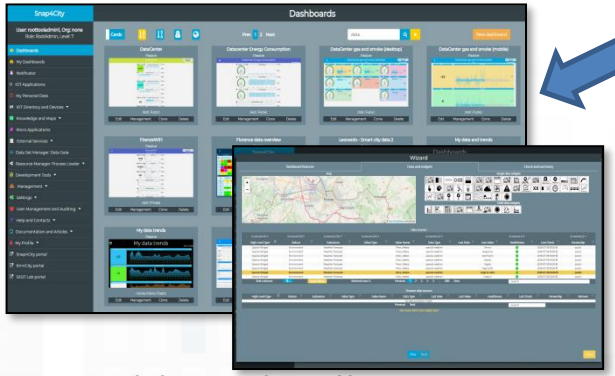
MicroServices collections



My IOT Applications







ServiceMap Discovery



Dashboard Collection, Editor and Wizard

IOT App. Editor



Generating IOT App With Dashboard

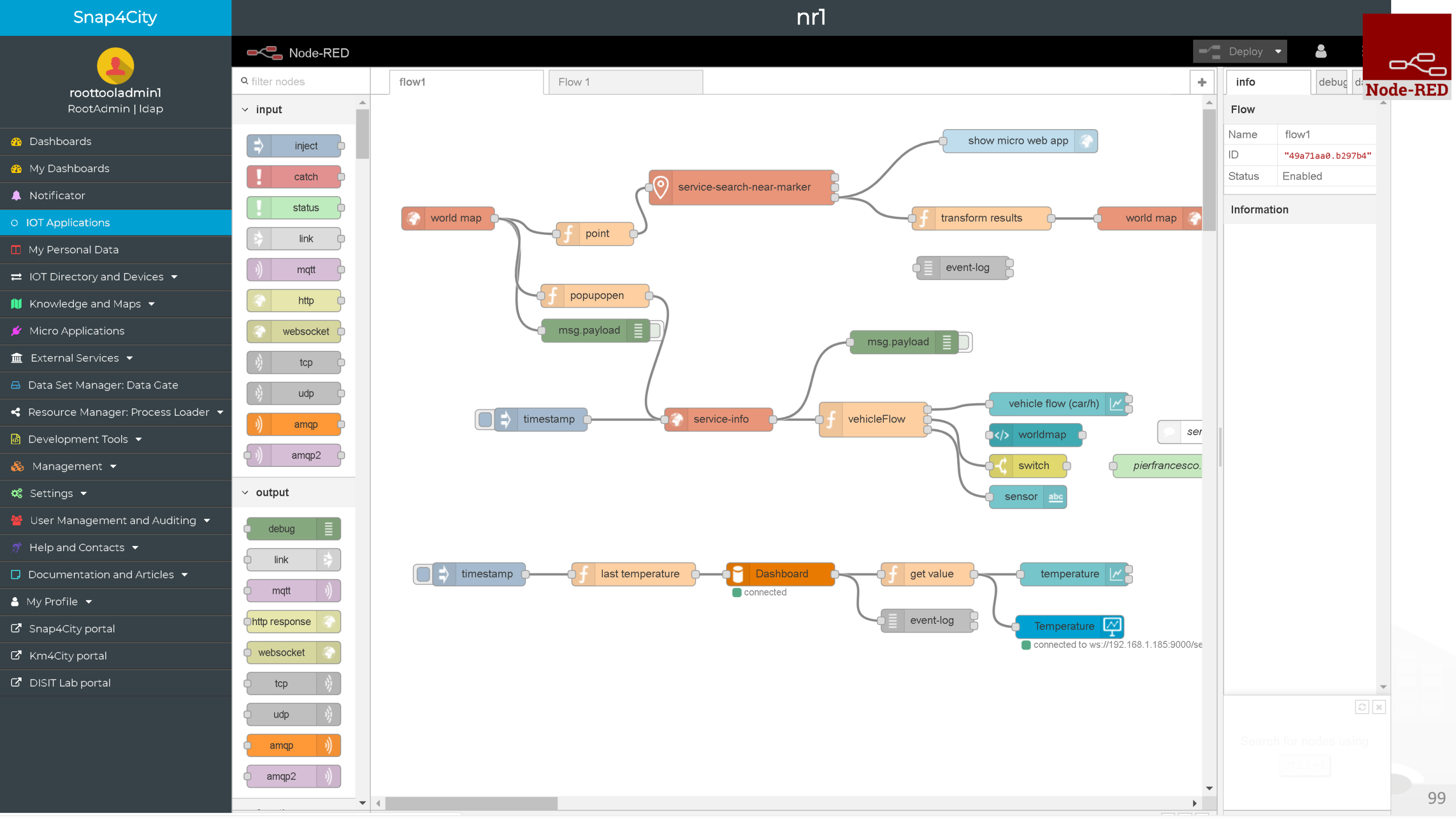


Sharing/saving reusing IOT App



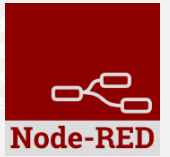
Resource Manager

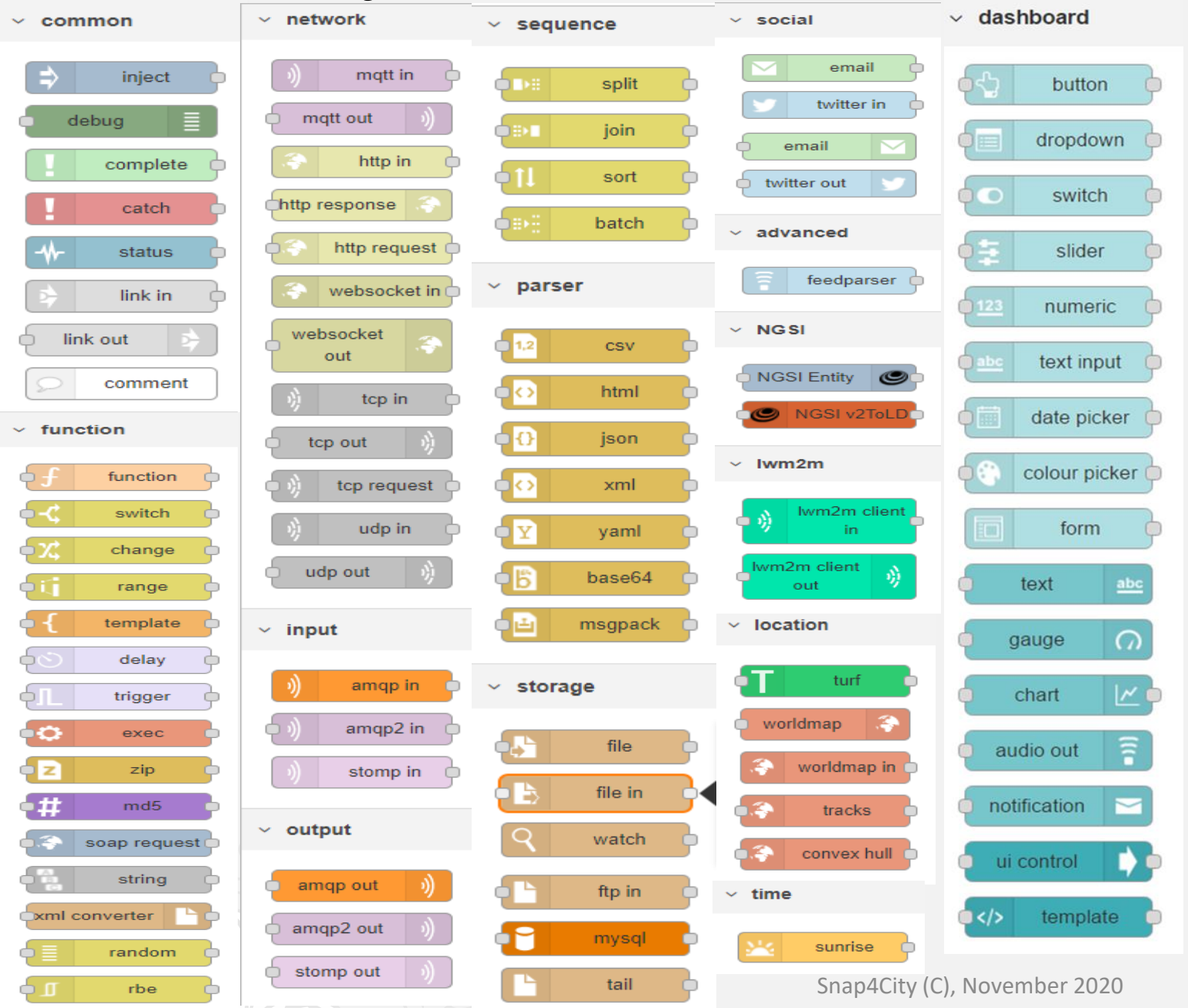



Snap4City
nr1
roottooladmin1
RootAdmin | ldap
Dashboards
My Dashboards
Notificator
IOT Applications
My Personal Data
IOT Directory and Devices
Knowledge and Maps
Micro Applications
External Services
Data Set Manager: Data Gate
Resource Manager: Process Loader
Development Tools
Management
Settings
User Management and Auditing
Help and Contacts
Documentation and Articles
My Profile
Snap4City portal
Km4City portal
DISIT Lab portal
Node-RED
Deploy
filter nodes
flow1
Flow 1
input
inject
catch
status
link
mqtt
http
websocket
tcp
udp
amqp
amqp2
output
debug
link
mqtt
http response
websocket
tcp
udp
amqp
amqp2
show micro web app
service-search-near-marker
world map
point
transform results
world map
event-log
popupopen
msg.payload
msg.payload
timestamp
service-info
vehicleFlow
vehicle flow (car/h)
worldmap
switch
sensor
timestamp
last temperature
Dashboard
connected
get value
temperature
event-log
Temperature
connected to ws://192.168.1.185:9000/se
info
debug
Flow
Name
flow1
ID
"49a71aa0.b297b4"
Status
Enabled
Information
Search for nodes using
Node-RED
99

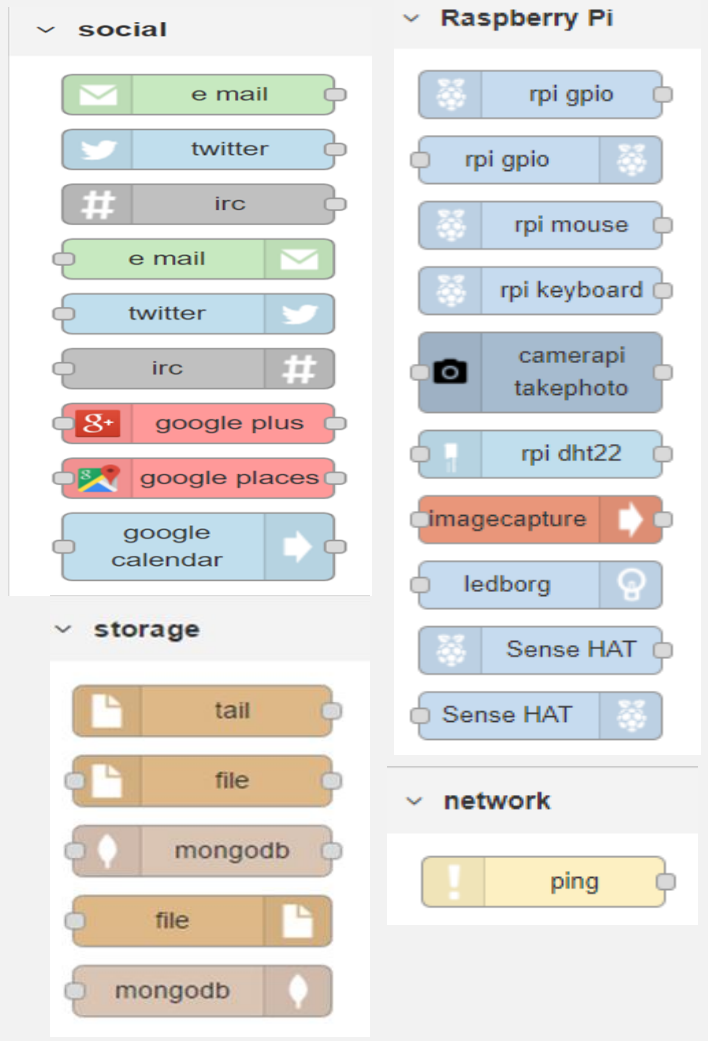
Basic Node.js Blocks on NodeRed on our Advanced IOT Apps

DISIT Lab, Distributed Data Intelligence and
ICT Systems and Internet
Information Engin
http://www.di





+ on IOT Edge Raspberry

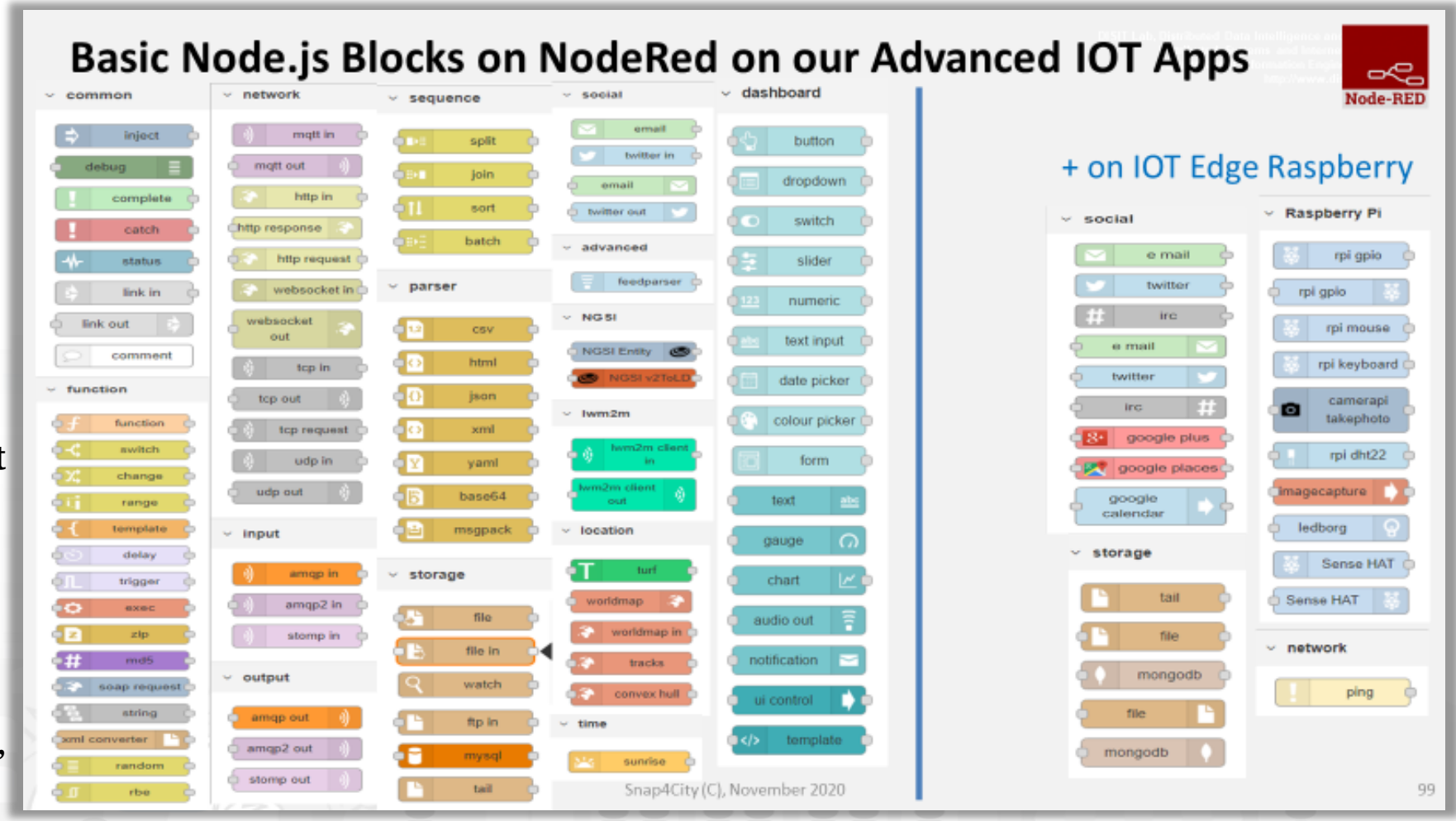


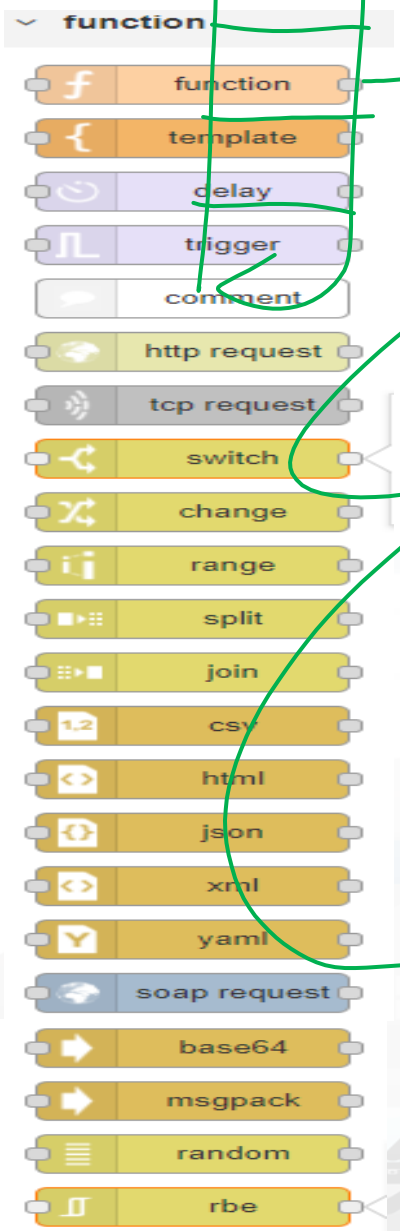
Node-RED Basic Blocks

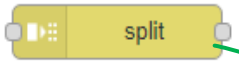
It is provided with **a minimum set** of functionalities (the building blocks/nodes) while other blocks can be easily added loading them from a **large library** made available by the **JS Foundation**.

Despite to its diffusion, for the usage in the context of Smart City it was **not powerful** to cope with the **basic requirements** of the domain.

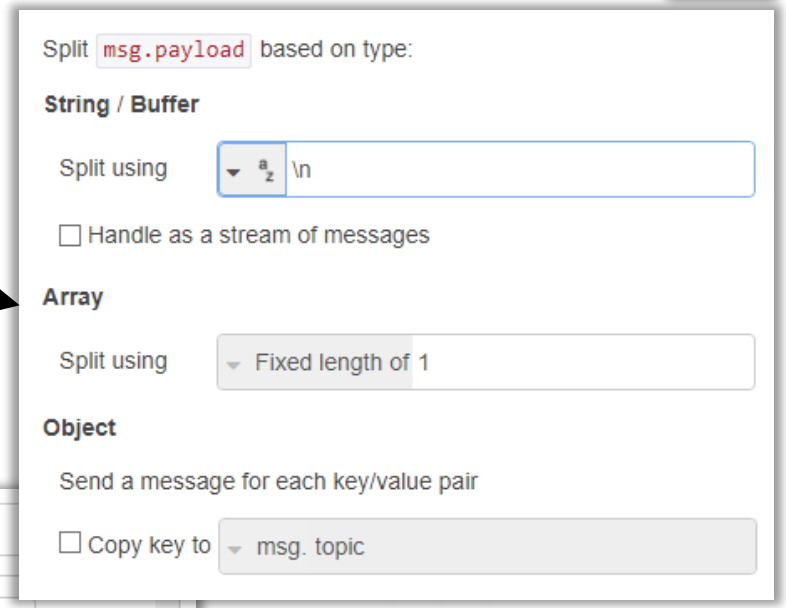
The classical nodes provided in the standard version can be classified as: input, output, function, social, storage, analysis, advanced, and dashboard.





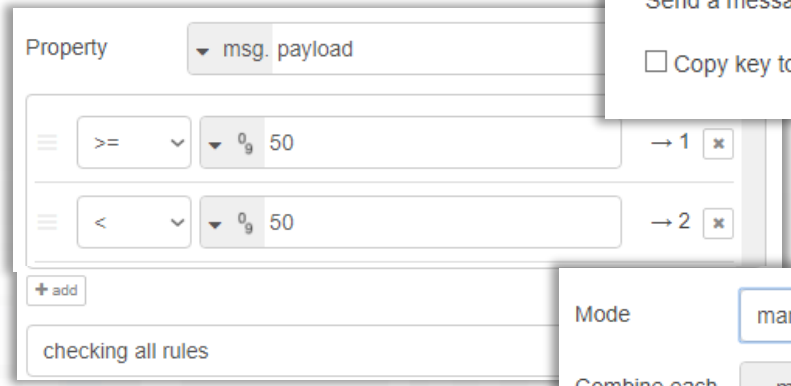


Divides the input message into multiple messages as indicated in the configuration. If you have an array at the input, you can configure it to send each element of the array individually at the output.



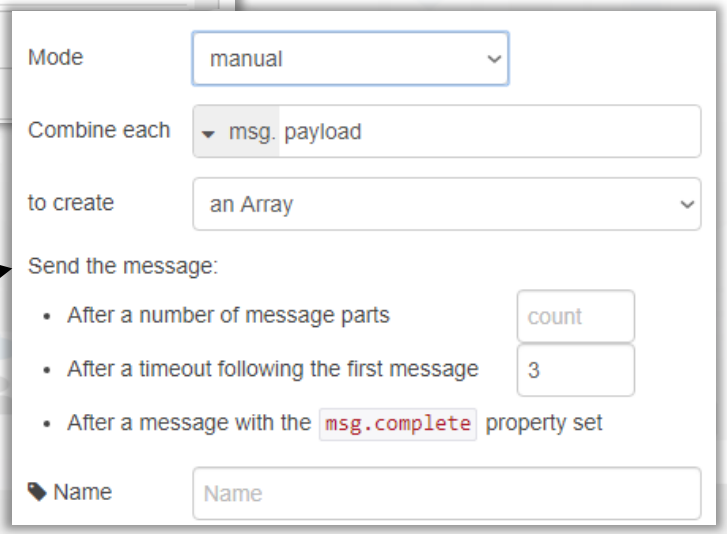


Treads the input message on possible different outputs based on a comparison made on the input message.



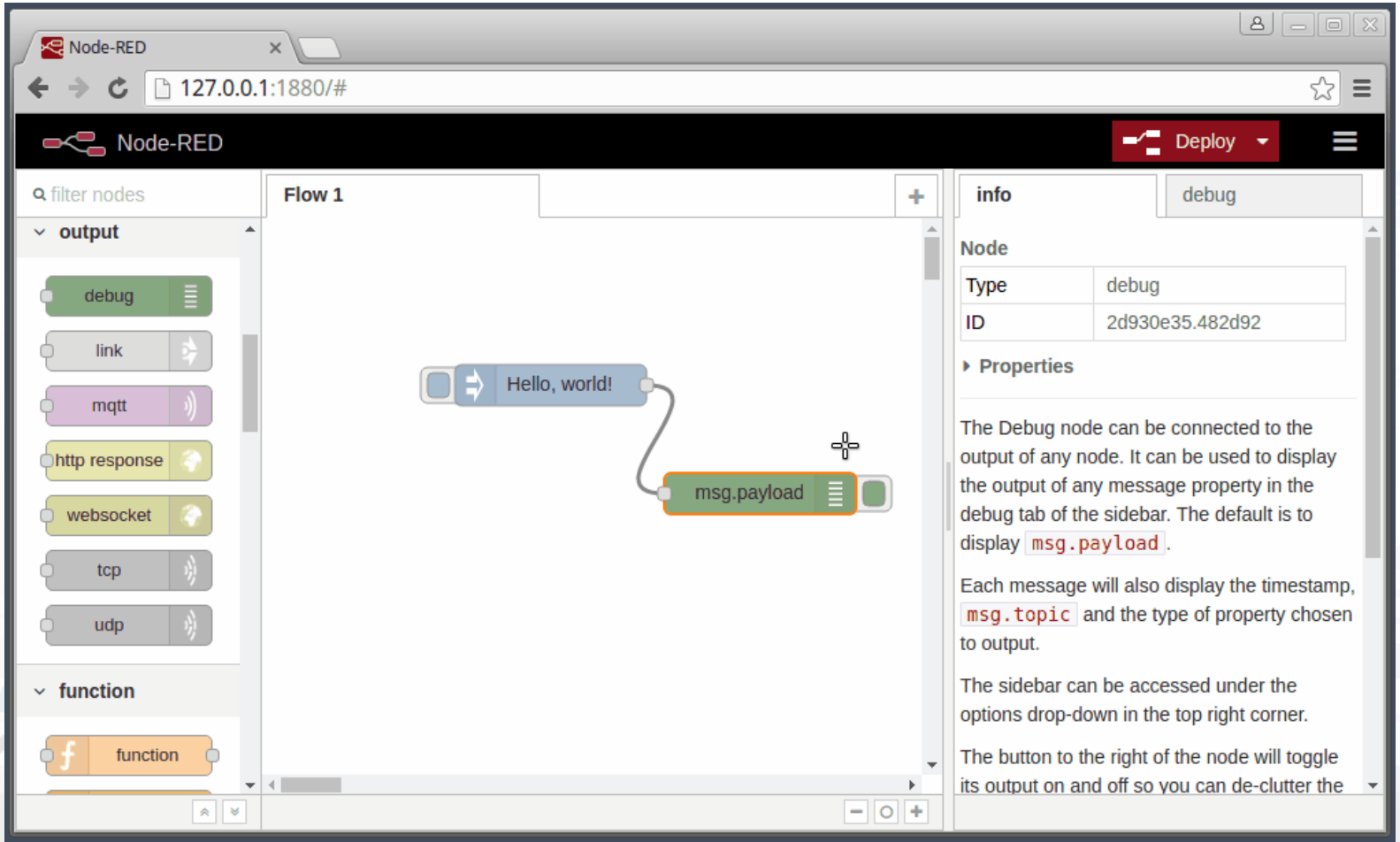


Operates in reverse order to the split. Joins the incoming messages in the mode indicated in the configuration.



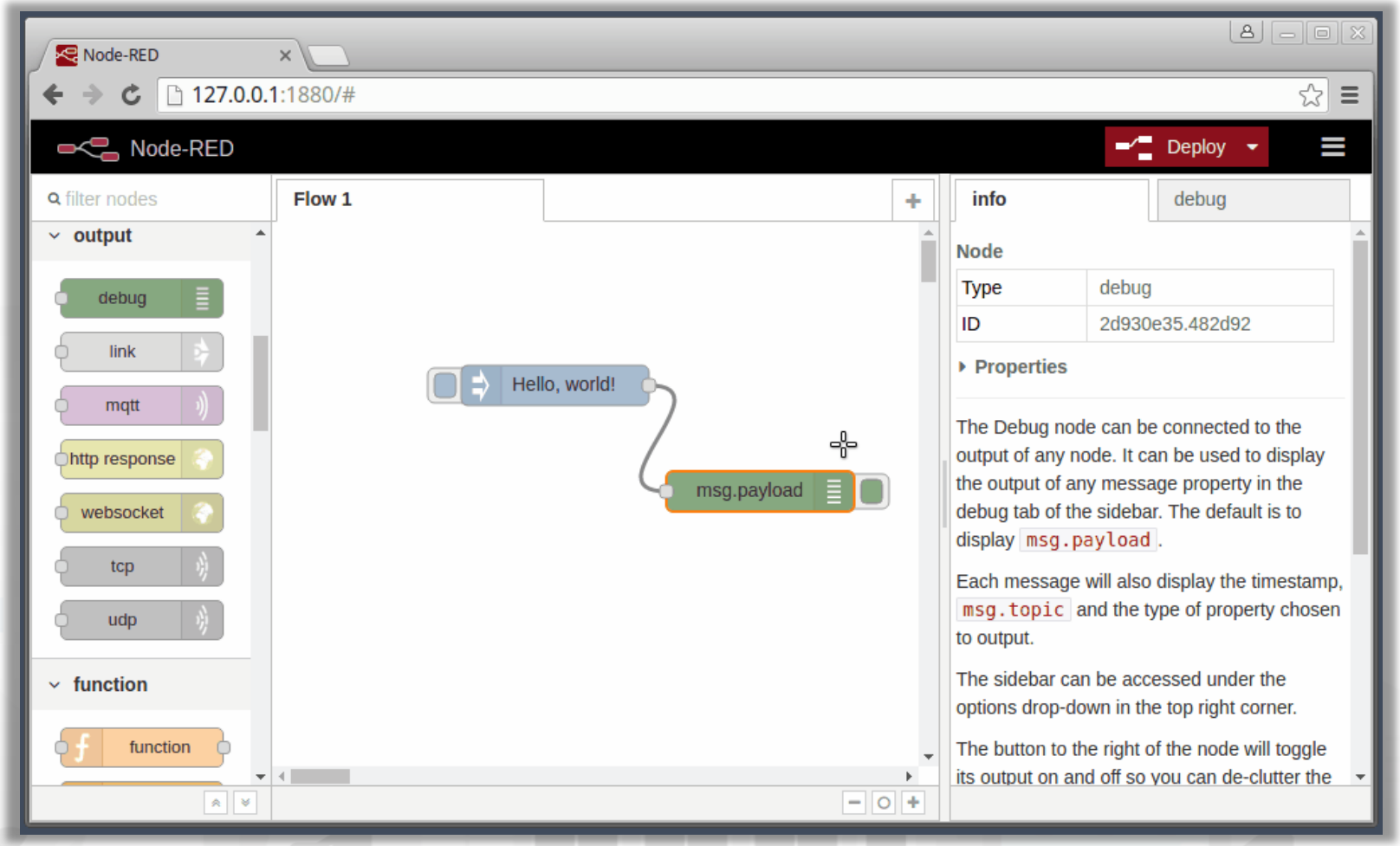
Hello World of Node-RED

- http://developer.opto22.com/nodered/general/getting-started/node-red-hello-world/



Node-RED

- Node-RED is a **flow-based** development tool for visual programming proposed by **JS Foundation**
- The Node-RED approach is a mix of **visual composition** of **nodes/blocks** to compose the so-called **flows** that are concurrently executed by an engine **Node.js**.
- It is quite diffuse being also directly provided into **official releases** of IOT devices as **Raspberry Pi** family
- Based on **Node.js**
- 100% **open source**



DEMO Section 1

Node-RED Hello World



Example of simple IOT Application

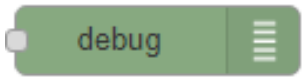
In this demo let's create an IoT Application that:

- generate a random value,
- the value is switched on the correct path
- the value is showed in the local dashboard of NodeRed

Nodes for flow



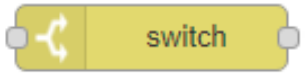
Generates an input for the other nodes. It can be repeated at predefined intervals, entered manually and of various types (timestamp, string, number, boolean, JSONetc)



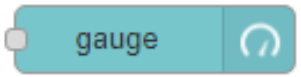
Each message that enters the debug node is shown in the "debug" tab on the right of nodered (you can choose which part of the message to show)



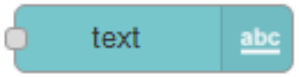
Generates a random number. You can configure the number generation interval and the type (integer or float).



Evaluates the input message and routes it to the correct output according to the desired configuration



Shows a number inside a gauge counter.



Shows a text inside the local dashboard

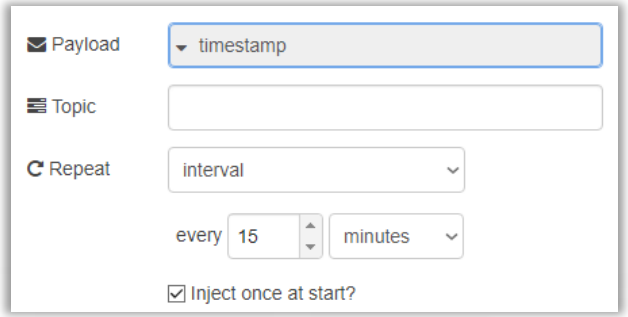
Step 1



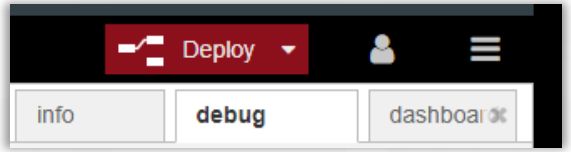
- Inject and Debug



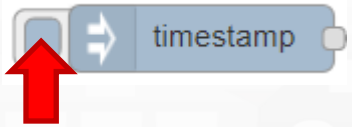
- Connect
- Configure



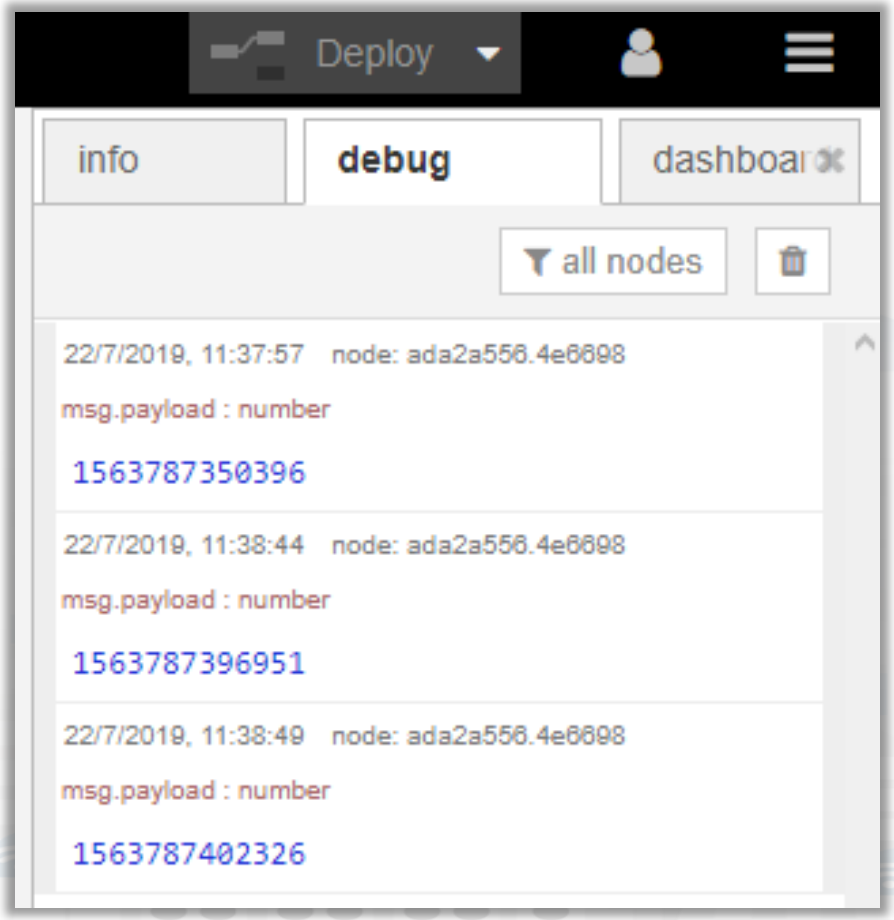
- Deploy



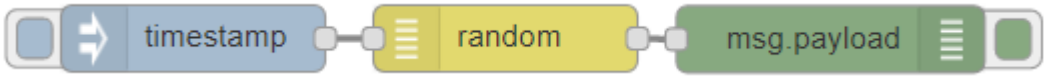
- Click



- Observe

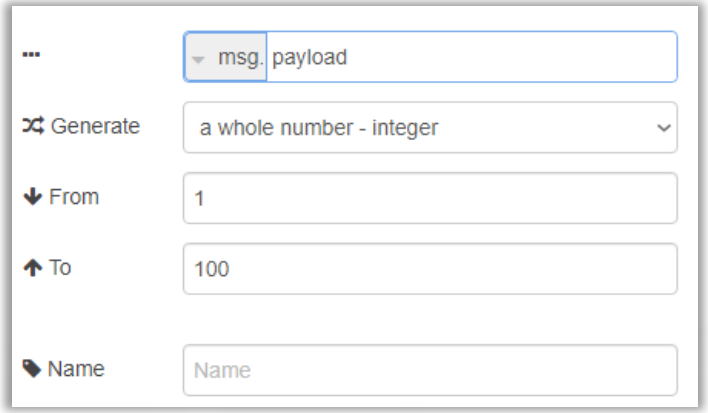


Step 2

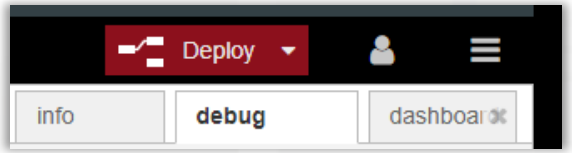


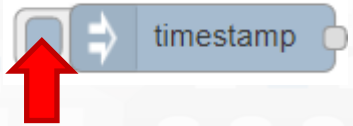
- Random
- Connect
- Configure

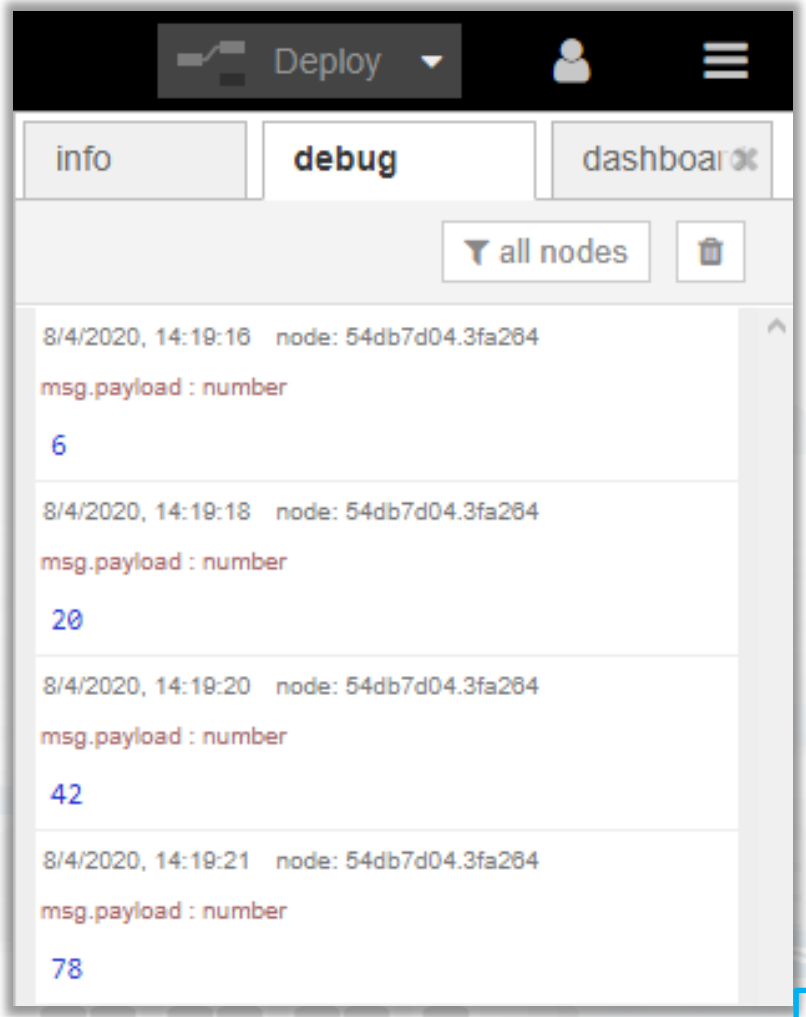




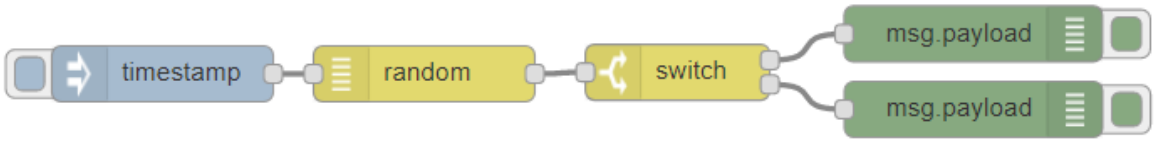
- Deploy
- Click
- Observe



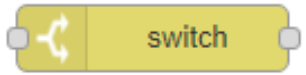




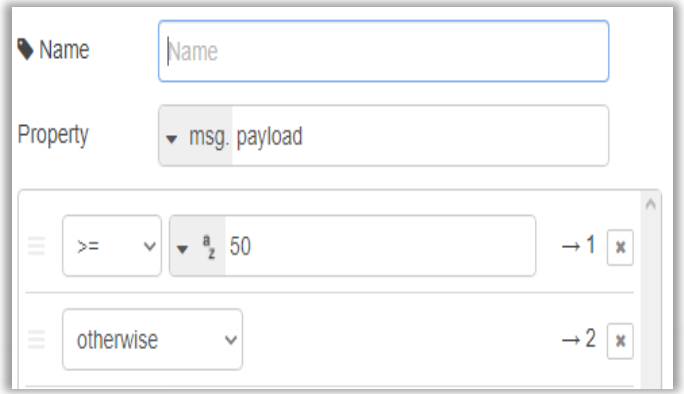
Step 3



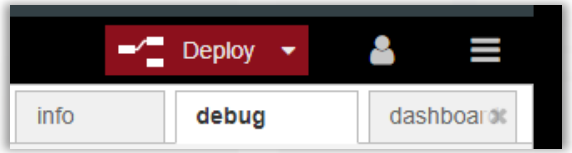
- Switch



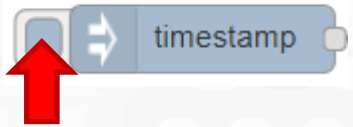
- Connect
- Configure



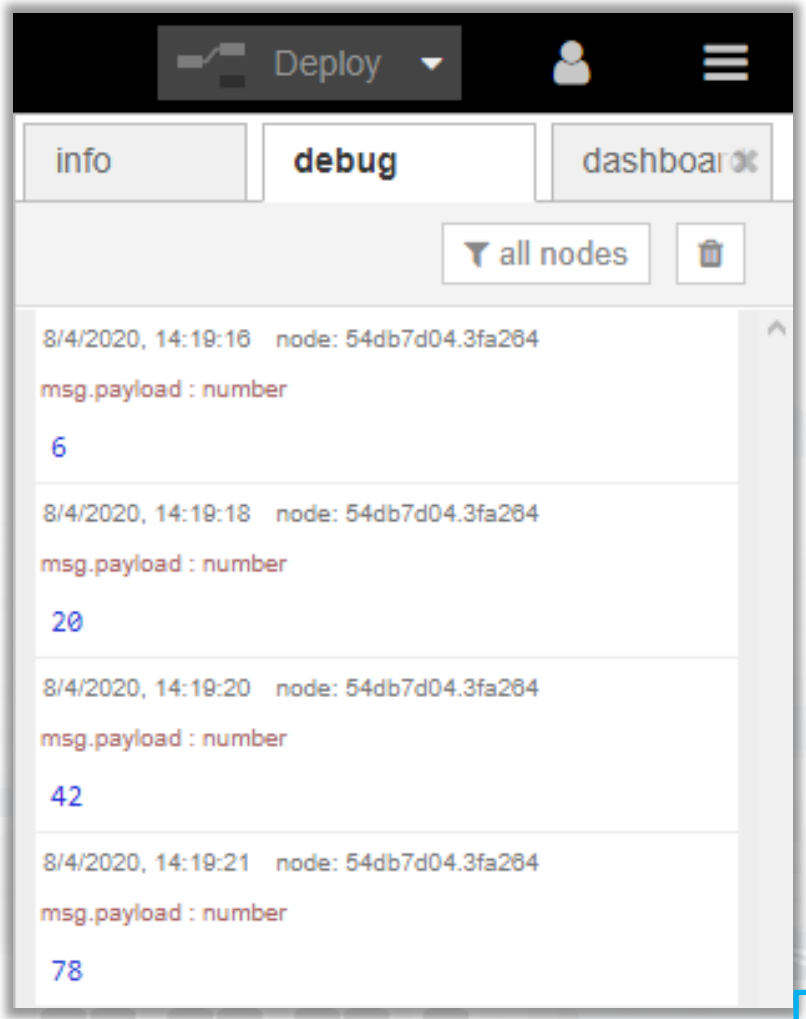
- Deploy



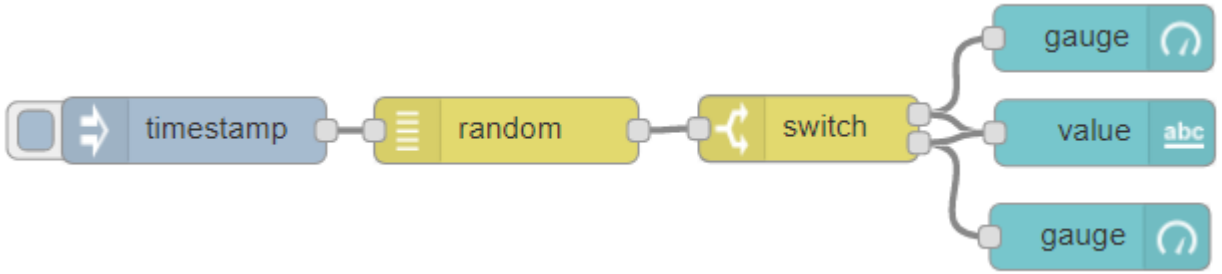
- Click



- Observe

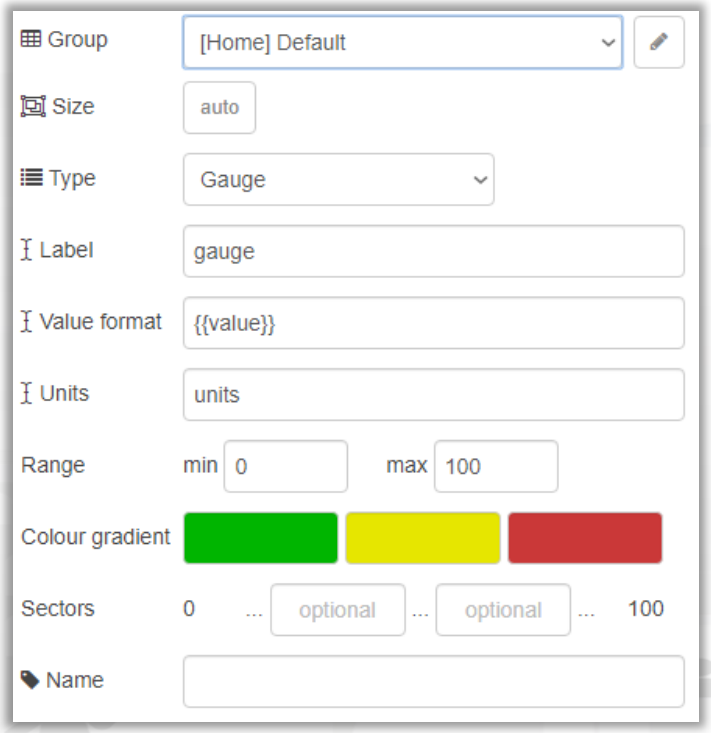


Step 4

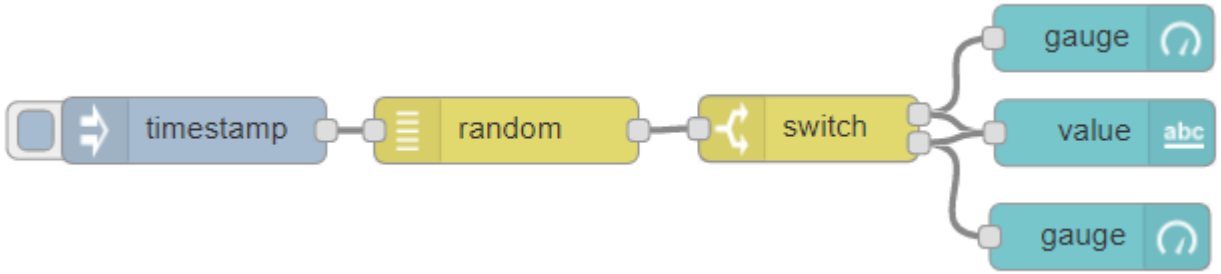


- Gauge and text 

- Connect
- Configure gauge

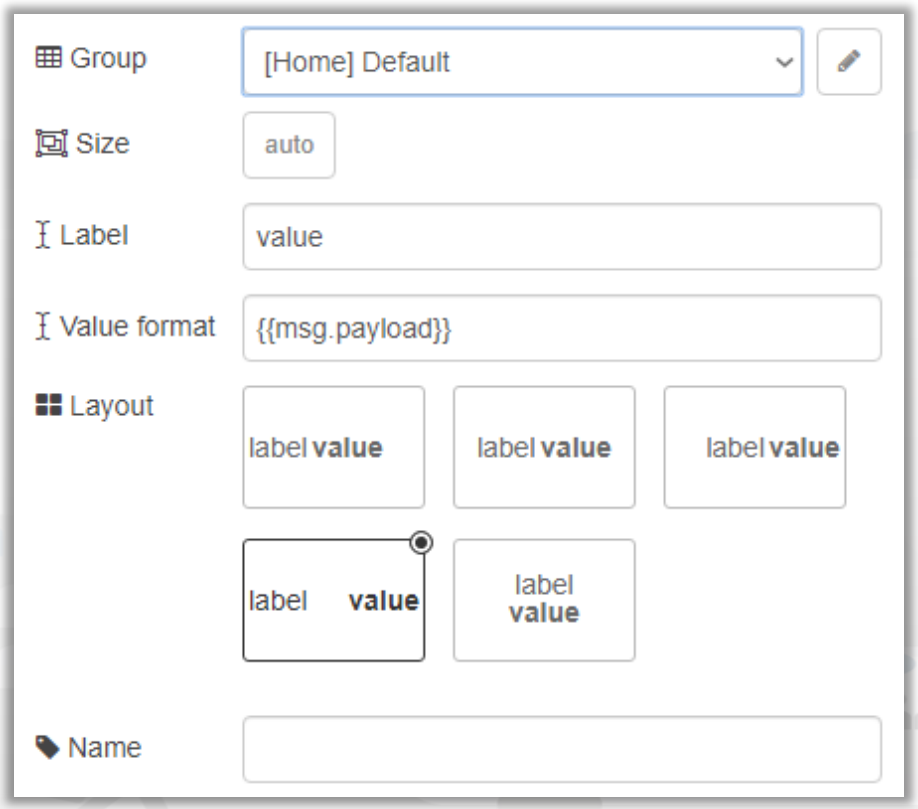


Step 4 Bis

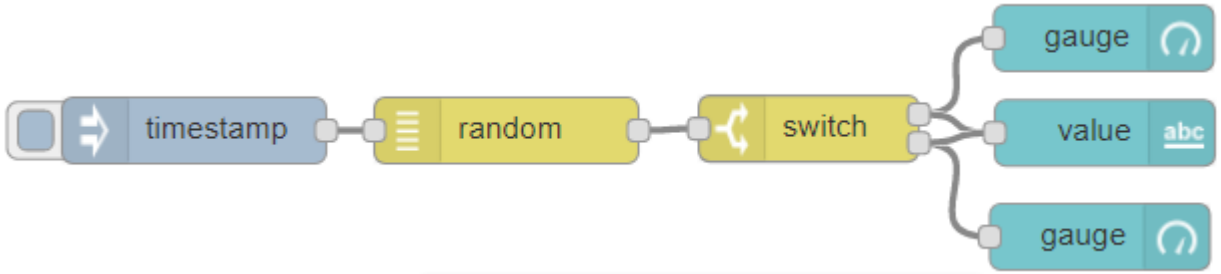


- Gauge and text 

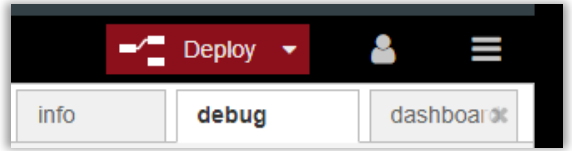
- Connect
- Configure text



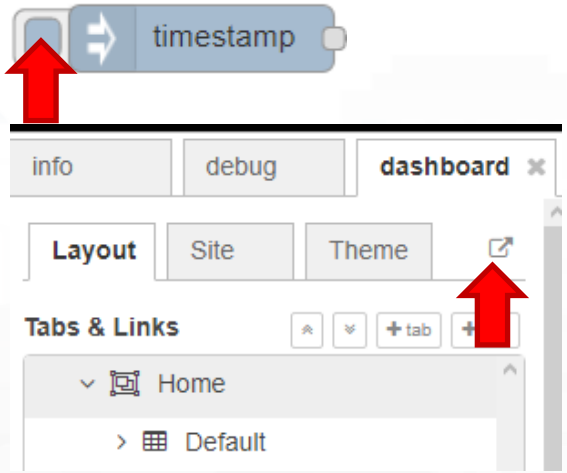
Step 5



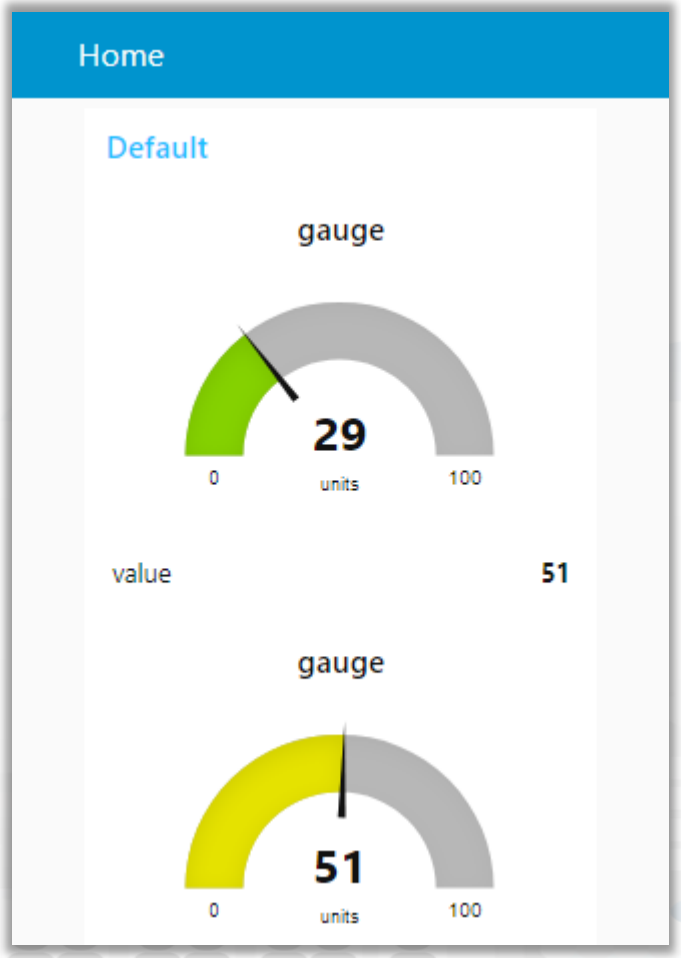
- Deploy



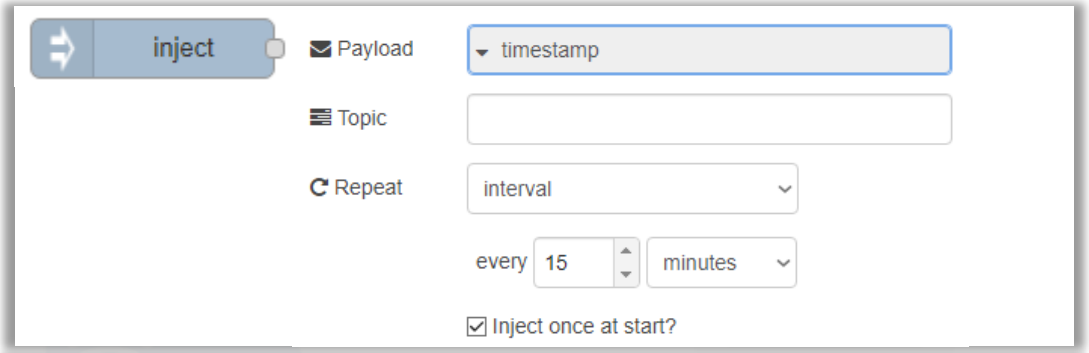
- Click

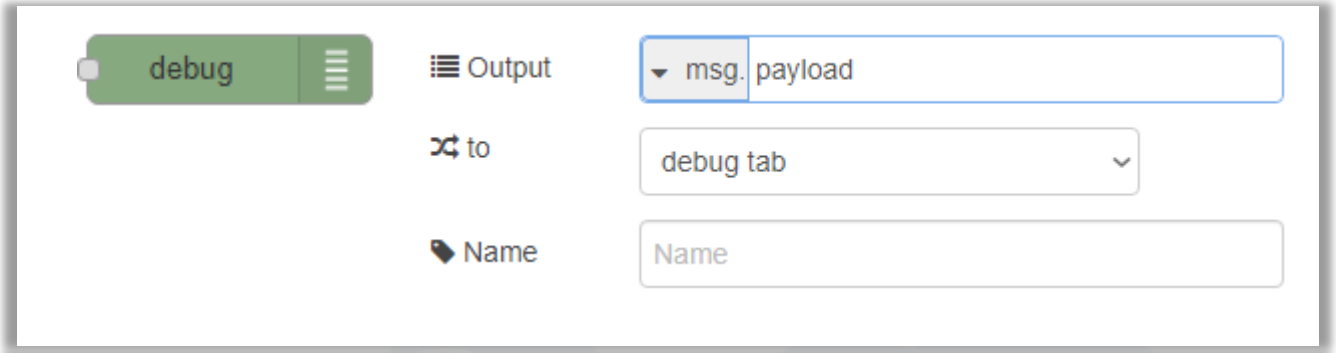


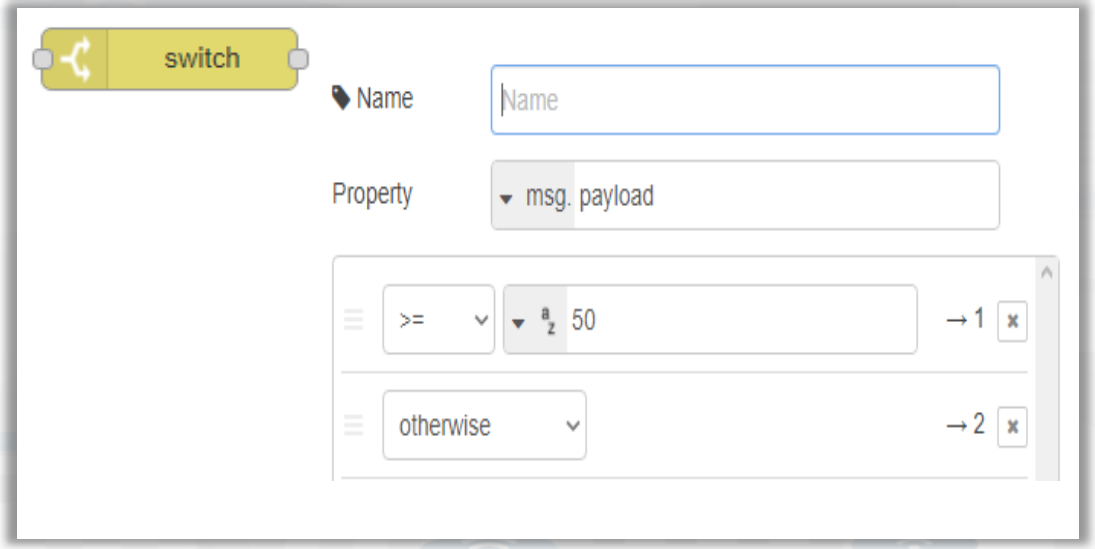
- Click
- Observe

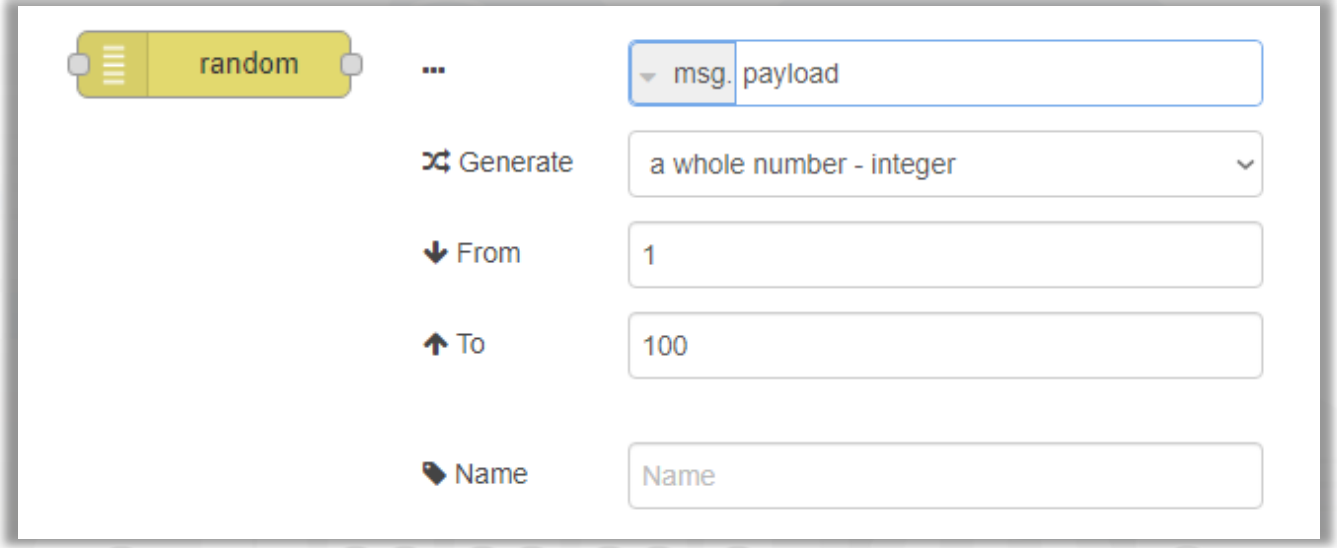


Nodes configuration 1/2

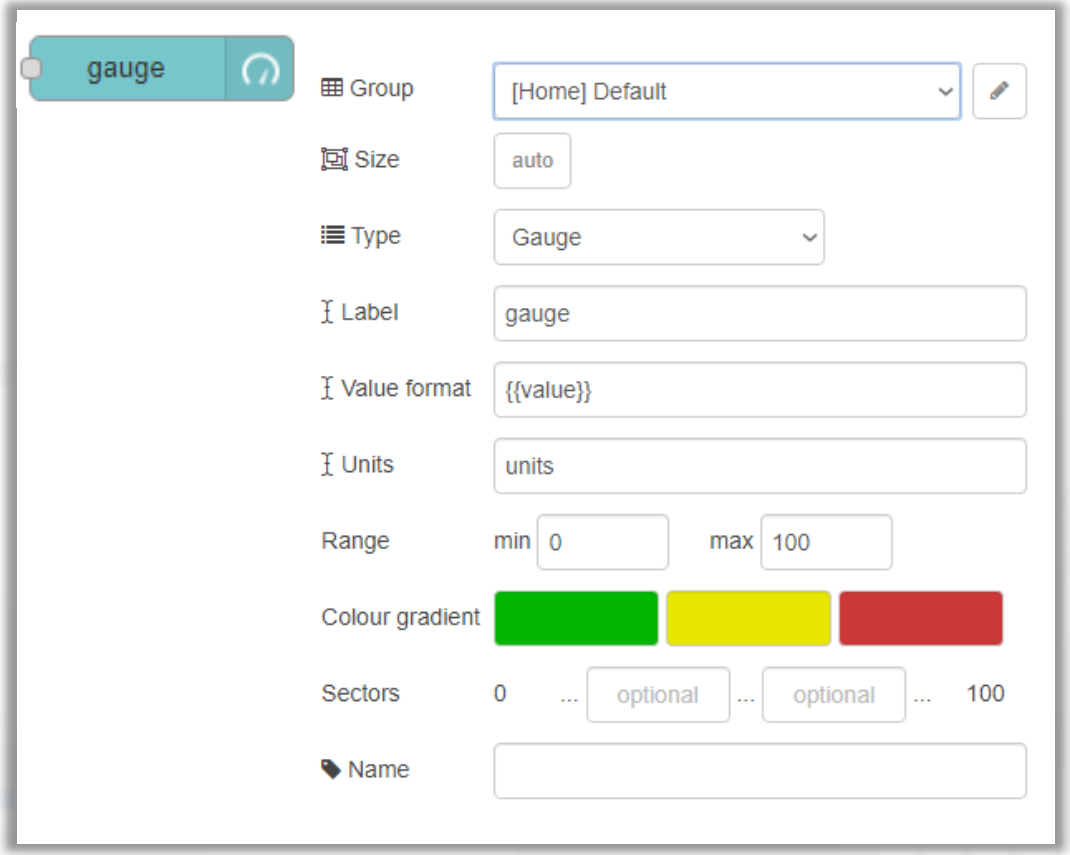


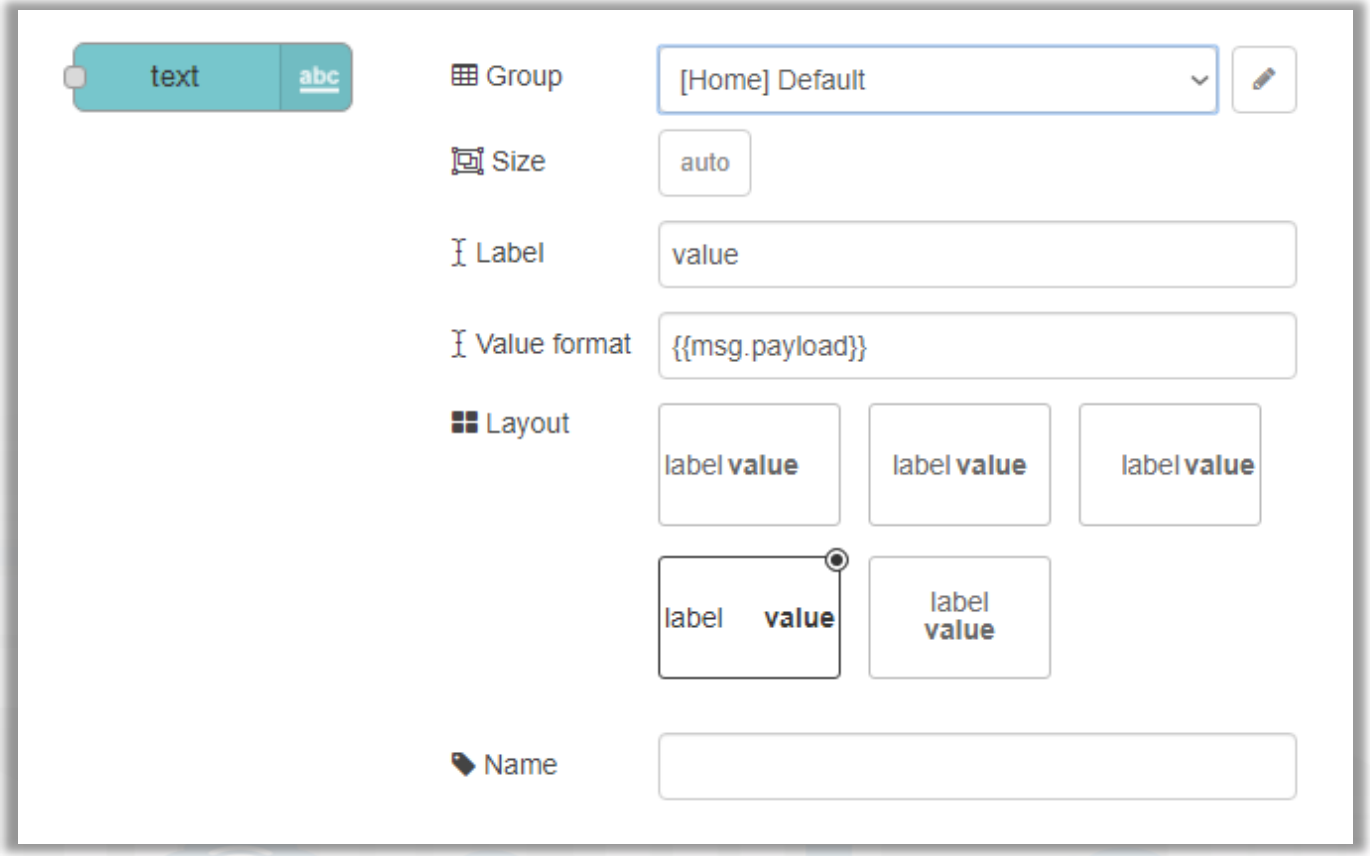




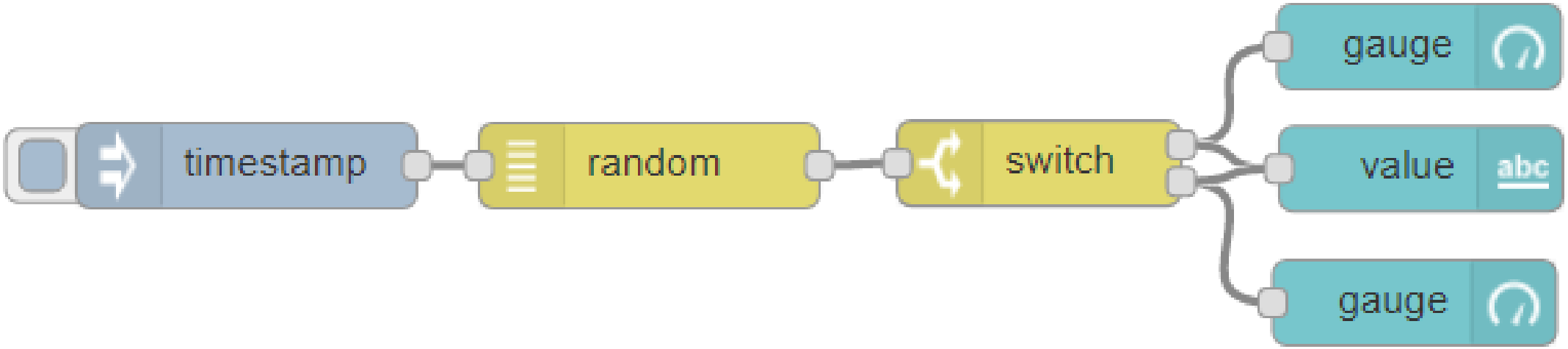


Nodes configuration 2/2





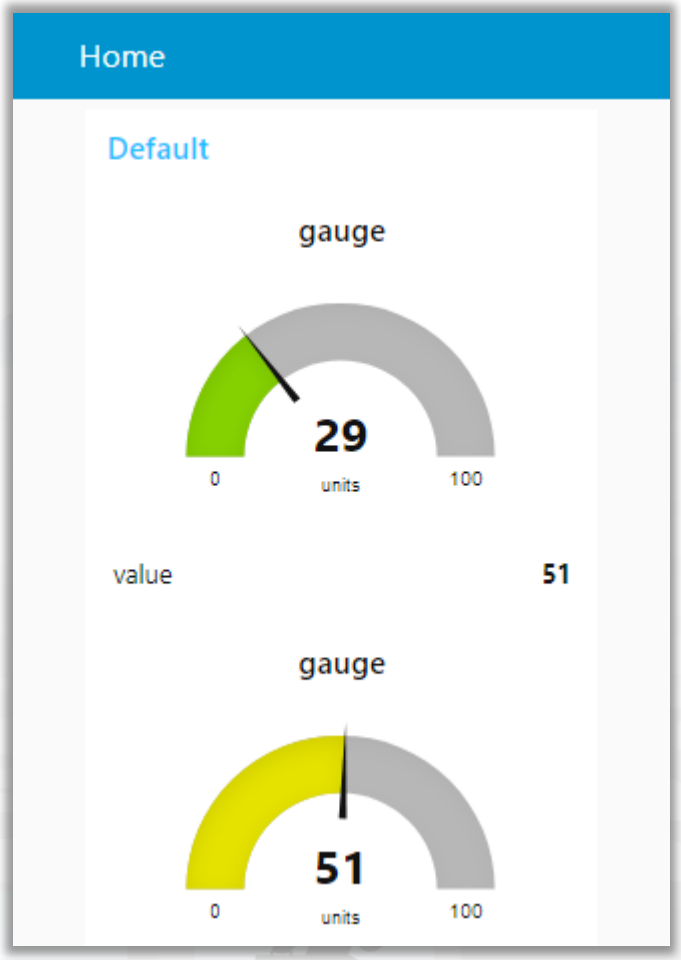
Nodes connections



Explaining: IOT Application Flow

- On Click or Every 15 minutes the ***timestamp*** node sends a message to the ***random*** node.
- When the message arrives, the ***random*** node generates a random number as output message.
- The **switch** node routes the value on the correct output based on the configuration .
- The Number can be sent to Different kinds of nodes to show it on NodeRed Dashboard.

Resulting Dashboard



This is a local Node-RED dashboard.

The dashboards created within the Snap4city platform are more :

- Powerful
- Flexible
- Secure

end DEMO Section 1

SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT CITIES

TOP

FORGING & MANAGING OPEN AND FLEXIBLE WEB AND MOBILE APPS

FROM CITY DASHBOARD TO APPLICATIONS

IOT APPLICATIONS VS IOT EDGE DEVICES

SNAP4CITY FOR BEGINNERS

TWITTER VIGILANCE: SOCIAL MEDIA ANALYSIS

SNAP4CITY ARCHITECTURE AND ECOSYSTEM. OPENED TO DEVELOPERS AND STAKEHOLDERS

SNAP4CITY AND KM4CITY PROJECTS

DATA GATHERING AND CITY DATA KNOWLEDGE MANAGEMENT

IOT APPLICATIONS, THE LOGIC AND THE SMARTNESS

ADVANCED SMART CITY API, MICROSERVICES, SNAP4CITY API

SNAP4CITY LIVING LAB FOR COLLABORATIVE WORK

DECISION SUPPORT SYSTEM AND CITY RESILIENCE

IOT App= Node-RED +Snap4City

Node-RED

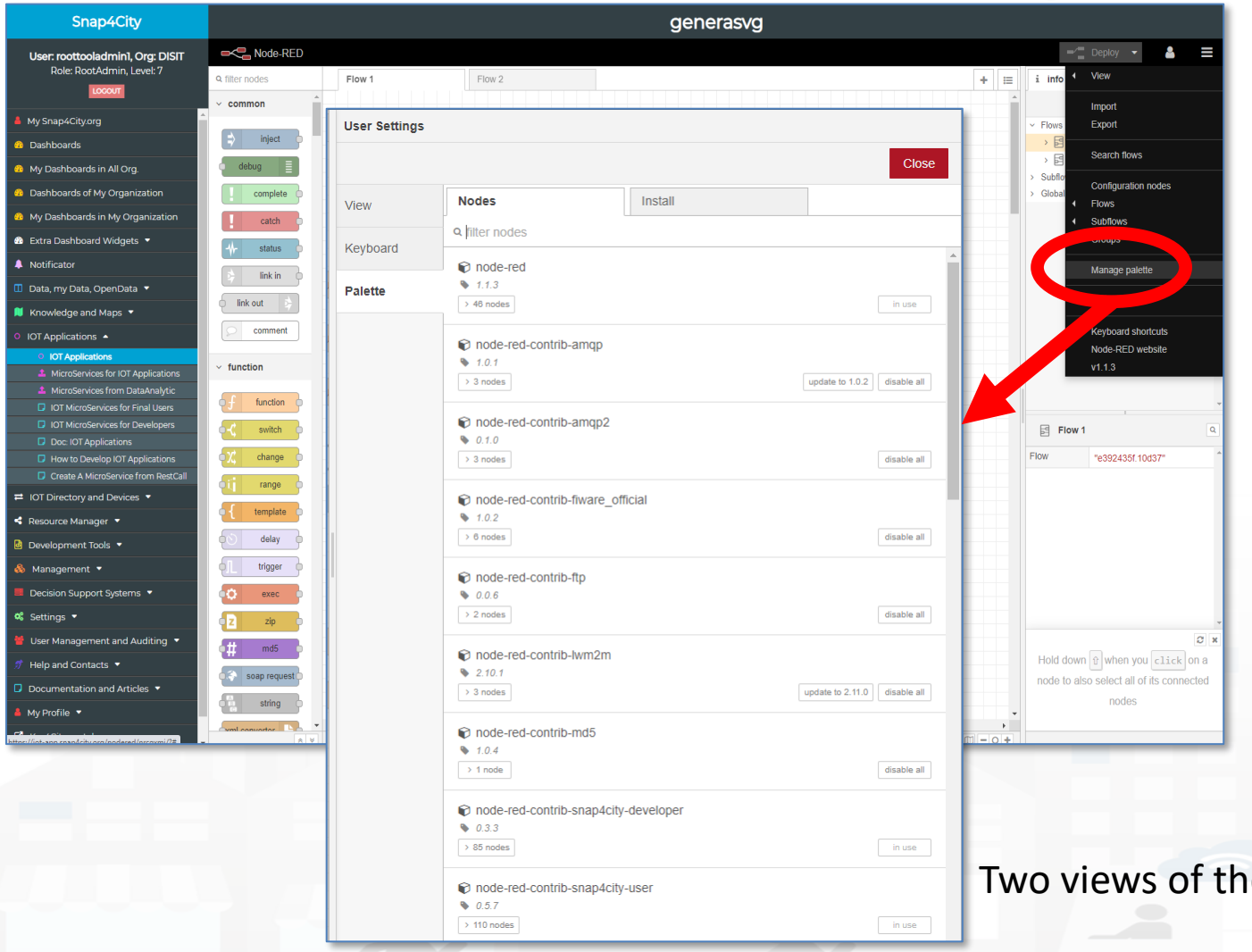
IOT Application Editor: NODE-RED

- In the **IOT Application of Snap4City**, it is possible to:
  - Execute flows that process data as: Event Driven, Batch (periodic or not)
  - Create multiple concurrent Flows for each IOT Application
  - Create subflows as macros to be reused
  - Create Groups of nodes as macro
  - **Save/load**, **share**, of nodes, flows and applications with other users via
    - the Snap4City Resource Manager or
    - with JS Foundation or
    - via email, skype, file sharing in general
  - ..

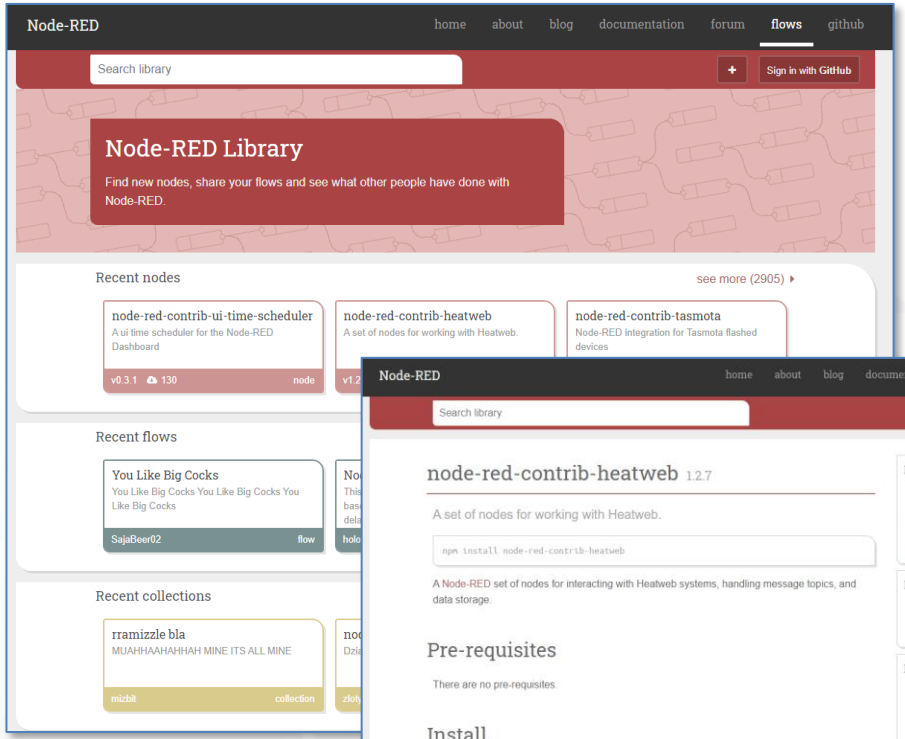
IOT Application Editor: NODE-RED

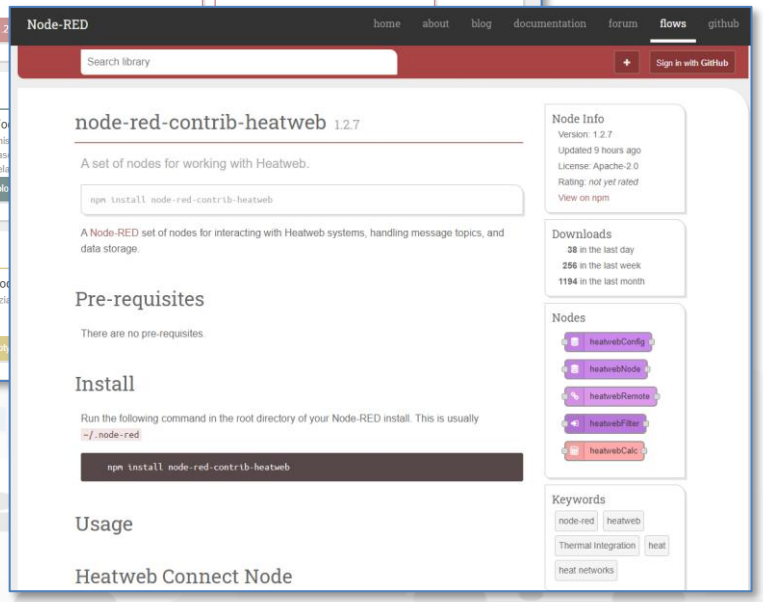
- In the **IOT Apps of Snap4City**, it is possible to Extend the Capabilities:
  - Load other Nodes, segments of flow and entire flows from several sources: email, libraries, S4C repository, etc.
  - Load other libraries of MicroServices/Nodes/Blocks from **Manage Palette**
    - A large set of Libraries of Node is available.
    - The loading may have some limitations for security reasons
  - **Get more IOT Apps** above the Limit that may depend on the organization and/or on personal authorizations, ask to Admin
  - ..

Load Library from Palette



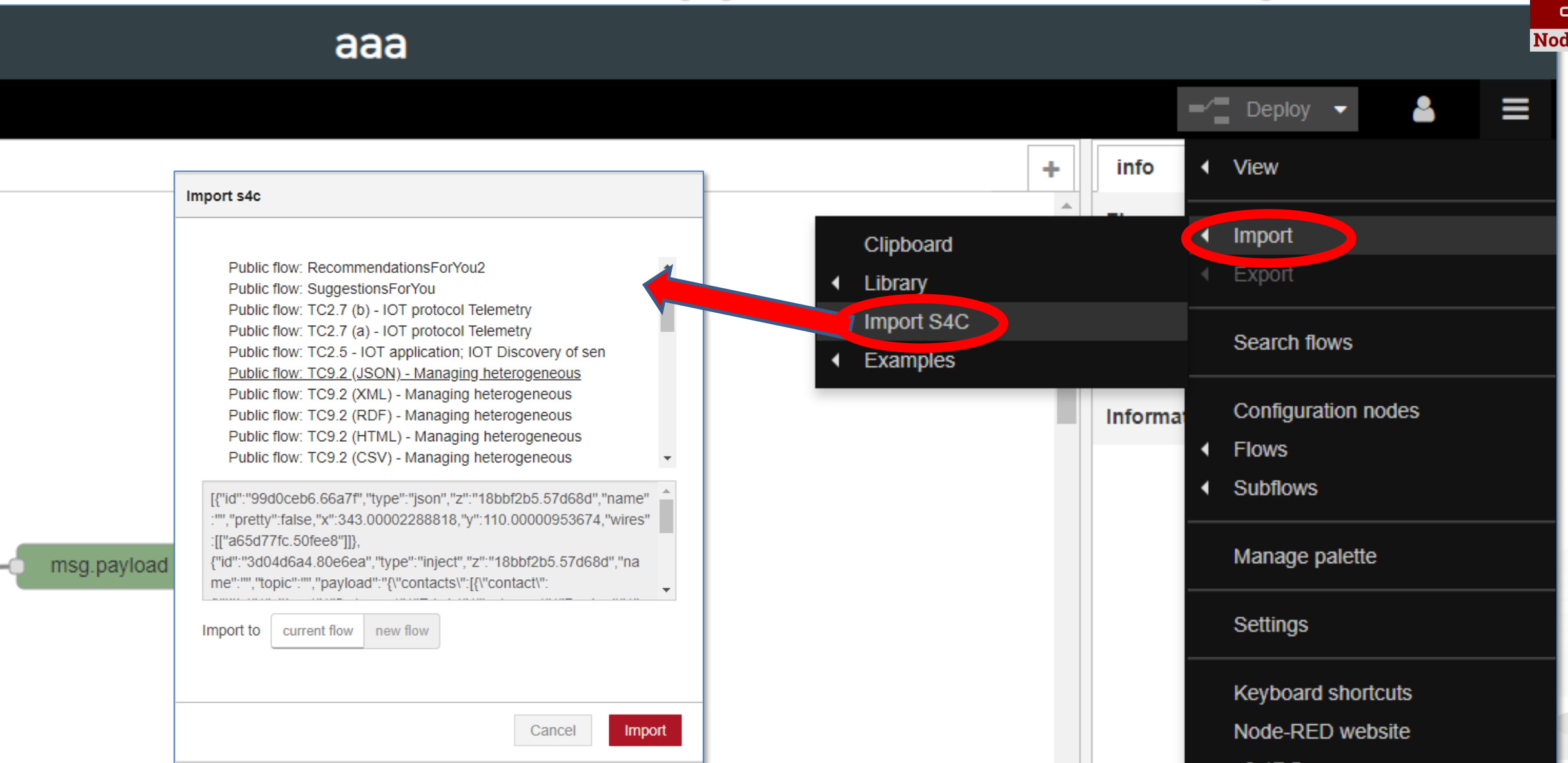
https://flows.nodered.org/





Two views of the same libraries

Load an IOT application of example

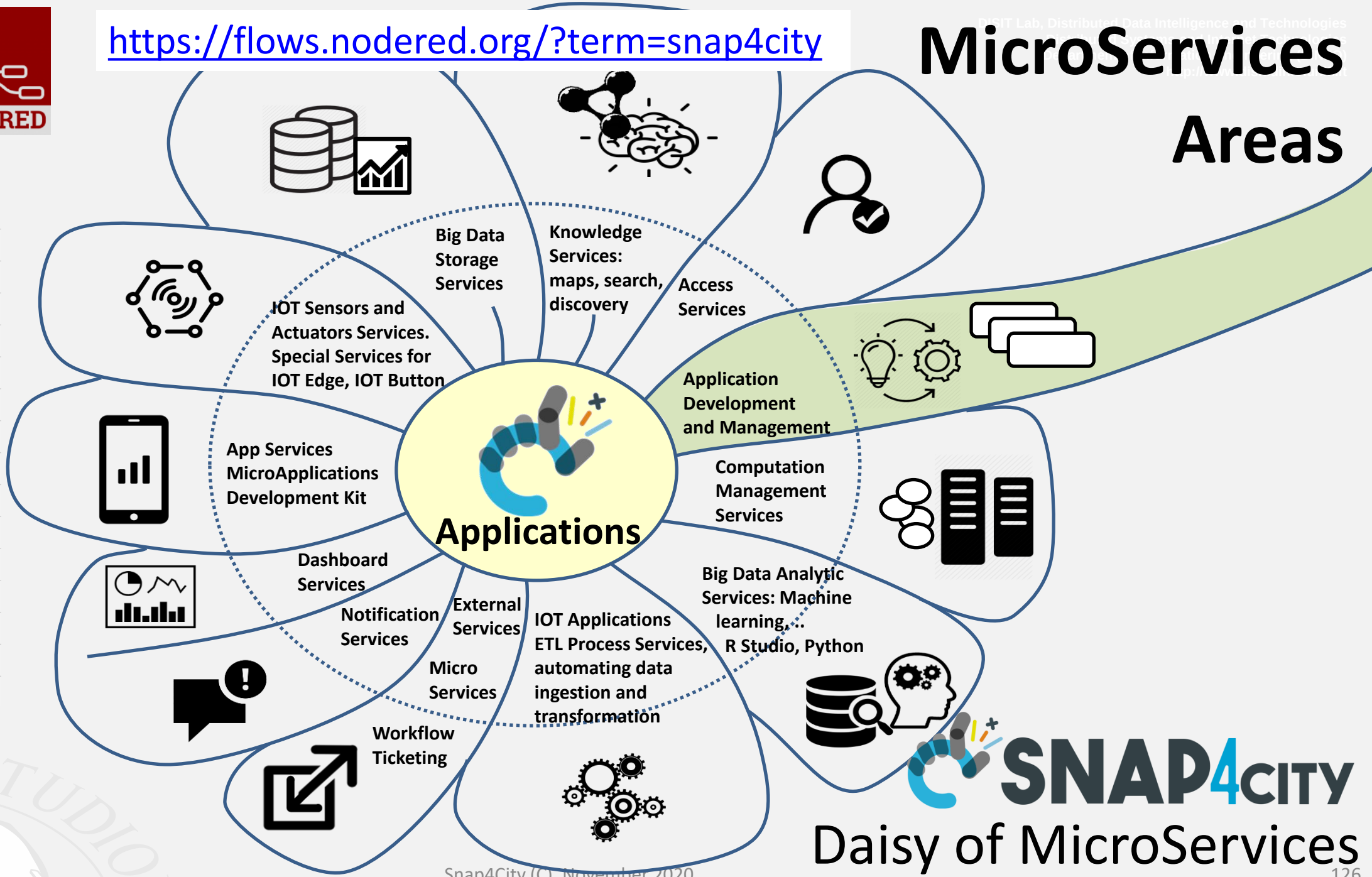


- input
- output
- function
- social
- storage
- analysis
- advanced
- NGSI
- Iwm2m
- S4CSearchDev
- S4CUtility
- S4CMapping
- S4CManagement
- S4CDataAnalytic
- S4CBigData
- S4CIOTApp
- S4CSearch
- S4CData
- S4CKPIData
- S4CDashboard
- S4CSigfox
- S4CIoT
- S4CLogDev
- S4CView
- S4CSocial
- location
- dashboard

Node-RED

https://flows.nodered.org/?term=snap4city

MicroServices Areas



SNAP4CITY

Daisy of MicroServices

Smart City and IOT main needs

Smart City Entities Search: search and access to city entities and their relationships in the city.

Historical Data: search and access to data collected over time into the smart city data aggregator.

Save and Get Personal Data: for many smart city applications, the possibility of saving and retrieval of personal data enables a large variety of smart scenarios for the final users and operators.

Advanced Dashboards: This means to have the possibility of developing a real user interface of the IOT App (to render and produce data for the IOT network).

Data Analytic: The real need in the context of smart City is to have the possibility for a data-analysts of creating some data analytic processes and use it into the flow as MicroService without the intervention of a programmer nor administrator.

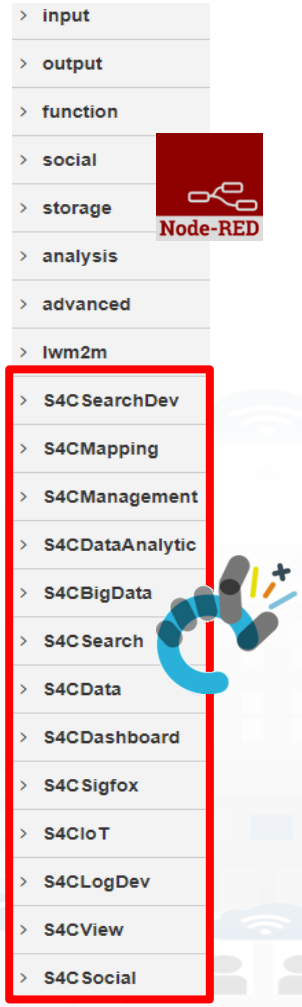
IOT Device Connection: This means that the developers expect to have the possibility of using nodes for connecting to a large set of IOT devices using different protocols, and thus connecting to different kind of IOT brokers.

IOT Directory: It should be a single point service for searching, managing and discovering all the IOT Devices which can be connected to the infrastructure by means of a large set of heterogenous IOT Brokers.

IOT Applications

IOT Applications = Node-RED + Snap4City Platform

- A collection of more than **150 MicroServices** have been developed covering the above-mentioned requirements and much more.
- The issue was not only to formalize the MicroServices, but also to create the infrastructure that enable their usage. In many cases, the simple MicroServices hide very **complex** and **sophisticate tools and algorithms (Snap4city Platform)**.
- They are formally distributed as two official libraries of Node-RED nodes (**Snap4City Basic and Advanced**) by the JS Foundation portal.
- They can be **directly installed** in any Node-RED tool of any operating system.



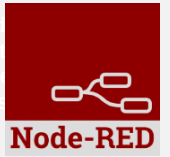


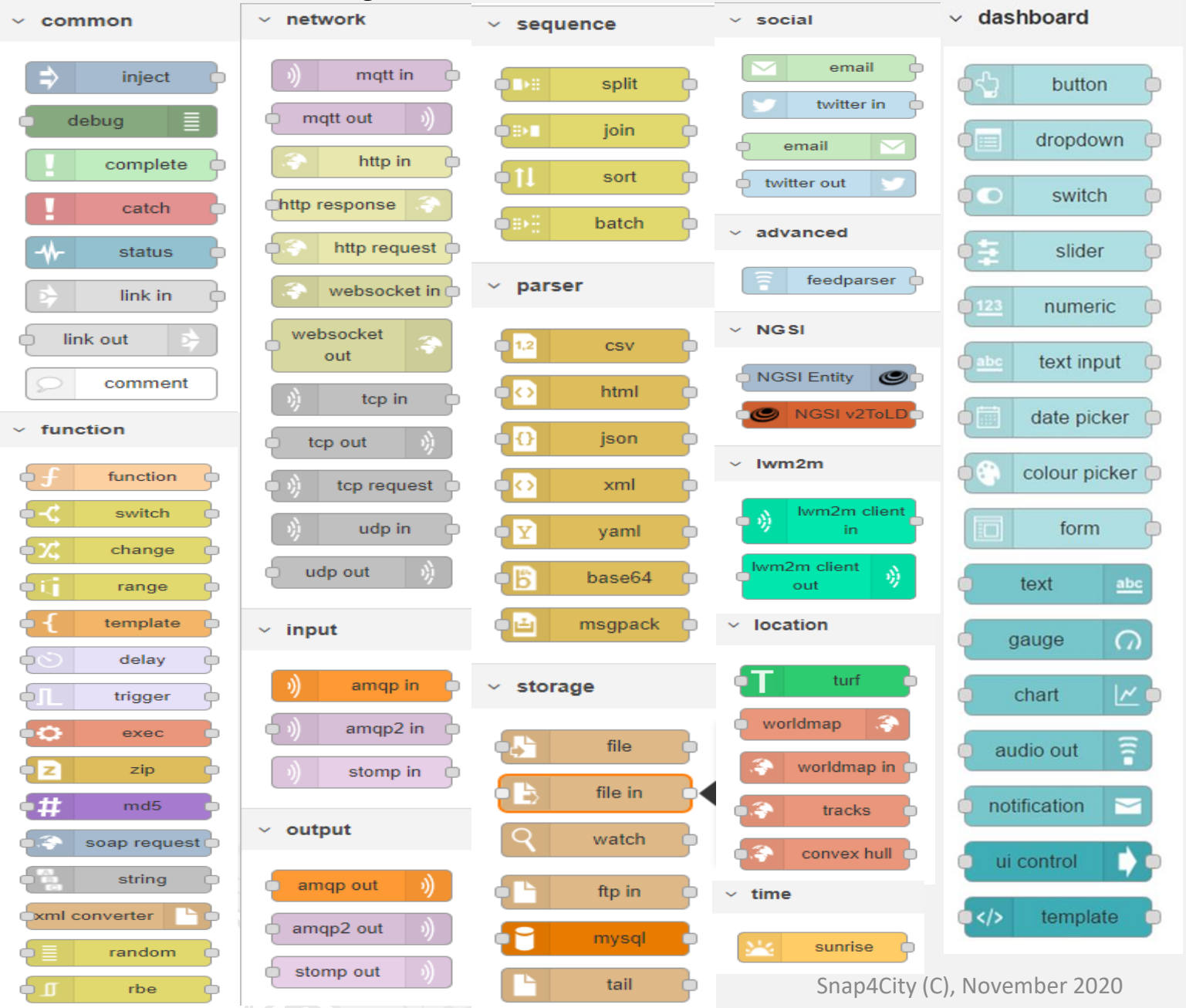
https://flows.nodered.org/?term=snap4city

node-red-contrib-snap4city-developer
Node-red nodes for developing IoT applications for smart cities. These nodes are
v0.1.5 18 node

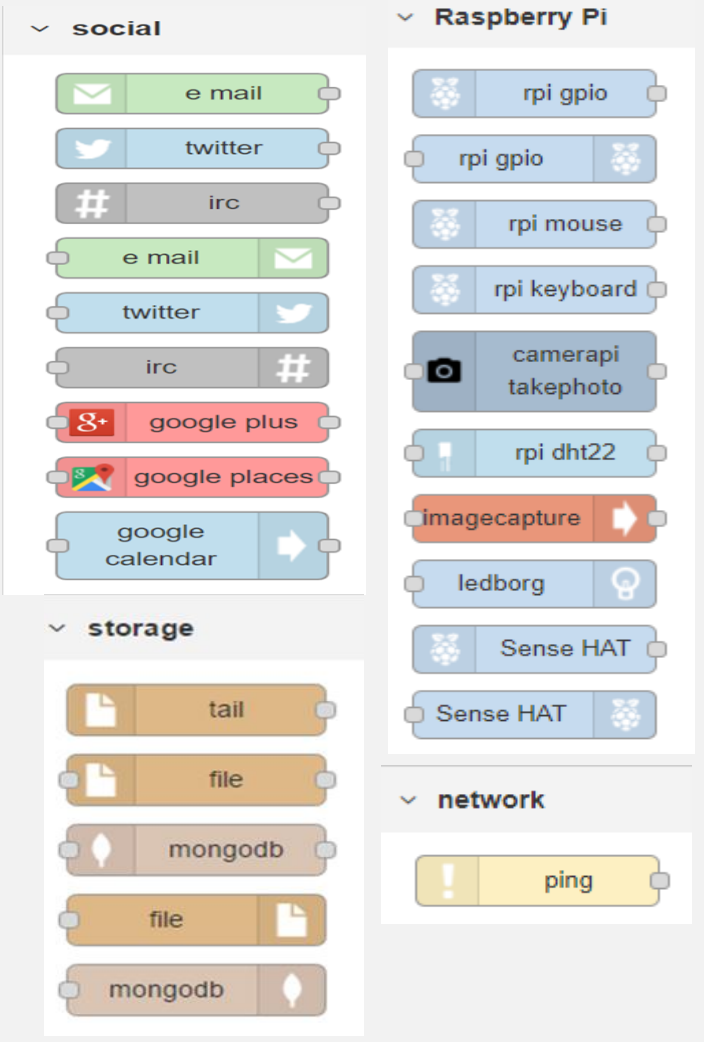
node-red-contrib-snap4city-user
Nodes for Snap4city project, targeted to standard user (no developer)
v0.2.0 27 ★5.0 (1) node

Basic Node.js Blocks on NodeRed on our Advanced IOT Apps





+ on IOT Edge Raspberry

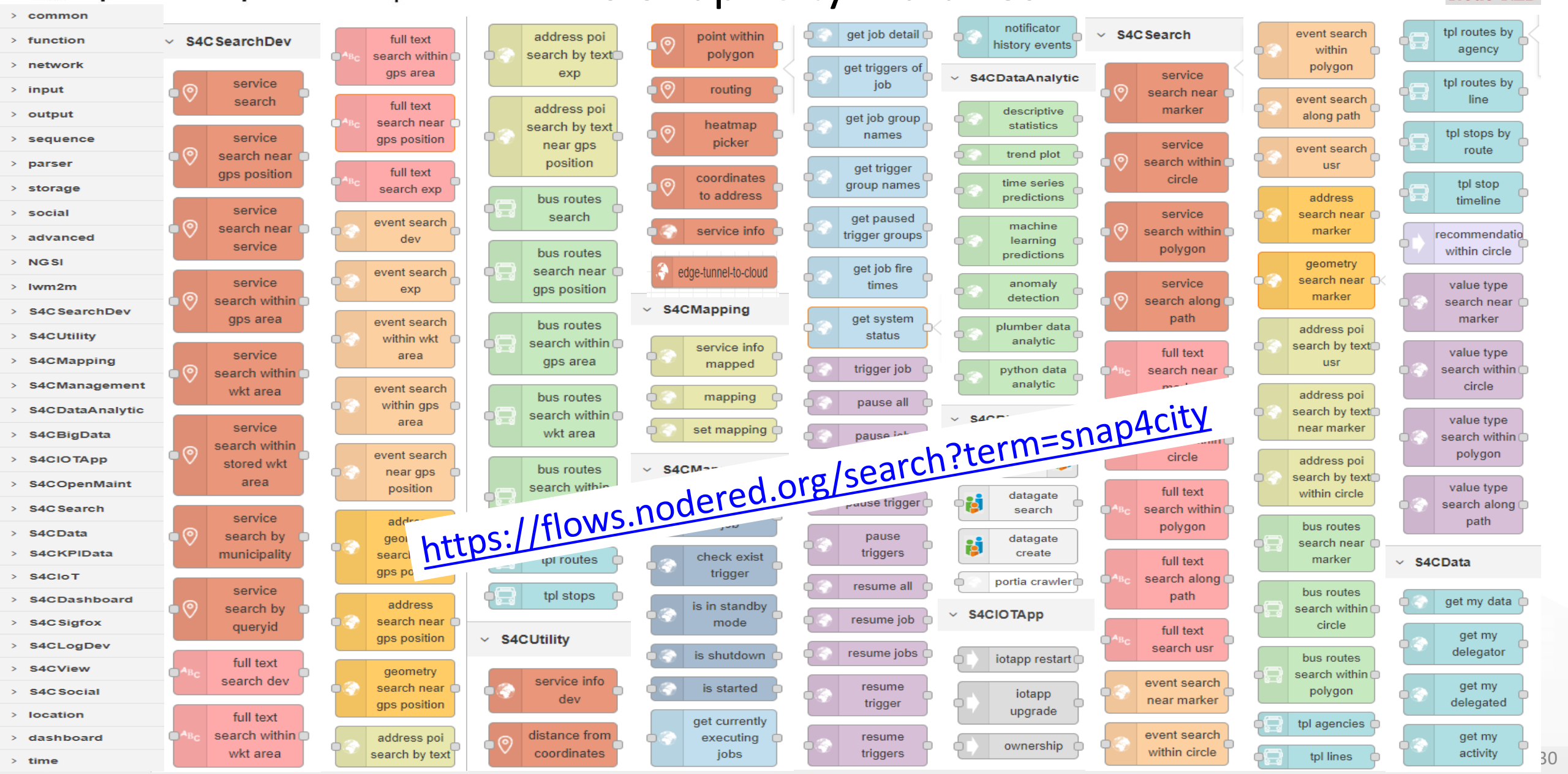


UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

Nov. 2020 collection

Two Snap4City Libraries

SNAP4CITY | Node-RED



https://flows.nodered.org/search?term=snap4city

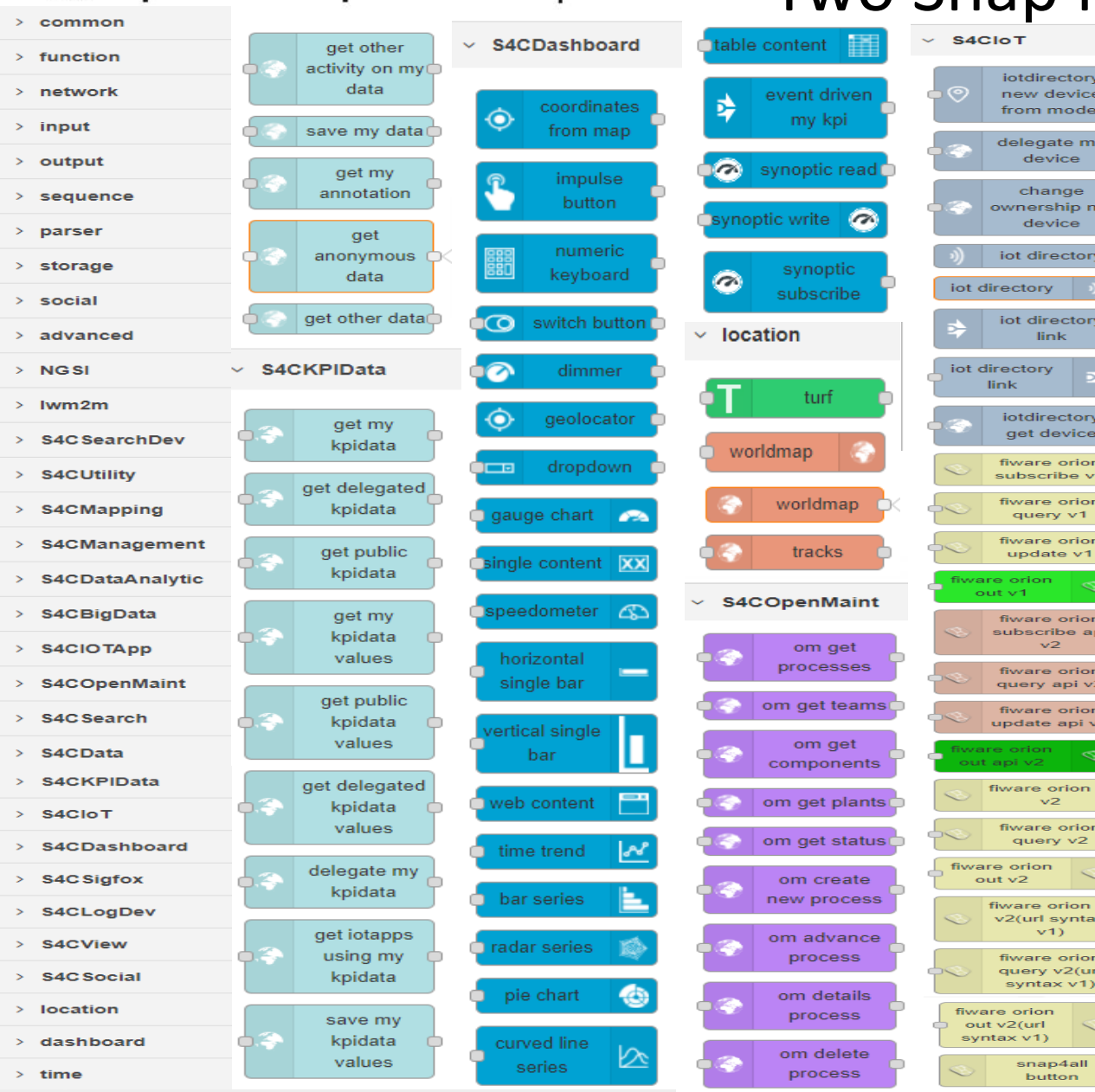
UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

Nov. 2020 collection

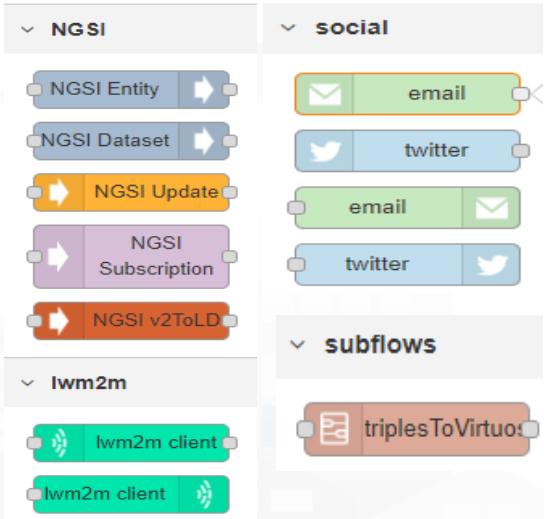
Two Snap4City Libraries

SNAP4CITY Node-RED

https://flows.nodered.org/search?term=snap4city



We suggest also to install:



UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

MicroServices Suite for Smart City Applications

- C. Badii, P. Bellini, A. Difino, P. Nesi, G. Pantaleo, M. Paolucci, Sensors, Vol.19, 2019, ISSN 1424-8220
- https://doi.org/10.3390/s19214798
- https://www.mdpi.com/1424-8220/19/21/4798/pdf



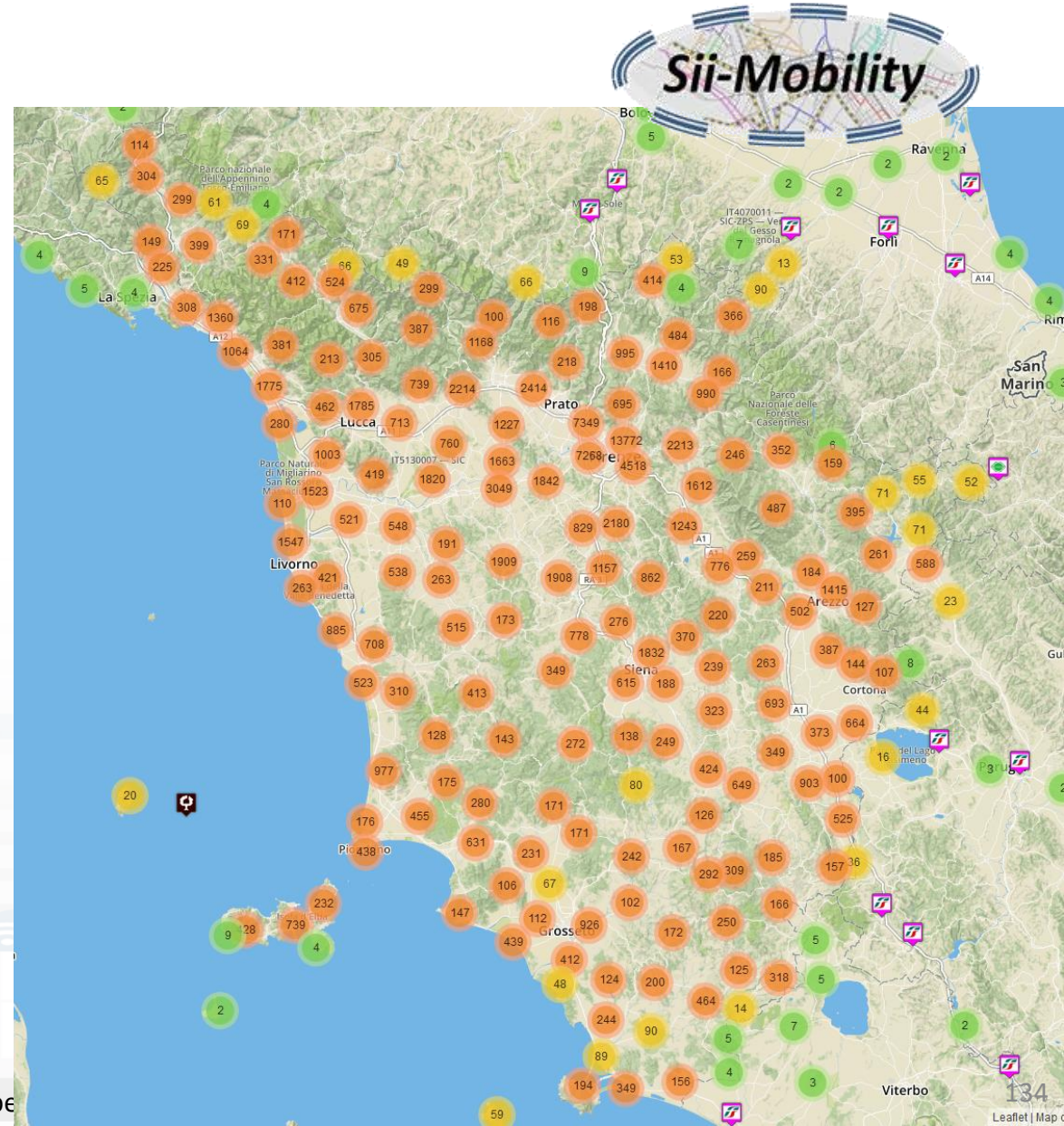
IOT App = Node-RED + Snap4City

search vs services, the ServiceURI



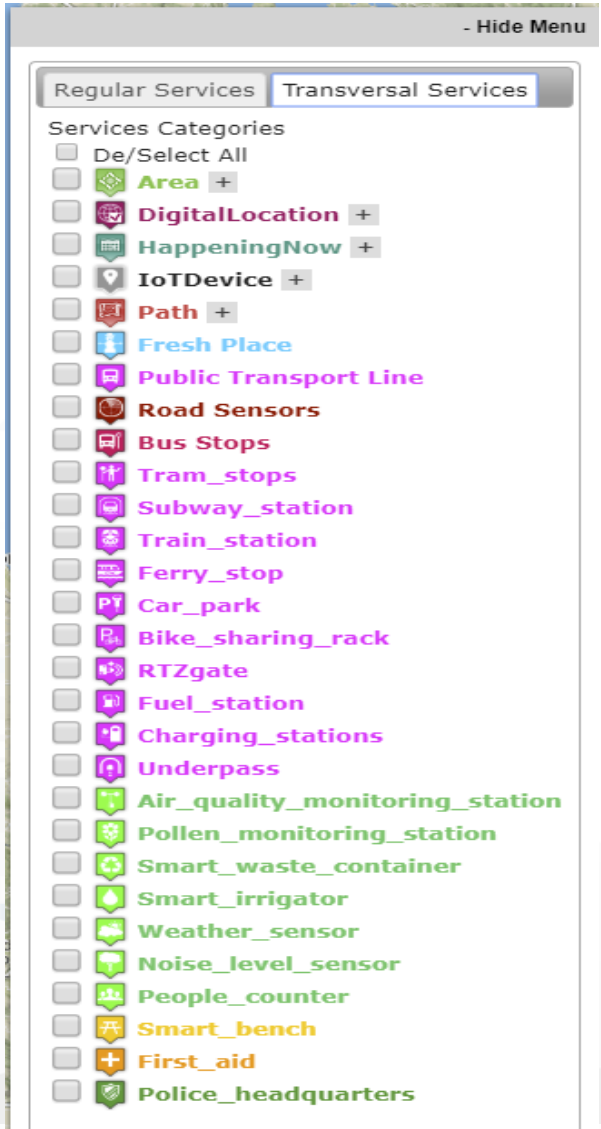
Data Domains

- **Street and geoinformation of the territory and details for routing, navigation, …**
- **GeoResolution, Environmental data**
- **Mobility and Transport**: public and private, public transport, parking status, fuel stations prices, traffic sensors, etc.
- **Culture and Tourism**: POI, churches, museum, schools, university, theatres, events in Florence
- **Environmental**: pollution real time, weather forecast, etc.
  - Environmental data geo resolution
- **Social Media**: twitter data
- **Health**: hospital, pharmacies, status of the first aid triage in major hospitals, …
- **Alarms**: civil protection alerts, hot areas, …



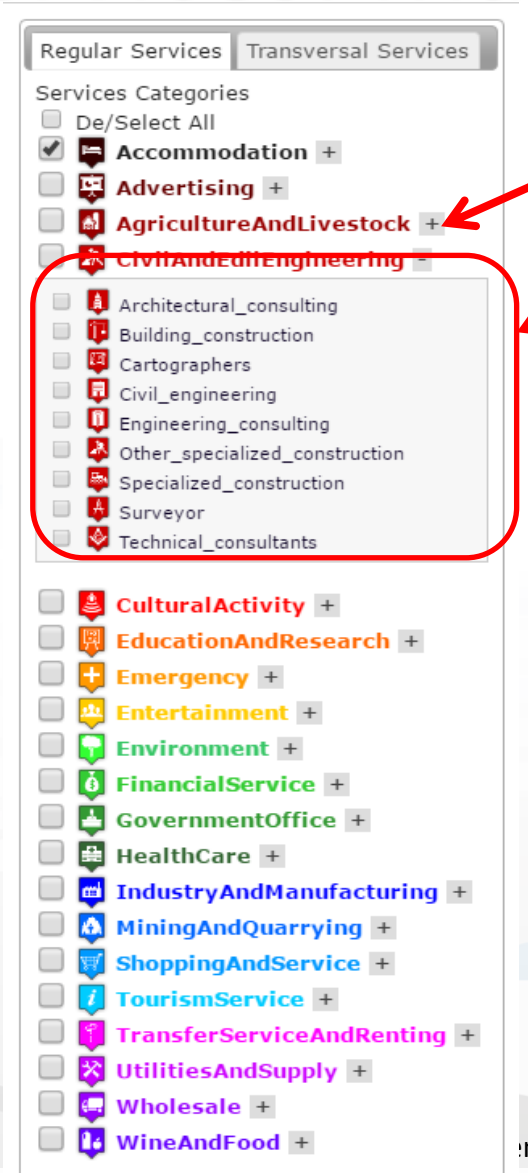
Access to Point of Interest information, POI

- **POI**: point of interest
- **type**: macro (nature) and subcategories (subnature)
- **Position**: GPS, address, telephone, fax, email, URL, …
- **Description**: textual, multilingual, with images, …
- **Link** to dbPedia, Linked Open Data
- **Links to other services**
- **Real time data if any:** sensors data, timeline, events, prices, opening time, rules of access, status of services, status of queue, etc..
- *See transversal services on ServiceMap*
  - *Regular and in test platform*



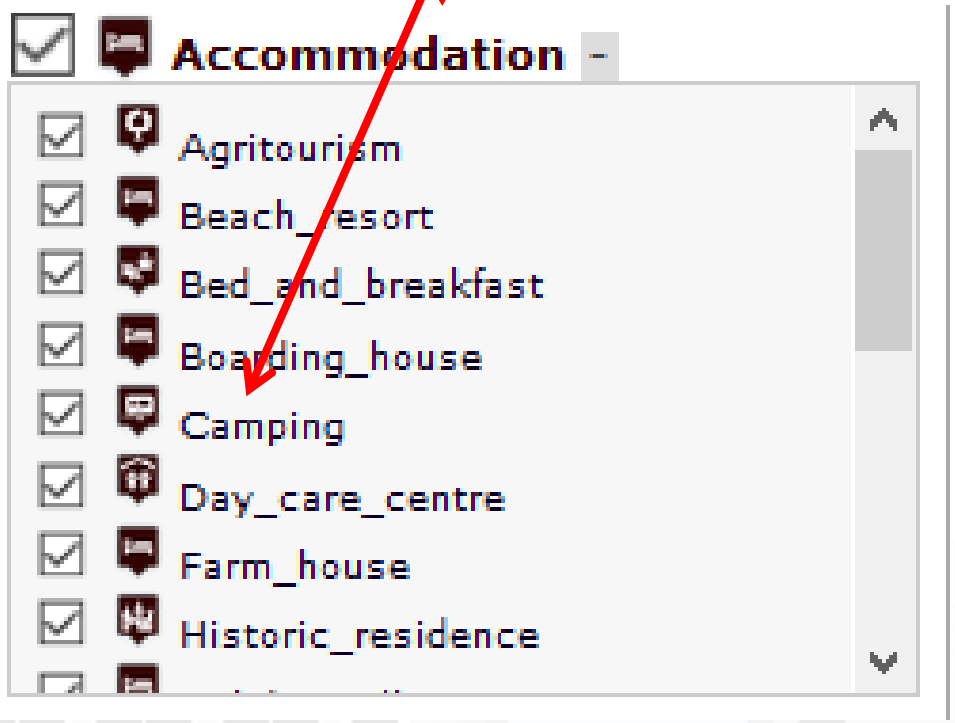
Concepts of Services: Macro and subcathegory

A SKOS area into the Km4CIty Ontology and Knowledge base for modeling POI and any element on map



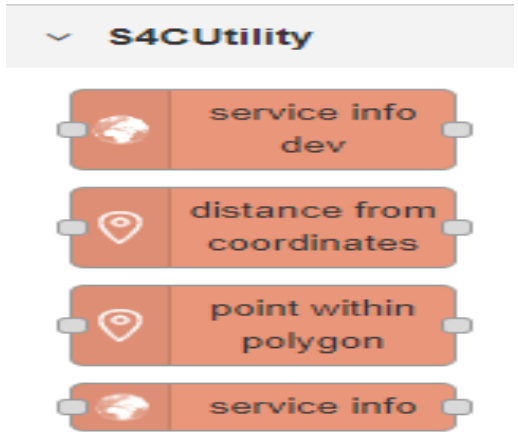
20 Service Macro Classes (The Nature)

Service subClasseses (the SubNature)

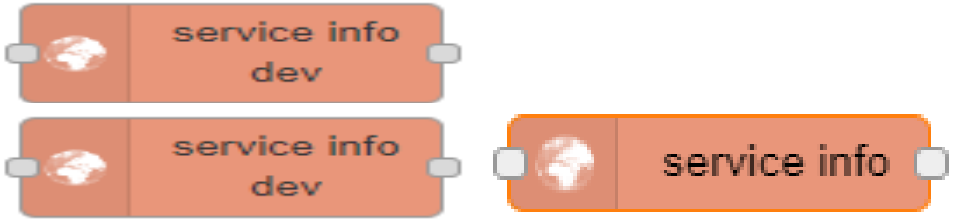


S4CSearch

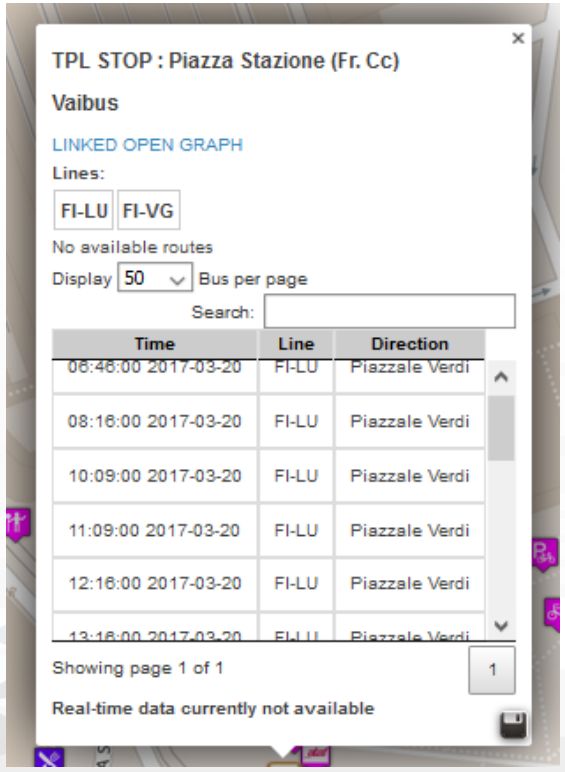
Node-RED

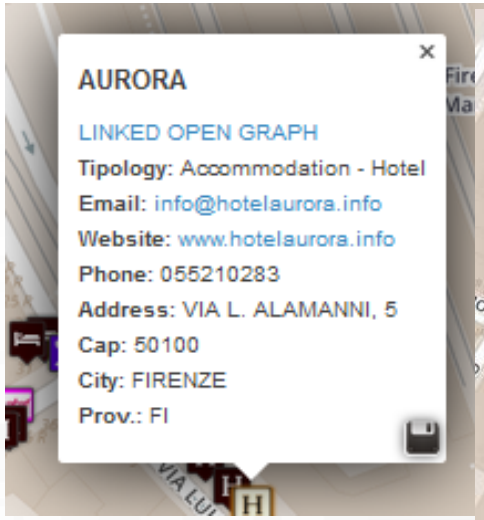


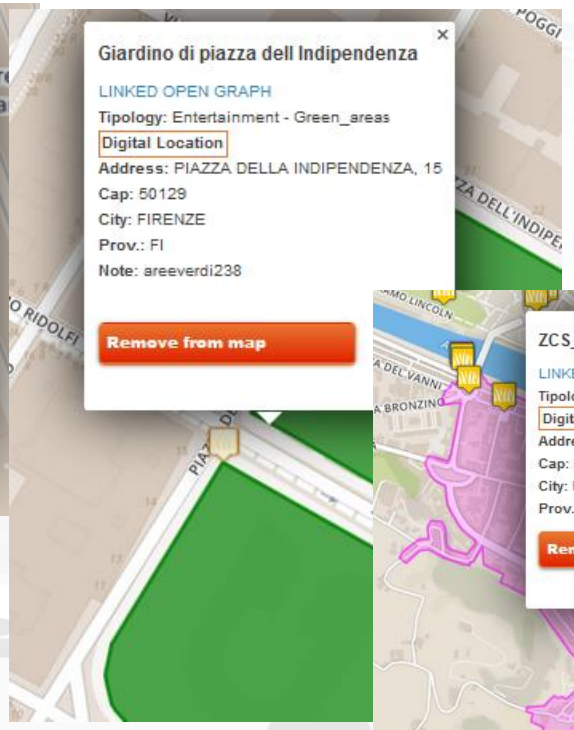
- ANY kind of sensors
- **To Get DATA of a Service / POI /sensor**
  - Historical and real time
  - Real Time

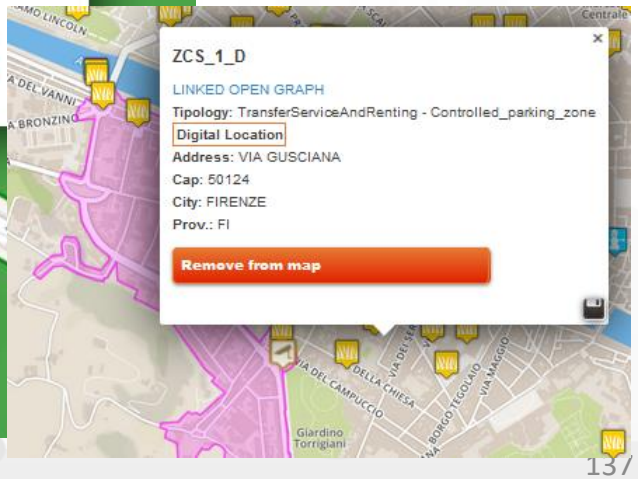




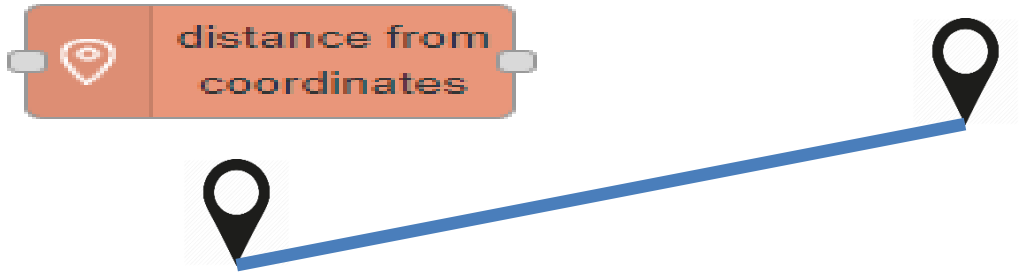




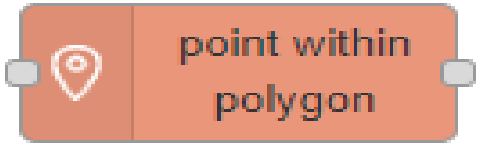




- **Distance from GPS point**



- **Point is in Polygon ?**
  - **Polyline as WKT**

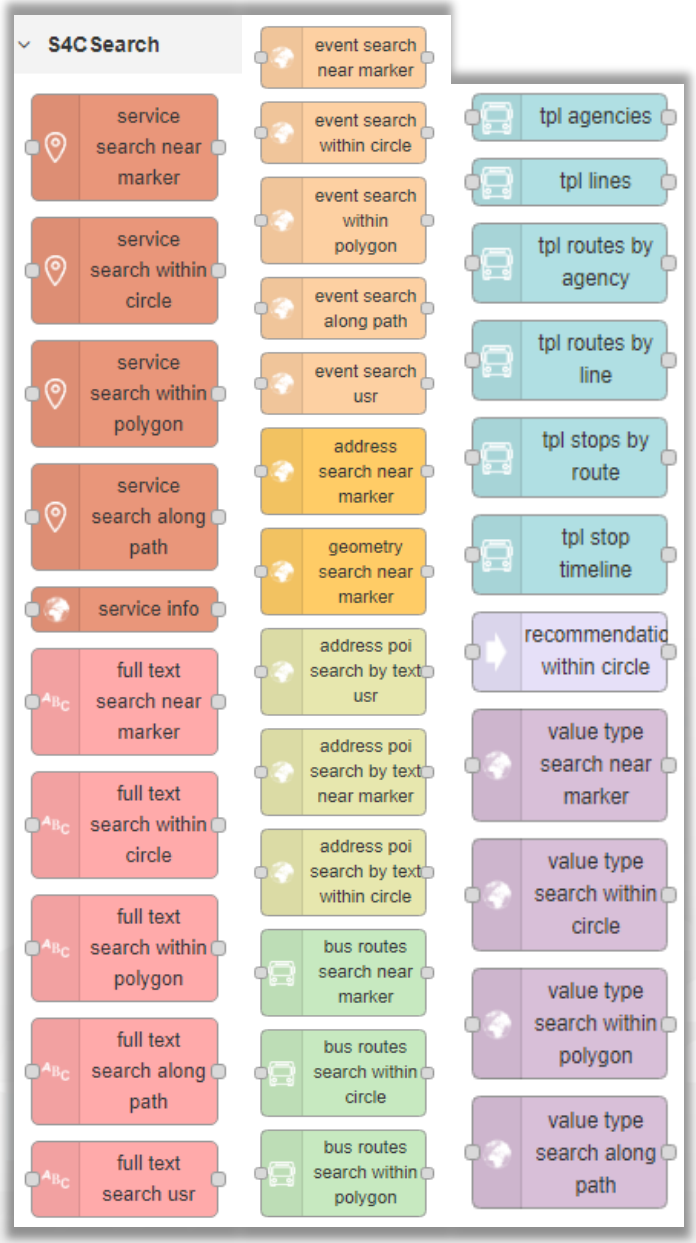




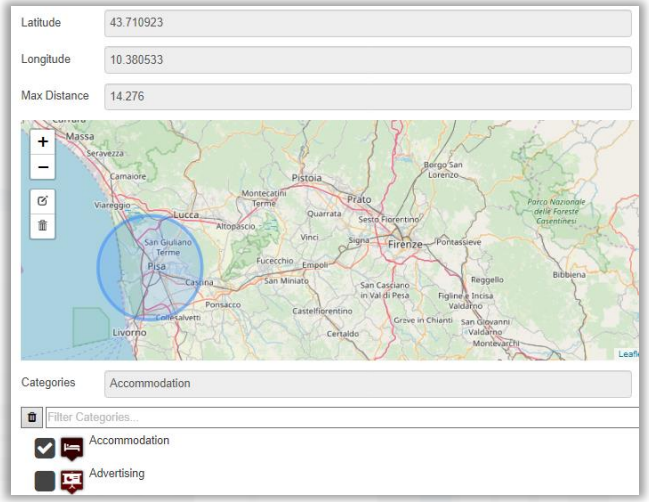
S4CSearch

Smart City Entities Search

Simple and Fast

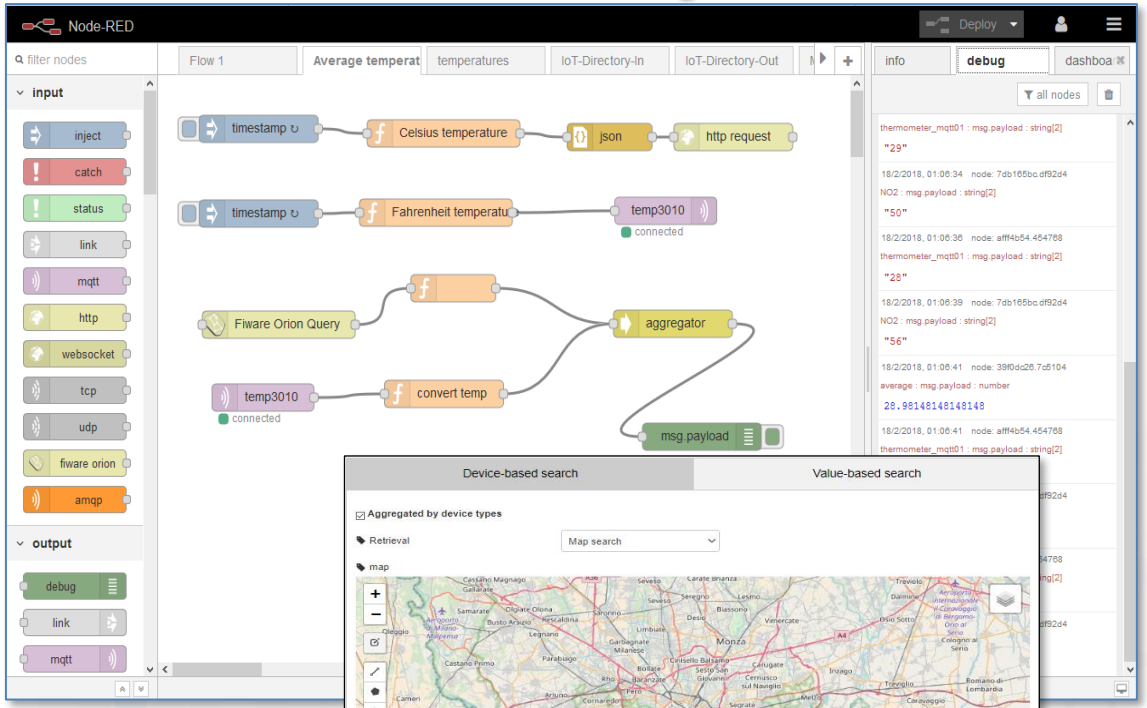


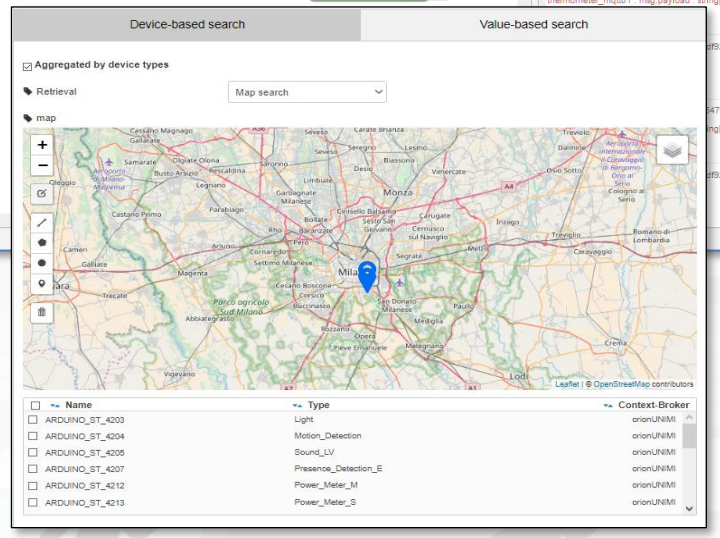
- **For example to search for:**
  - POIs:
    - near a GPS position, from text, along a path, in an area, etc..
  - Public Transport information / data
  - Suggestions
  - Public Transport Means Routes/Paths
  - Events in the area
  - Value Type (kind of data)
  - Etc.
- **To Get DATA of a Service / POI /sensor**
  - Real Time
  - ANY kind of senso

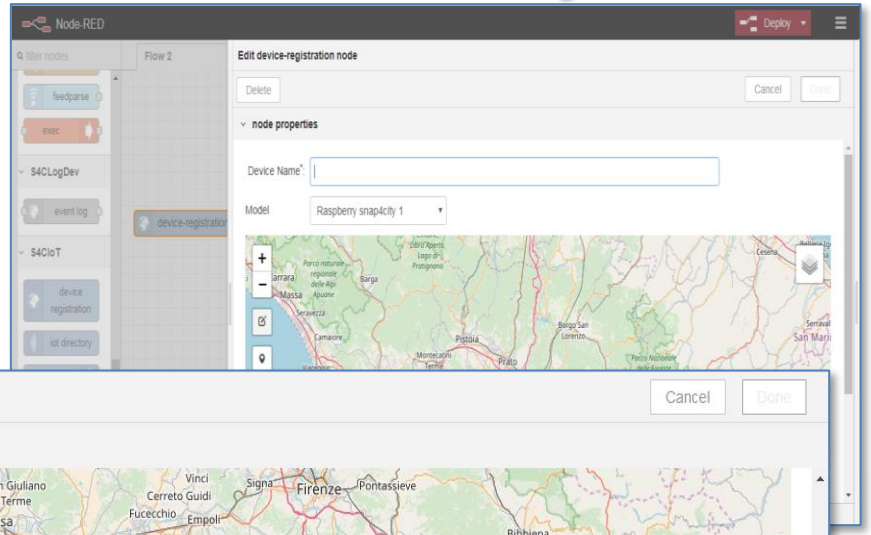


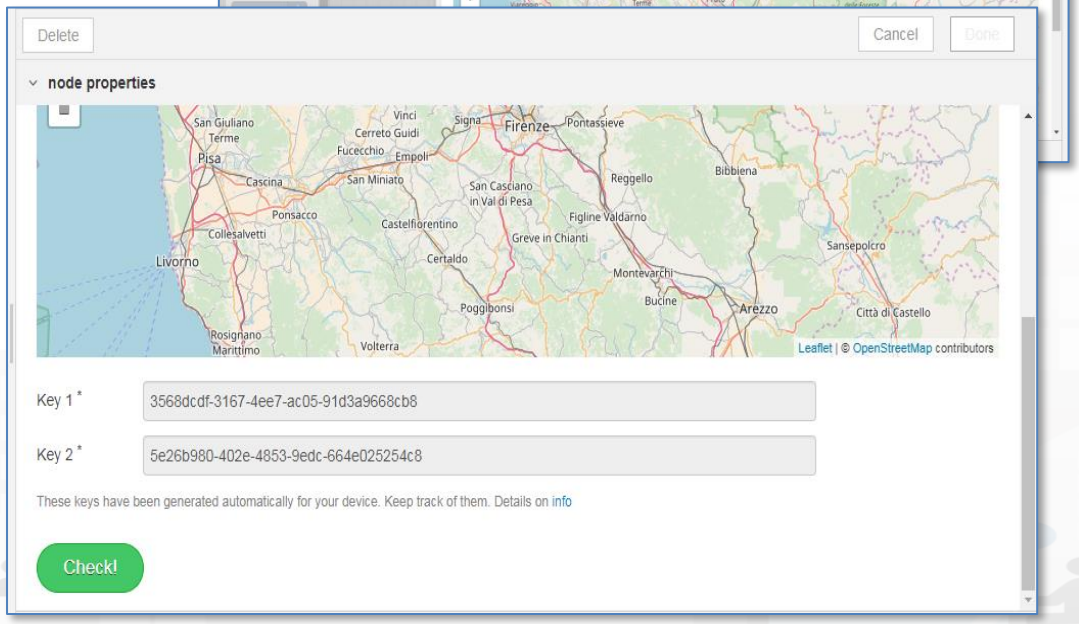


IOT Discovery on IOT Application Development

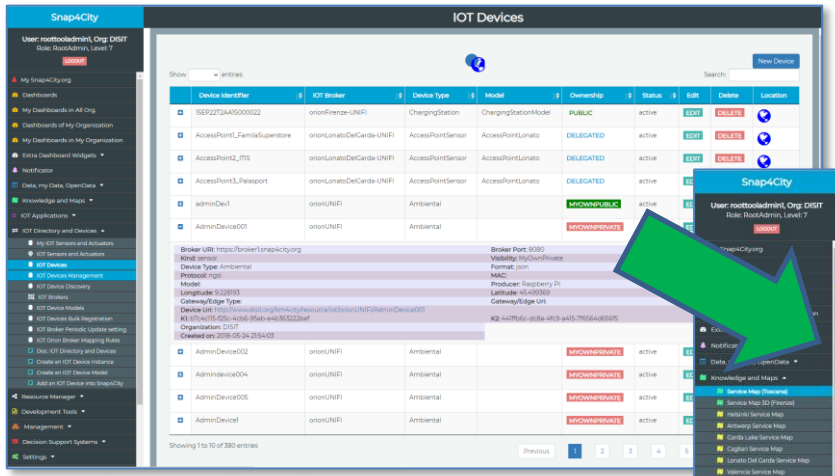




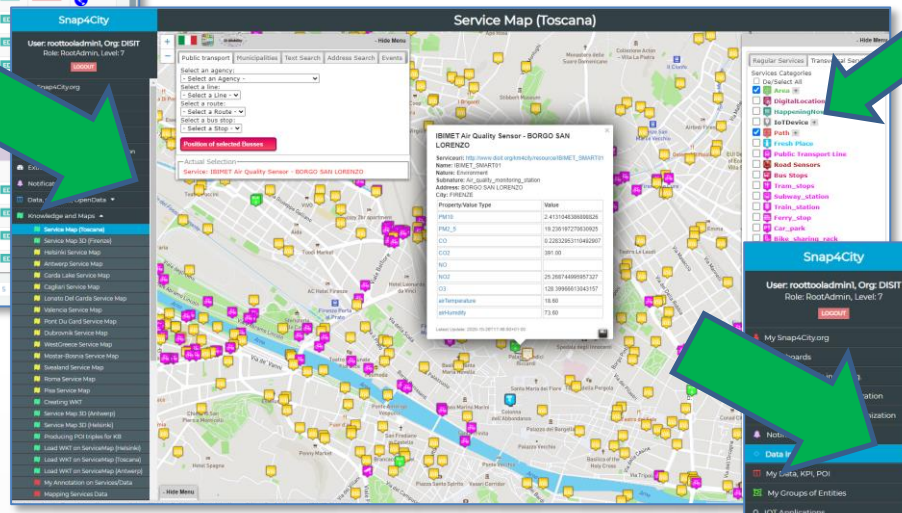




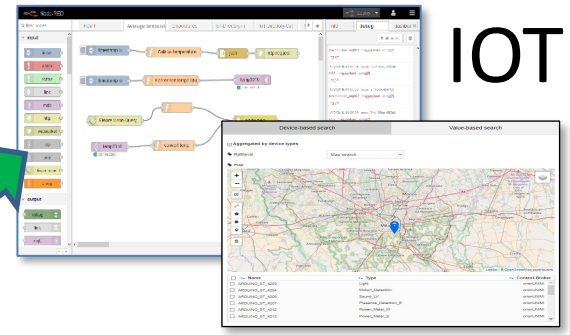
Data Registration Flow at a Glance



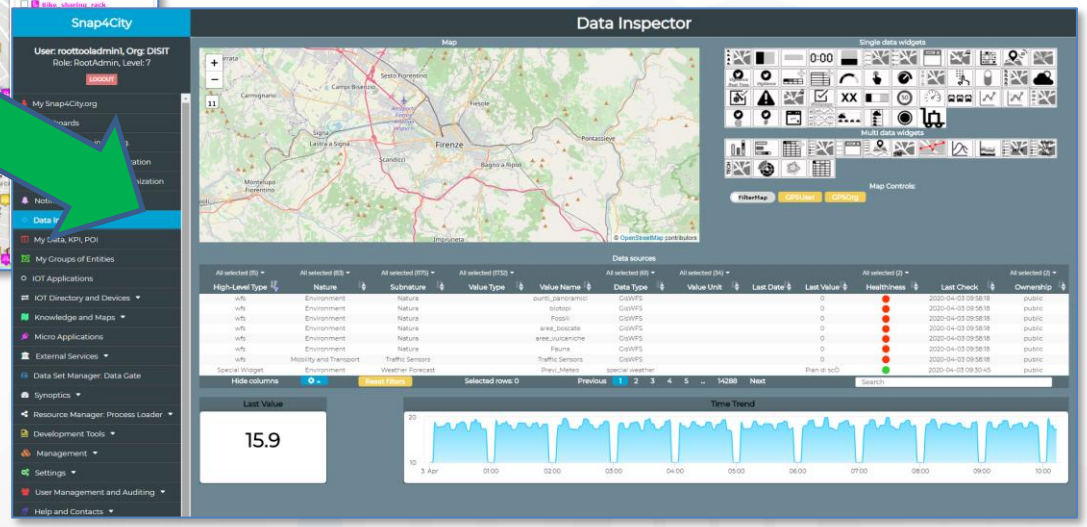
IOT Directory:
Devices…
Sensors..
Actuators…



Knowledge Base,
ServiceMap,
SmartCity API,
ASCAPI



IOT Apps



DataInspector
Dashboard Wizard

Notation Terminology

WHERE	Are synonymous at level of service which can be IOT device or entity with data	Are synonymous at level of the single attribute of the entity, device, service, etc.
IOT Directory	IOT Device	Sensor, Actuator, Attributes, Values (value name)
Knowledge Base, ServiceMap, SmartCity API, ASCAPI	Service, ServiceURI, SURI	Attribute, Metric
DataInspector, Wizard, Dashboard	ValueName	Sensor, Sensor Actuator, ValueType
IOT Applications, Node-RED	ServiceURI, SURI	SURI and its real time results of the objects into the data structure

ServiceURI, SURI of a sensor device:

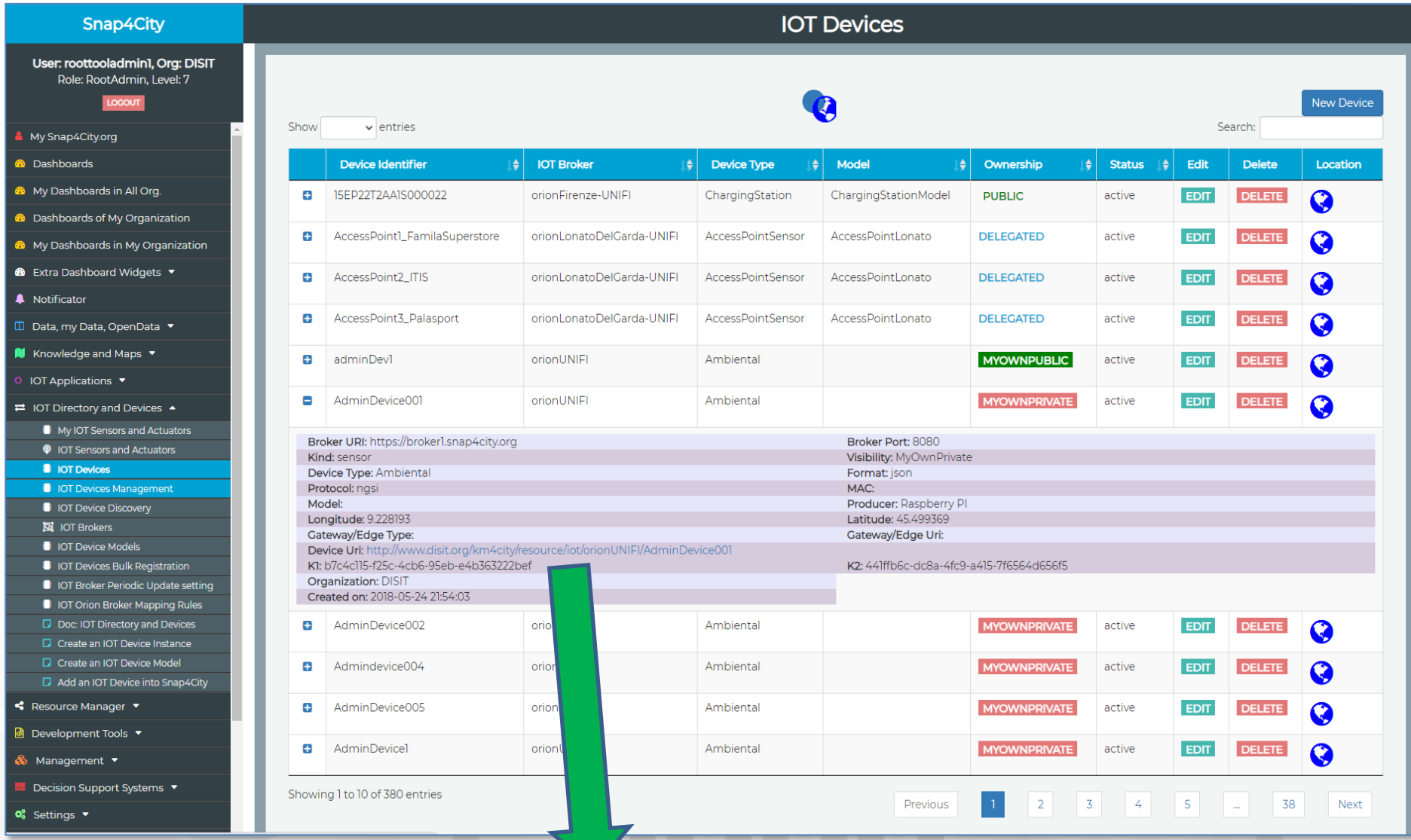
- http://www.disit.org/km4city/resource/METRO759
- http://www.disit.org/km4city/resource/iot/orionCAPELON-UNIFI/CAPELON/Streetlight%3A90FD9FFFFEBD5A7F

ServiceURI, SURI extended with attribute/variable/value:

- http://www.disit.org/km4city/resource/METRO759&metric=vehicleFlow
- http%3A%2F%2Fwww.disit.org%2Fkm4city%2Fresource%2FMETRO759&metric=vehicleFlow
- In some cases
  - http://www.disit.org/km4city/resource/METRO759/vehicleFlow

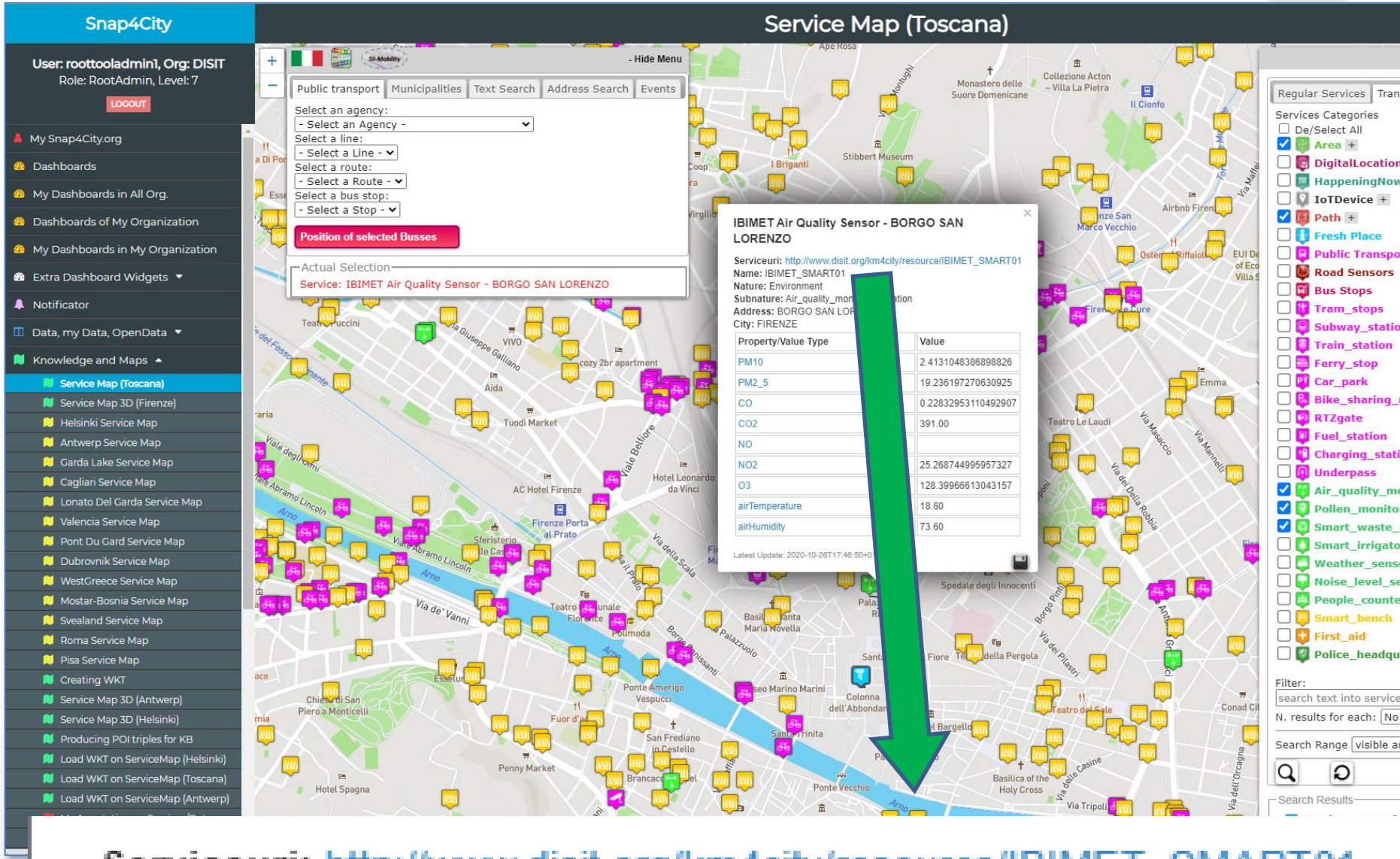
IOT Directory

- **For:**
  IOT Devices, Sensors, Sensor mobile, Actuators, Virtual Sensors, etc.
- Accessible as
  - ServiceURI
  - **Device URI**

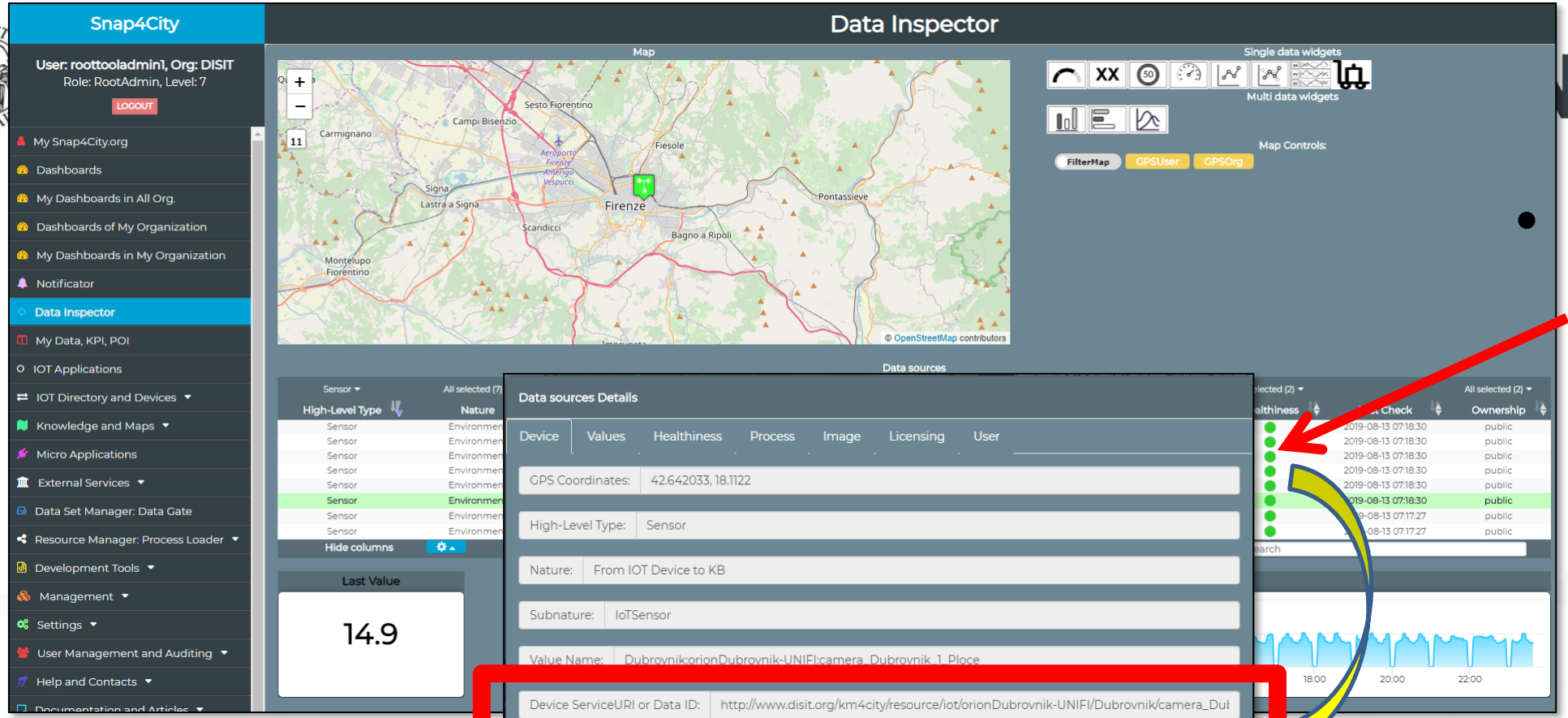


ServiceMap

- **For PUBLIC**:
  - IOT Devices, Sensors,
  - Sensor mobile,
  - Actuators,
  - Virtual Sensors,
  - POI, etc.
- See as
  - ServiceURI



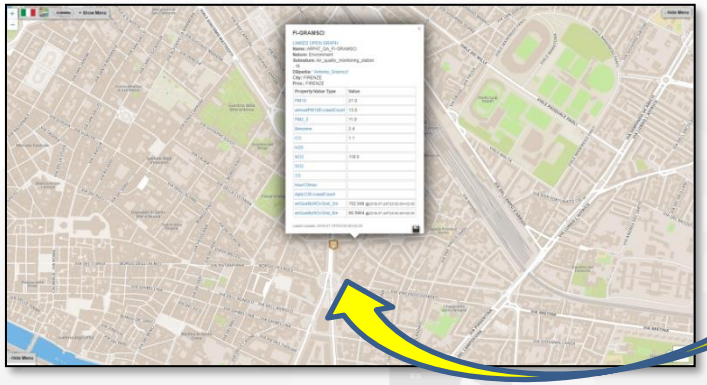
Serviceuri: http://www.disit.org/km4city/resource/IBIMET_SMART01

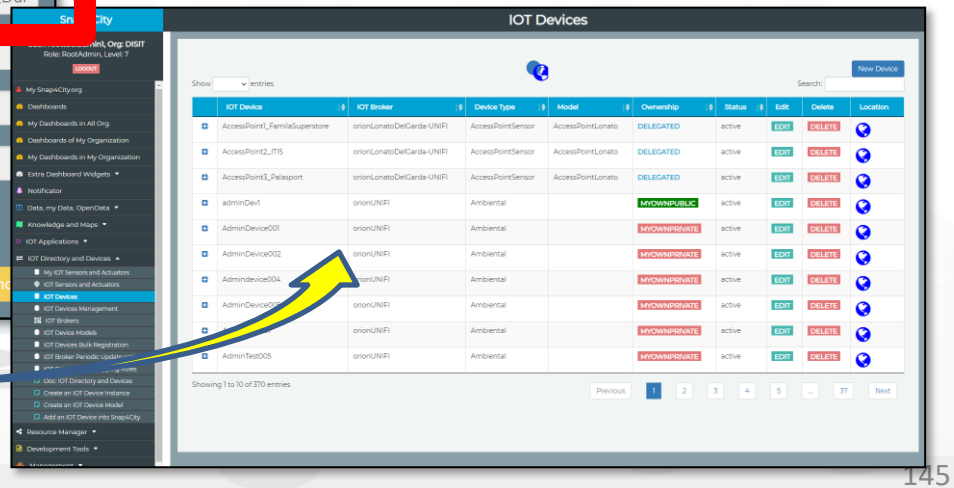


Knowledge Base view

- Click with the mouse on it

Data Inspector Wizard





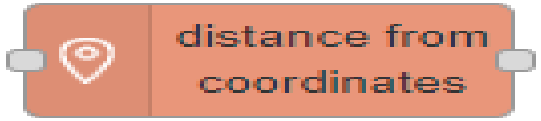
Some functionalities are limited to certain roles

S4CSearch



- **For example to search for:**
  - POIs:
    - near a GPS position, from text, along a path, in an area, etc..
  - Public Transport information / data
  - Suggestions
  - Public Transport Means Routes/Paths
  - Events in the area
  - Value Type (kind of data)
  - Etc.
- **To Get DATA of a Service / POI /sensor**
  - Real Time
  - ANY kind of sensors
- **Distance from GPS point**





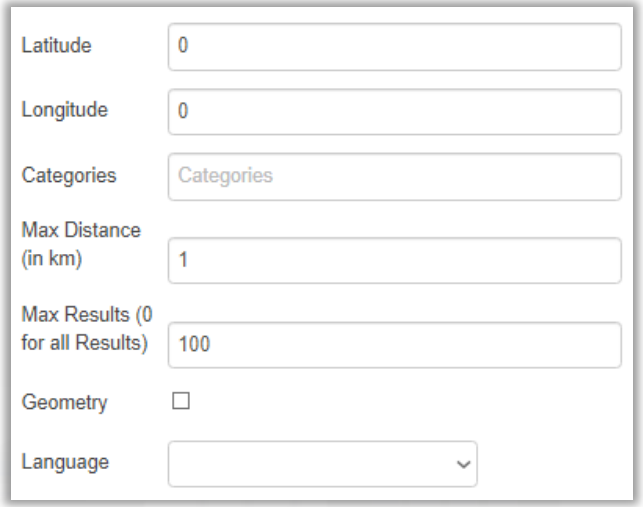
S4CSearch Adv

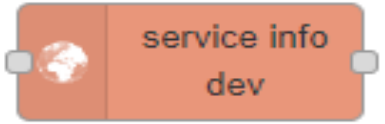
Smart City Entities Advanced Search

Flexiblity



- Similar to basic Search functions but with more flexibility of the function for programming the search
- Adding Dynamic behavior:
  - Getting in input JSON with parameters
- **To Get DATA of a Service / POI /sensor**
  - Historical and real time
  - ANY kind of sensors

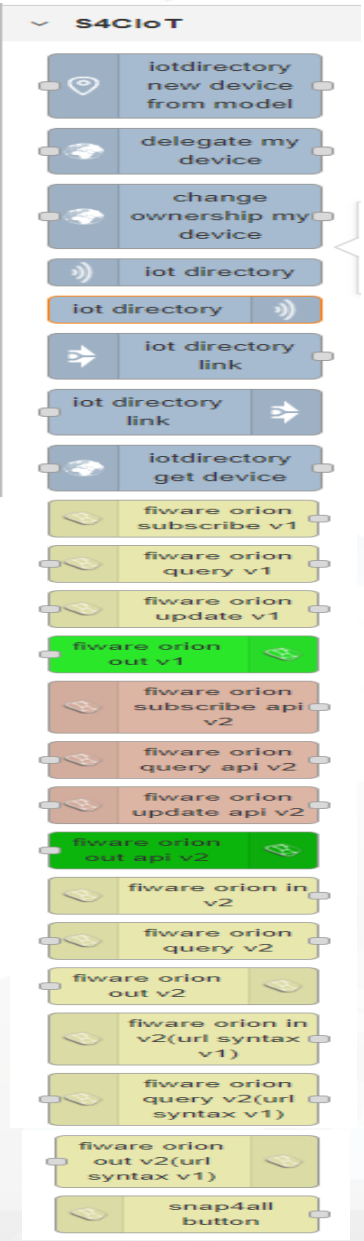




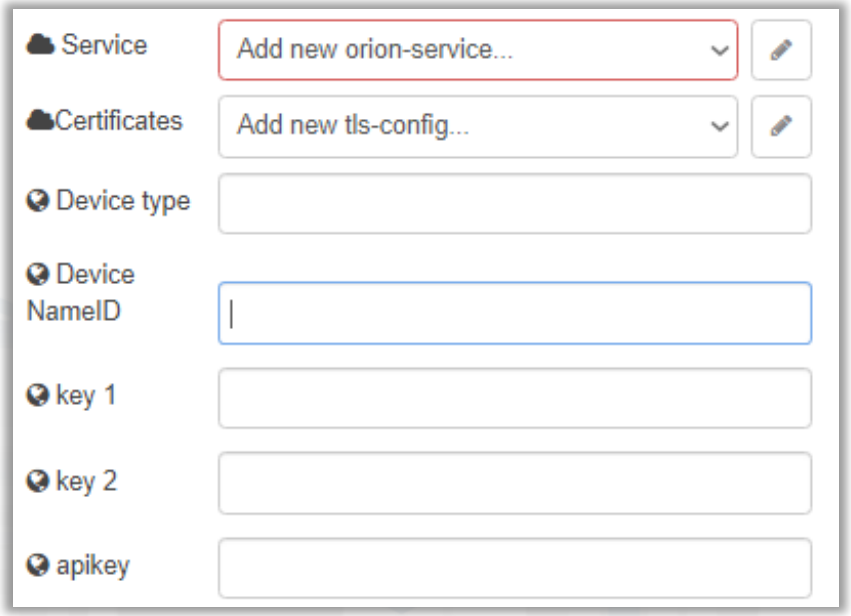
IOT App = Node-RED + Snap4City

IOT Devices NGSI just list see later





- **Search** for IOT Devices in a given area, or for kind (temperature, model, location, producer, Broker, …)
- **Subscribe** to one or more IOT Devices independently on their protocol, broker, owner, etc.
- **Send** data to IOT devices
- Establish with IOT Devices **Secure** certified Connections



- Please note that many other protocols can be also added, adding mode nodes, or registering IOT brokers to the Snap4City IOT Directory

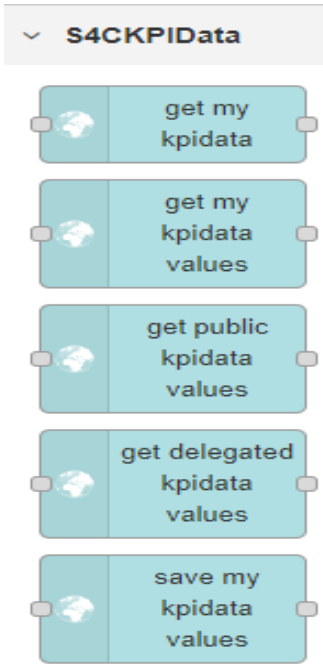


IOT App = Node-RED + Snap4City

Others nodes



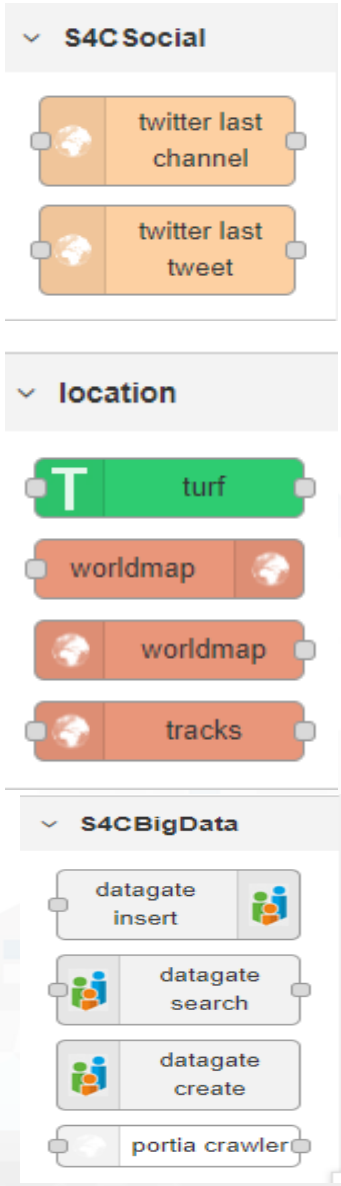
S4CKPIData



- Save and retrieve MyKPI into the safe personal data storage
- Access to MyKPI and to those that other user have delegated to Me
- **MyKPI are:**
  - Time series of data with GPS coordinates that can chage over time
  - Suitable for: moving sensors, trajectories, data from OBU, data from mobile, sensor data (if needed), etc. etc.
- **MyPOI are:**
  - POI with full metadata description and static coordinates

Mixt

Node-RED

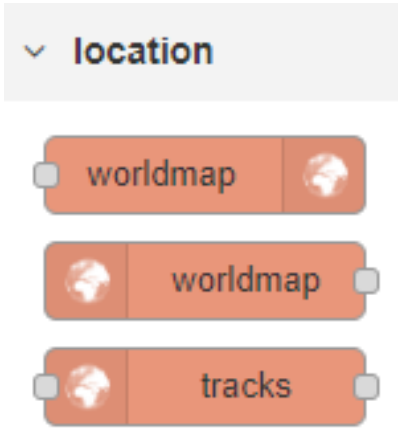


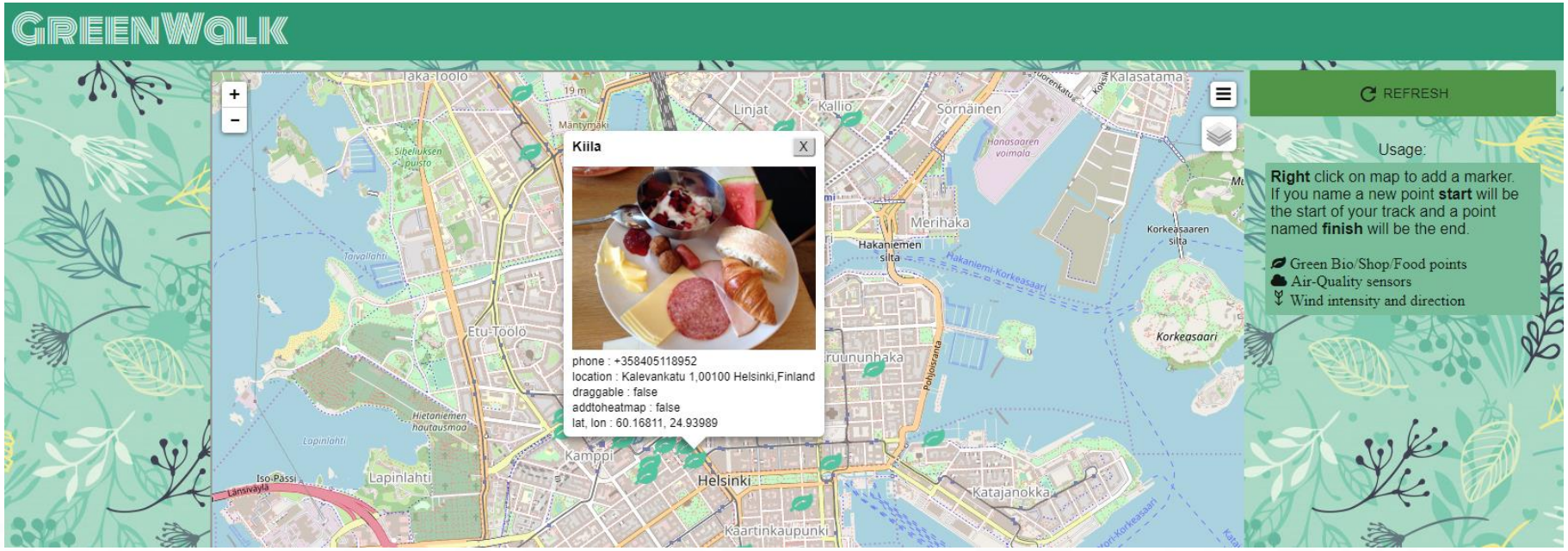
- Request metrics from Twitter Vigilance Channel service and engine of DISIT Lab
  - Different Twitter Vigilance services may be attached according to the Organization, different metrics and values

- Location services
- Maps and get position (raw solution)

- Getting data from DataGate/CKAN
- Publishing data to DataGate/CKAN
- Managing time series on DataGate/CKAN

Third party solution to: Control Maps from IOT Apps



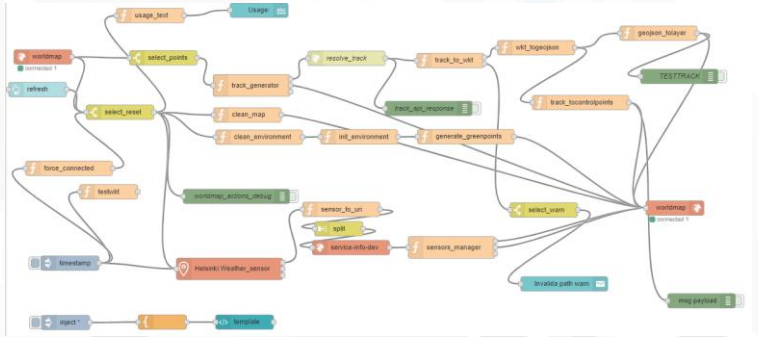


https://iot-app.snap4city.org/nodered/nrve0e3/ui/#!/0

- Show points on maps
- Get Points
- Tracks
- See examples on:

https://www.snap4city.org/409

https://www.snap4city.org/417



IOT App = Node-RED + Snap4City

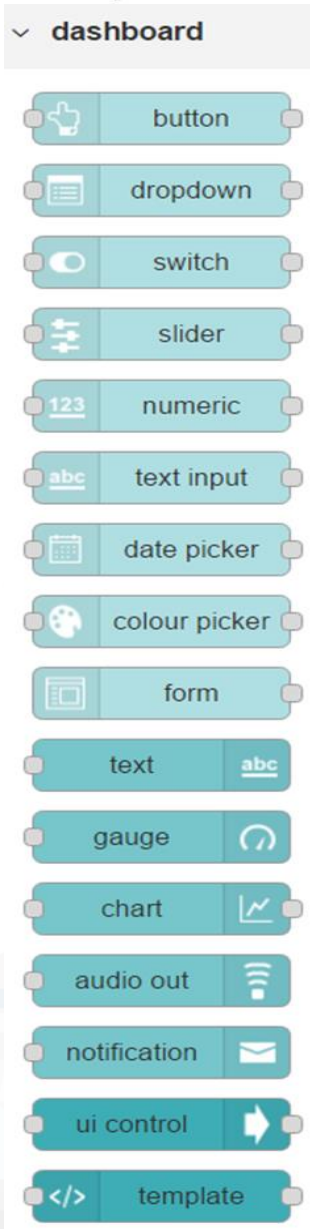
Dashboard Integration



UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

Dashboard

Native Local or Snap4City

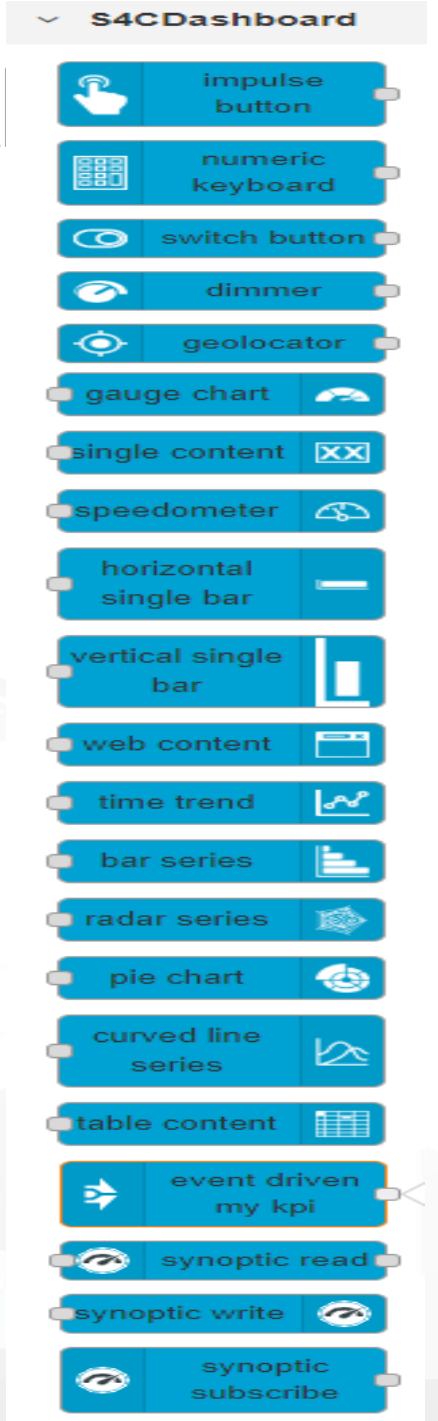


- Input/output
- non secure
- Limited in graphics
- No authentication
- No HLT
- No integration
- No historical data
- No Synoptics
- Etc..

- Local on IOT Edge

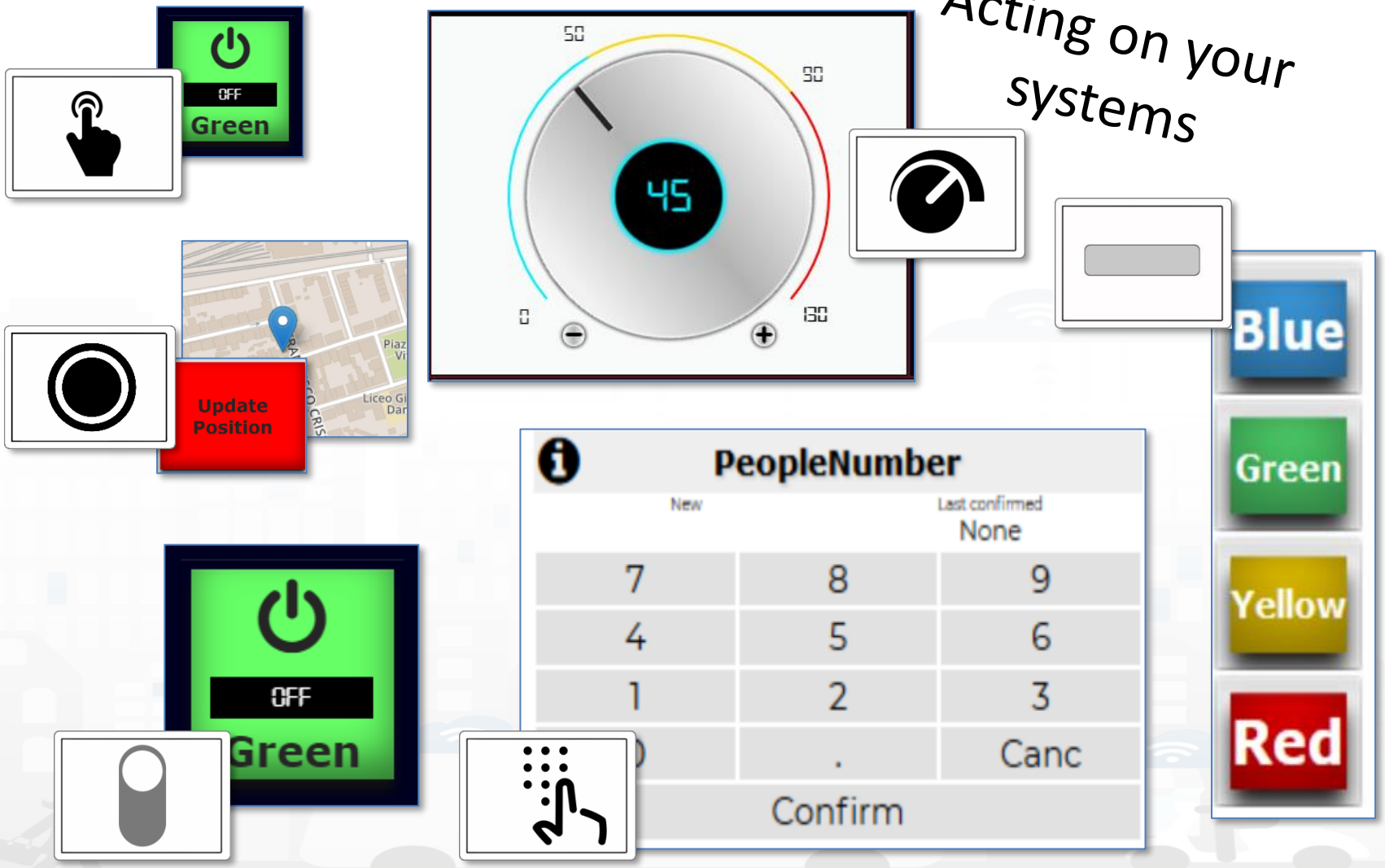
- Input/output
- Secure
- Advanced in graphics
- Single Sign On
- Several HLT
- Fully integrated
- Historical data
- Full Synoptics
- Etc..

- Remote for IOT Edge via WebSocket Secure



From Dashboard to IOT Devices

- **Widgets:**
  - Impulse Button
  - Button
  - Switch
  - Dimer/Knowb
  - KeyPad
  - geolocator
- **Registered** on some IOT brokers with NGSI mutual authentication

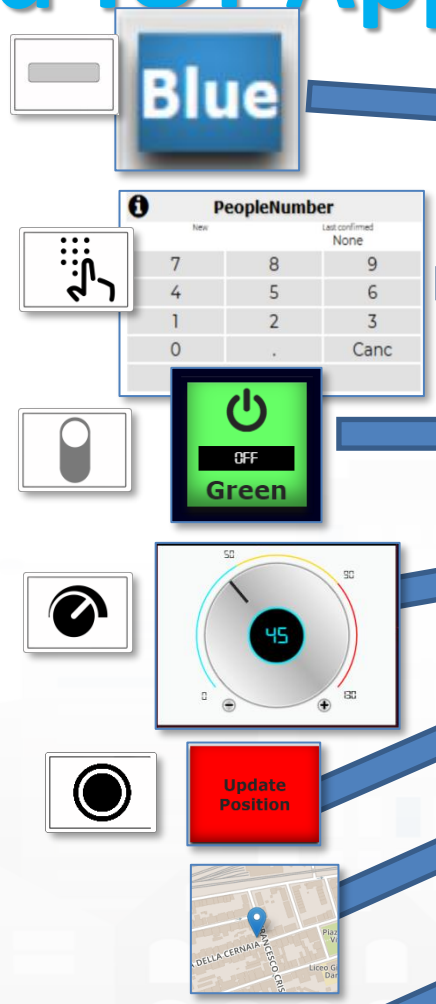


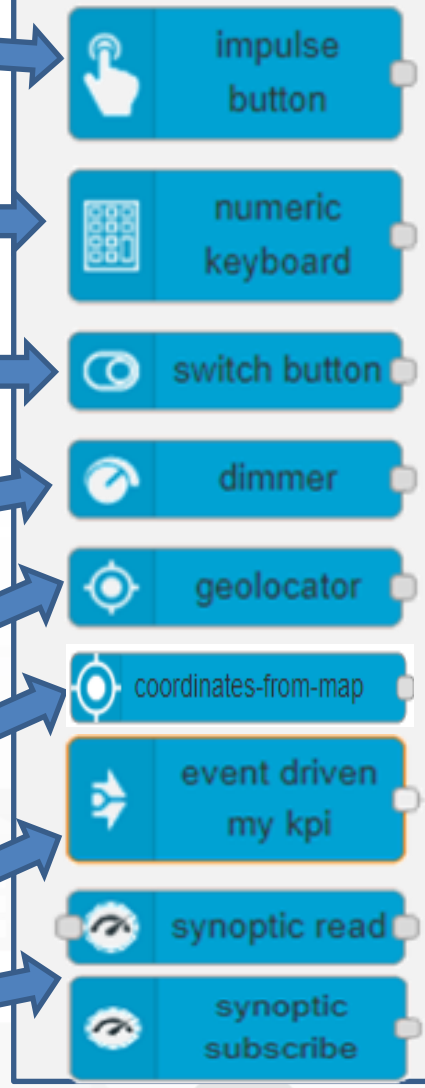
Nature

Dashboard-IOT App

From Dashboard to IOT App

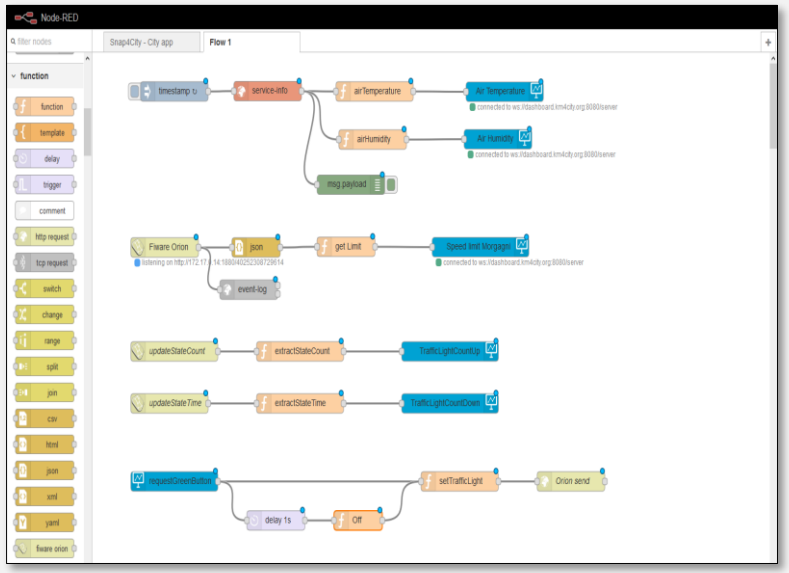






MyKPI variable

Synoptics



IOT Application

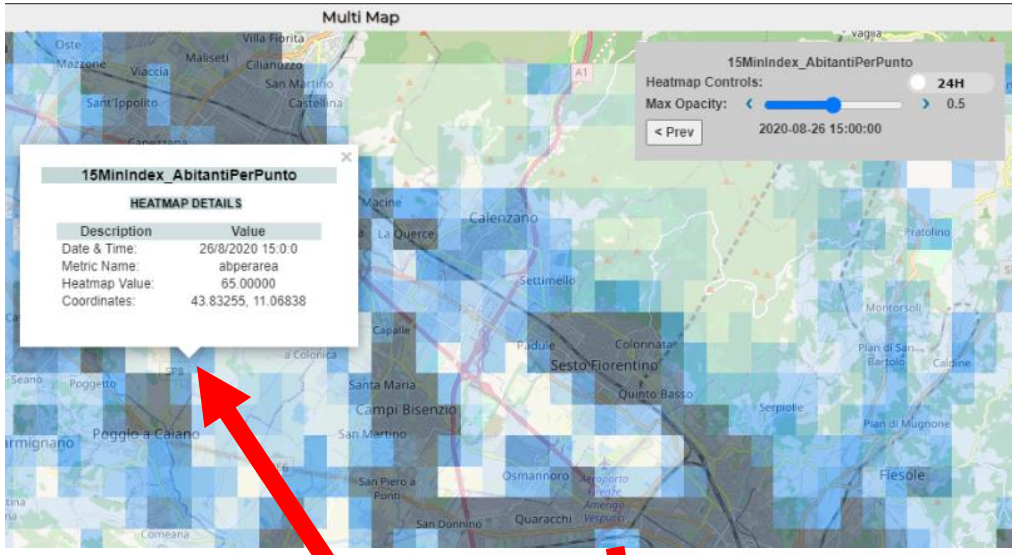
Geolocation of Mobile Device



Web Browser GPS data rendering the Snap4City Dashboard can be passed to IOT Applications and saved ☺

- Complete message
  - Returns a JSON containing all information about geolocation
- Latitude
  - Returns the latitude
- Longitude
  - Returns the longitude
- Accuracy
  - Returns the accuracy of latitude and longitude
- Altitude
  - Returns the altitude
- Altitude Accuracy
  - Returns the altitude accuracy
- Heading
  - Returns the heading
- Speed
  - Returns the speed

Multi Data Map GPS Location Picking

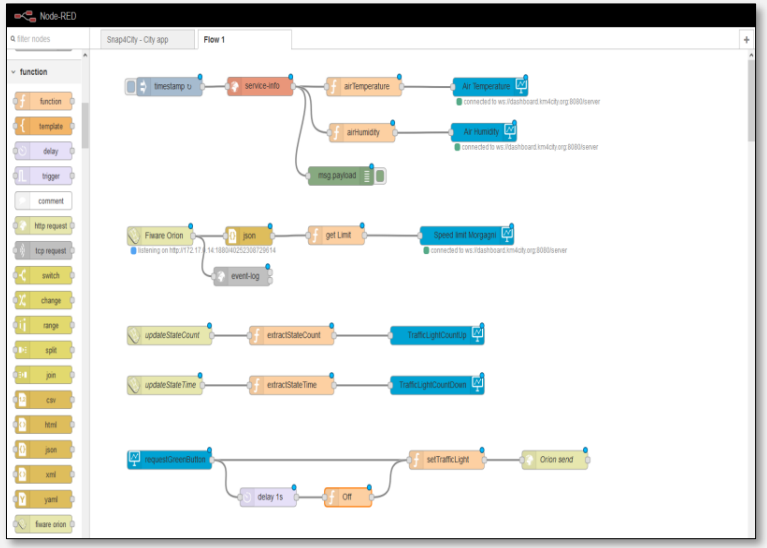


- 3) The click on the map can be used for passing GPS coordinates into IOT App to:
  - search for location
  - picking the value of one or more heatmaps
  - dynamically change data on widgets
  - Etc.

Nature

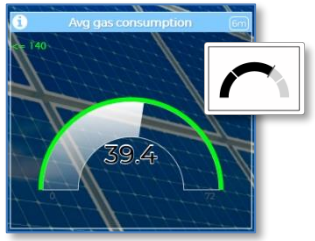
Dashboard-IOT App

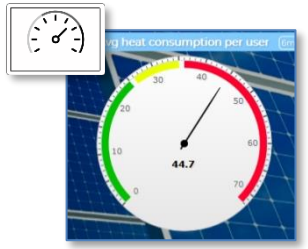
From IOT App to Dashboard



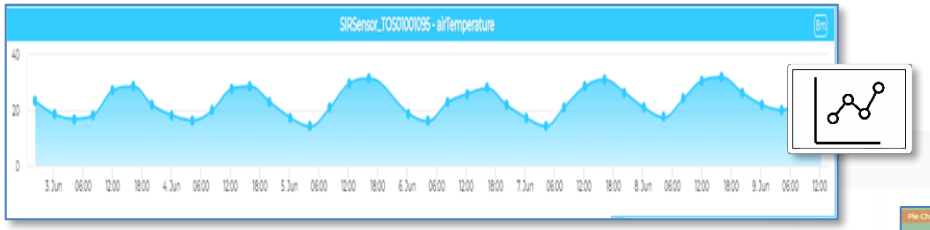
IOT Application

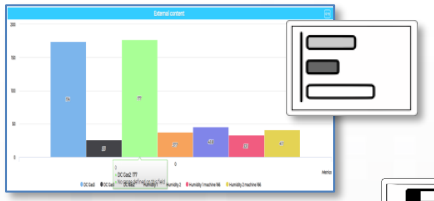
- gauge chart
- single content
- speedometer
- horizontal single bar
- vertical single bar
- web content
- time trend
- bar series
- radar series
- pie chart
- curved line series
- table content
- synoptic write

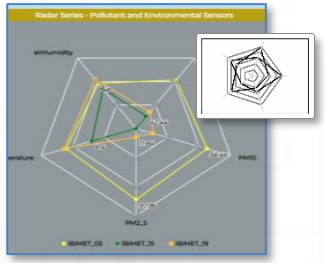


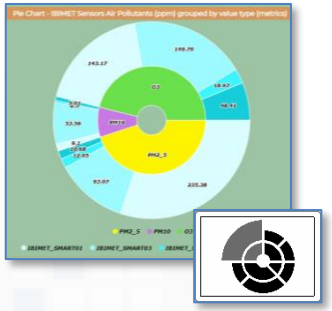


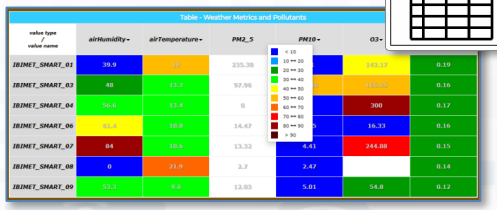


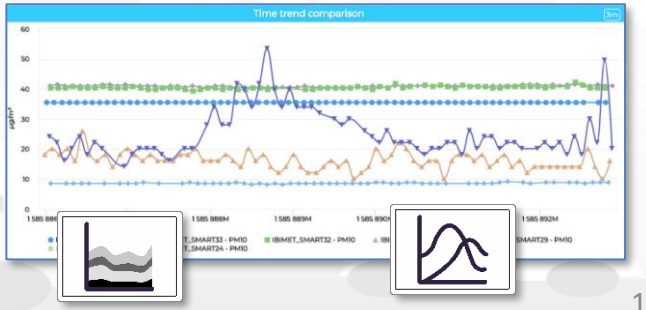


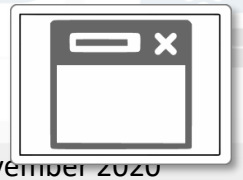












Single Content Widget (flexibility)

From Dashboard Editor and IOT Applications, accepts in input:

- Numbers
- String
- HTML code



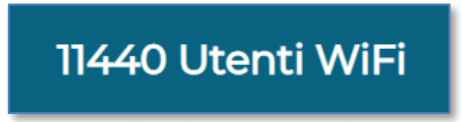




















https://www.snap4city.org/578

Dynamic Widgets data on Dashboard from IOT Applications



How the Dashboards exchange data

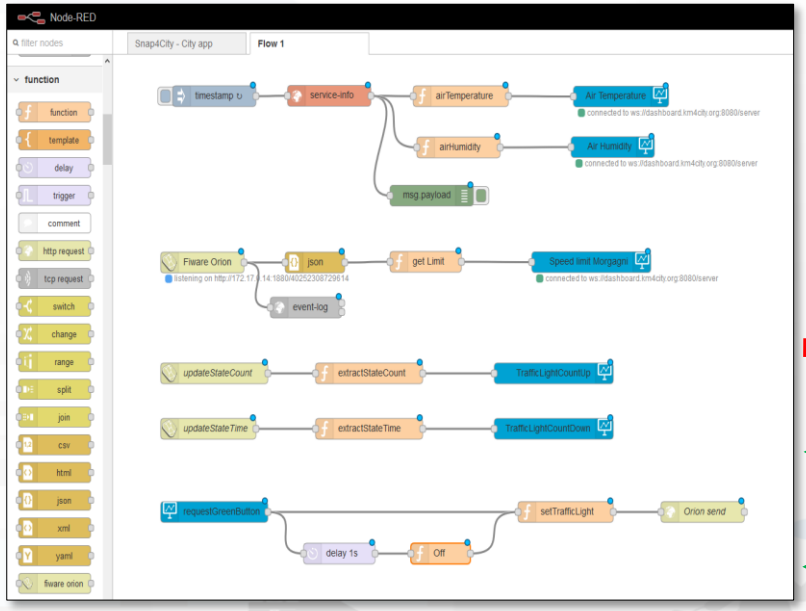
Snap4City BigData Storage and KB

Metric, KPI

MyKPI, MyPOI, …

API, External Services, MicroApp

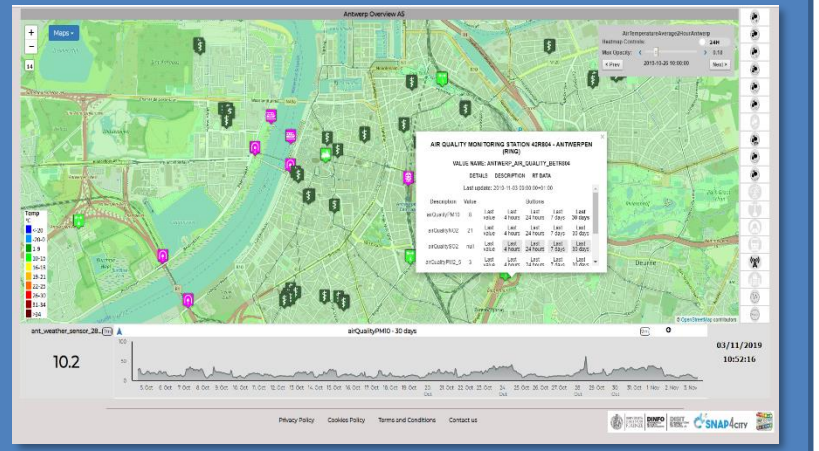
- Req. ServiceURI
- Req. KPI, Metric ID
- Req. MyKPI ID
- GIS, HTTPs URLs

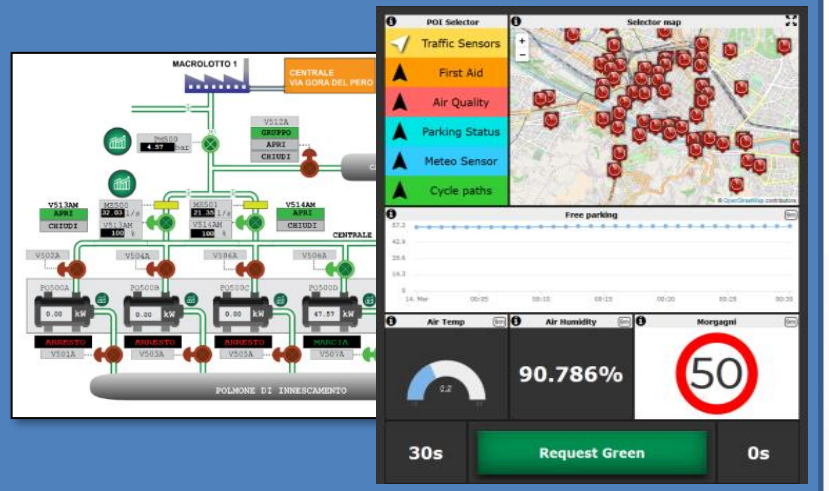


IOT Application

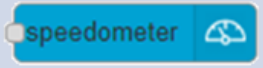
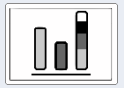
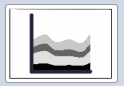
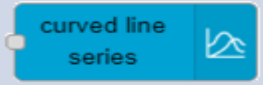
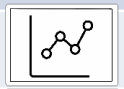
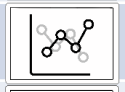
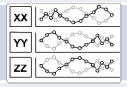
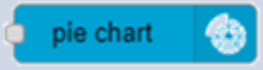
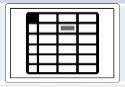
- ServiceURI (ID)
- MyKPI, Metric (ID)
- Dynamic Data, computed into IOT Application
- Rx. Dynamic Data
- Event Driven Synoptics
- Actions, Show

Dashboards





Dynamic (+)

Widgets ICONS	Widget Name, Description	IOT App	Dashboard-IOT App	KPI (metric)	MyPersonal Data	MyData	My KPI	Sensor
	Single Content 	X (cs)	X (DD)	X	X	X	X	X
	Speed Limit			X				X
	Speedometer 	X (cs)	X (DD)	X	X	X	X	X
	Gauge 	X (cs)	X (DD)	X	X	X	X	X
 	Single Bar, V/H  	X	X (DD)	X				
 	Single and Multiple Bars, stacked or not 	X (cs)	X (DD)	X	X	X	X	X
 	MultiSeries, shaded, staked and non staked 	X (cs)	X	X	X	X	X	X
	Time Trend (single) 	X	X (DD)	X	X	X	X	X
 	Time trend Compare			X	X			X
	SpiderNet, radar, Kiviat 	X (cs)	X (DD)	X	X	X	X	X
	Pie, Donut, 2 layers Donut 	X (cs)	X	X	X	X	X	X
	Table 	X (cs)	X	X	X	X	X	X

DD: Data Driven

Legenda

- **IOT APP column in previous table:**
  - **X**: means that from the IOT App you can send a new value or array to the widget directly, without the need to have is stored into Sensor or MYKPI variable, etc.
  - **CS, widget supports Change (data) Source**, in the sense that: from the IOT App is possible to send a command to the Widget to change the data source. E.g., selecting sources among: Sensors (serviceURI), MyKPI (ID), any value produced on the IOT App directly. (cs) recent additions
- **Dashboard IOT App column in previous table**:
  - **X:** there is a MicroService / node on IOT App to act on those widgets on dashboard. The data are visualized.
  - **DD, widget is Data Driven**, in the sense that new data in push can be sent and the widget is updated in real time on web page without web page realoading

TC4.9: New Support Widgets for Bars, Barseries, Trend, and Series, on Dashboards and IOT Applications (partially obsolete)

Dynamic Widget data

ServiceURI (ID)

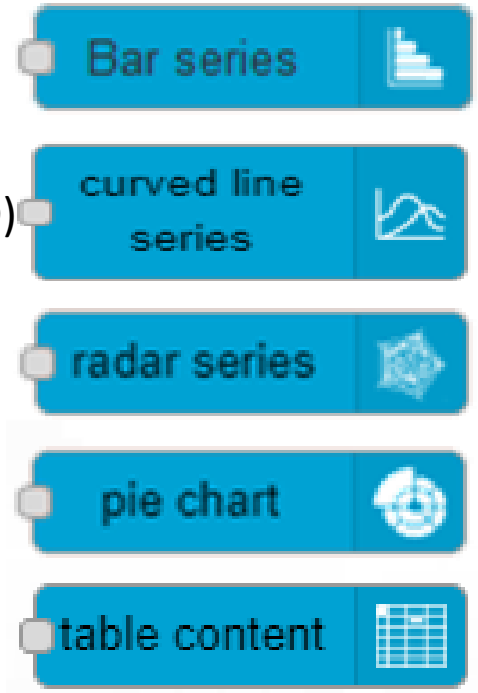
ServiceURI (ID)

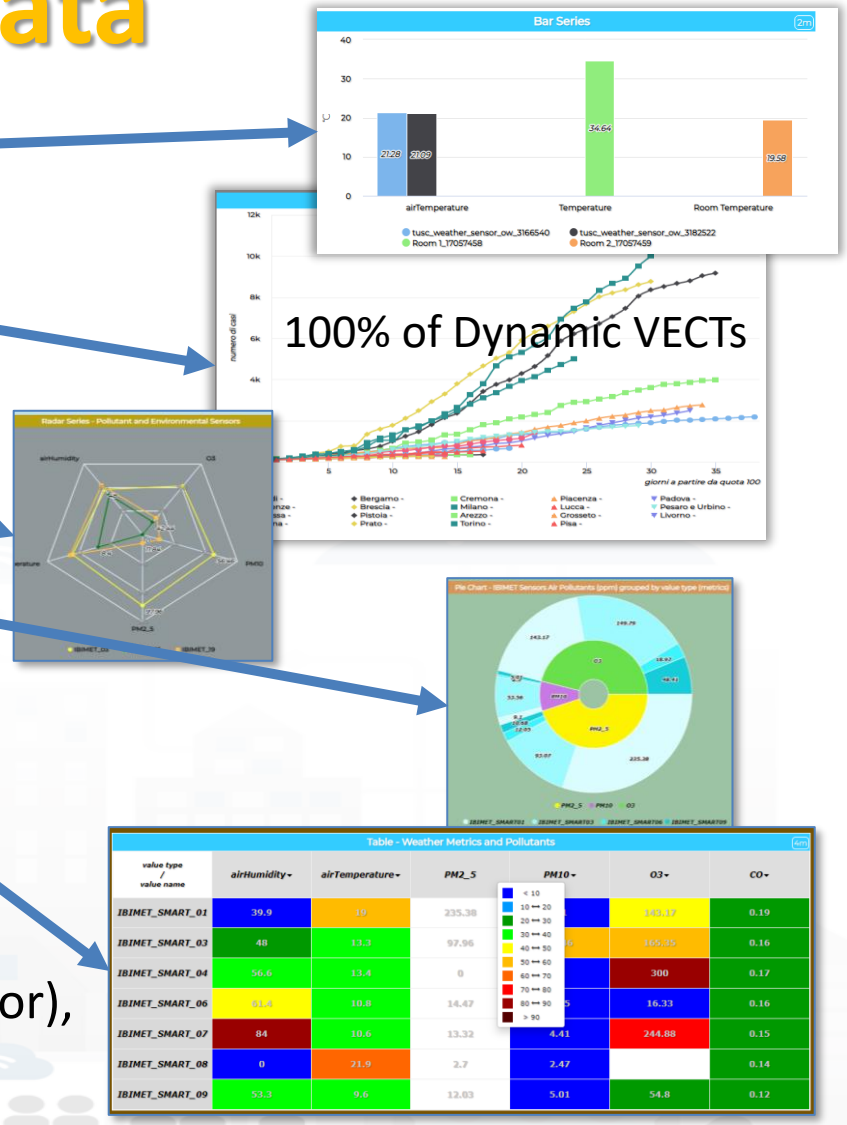
MyKPI (ID)

Dynamic

Dynamic


```
▾0: object
    metricId: "http://www.disit.org/km4city/resource/tusc_weather_sensor_ow_3166540"
    metricHighLevelType: "Sensor"
    metricName: "tusc_weather_sensor_ow_3166540"
    metricType: "airTemperature"
▾1: object
    metricId: "https://servicemap.disit.org/WebAppGrafo/api/v1/?
    serviceUri=http://www.disit.org/km4city/resource/tusc_weather_sensor_ow_3182522&format=json"
    metricHighLevelType: "Sensor"
    metricName: "tusc_weather_sensor_ow_3182522"
    metricType: "airTemperature"
▾2: object
    metricId: "17056320"
    metricHighLevelType: "MyKPI"
    metricName: "SiiMTuscanyTrackerLocation"
    metricType: "Velocity"
▾3: object
    metricId: ""
    metricHighLevelType: "Dynamic"
    metricName: "BatteryTemperatureGalaxyNote"
    metricType: "Gradi Centigradi"
    metricValueUnit: "°C"
    measuredTime: "2019-11-21T14:51:00Z"
    value: 6.688898111364505
▾4: object
    metricId: ""
    metricHighLevelType: "Dynamic"
    metricName: "BatteryTemperaturemia"
    metricType: "Gradi Centigradi"
    metricValueUnit: "°C"
    measuredTime: "2019-11-21T14:51:00Z"
    value: 62.8502788741156
```



- **ServiceURI** (ID)
- **MyKPI (ID)**, Metric (ID)
- **Dynamic** Data in JSON (single or Vector), computed into IOT Application

[TC4.9: New Support Widgets for Bars, Barseries, Trend, and Series, on Dashboards and IOT Applications]

Example of: Dynamic Widgets data on Dashboard from IOT Applications



How to send the Dynamic Data to Widgets

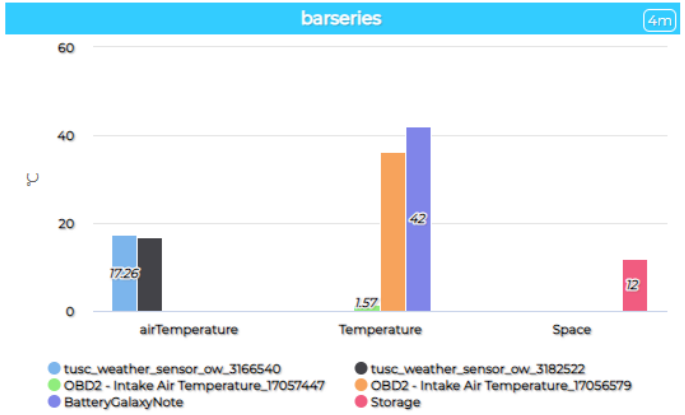
- TC4.9: New Support Widgets for Bars, Barseries, Trend, and Series, on Dashboards and IOT Applications

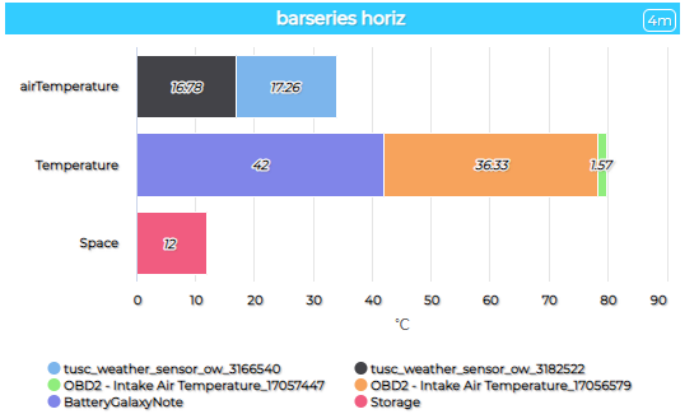
- Dynamic Data are used to control the Widget from the IOT App. To dynamically change:
  - **ServiceURI** (as metricID) to change the data source of a Dashboard Widget
  - **MyKPI** (as metricID) to change the data source of a Dashboard Widget
  - **Dynamic**, data computed somehow into IOT App, and sent to some Dashboard Widget without to save them on some Storage

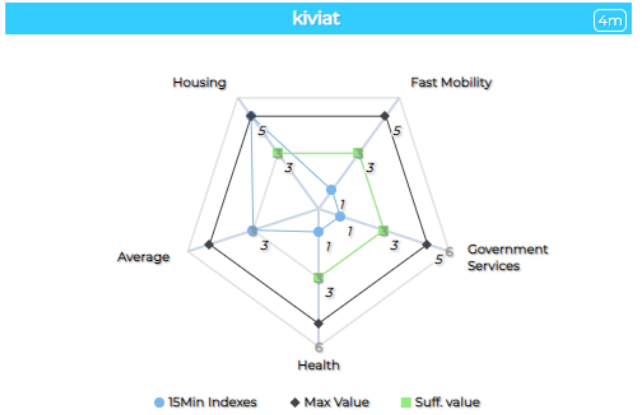
Dashboard with Dynamic Data Managed by IOT App

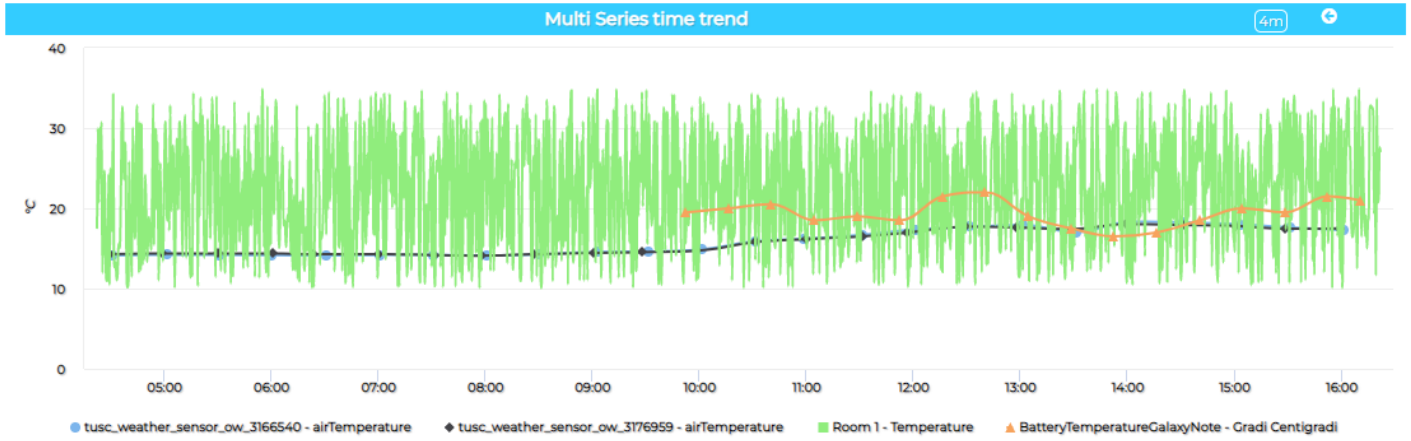
Dynamic Widget data

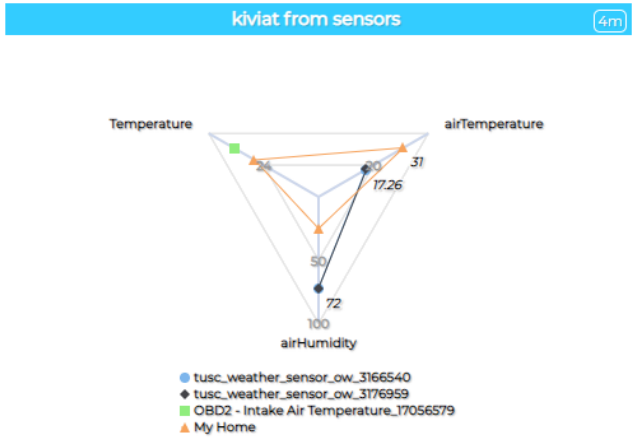
Mon 2 Nov 16:22:49





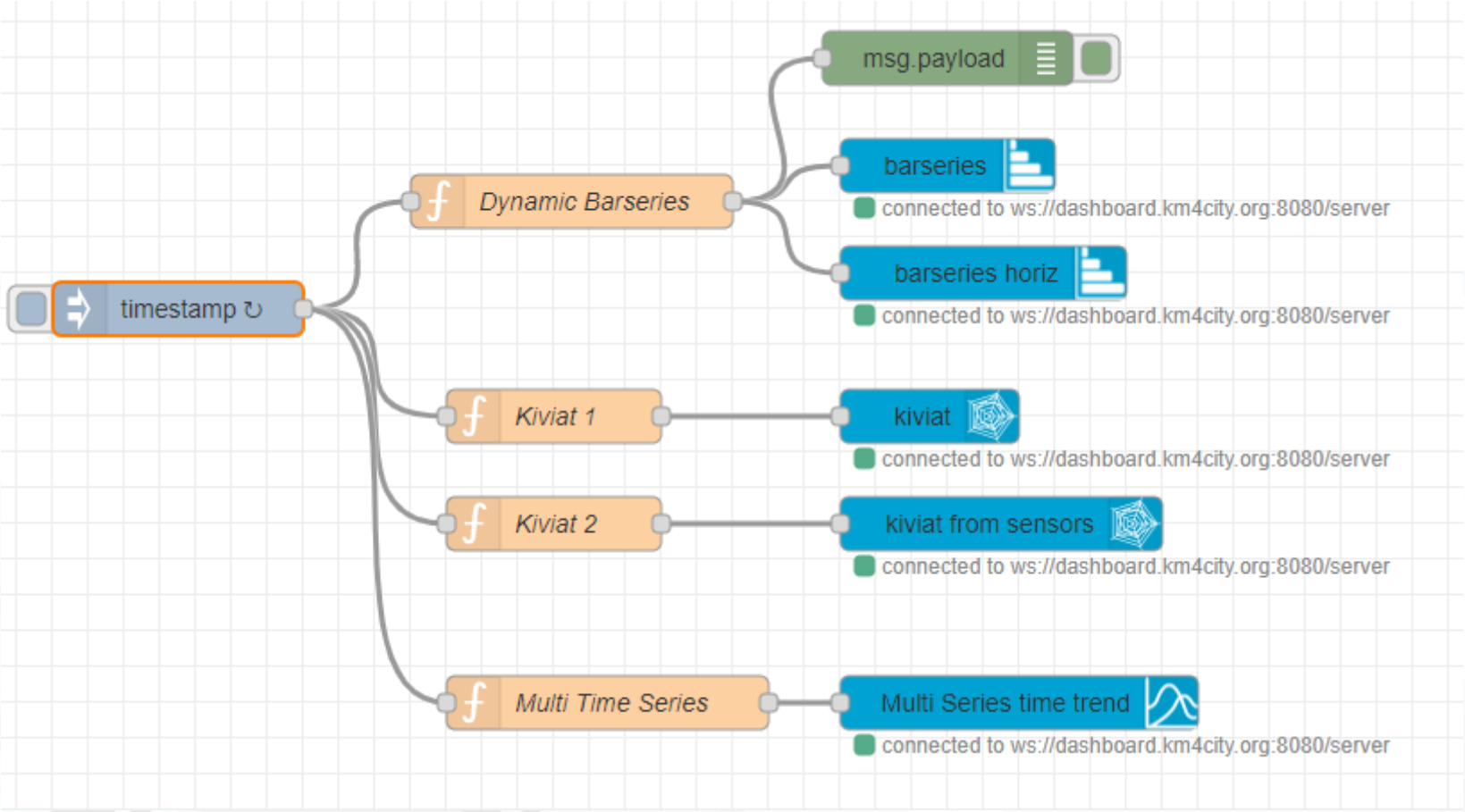






https://main.snap4city.org/view/index.php?iddasboard=Mjk5NQ==

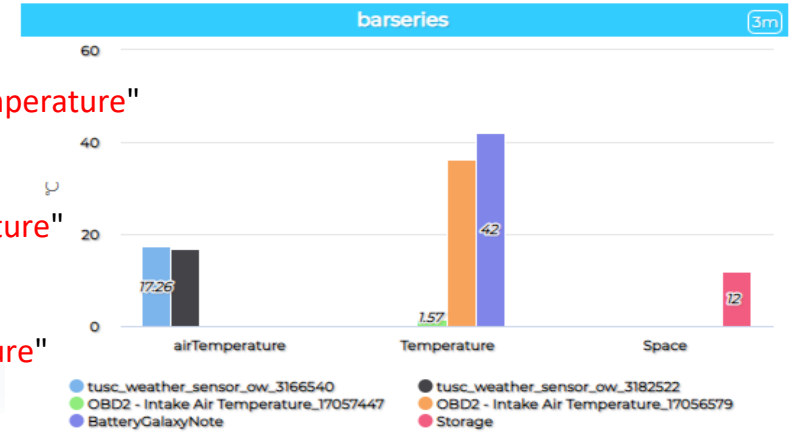
The IOT App controlling the Dashboard data

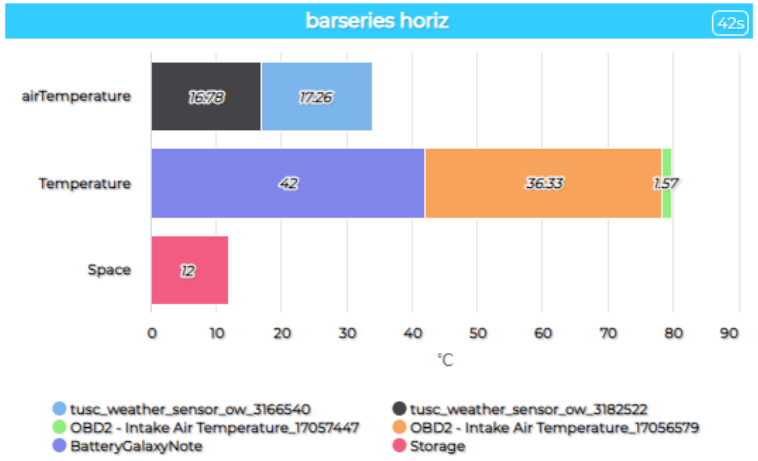


Dynamic BarSeries

```
msg.payload =[
  {    "metricId":"http://www.disit.org/km4city/resource/tusc_weather_sensor_ow_3166540",
   "metricHighLevelType":"Sensor",
   "metricName":"tusc_weather_sensor_ow_3166540",     "metricType":"airTemperature"
  },
  {    "metricId":"http://www.disit.org/km4city/resource/tusc_weather_sensor_ow_3182522",
   "metricHighLevelType":"Sensor",     "metricName":"tusc_weather_sensor_ow_3182522",     "metricType":"airTemperature"
  },
  {    "metricId":"17057447",
   "metricHighLevelType":"MyKPI",     "metricName":"OBD2 - Intake Air Temperature",          "metricType":"Temperature"
  },
  {    "metricId":"17056579",
   "metricHighLevelType":"MyKPI",    "metricName":"OBD2 - Intake Air Temperature",          "metricType":"Temperature"
  },
  {    "metricId":"",
   "metricHighLevelType":"Dynamic",  "metricName":"BatteryGalaxyNote",          "metricType":"Temperature",
   "metricValueUnit":"°C",
   "measuredTime":"2019-11-21T14:51:00Z",
   "value":42
  },
  {    "metricId":"",
   "metricHighLevelType":"Dynamic",     "metricName":"Storage",    "metricType":"Space",
   "metricValueUnit":"Gb",
   "measuredTime":"2019-11-21T14:51:00Z",
   "value":12
  }
]
return msg;
```







Sensors, MyKPI and dynamic in this case

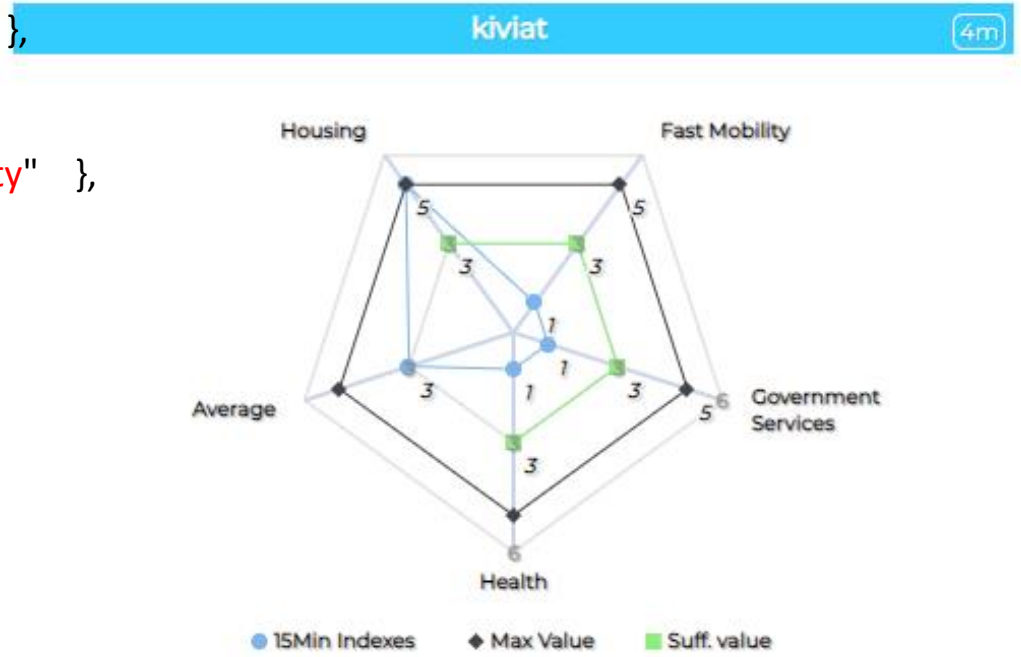
May be aggregated by Metric Type or by ValueName, staked, oriented .. see More Options

Kiviat 1

```
msg.payload = [
  { "mapName":"15MinIndex_FastMobilityIndex",    "metricName":"15Min Indexes",    "value":1,
    "metricHighLevelType":"Dynamic",    "metricValueUnit":"",    "metricType":"Fast Mobility"   },
  { "mapName":"15MinIndex_GovernmentServicesIndex",   "metricName":"15Min Indexes",    "value":1,
    "metricHighLevelType":"Dynamic",  "metricValueUnit":"",   "metricType":"Government Services"   },
  { "mapName":"15MinIndex_HealthIndex",    "metricName":"15Min Indexes",    "value":1,
    "metricHighLevelType":"Dynamic",    "metricValueUnit":"",    "metricType":"Health"  },
  { "mapName":"15MinIndex_AverageIndex",   "metricName":"15Min Indexes",    "value":3,
    "metricHighLevelType":"Dynamic",   "metricValueUnit":"",    "metricType":"Average"  },
  { "mapName":"15MinIndex_HousingIndex",   "metricName":"15Min Indexes",    "value":5,
    "metricHighLevelType":"Dynamic",   "metricValueUnit":"",    "metricType":"Housing"  },

  { "mapName":"15Min Indexes",   "metricName":"Max Value",    "value":5,
   "metricHighLevelType":"Dynamic",  "metricValueUnit":"",   "metricType":"Fast Mobility"   },
  {   "mapName":"15Min Indexes",     "metricName":"Max Value", ……..
……..

  { "mapName":"15Min Indexes",  "metricName":"Suff. value",  "value":3,
   "metricHighLevelType":"Dynamic",  "metricValueUnit":"",  "metricType":"Health","   },
  ……..
]
return msg;
```



Aggregated by MetricType



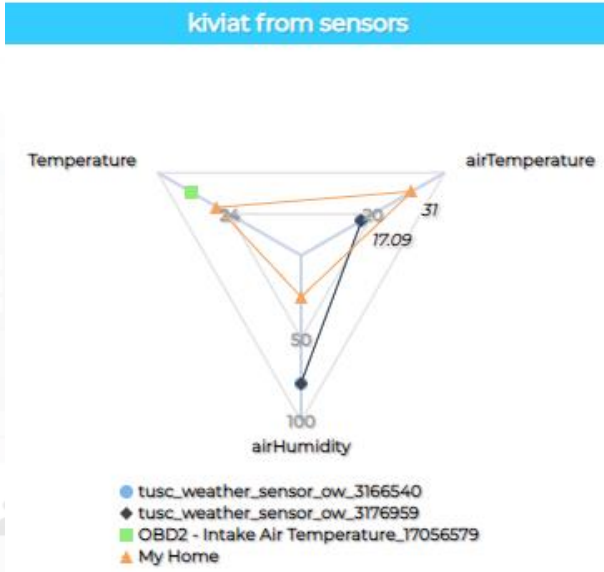
All dynamic in this case

Kiviat 2



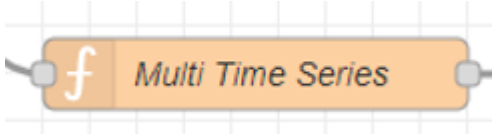

```
msg.payload = [
{      "metricId": "http://www.disit.org/km4city/resource/tusc_weather_sensor_ow_3166540",
"metricHighLevelType": "Sensor",
"metricName": "tusc_weather_sensor_ow_3166540",
 "metricType": "airTemperature"
},
{      "metricId": "http://www.disit.org/km4city/resource/tusc_weather_sensor_ow_3166540",
"metricHighLevelType": "Sensor",      "metricName": "tusc_weather_sensor_ow_3166540",      "metricType":
"airHumidity"    },
 {      "metricId": "http://www.disit.org/km4city/resource/tusc_weather_sensor_ow_3176959",
"metricHighLevelType": "Sensor",      "metricName": "tusc_weather_sensor_ow_3176959",      "metricType":
"airHumidity"    },
{      "metricId": "http://www.disit.org/km4city/resource/tusc_weather_sensor_ow_3176959",
"metricHighLevelType": "Sensor",      "metricName": "tusc_weather_sensor_ow_3176959",      "metricType":
"airTemperature"    },
{      "metricId": "17056579",      "metricHighLevelType": "MyKPI",      "metricName": "OBD2 - Intake Air
Temperature",      "metricType": "Temperature"    },
{      "metricId": "",      "metricHighLevelType": "Dynamic",      "metricName": "My Home",      "metricType":
"airTemperature",      "metricValueUnit": "°C",      "measuredTime": "2019-11-21T14:51:00Z",      "value": 31
},
{      "metricId": "",      "metricHighLevelType": "Dynamic",      "metricName": "My Home",      "metricType":
"Temperature",      "metricValueUnit": "°C",      "measuredTime": "2019-11-21T14:51:00Z",      "value": 28    },
{      "metricId": "",      "metricHighLevelType": "Dynamic",      "metricName": "My Home",      "metricType":
"airHumidity",      "metricValueUnit": "%",      "measuredTime": "2019-11-21T14:51:00Z",      "value": 25    }]

return msg;
```


Aggregated by MetricType

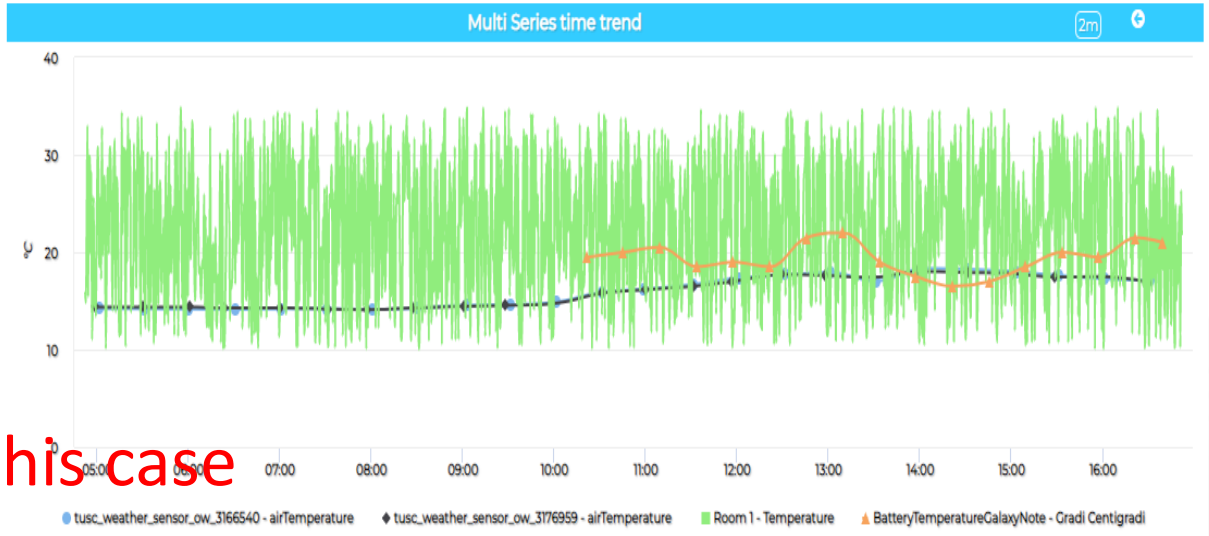


Sensors, MyKPI and dynamic in this case

Multi TimeSeries




```
var now = new Date();
var base =60*60*100;
msg.payload =[
  { "serviceUri":"http://www.disit.org/km4city/resource/tusc_weather_sensor_ow_3166540",
    "metricHighLevelType":"Sensor",     "metricName":"tusc_weather_sensor_ow_3166540",
    "smField":"airTemperature" // as MetricType
  },
  {  "serviceUri":"http://www.disit.org/km4city/resource/tusc_weather_sensor_ow_3176959",
    "metricHighLevelType":"Sensor",     "metricName":"tusc_weather_sensor_ow_3176959",
    "smField":"airTemperature"
  },
  {  "serviceUri":"17057458",     "metricHighLevelType":"MyKPI",     "metricName":"Room 1",     "metricType":"Temperature"  },
  {    "metricId":"",     "metricHighLevelType":"Dynamic",     "metricName":"BatteryTemperatureGalaxyNote",     "smField":"Gradi Centigradi",
    "metricValueUnit":"°C",
    "values":[
        [now-64*base, 19.5],  [now-60*base, 20.0],   [now-56*base, 20.5],
        [now-52*base, 18.5], [now-48*base, 19], [now-44*base, 18.5],
        [now-40*base, 21.5],  [now-36*base, 22.0],   [now-32*base, 19],
        [now-28*base, 17.5],  [now-24*base, 16.5],  [now-20*base, 17.0],
        [now-16*base, 18.5],  [now-12*base, 20.0],   [now-8*base, 19.5],
       [now-4*base, 21.5],   [now-1*base, 21]
         ]
  }
]
return msg;
```

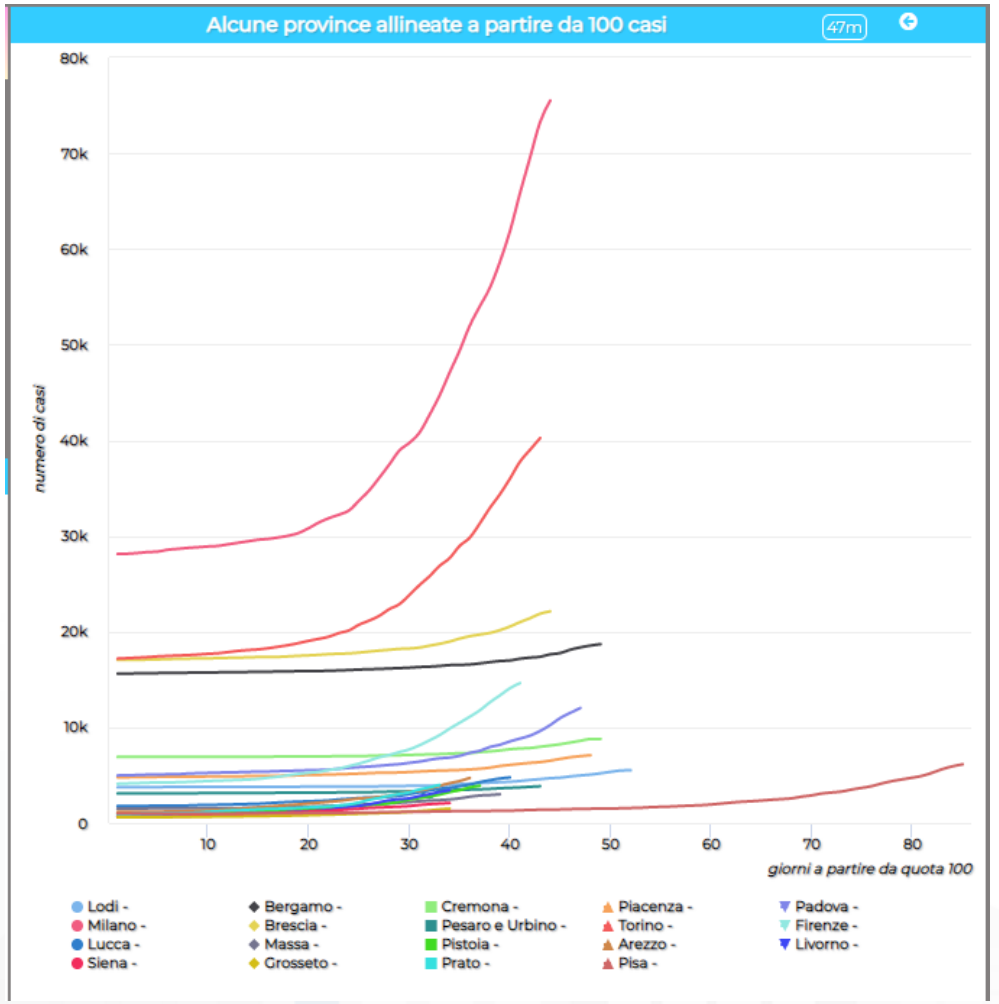


Sensors, MyKPI and dynamic in this case

Multi Series without Time

- multi series with ordinal data

```
{    "metricId":"",     "metricHighLevelType":"Dynamic",
"metricName":"BatteryTemperatureGalaxyNote",
"smField":"Gradi Centigradi",
     "metricValueUnit":"°C",
     "values":[
          [1, 19.5],   [2, 20.0],    [3, 20.5],
          [4, 8.5], [5, 19], [6, 18.5],
.............
          [50,5],    [51, 21]
           ]
   }
```



- You can set Staked via MoreOption

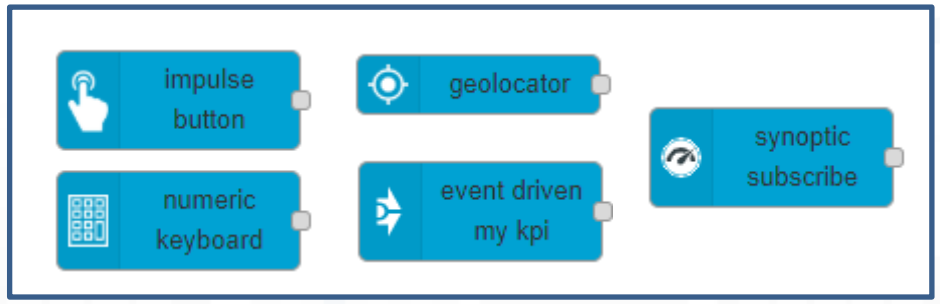
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjU3NQ==

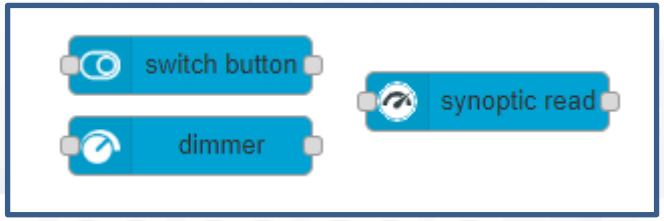
IOT Application stressing Virtual Sensors Actuators concepts

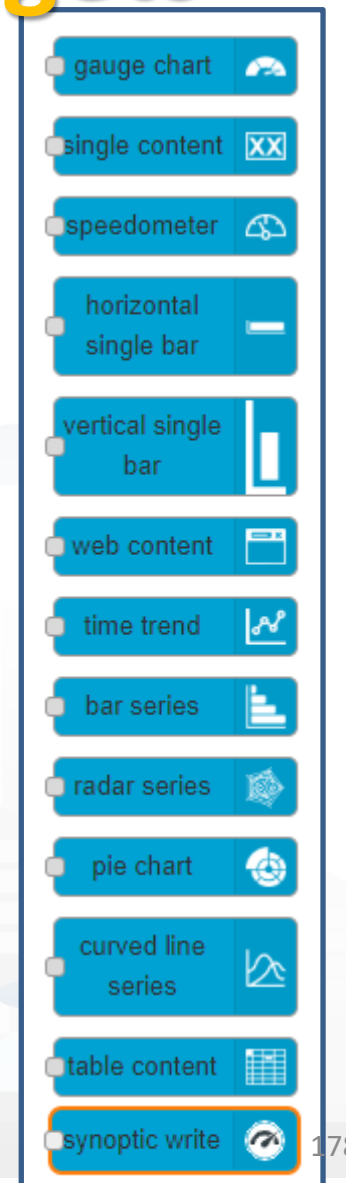


Advanced Feature of Snap4City Dash Widgets

- Dashboard widgets can be classified in:
  - **Virtual Sensors**
    - Those that produce the data
      From IOT App on Dashboard
  - **Virtual Actuators**
    - Those that produce the data
      From Dashboard to IOT App
  - **Virtual Sensors Actuators**
    - Those that produce/receive the data
      From/to Dashboard to/from IOT App

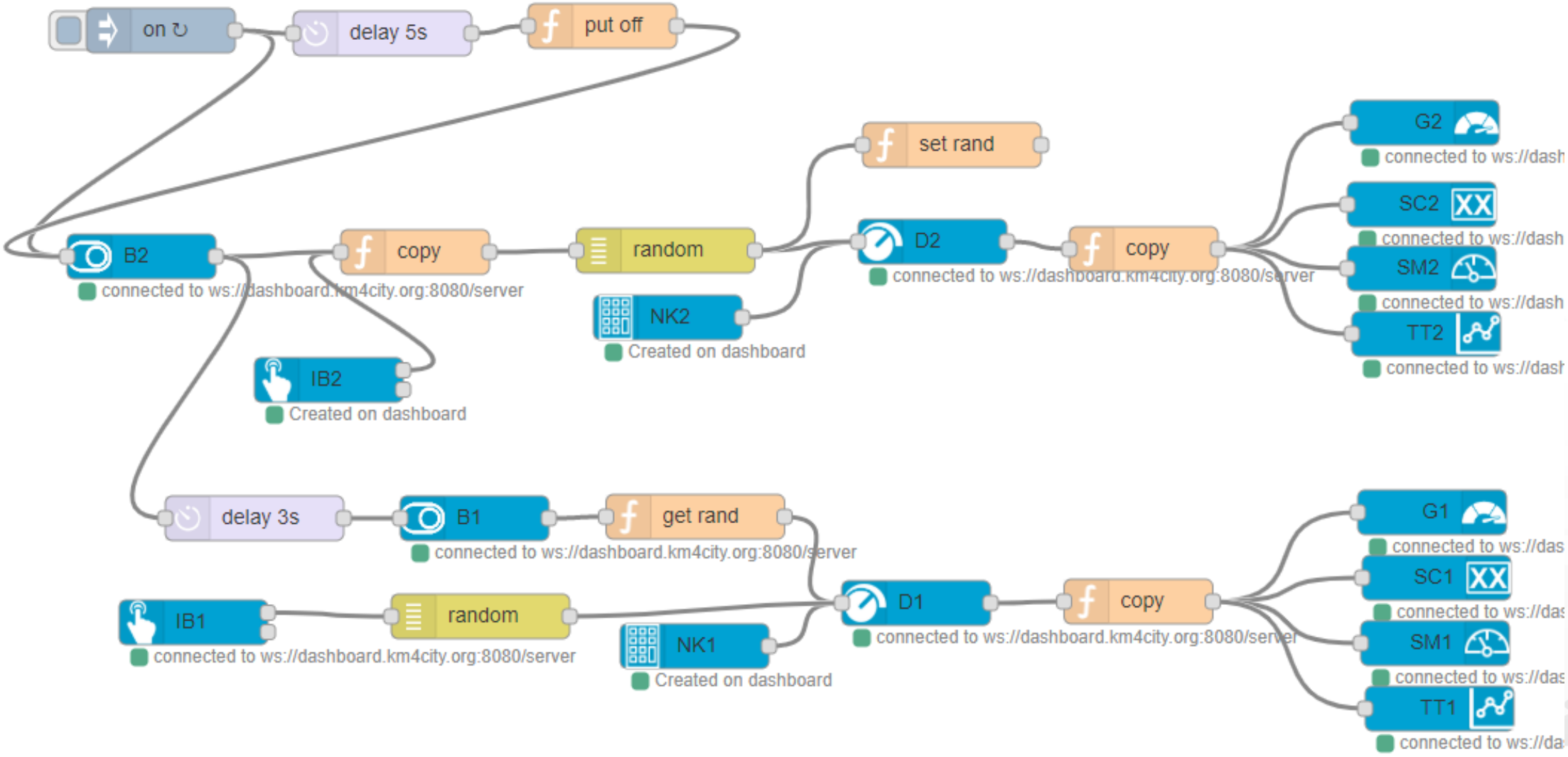




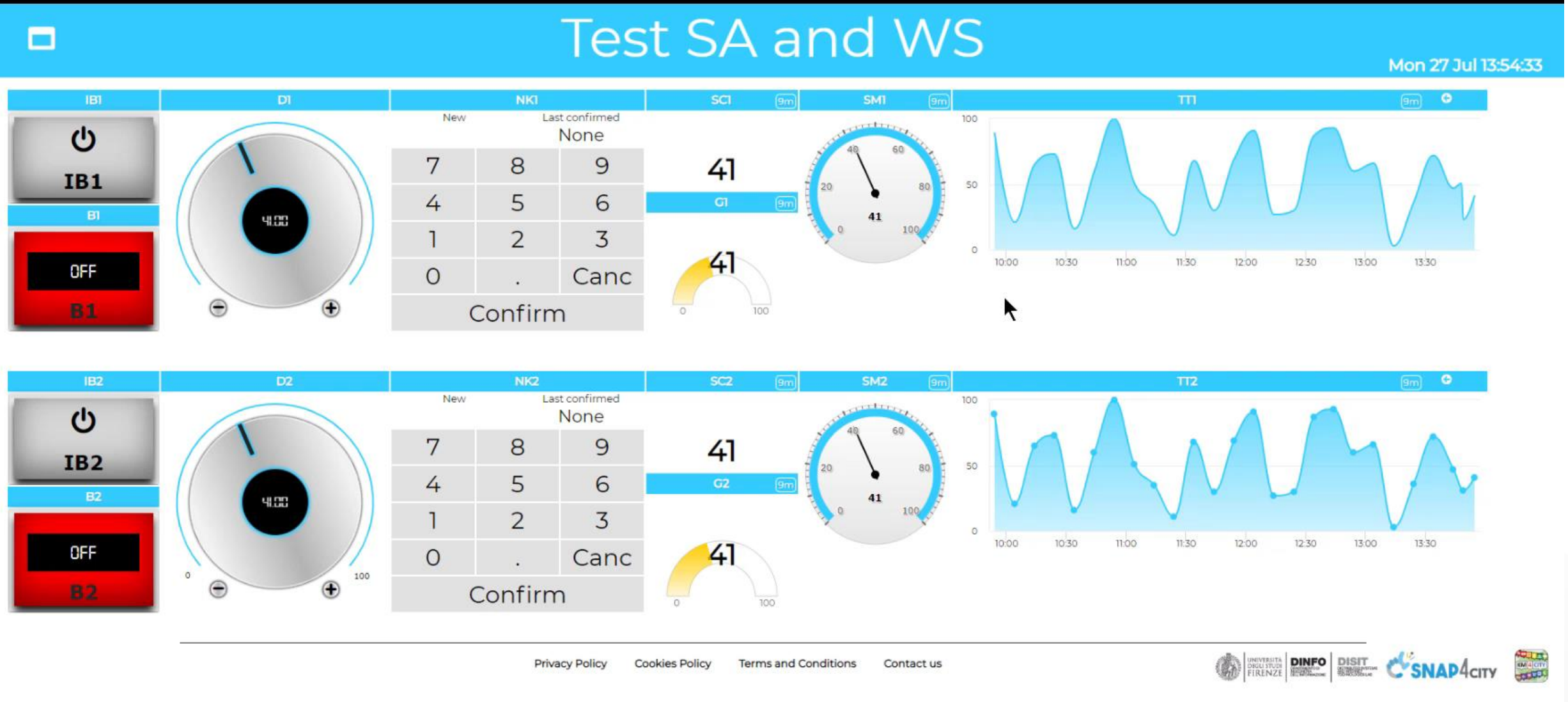


[TC9.18: Advanced Features of Sensors Actuators of Dashboards vs IOT App]

Sensors Actuators Allow to change the set up

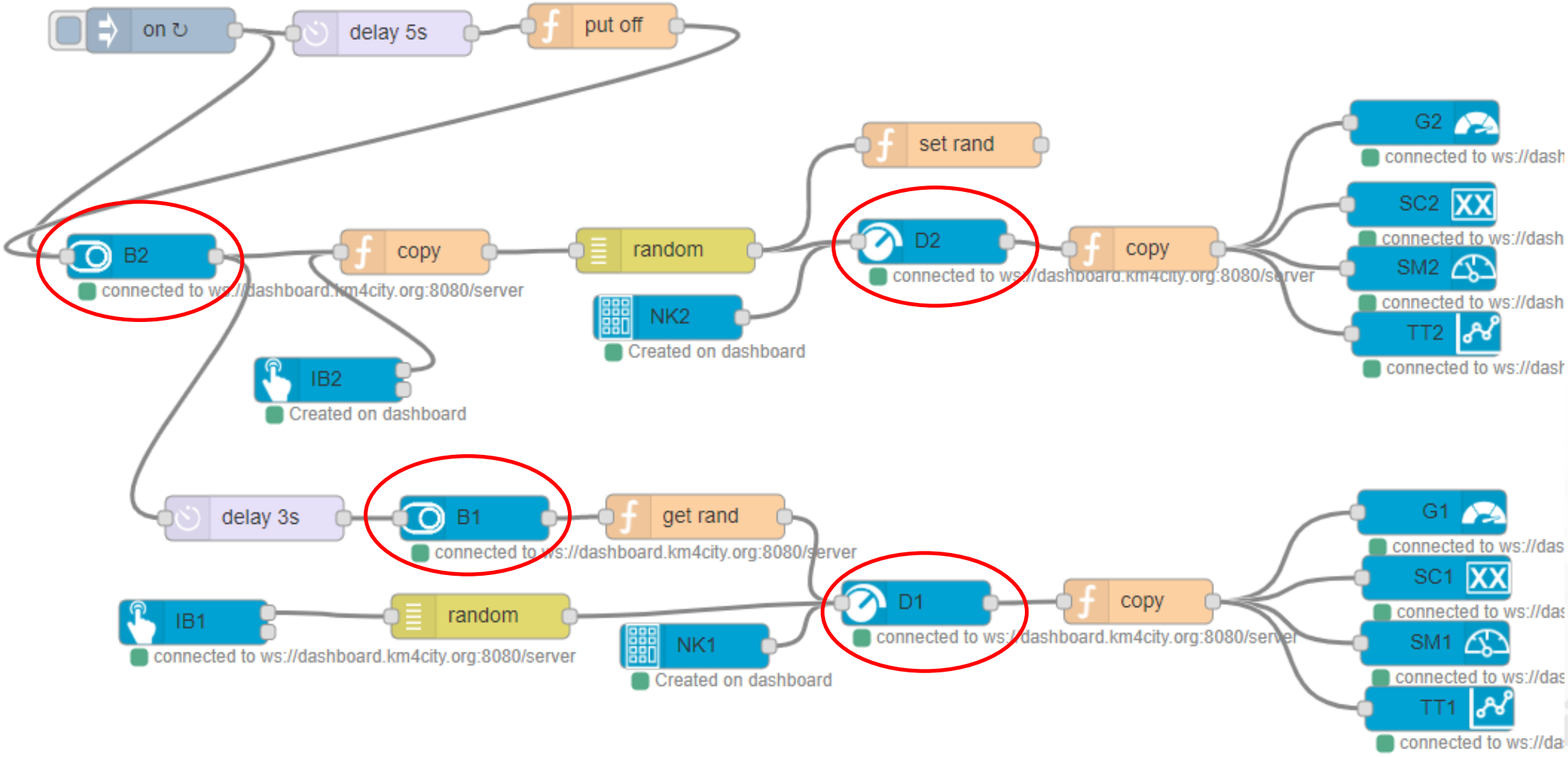


The SATEST Dash



https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=Mjc2NA==

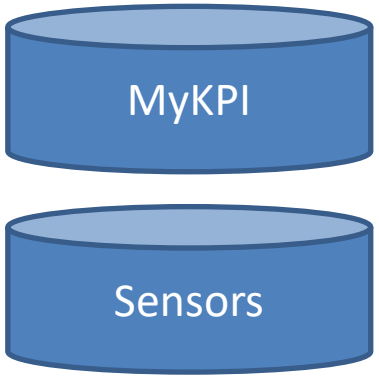
Sensors Actuators Allow to change the set up



IOT Application Integration with Synoptics



From-To Custom Widgets / Synoptics to Storage in WS



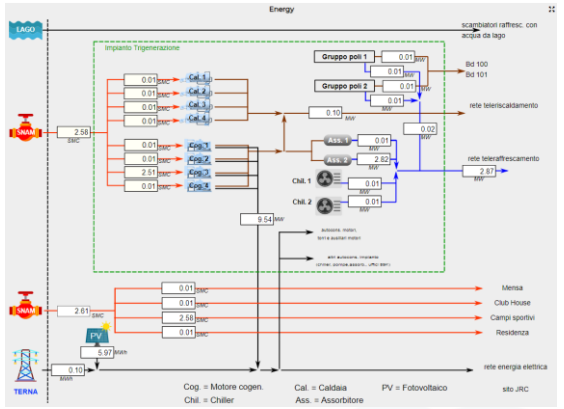
MyKPI

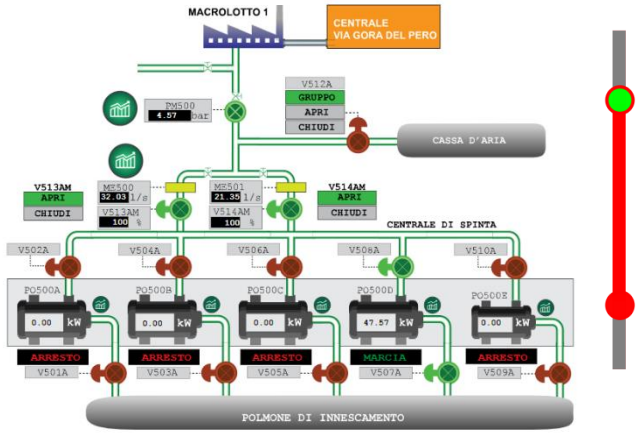
Sensor

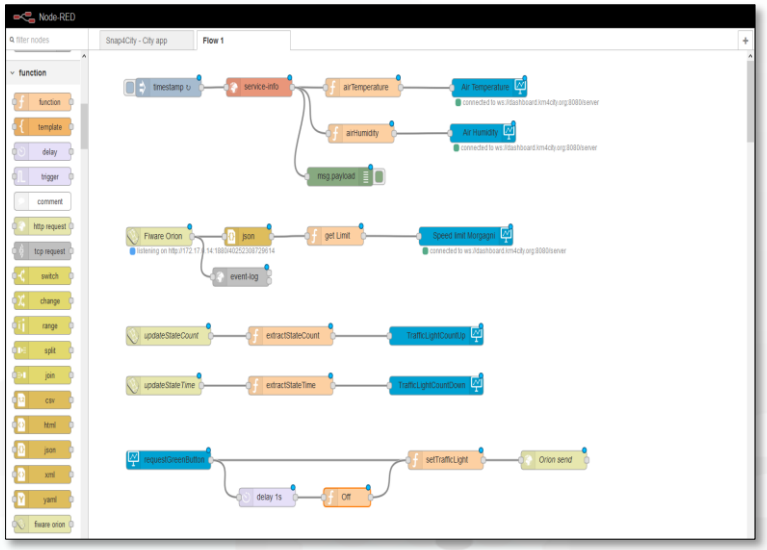
[ ]

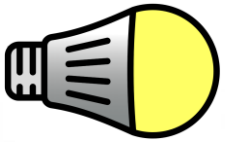
New Shared Variables

Constant Values

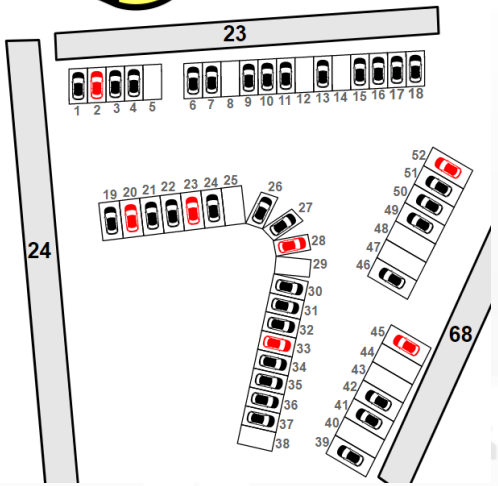


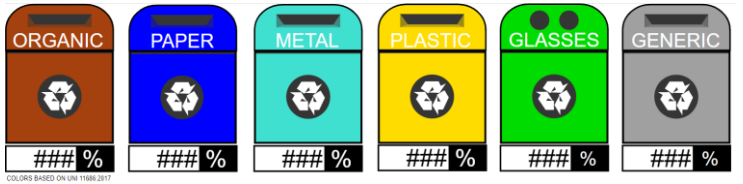


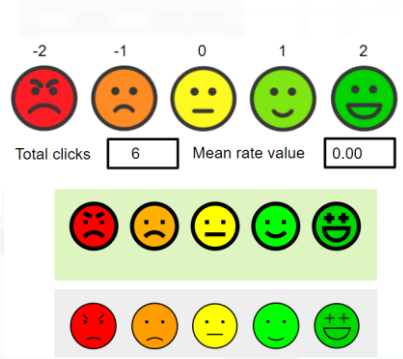


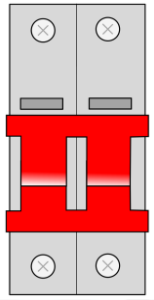


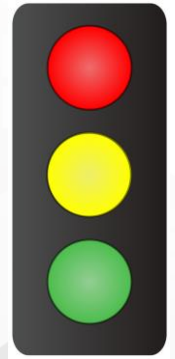








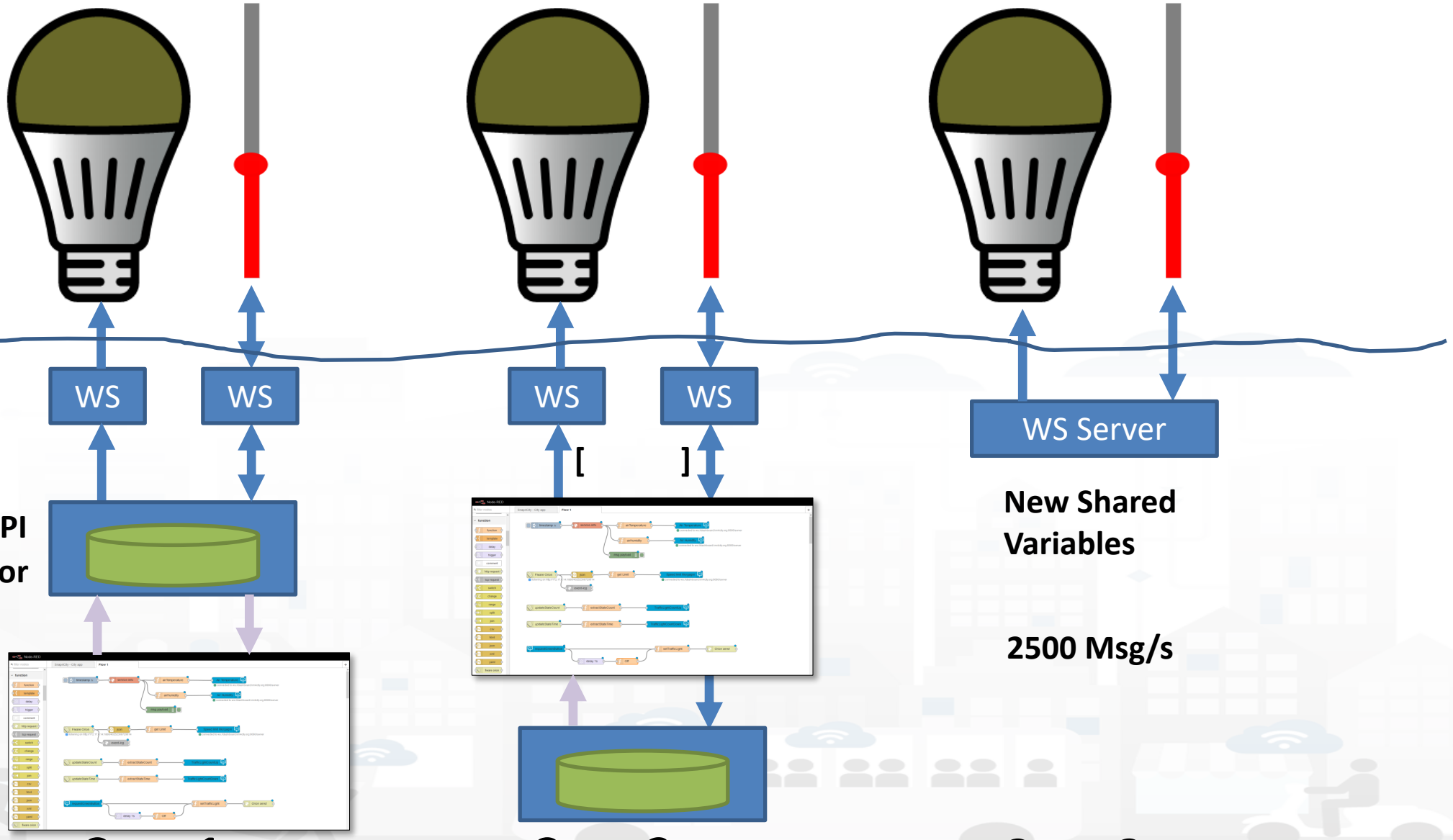




Dashboard on Browser

Internet

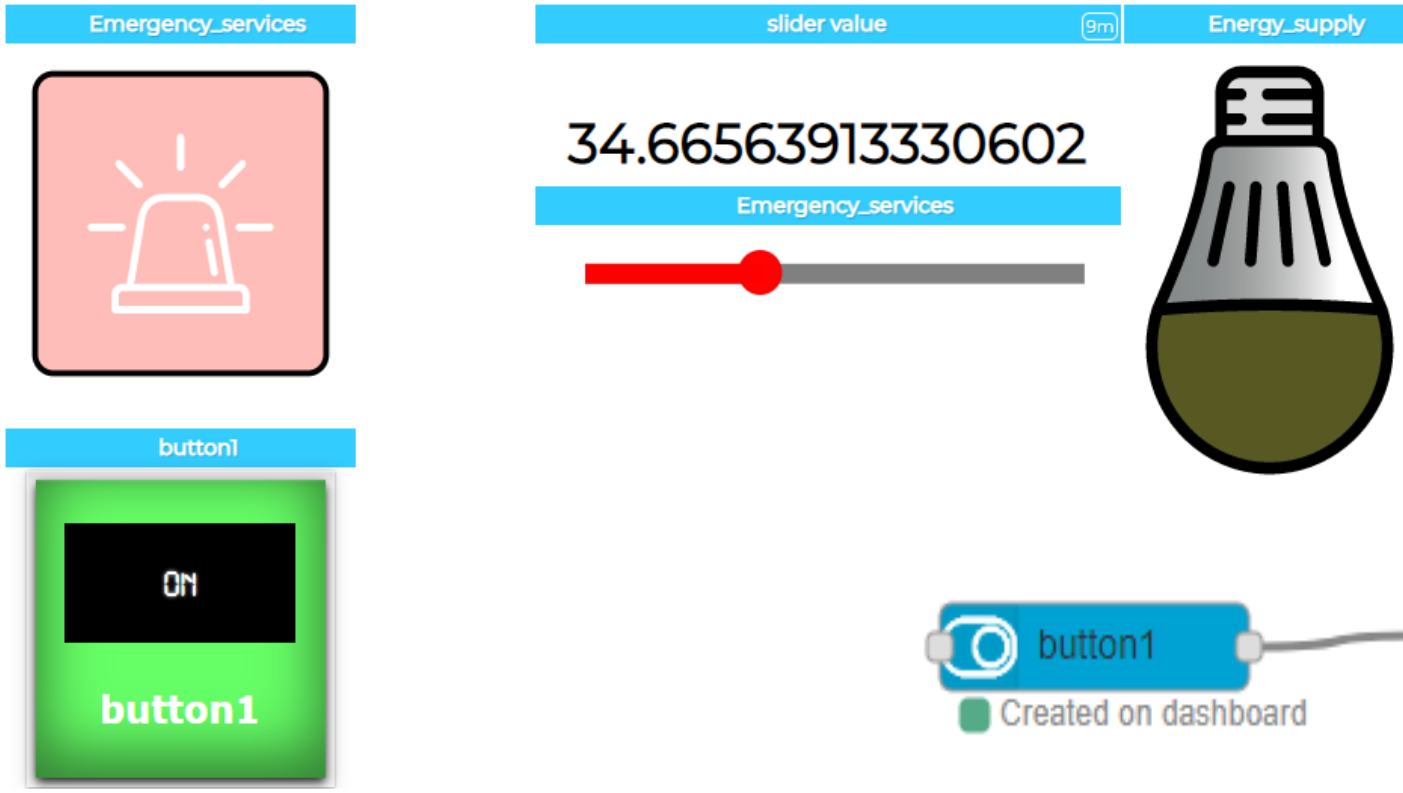
Storage and IOT App on cloud or on Premise



Case 1 Case 2 Case 3

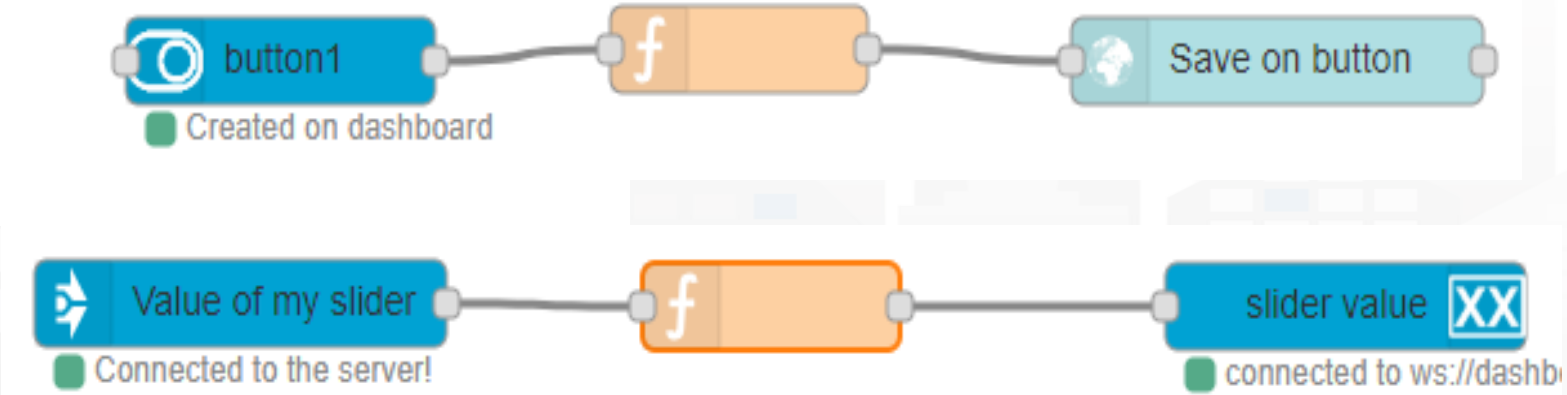
Case 1

Case 1 SVG ws3



https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=Mjc4NA==

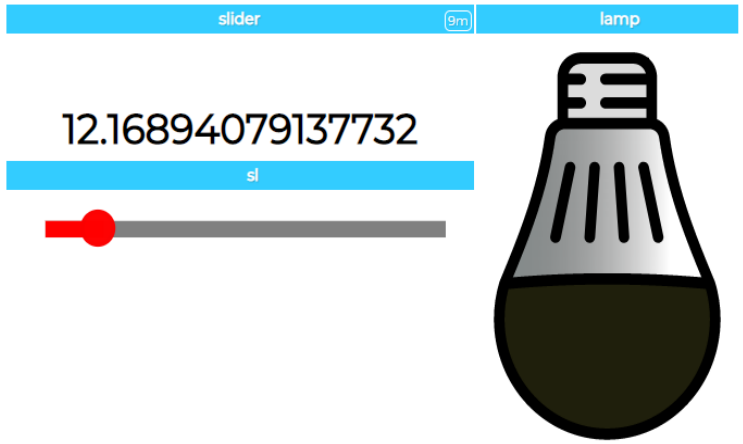
10 WS messages per second



Case 2: Event Driven 100%

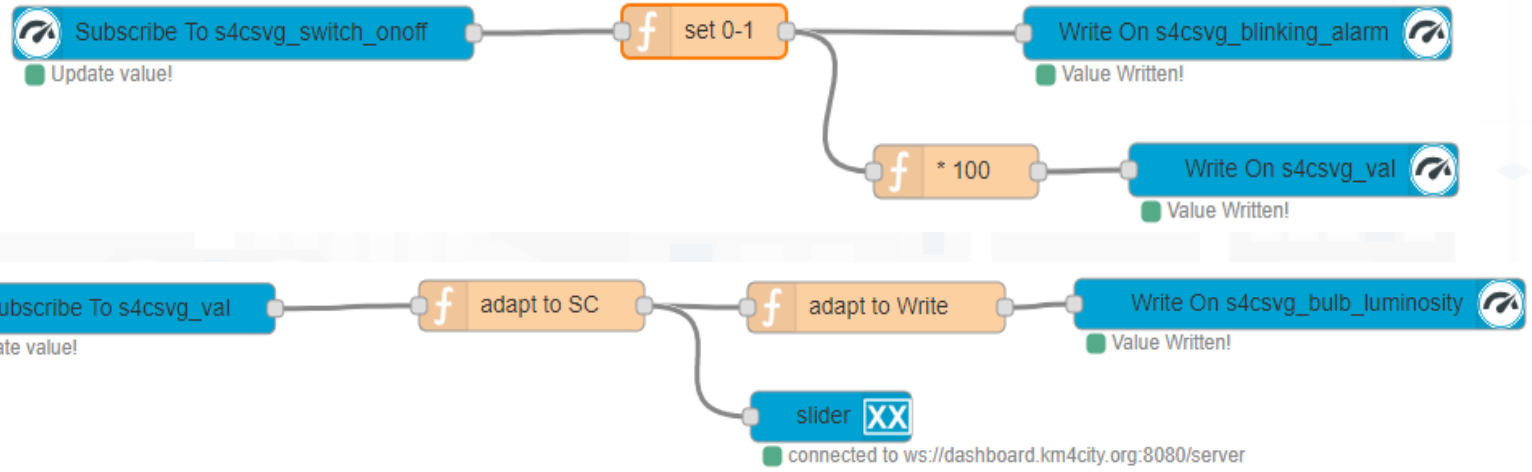
case 2 SVG WS3





https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=Mjc4NQ==

40 messages per second



Read more on

- TC9.19: Custom Widgets / Synoptics controlled by IOT Applications
- Custom Synoptics and Widgets for Dashboards
- Scenario: 5G Enabled Water Cleaning Control
- Snap4Industry: Snap4City for Industry 4.0
- TC1.22: Create and configure a Snap4City SVG Custom Widget for real-time interaction

DEMO Section 2

Demo IOT Application exploiting Snap4City Dashboard



Example of complex IOT Application

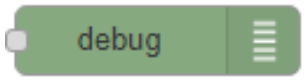
In this demo let's create an IoT Application that:

- send random values on Snap4city's Dashboard
- create complex widget based on MyKPI e SURI

Nodes for flow



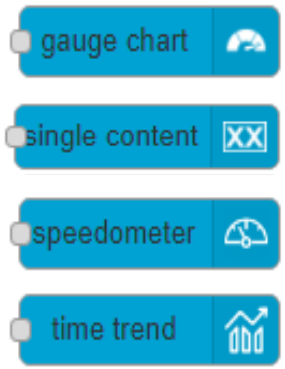
Generates an input for the other nodes. It can be repeated at predefined intervals, entered manually and of various types (timestamp, string, number, boolean, JSONetc)



Each message that enters the debug node is shown in the "debug" tab on the right of nodered (you can choose which part of the message to show)



Generates a random number. You can configure the number generation interval and the type (integer or float).



Display values in different modes on a dashboard. The node called single content accepts strings, numbers and html. The others only accept numbers.

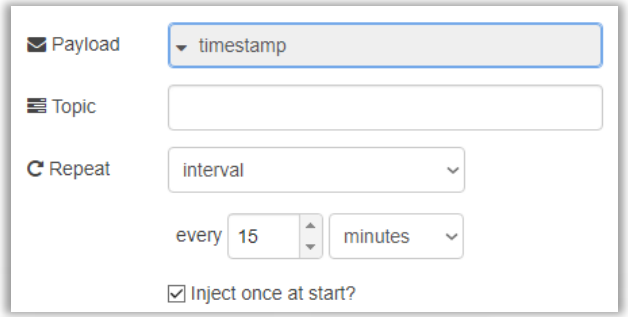
Step 1



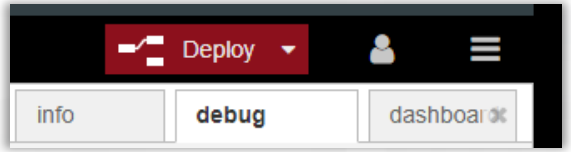
- Inject and Debug



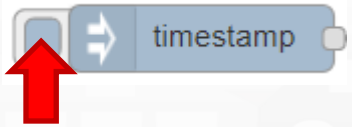
- Connect
- Configure



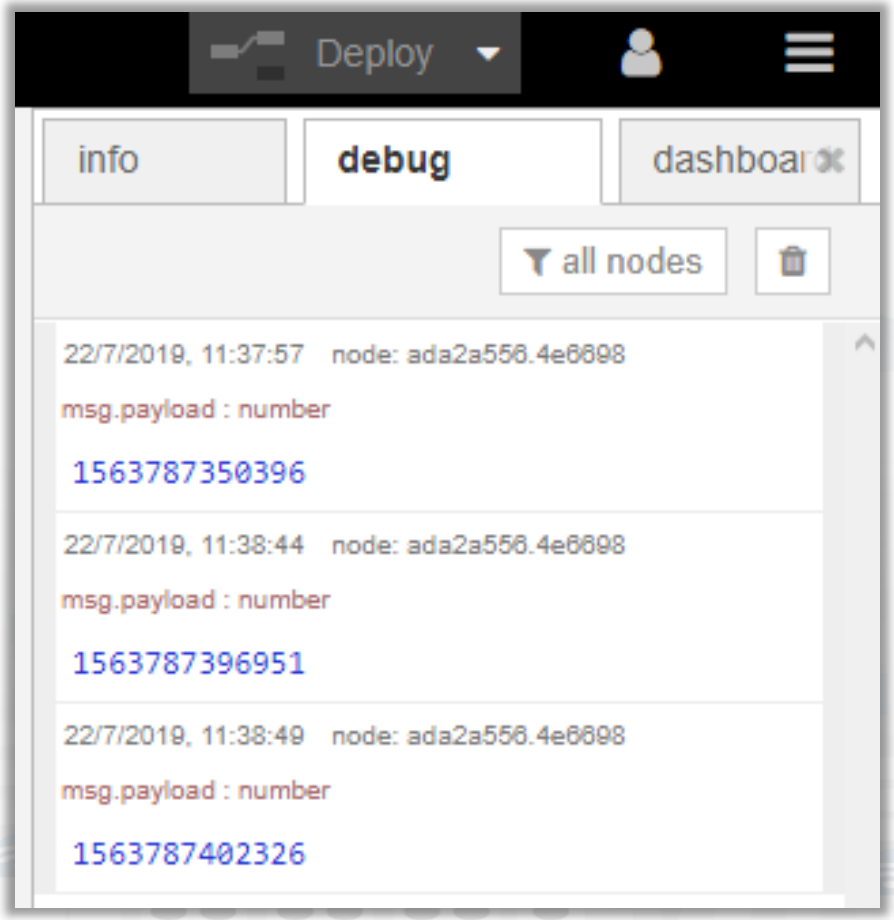
- Deploy



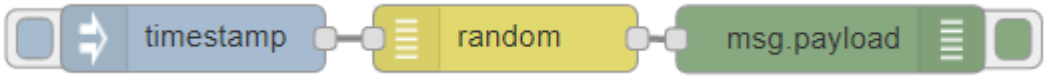
- Click



- Observe

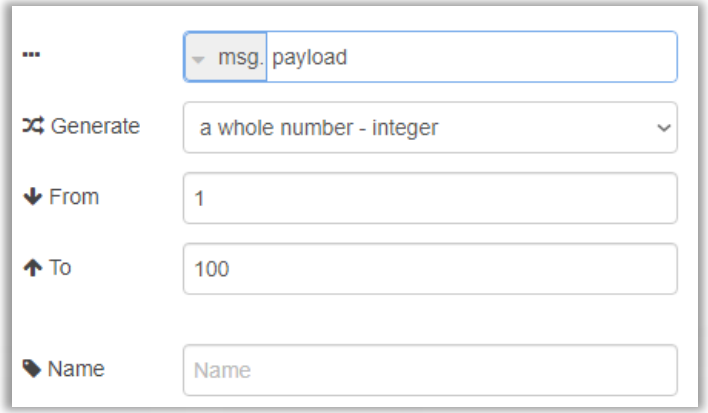


Step 2

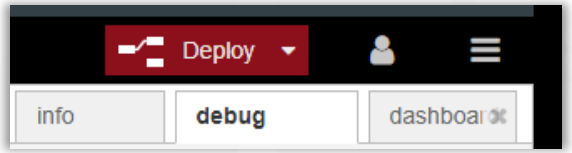


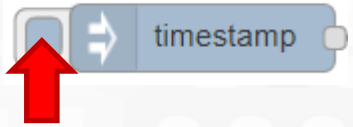
- Random
- Connect
- Configure

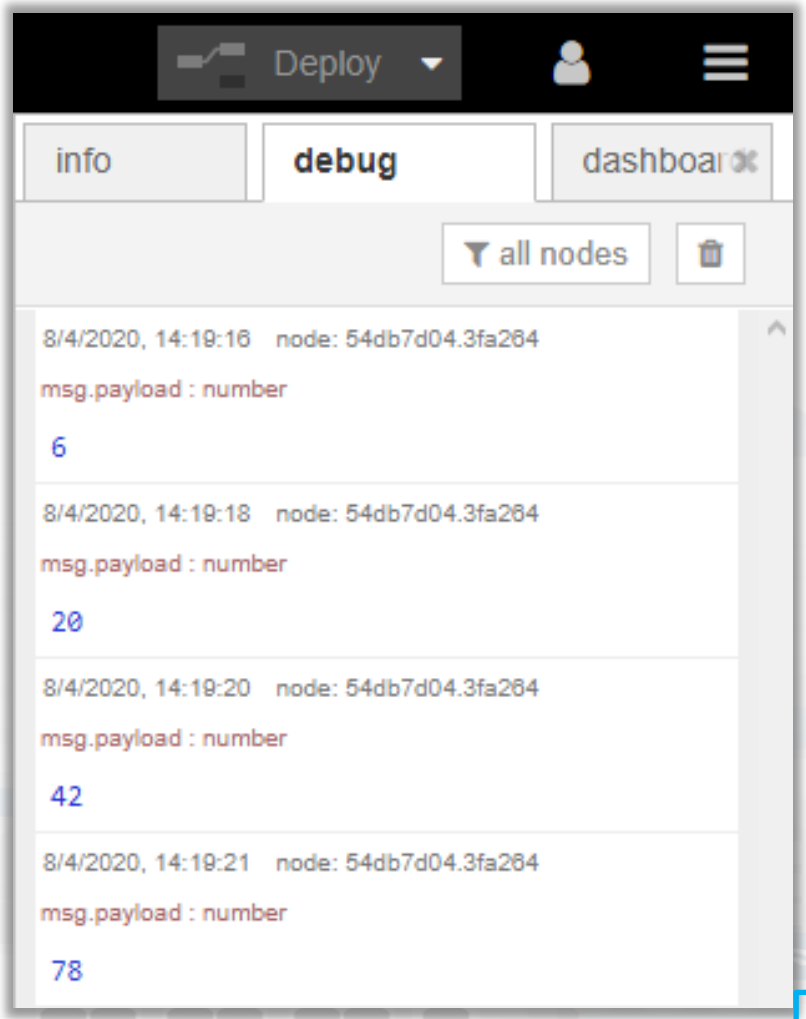




- Deploy
- Click
- Observe







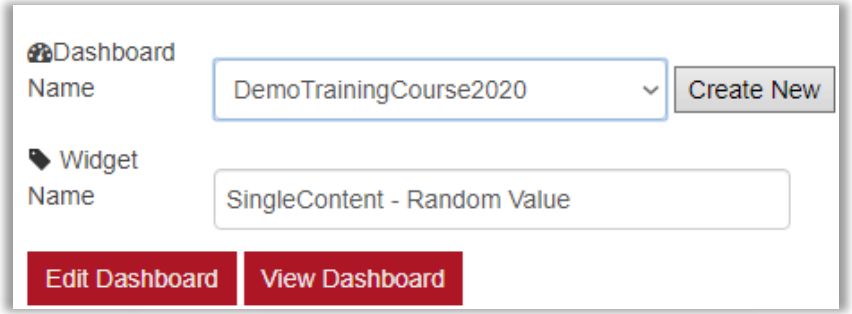
Step 3



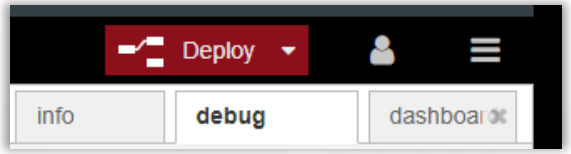
- Single content

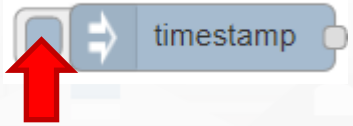


- Connect
- Configure

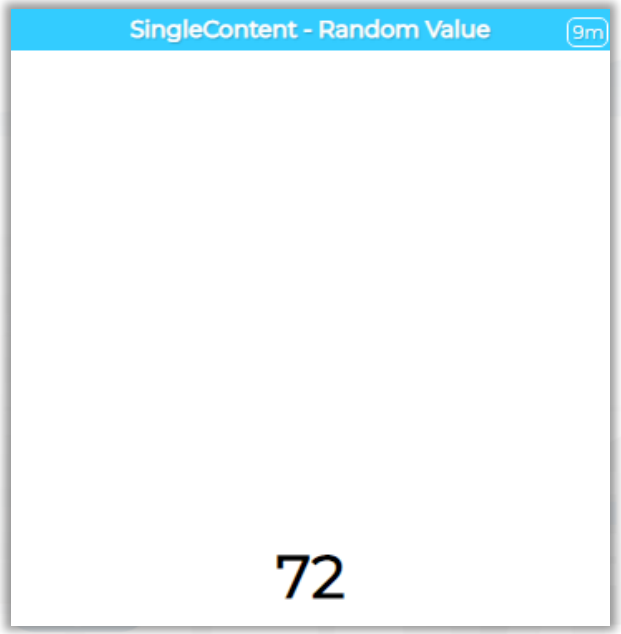


- Deploy

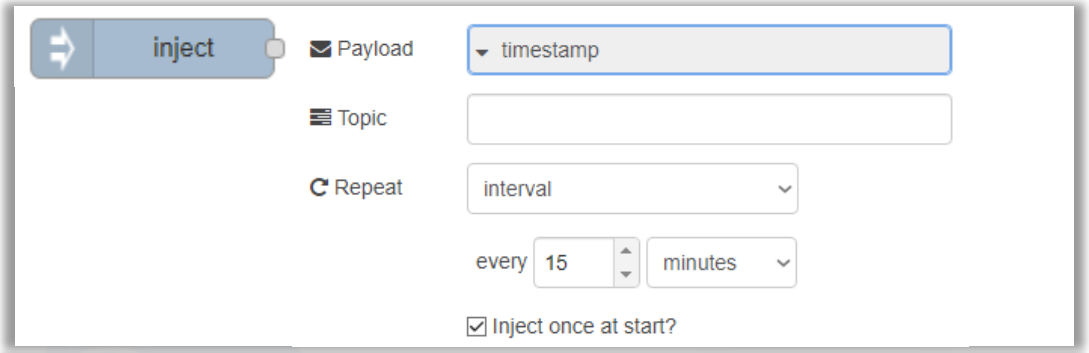


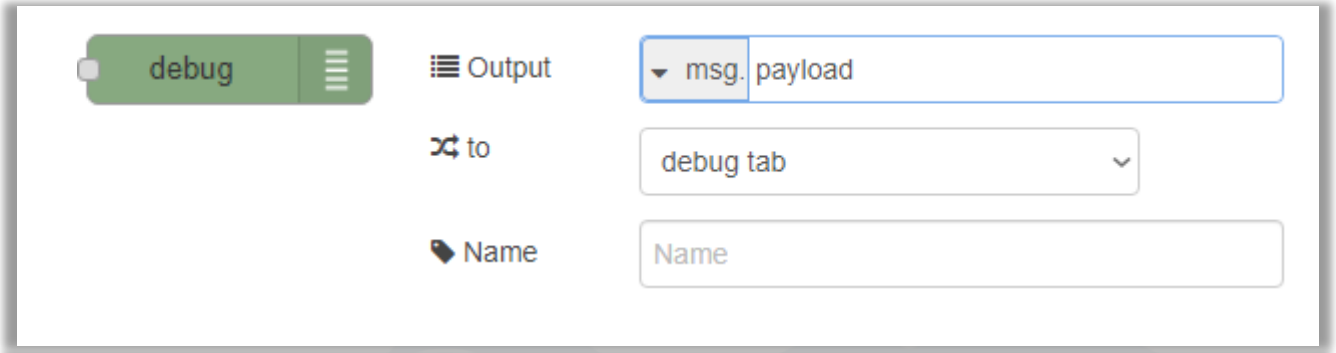
- Click  Click 

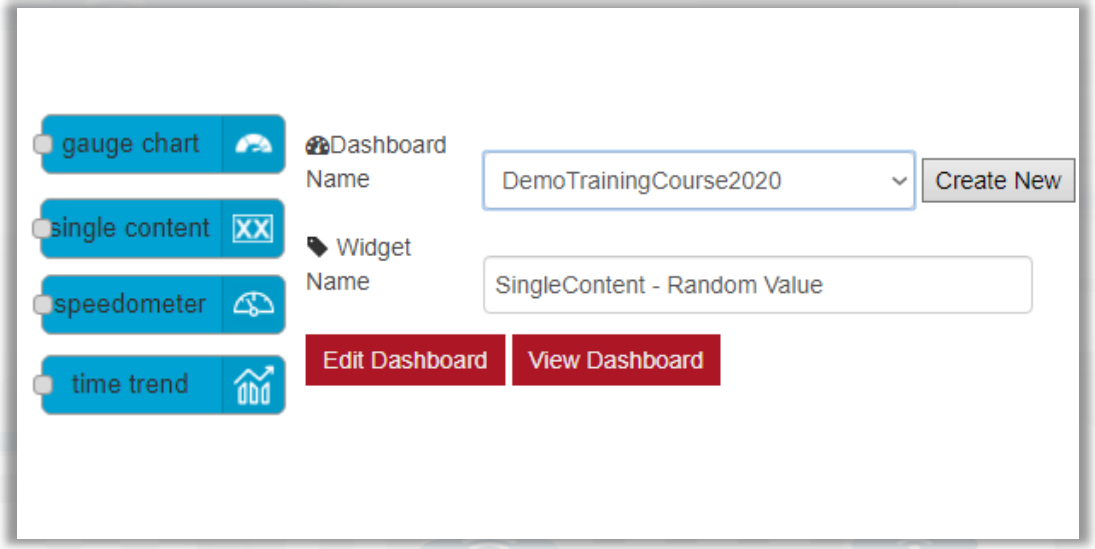
- Observe

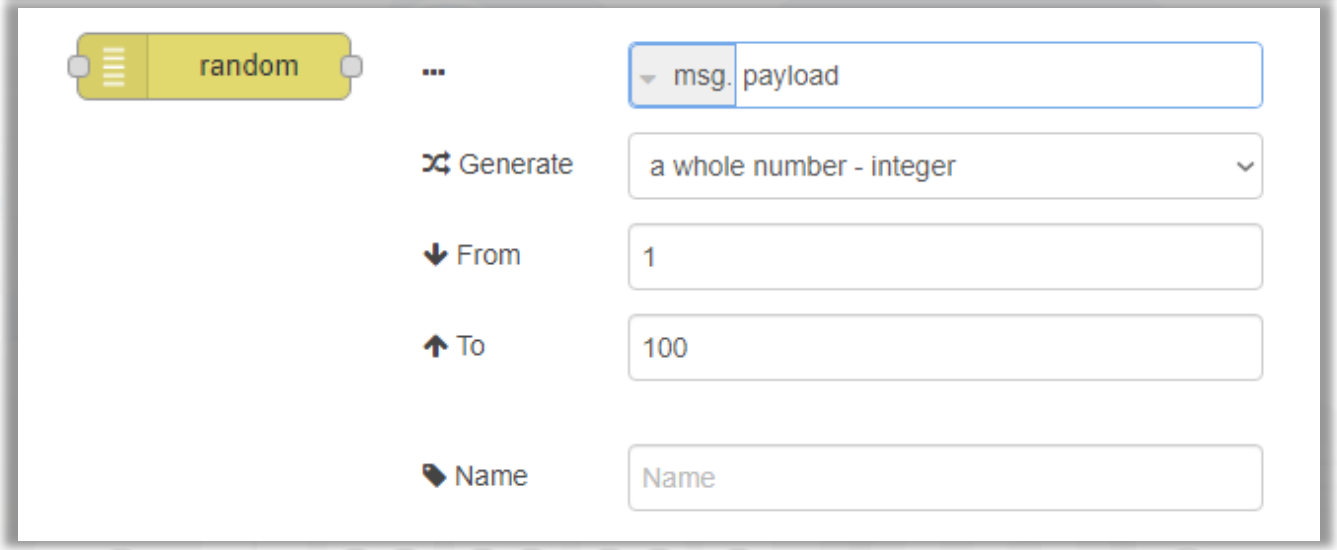


Nodes configuration

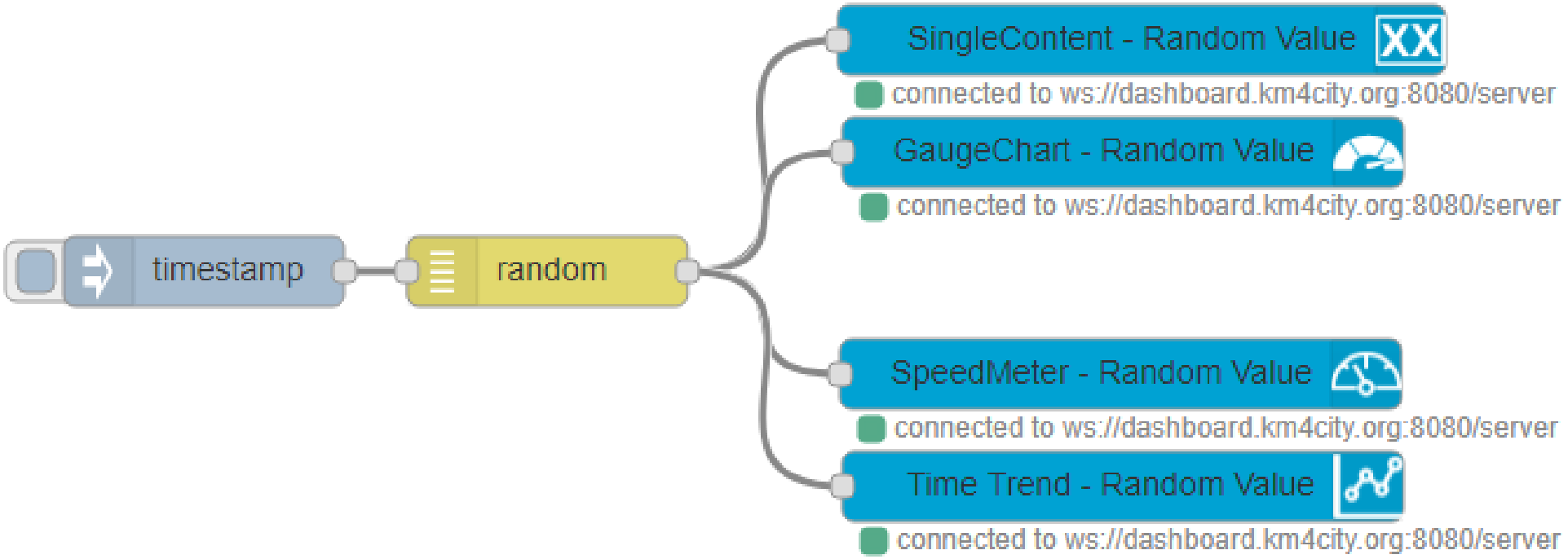








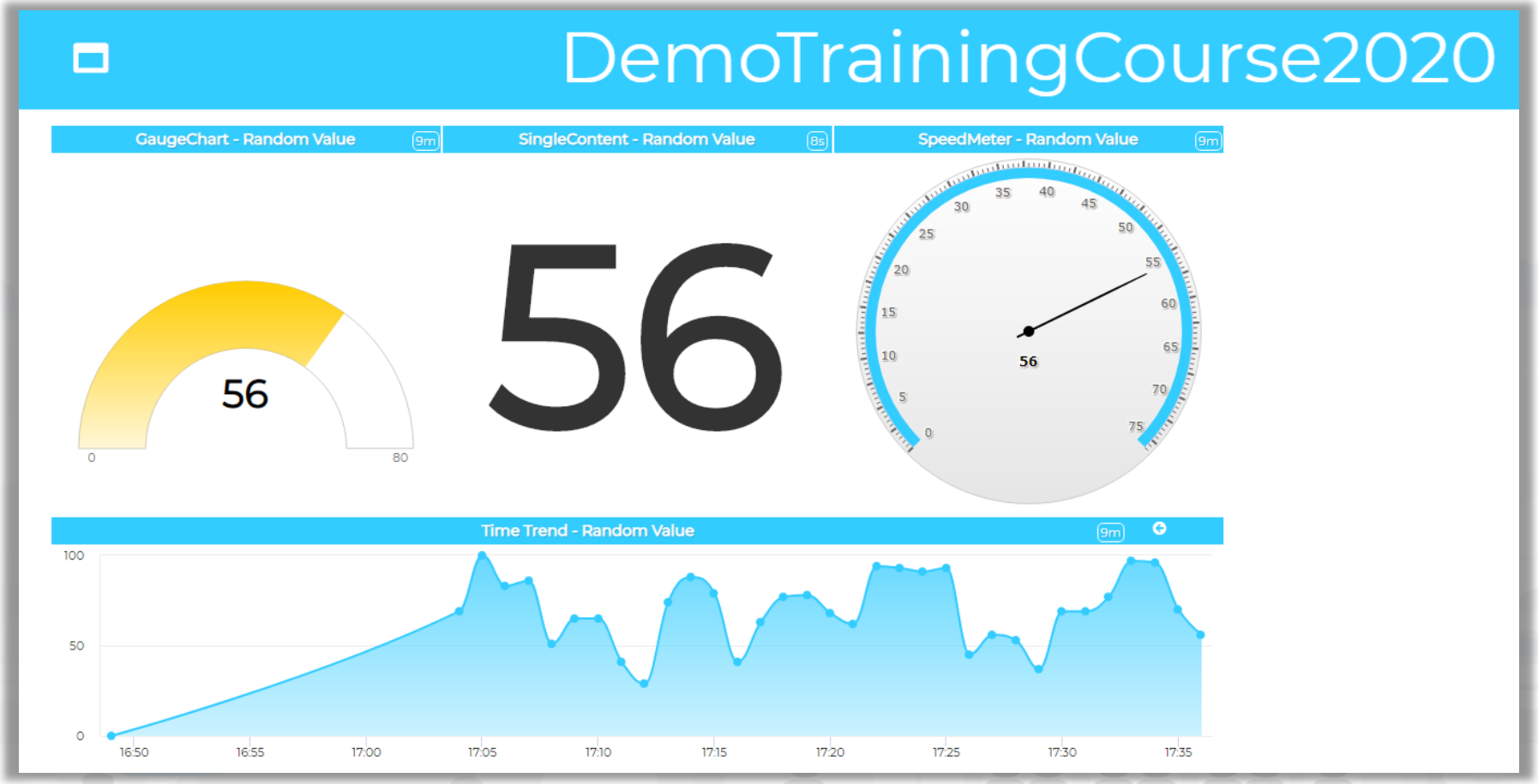
Nodes connections



Explaining: IOT Application Flow

- On Click or Every 15 minutes the ***timestamp*** node sends a message to the ***random*** node.
- When the message arrives, the ***random*** node generates a random number as output message.
- The Number can be sent to Different kinds of nodes to show it on NodeRed Dashboard.

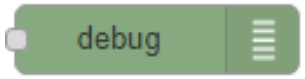
Resulting Dashboard



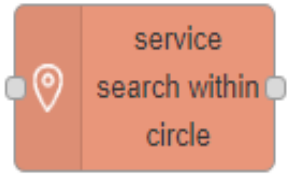
Nodes for flow



Generates an input for the other nodes. It can be repeated at predefined intervals, entered manually and of various types (timestamp, string, number, boolean, JSONetc)

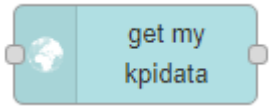


Each message that enters the debug node is shown in the "debug" tab on the right of nodered (you can choose which part of the message to show)



Search in around a certain point of the indicated service. It returns:

- servicesUri of all the services found,
- a GeoJSON containing a minimum of information about the services found, including the coordinates and the name of the service.



Retrieve the information about My KPIData saved on the Snap4city platform



Display values in different modes on a dashboard. Check info of the node in the Node-RED tab.

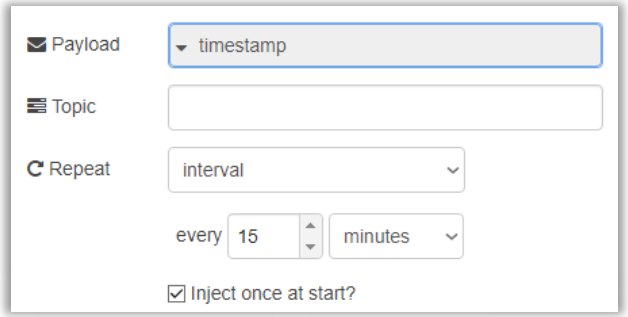
Step 1



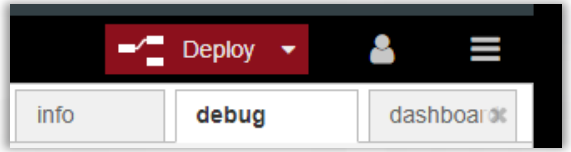
- Inject and Debug



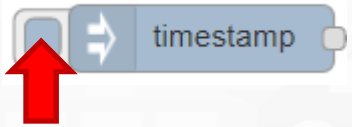
- Connect
- Configure



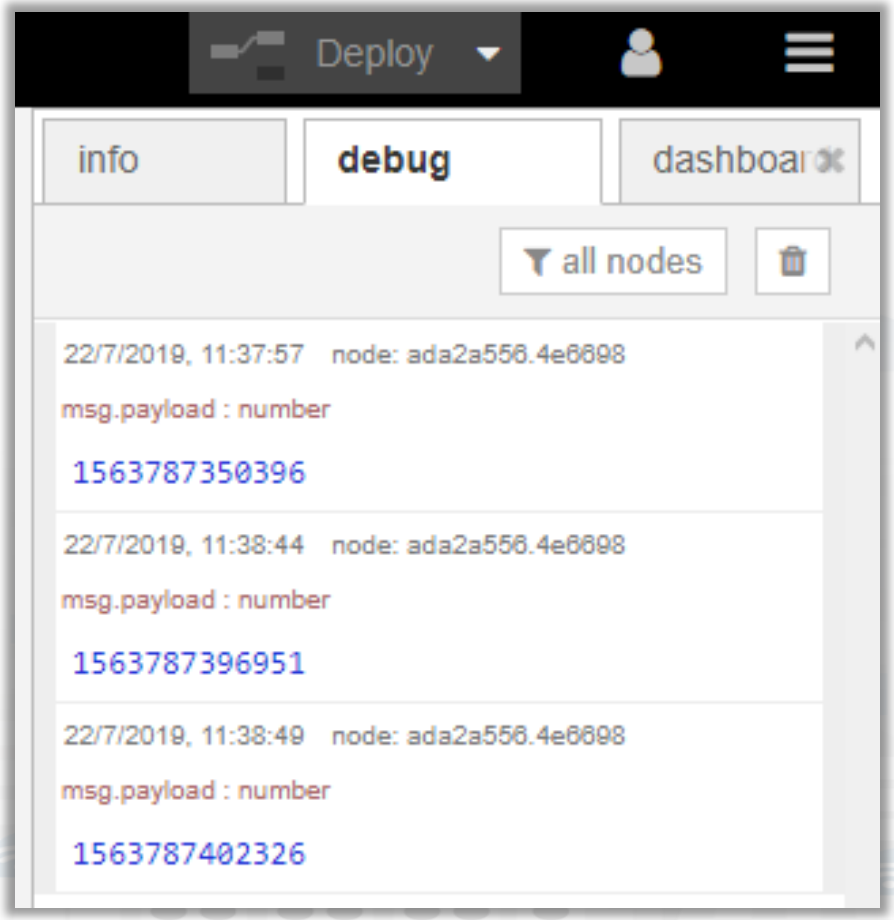
- Deploy



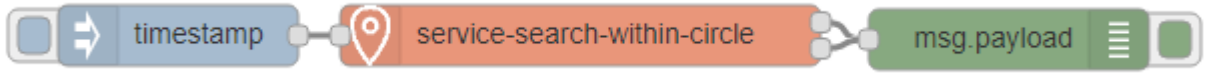
- Click



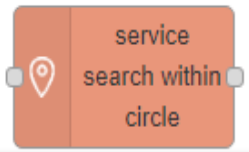
- Observe



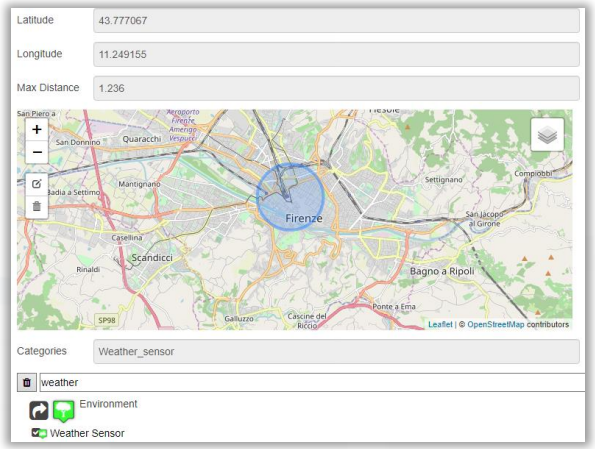
Step 2



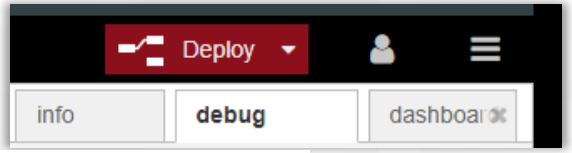
- Service Search Within Circle



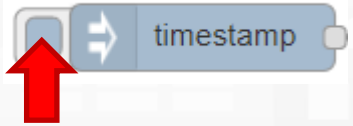
- Connect
- Configure



- Deploy



- Click



- Observe



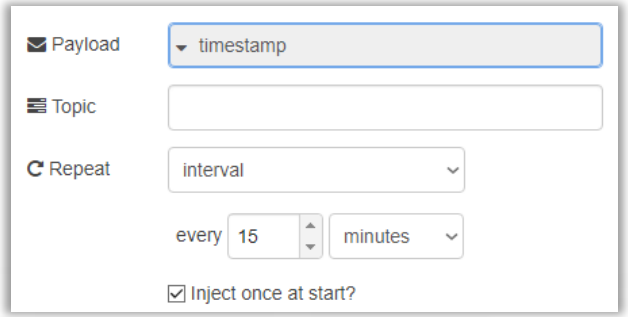
Step 1 Bis



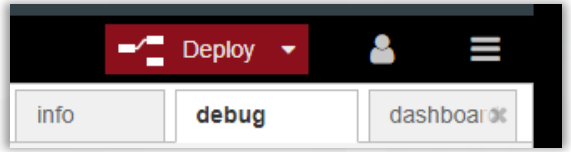
- Inject and Debug



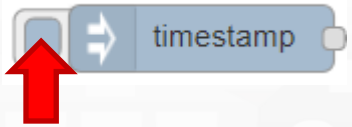
- Connect
- Configure



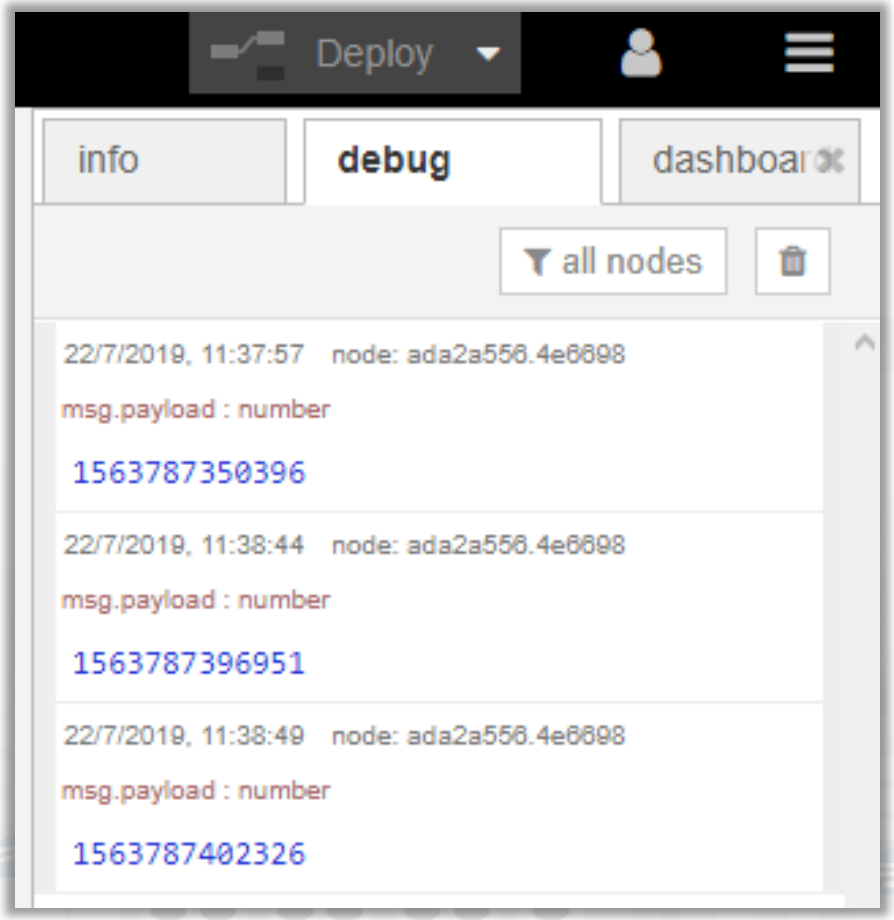
- Deploy



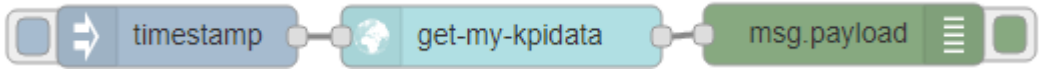
- Click

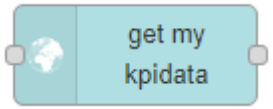


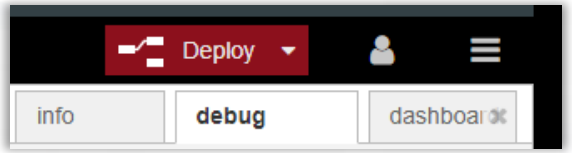
- Observe

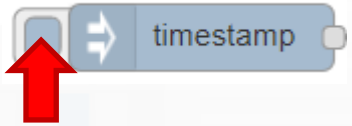


Step 2 Bis



- Get My KPIData 

- Connect
- Deploy 

- Click 

- Observe



Step 3



- Inject Node 

- Configure with data of Weather Sensors and MyKPI retrieved at the previous steps

```
[
    {
        "metricId": "http://www.disit.org/km4city/resource/tusc_weather_sensor_ow_3166540",
        "metricHighLevelType": "Sensor",
        "metricName": "tusc_weather_sensor_ow_3166540",
        "metricType": "airTemperature"
    },
    {
        "metricId": "http://www.disit.org/km4city/resource/tusc_weather_sensor_ow_3182522",
        "metricHighLevelType": "Sensor",
        "metricName": "tusc_weather_sensor_ow_3182522",
        "metricType": "airTemperature"
    },
    {
        "metricId": "17057458",
        "metricHighLevelType": "MyKPI",
        "metricName": "Room 1",
        "metricType": "Temperature"
    },
    {
        "metricId": "17057459",
        "metricHighLevelType": "MyKPI",
        "metricName": "Room 2",
        "metricType": "Room Temperature"
    }
]
```

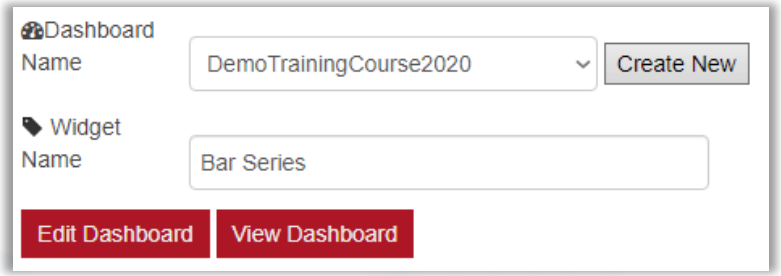
Step 4



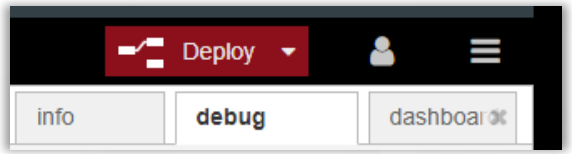
- Bar Series



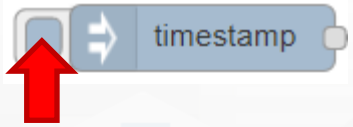
- Connect
- Configure



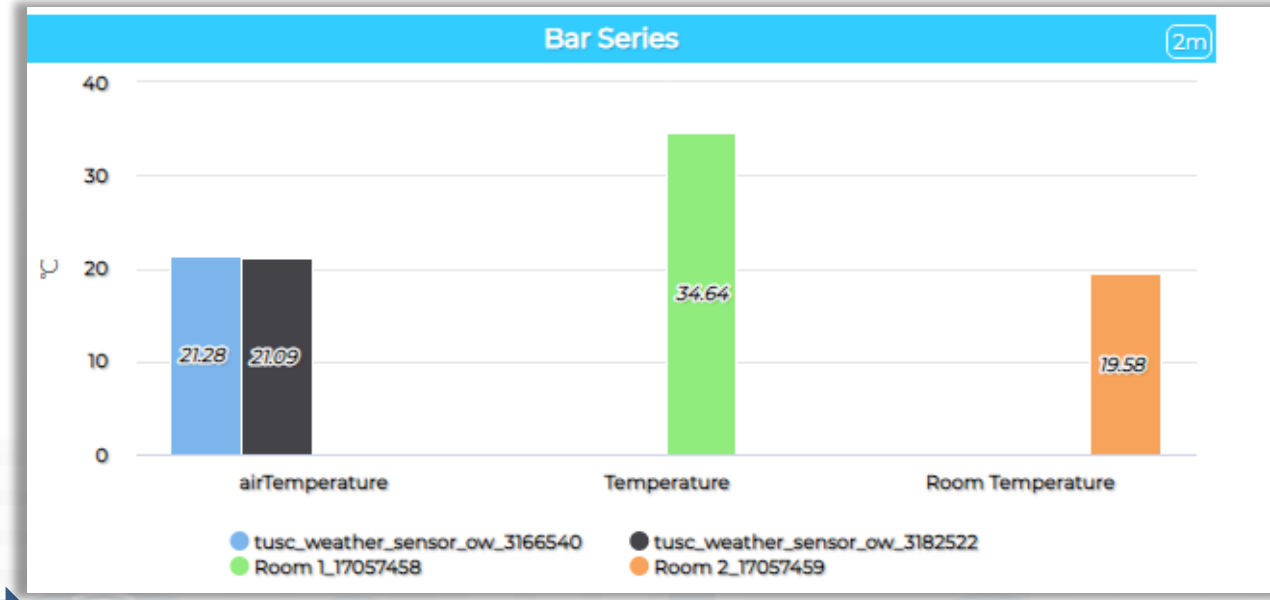
- Deploy



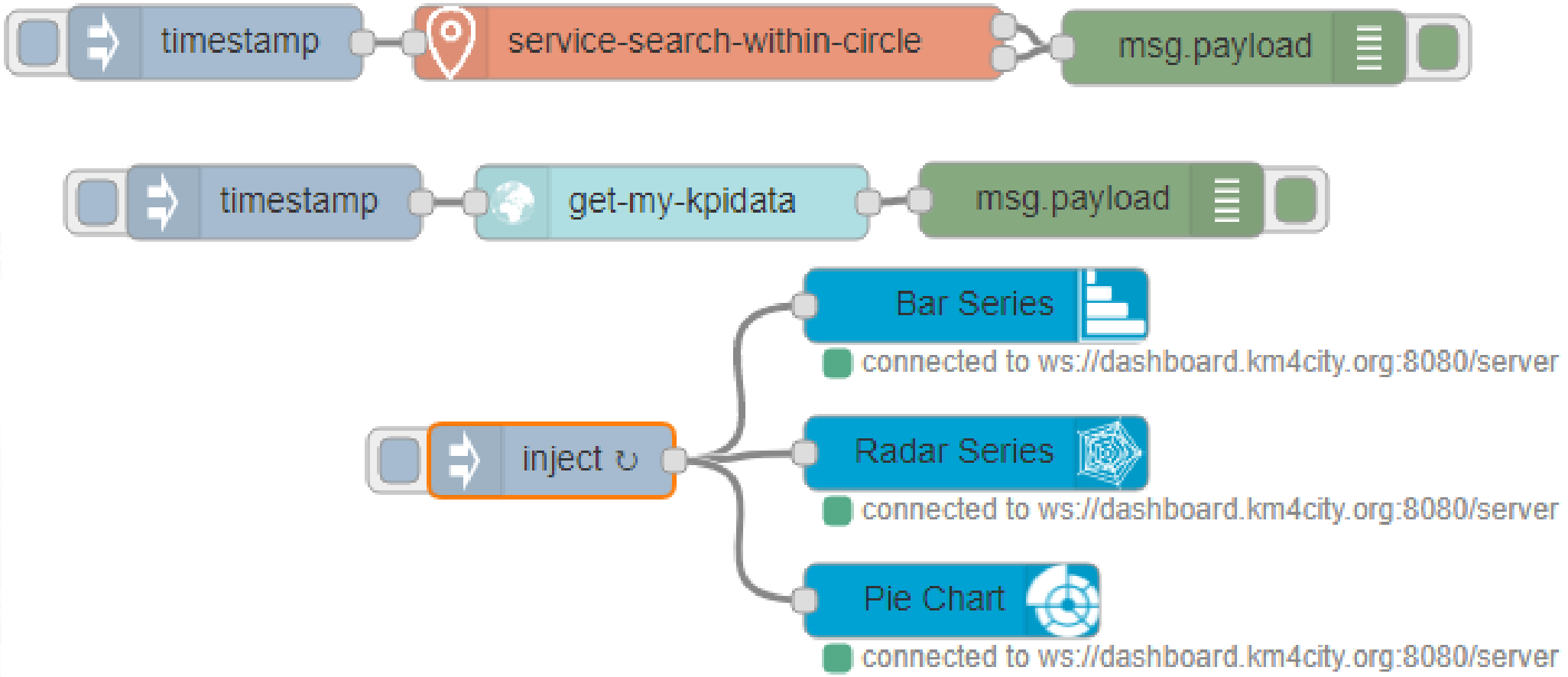
- Click



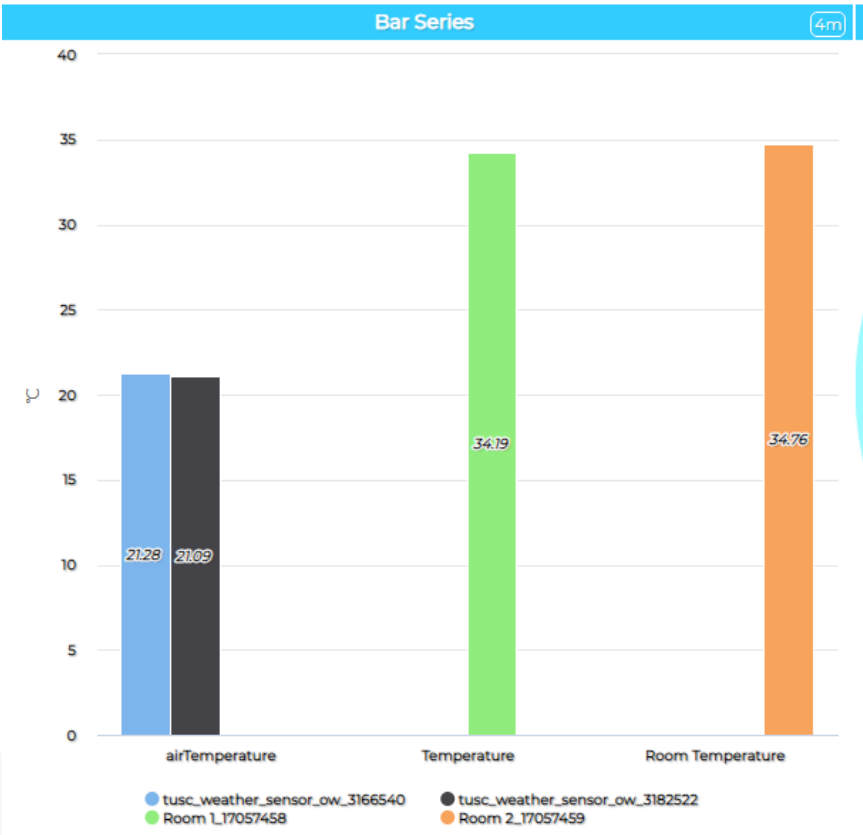
- Observe

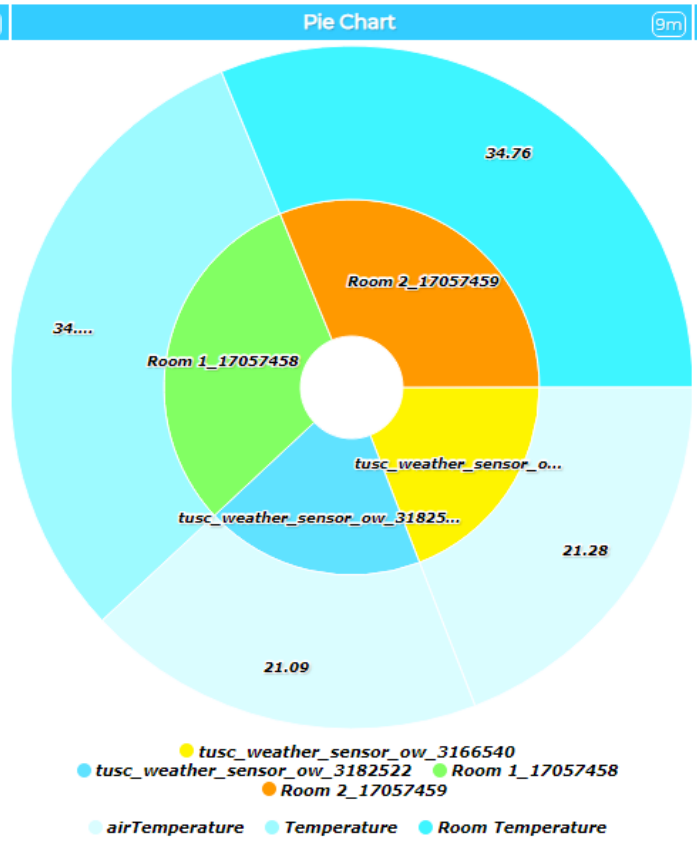


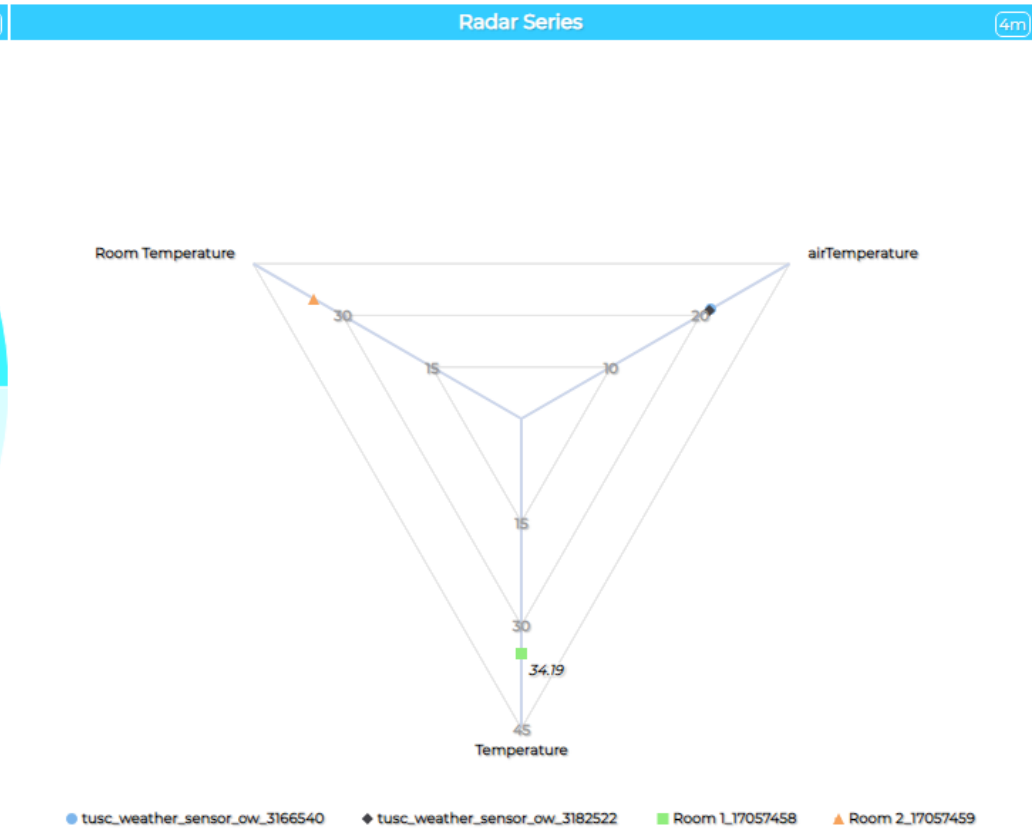
Nodes connections



Resulting Dashboard







end DEMO Section 2

TOP

IOT App = Node-RED + Snap4City

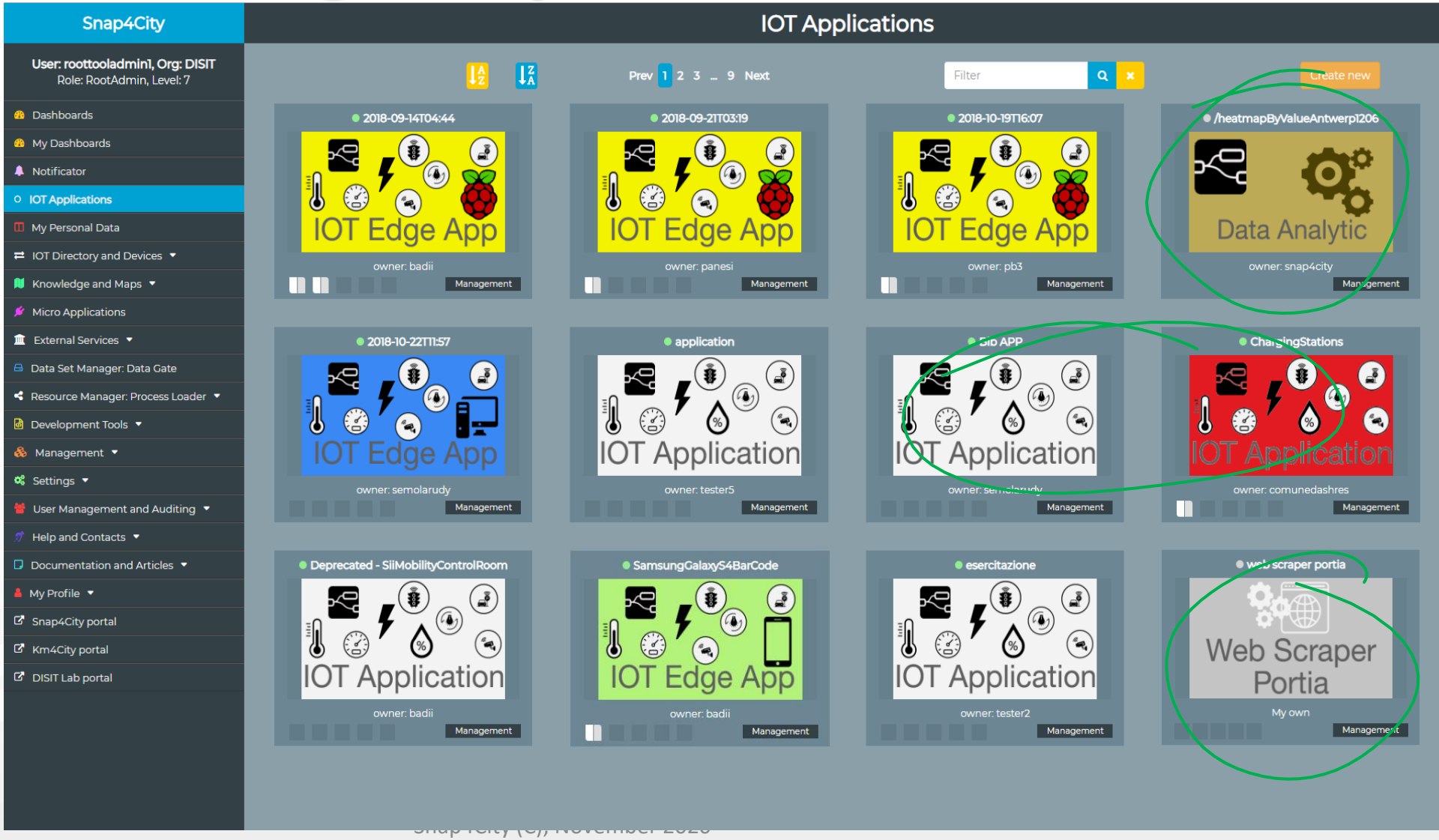
Data Analytics and WebScraping





IOT Application Listing, they can be

- Basic (white)
- Advanced (red)
- IOT Edge
  - Raspberry Pi
  - Android
  - Win/Linux
- **Data Analytic (Plumber)**
- **Web Scraper (Portia)**

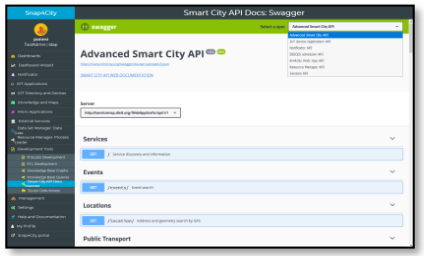


IOT App vs Data Analytics in R-Studio

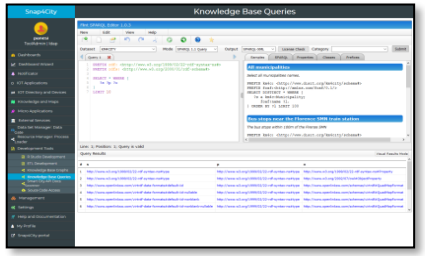


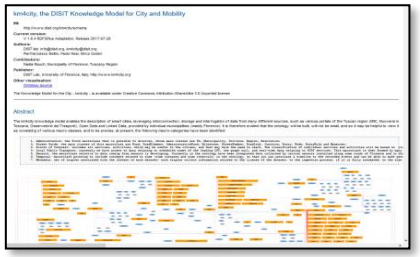
Data Analytics Dev. in R Studio and/or Tensor Flow

Swagger



SPARQL, FLINT





Ontology Schema



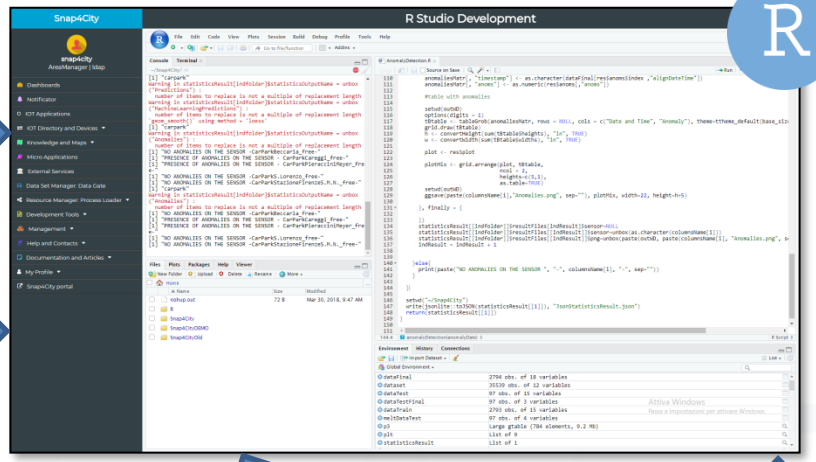
LOG.disit.org

Knowledge Base, Km4City



Big Data Store Facility

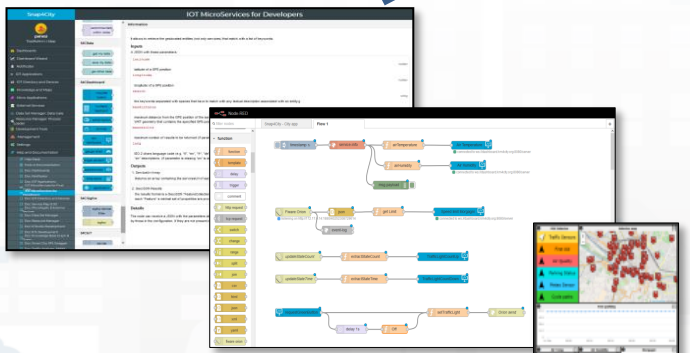
Smart City API from Knowledge Base and other tools



R Studio



Creating Micro Services



Using them into IOT Applications

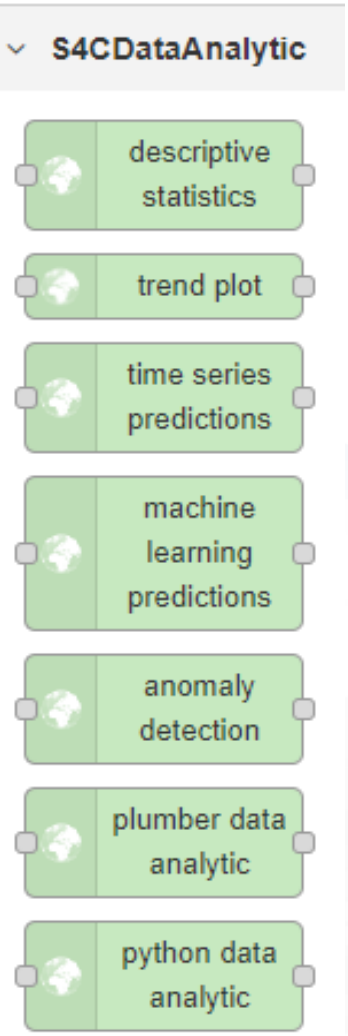
Saving
Sharing
Reusing



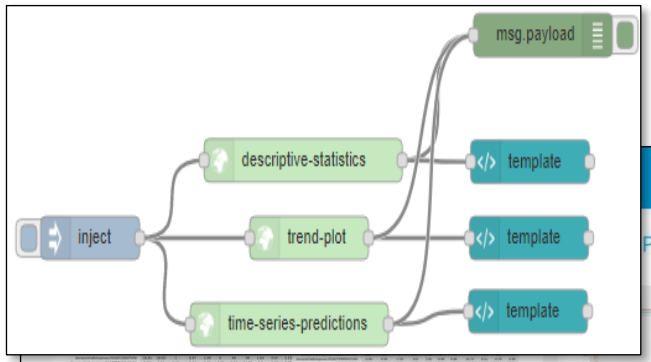
Resource Manager



Data Analytics to MicroServices

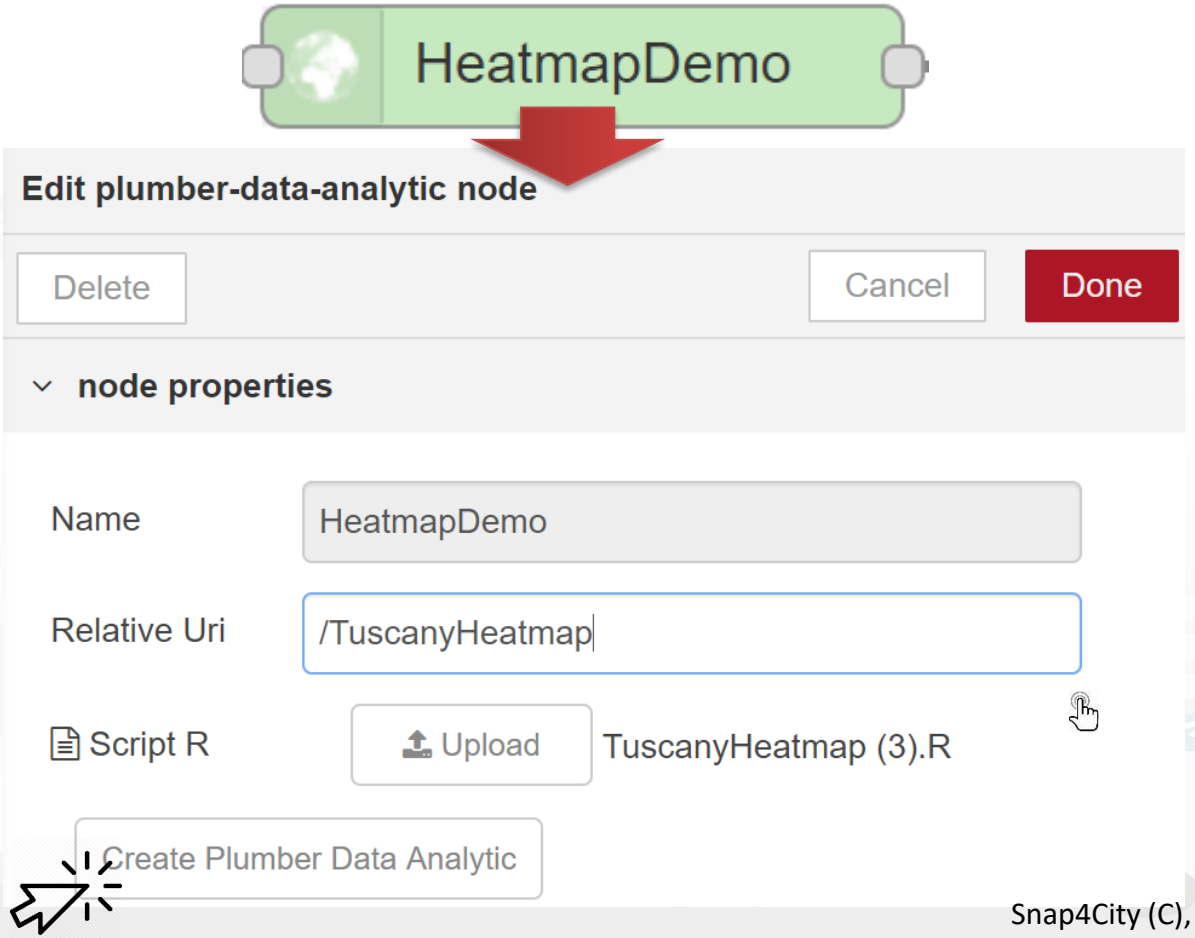


R Studio and Python algorithms are **automatically** transformed into **MicroServices** for your IOT Applications



Data Analytics to MicroServices with Plumber

How to configure the **plumber data analytic** node:



Relative Uri is the same of the R @get annotation:

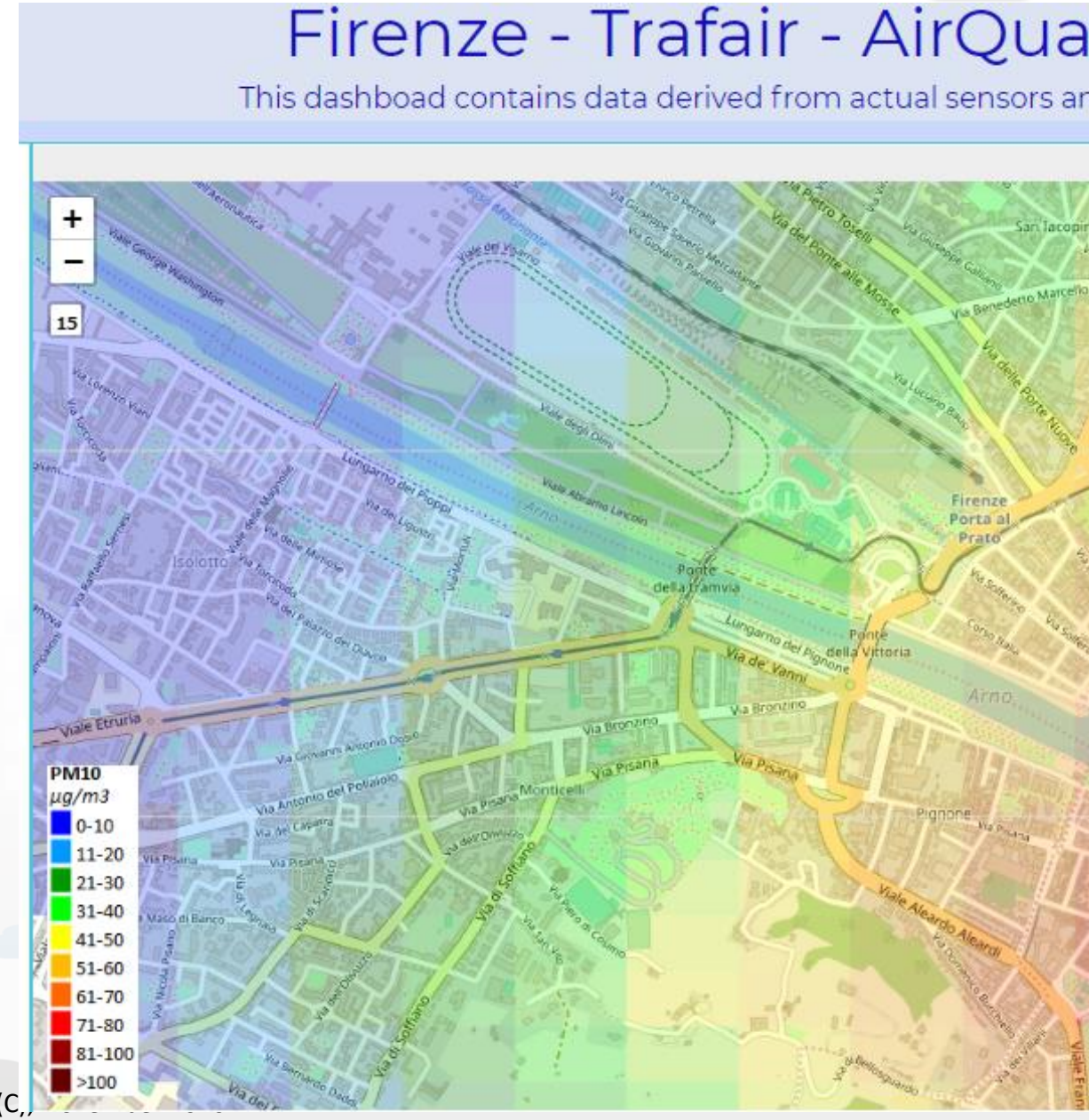
```
#' @get /TuscanyHeatmap
```

Automatic_IDW_Heatmaps_Creation.R

```
#' @get /heatmapIDW
#' @serializer unboxedJSON
heatmapIDW <- function (city, long_min,
long_max, lat_min, lat_max, epsgProjection,
subnature, valueType, fromDateTime,
toDateTime, heatmapName, colorMap) {
...
}
```

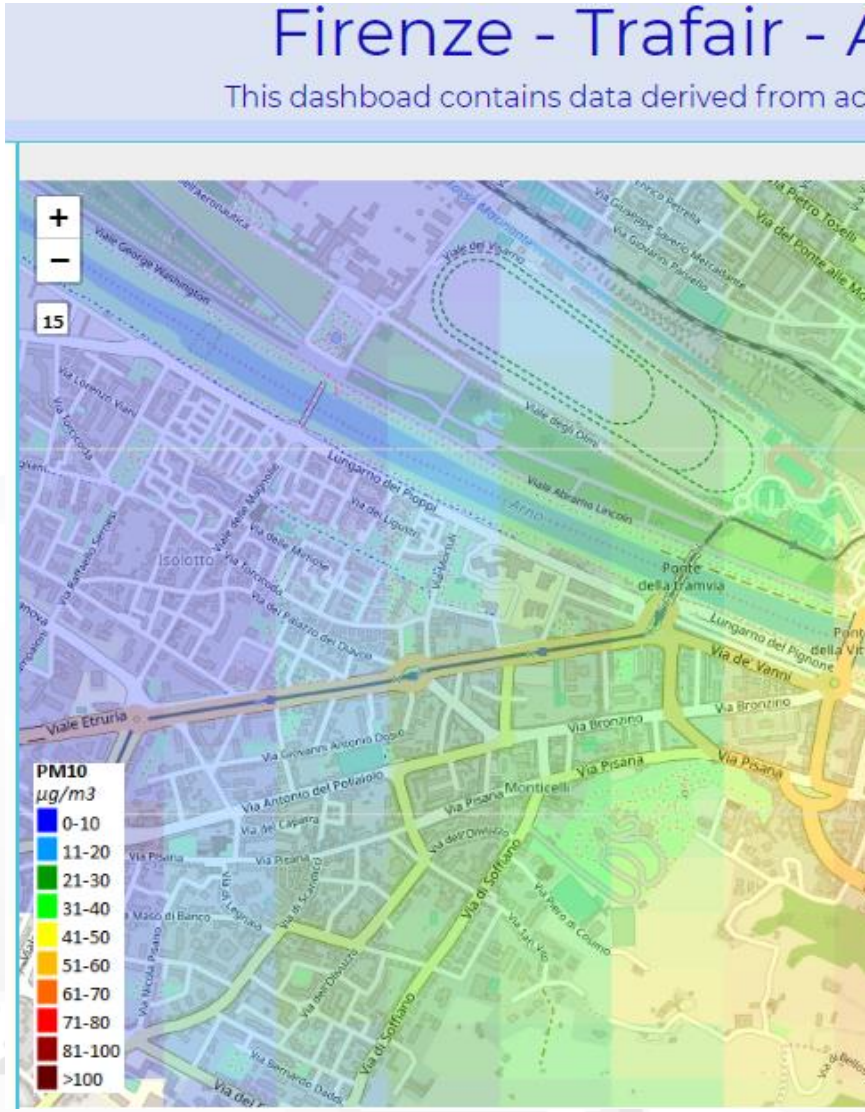
More Information

- HOW TO: produce heatmaps, custom heatmaps on any data
- TC1.19: Creating and Exploit heatmaps for Dashboards and as reference data services



PARAMETERS EXPLANATION

- city = name of the city related to the heatmap bbox (e.g., Rome, Florence, Helsinki, Antwerp etc.)
- long_min, long_max, lat_min, lat_max = heatmap bbox parameters (latitude and longitude coordinates)
- epsgProjection = UTM Projection related to the heatmap zone
- subnature = subnature of the sensor of interest (e.g., airQuality)
- valueType = single parameter or parameters array with the names of the measure of interest (the case of the array is related to multiple names for a single measure and NOT to multiple different measures - e.g, NO2, no2, airQualityNO2 etc.)
- fromDateTime = start date and time interval in timestamp format
- fromDateTime = end date and time interval (for example 1-hour or 1-day or the timestamp format 2020-05-25T15:00:00)
- heatmapName = name of the heatmap
- colorMap = name of the color map associated to the type of measure of interest

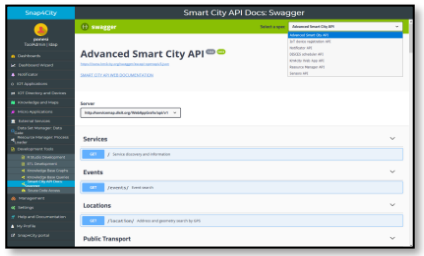


IOT App vs Data Analytics in Python

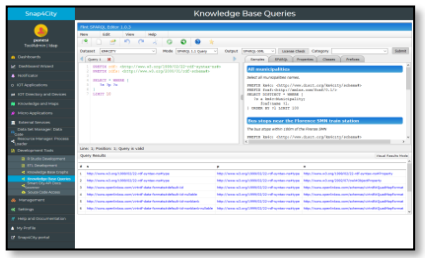


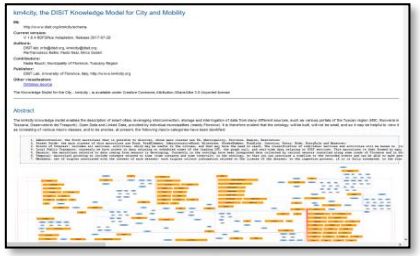
Data Analytics Development in Python,

Swagger



SPARQL, FLINT





Ontology Schema



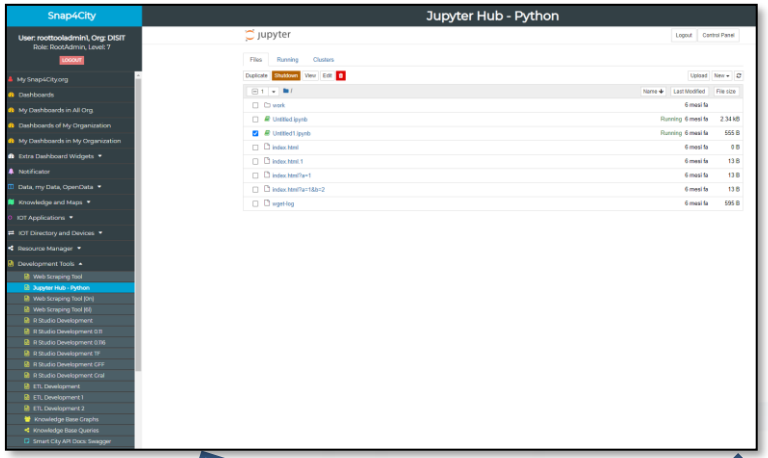
LOG.disit.org

Knowledge Base, Km4City



Big Data Store Facility

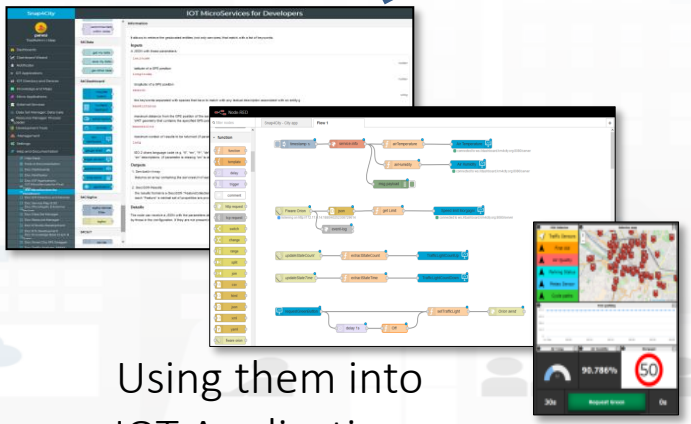
Smart City API from Knowledge Base and other tools



Coding
Testing



Creating
Micro Services



Using them into
IOT Applications

Saving
Sharing
Reusing



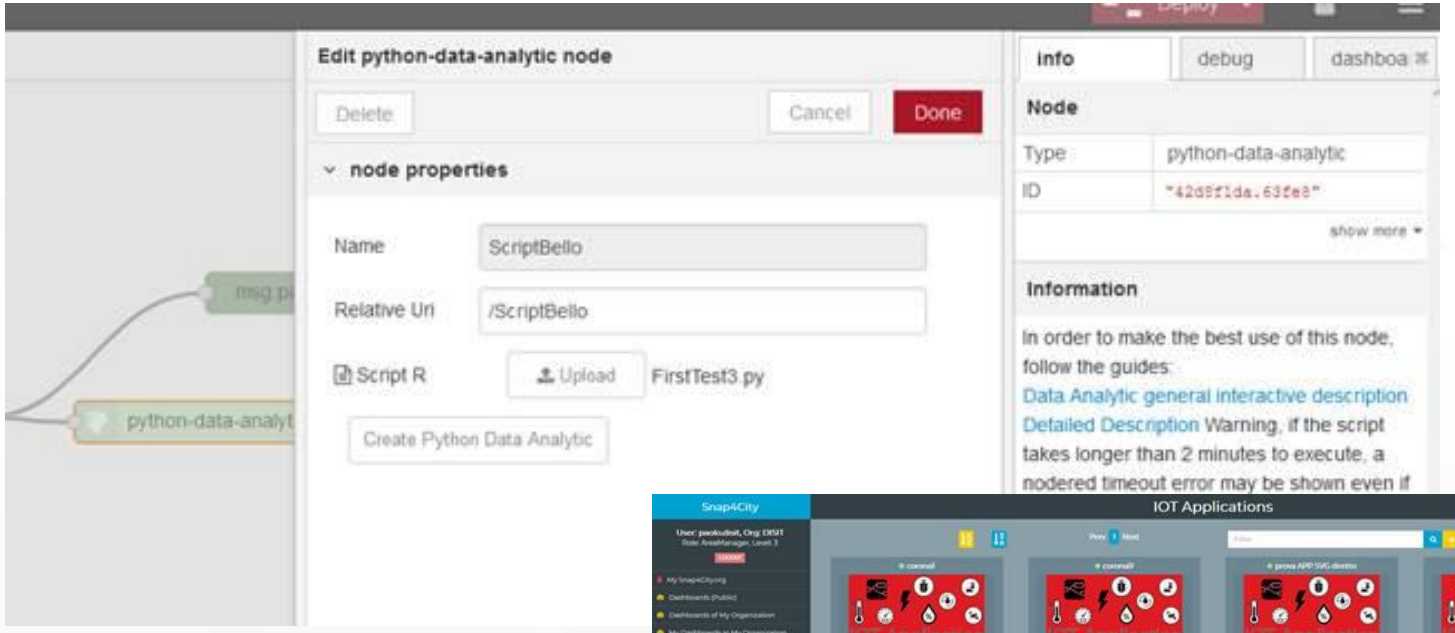
Resource Manager



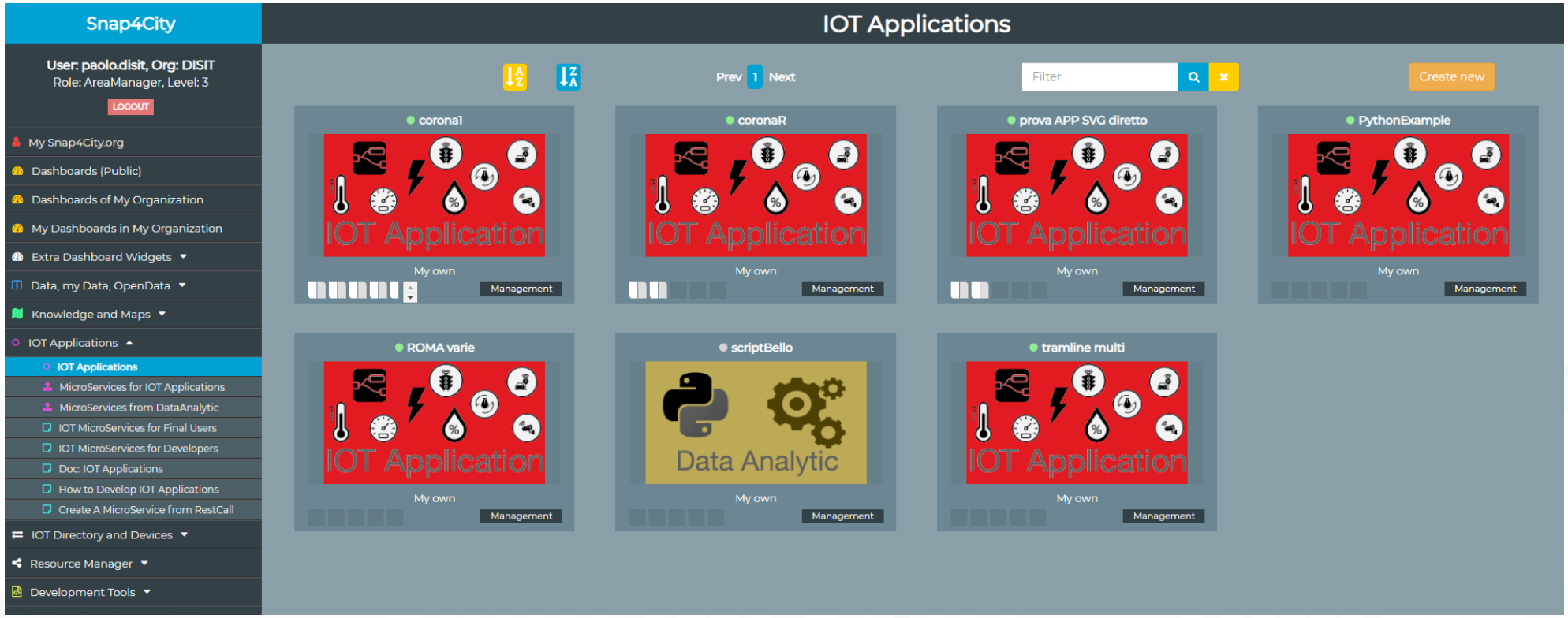
Python process

- Develop Python code exploiting Flask calls
- Test on local for the Call
- Test on Cloud for API
- Deploy via IOT App









More information

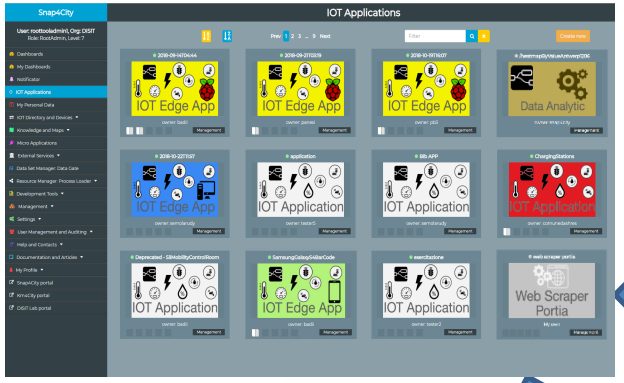
- [HOW TO: develop DataAnalytic in Python and manage them via IOT App]

IOT App vs Web Scraping

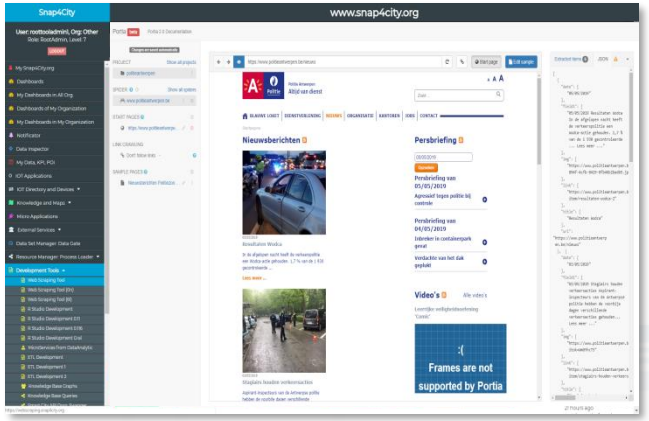


Web Scraping from Portia

My Scraping process



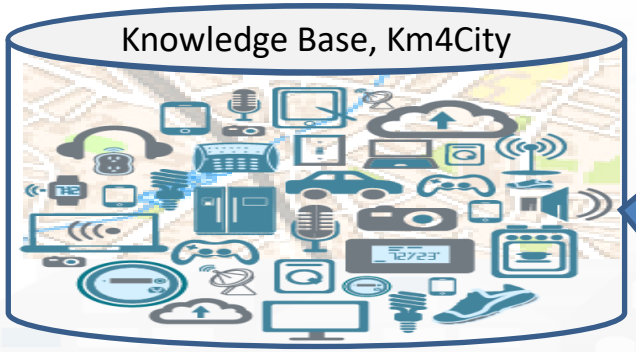
Web Scraper PORTIA

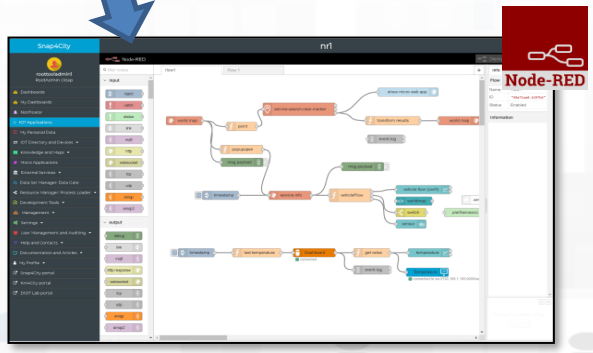


Generating WEB Scraping



Knowledge Base, Km4City





IOT App. Editor

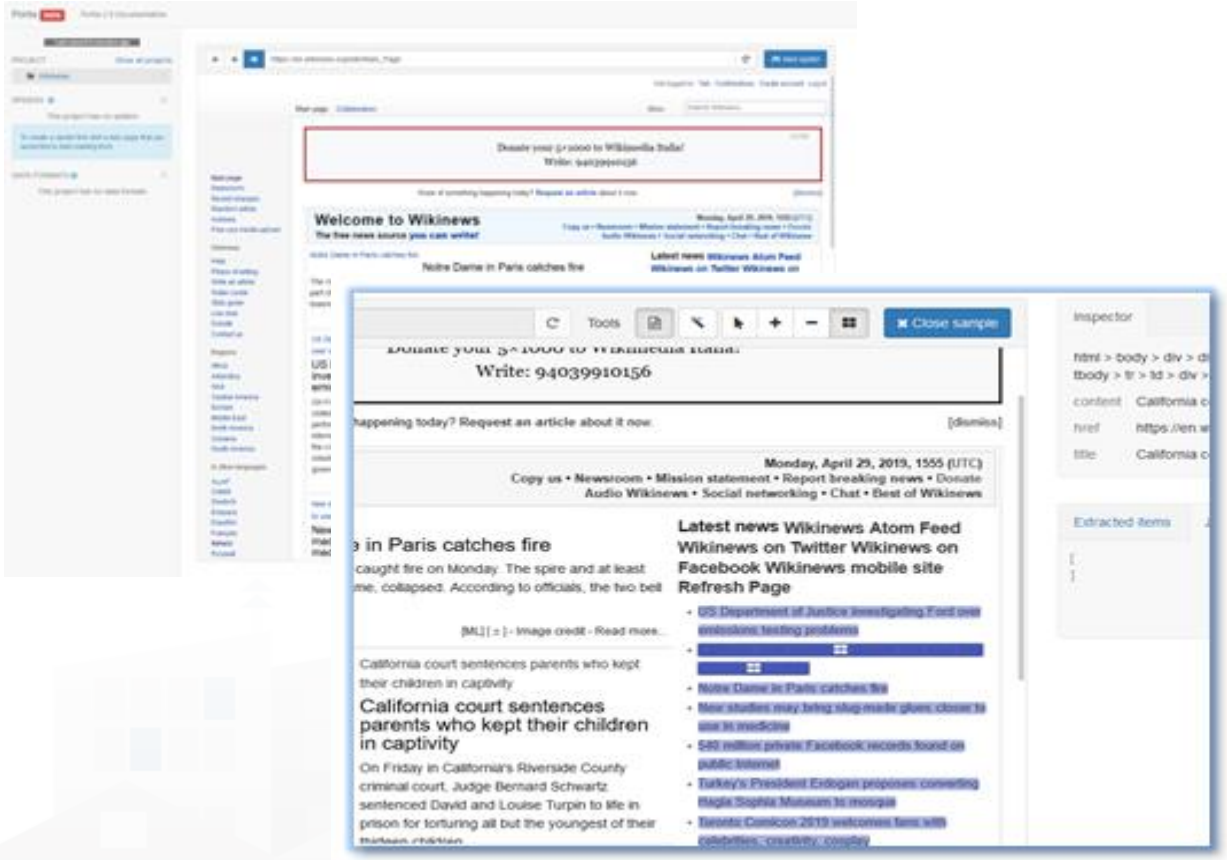
Sharing/saving reusing Scraping



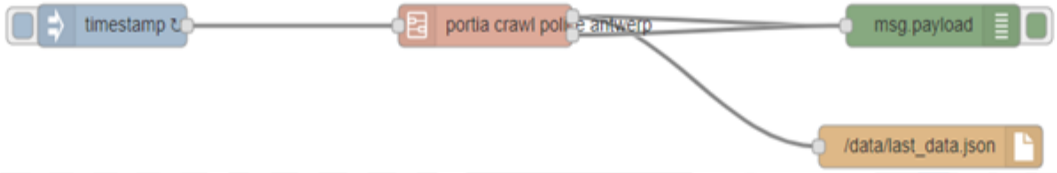
Resource Manager



Web scraping

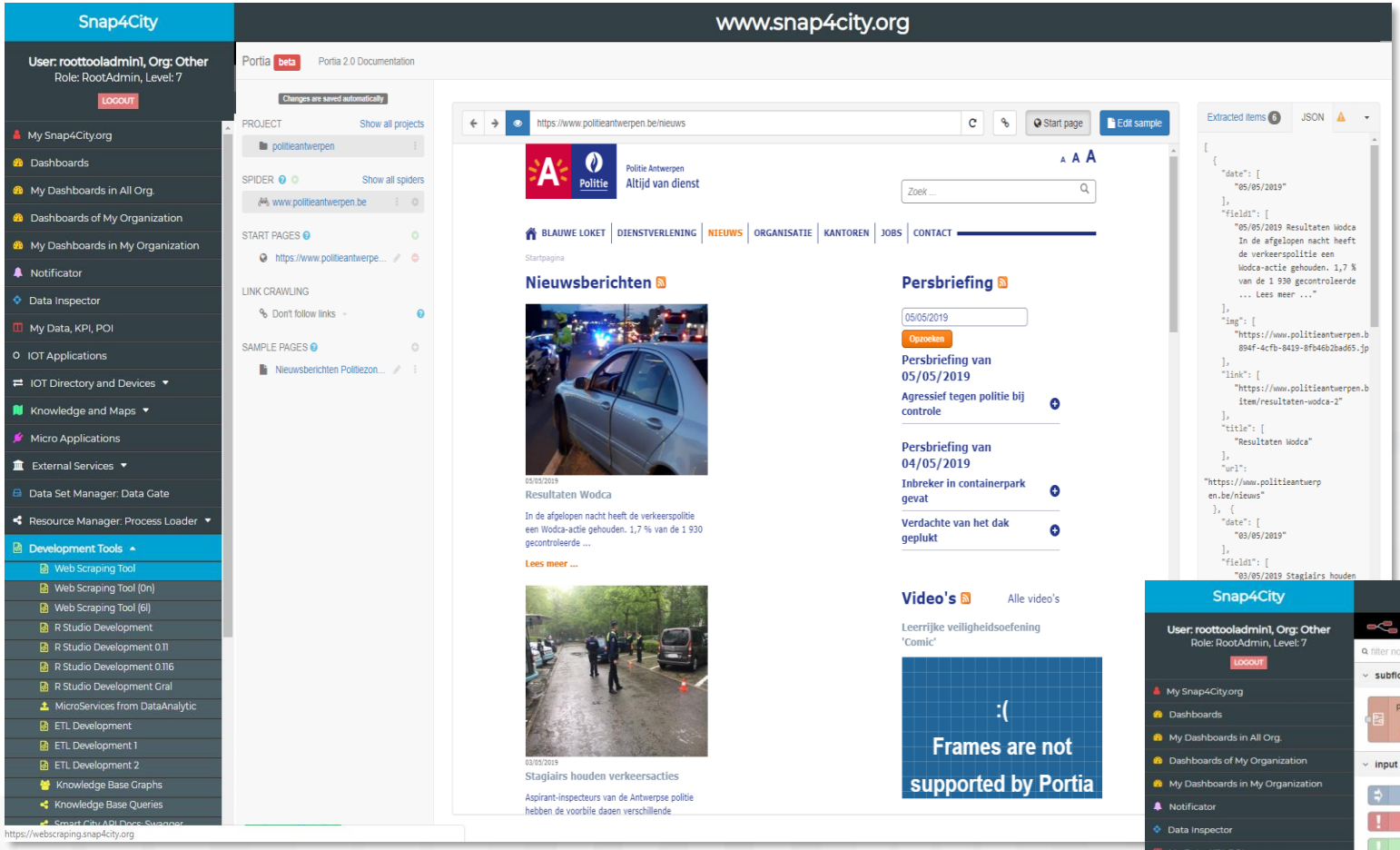


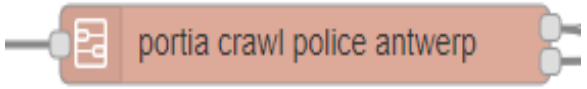


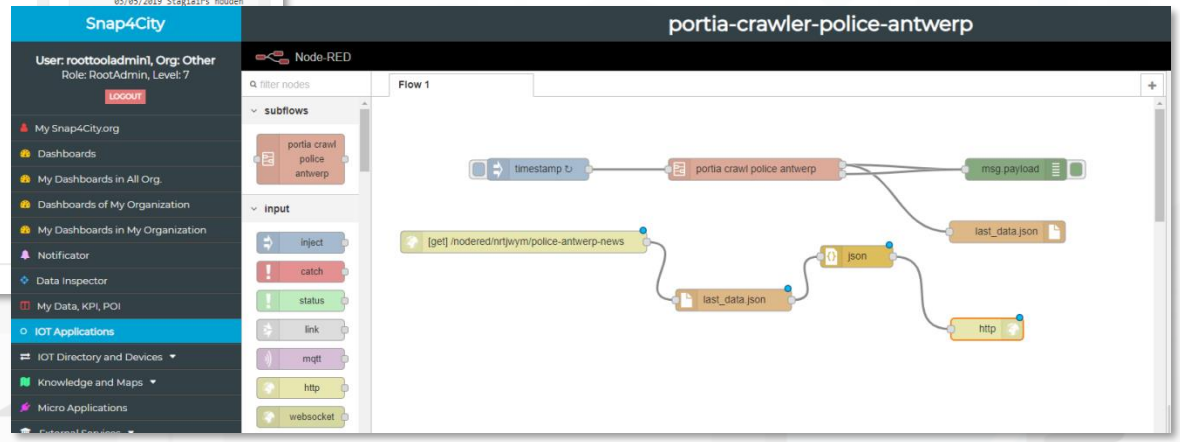


- [TC9.16 – Web Scraping to get data from web pages]

Web Scraping







IOT App self training



Self Training main path

- **Please start a fully guided training cases:**
- HOW TO: create a Dashboard in Snap4City
- HOW TO: add a device to the Snap4City Platform
- HOW TO: add data sources to the Snap4City Platform
- HOW TO: define privacy rules for personal data, produced by the end-users own device
- HOW TO: Develop Smart Applications, Snap4City development Life Cycle
- HOW TO: HLT vs Ingestion, and HLT vs Widgets
- HOW TO: Develop an IOT Application for Data Ingestion
- HOW TO: Upload data into Knowledge Base, ServiceMap (triple upload)
- HOW TO: Create as set of Devices with BulkProcessing
- HOW TO: Create an IOT Device Model
- HOW TO: Create an IOT Device Instance from IOT Directory tool

IOT Applications vs Dashboards (self training)

- IOT Applications, realized by using Snap4City Node-RED and integrated with Snap4City Nodes/MicroServices block, can be behind dashboards to get data from them with Virtual Sensors and Actuators.
  - Dashboards may be connected to multiple IOT Applications and IOT devices
  - IOT Applications may be connected with multiple Dashboards and IOT devices
- A network of Dashboards, IOT Apps and IOT Dev and data is easily realized exchanging data via secure connections.
- see the following Training Cases
  - **US2. Using and Creating Snap4City Applications with Dashboards**
  - **TC2.3 - List of MicroServices and the Help, for Final Users and Developers**
  - **TC2.4 - The daisy of MicroServices for Snap4City Dashboard and IOT App**
  - **TC2.28 - Snap4City MicroServices for Snap4City platform management from IOT Applications, feature of reflection**

IOT Applications vs Dashboards (self training)

- see the following Training Cases
  - US2. Using and Creating Snap4City Applications with Dashboards
  - TC2.3 - List of MicroServices and the Help, for Final Users and Developers
  - TC2.4 - The daisy of MicroServices for Snap4City Dashboard and IOT App
  - TC2.28 - Snap4City MicroServices for Snap4City platform management from IOT Applications, feature of reflection
  - TC2.24 - IOT Applications developed exploiting MicroServices, also supporting GDPR, real time, data sharing, etc.
  - US9. Creating Snap4City IOT Applications, different formats, protocols, brokers, communications
  - TC6.8 - ETL processes for data transformation, and exploiting MicroServices/API/RestCall
  - TC2.13 - Import of any new Block/MicroService or library of MicroServices into IOT Application Builder tools

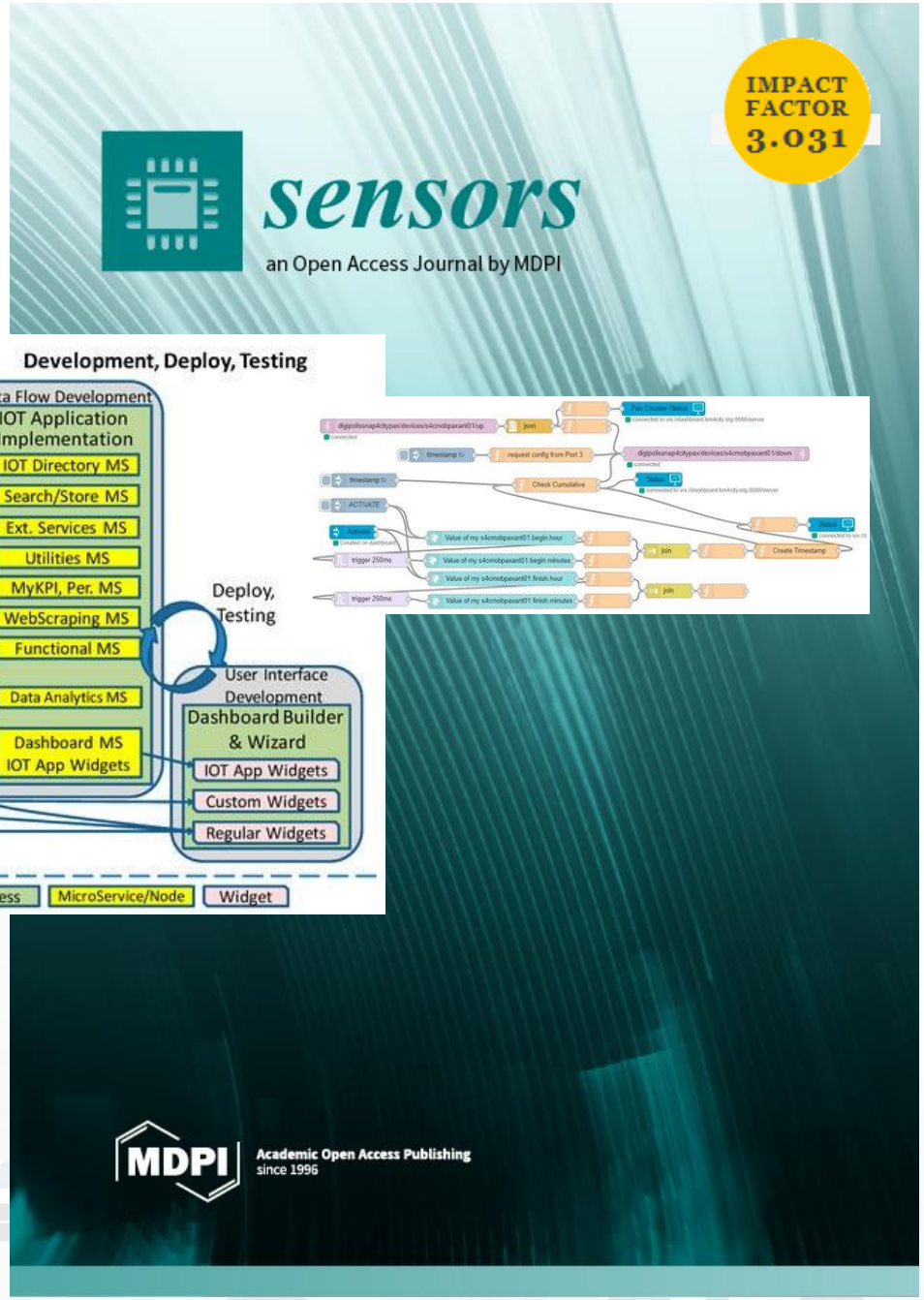
Self Training articles

- C. Badii, P. Bellini, A. Difino, P. Nesi, "Smart City IoT Platform Respecting GDPR Privacy and Security Aspects", accepted for publication on IEEE Access, 2020. 10.1109/ACCESS.2020.2968741 https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8966344
- C. Badii, P. Bellini, A. Difino, P. Nesi, G. Pantaleo, M. Paolucci, MicroServices Suite for Smart City Applications, Sensors, MDPI, 2019. https://doi.org/10.3390/s19214798
- C. Badii, P. Bellini, A. Difino, P. Nesi, "Sii-Mobility: an IOT/IOE architecture to enhance smart city services of mobility and transportation", Sensors, MDPI, 2019. https://doi.org/10.3390/s19010001 https://www.mdpi.com/1424-8220/19/1/1/pdf
- See also courses in ITALIANO: https://www.snap4city.org/485

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

MicroServices Suite for Smart City

- Badii, C.; Bellini, P.; Difino, A.; Nesi, P.; Pantaleo, G.; Paolucci, M. MicroServices Suite for Smart City Applications.
- *Sensors* **2019**, *19*, 4798.
- https://www.mdpi.com/1424-8220/19/21/4798/pdf



TOP

IOT App = Node-RED + Snap4City examples

FORGING & MANAGING OPEN AND FLEXIBLE WEB AND MOBILE APPS

DATA GATHERING AND CITY DATA KNOWLEDGE MANAGEMENT

IOT/IOE DEVICES AND NETWORKS

IOT APPLICATIONS, THE LOGIC AND THE SMARTNESS

SNAP4CITY LIVING LAB FOR COLLABORATIVE WORK

SNAP4CITY FOR BEGINNERS

DATA ANALYTICS, BUSINESS INTELLIGENCE, WHAT-IF AND SIMULATION

TWITTER VIGILANCE: SOCIAL MEDIA ANALYSIS

SNAP4CITY ARCHITECTURE AND ECOSYSTEM. OPENED TO DEVELOPERS

DECISION SUPPORT SYSTEM AND CITY RESILIENCE

HOW TO ADOPT SNAP4CITY, AND OUR ROADMAP

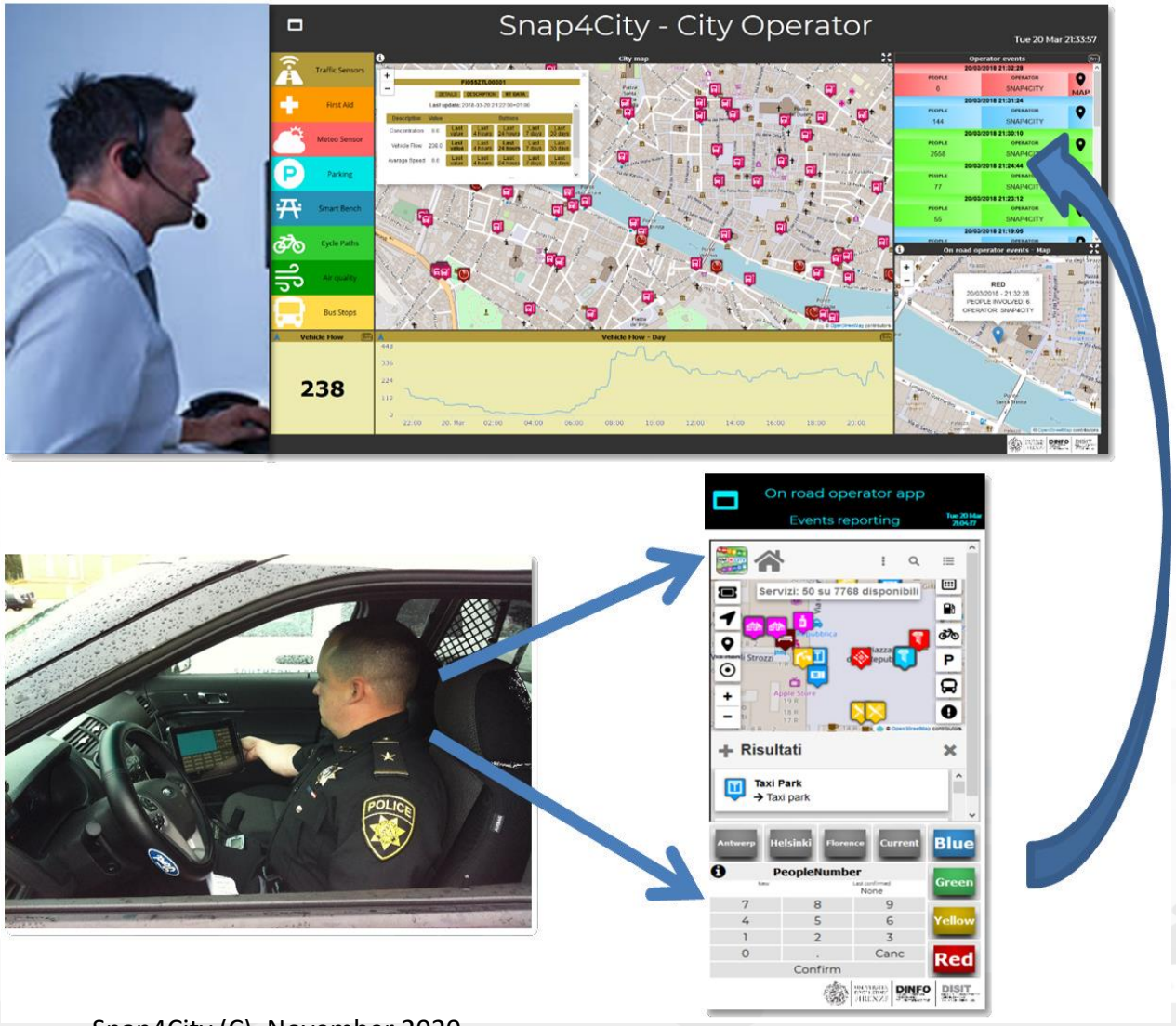
SNAP4CITY AND KM4CITY

Node-RED

Alerting about critical events involving people in a specific area

A public operator (Road Operator) on the field, like a policeman or a public transport driver, **notifies to a control room** operator (City Operator) a **critical event** in the city.

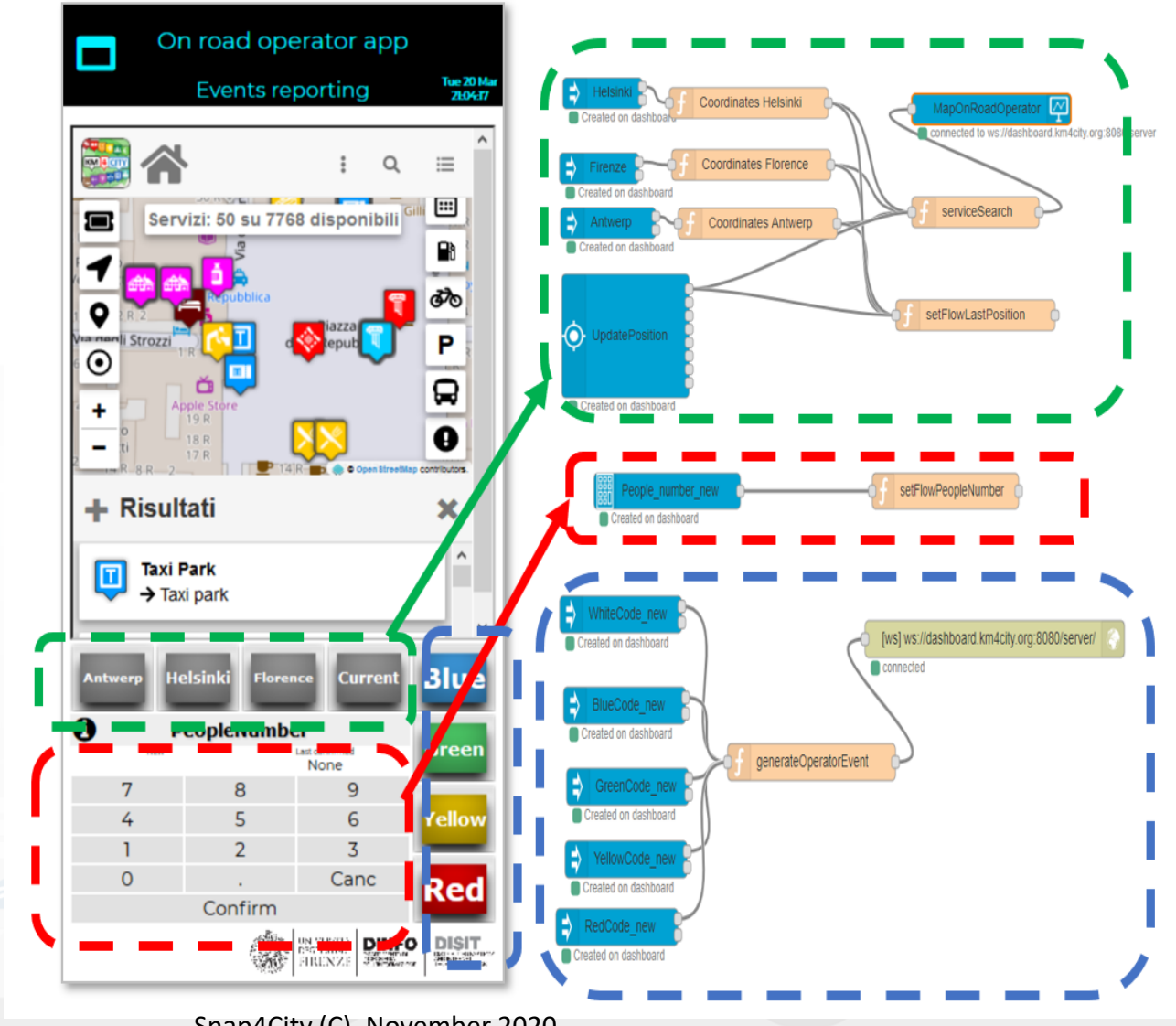
The notification includes the reporting in real time of the **event position, the number of involved people** and the **gravity** of the event.



Alerting about critical events involving people in a specific area

A public operator (Road Operator) on the field, like a policeman or a public transport driver, **notifies to a control room** operator (City Operator) a **critical event** in the city.

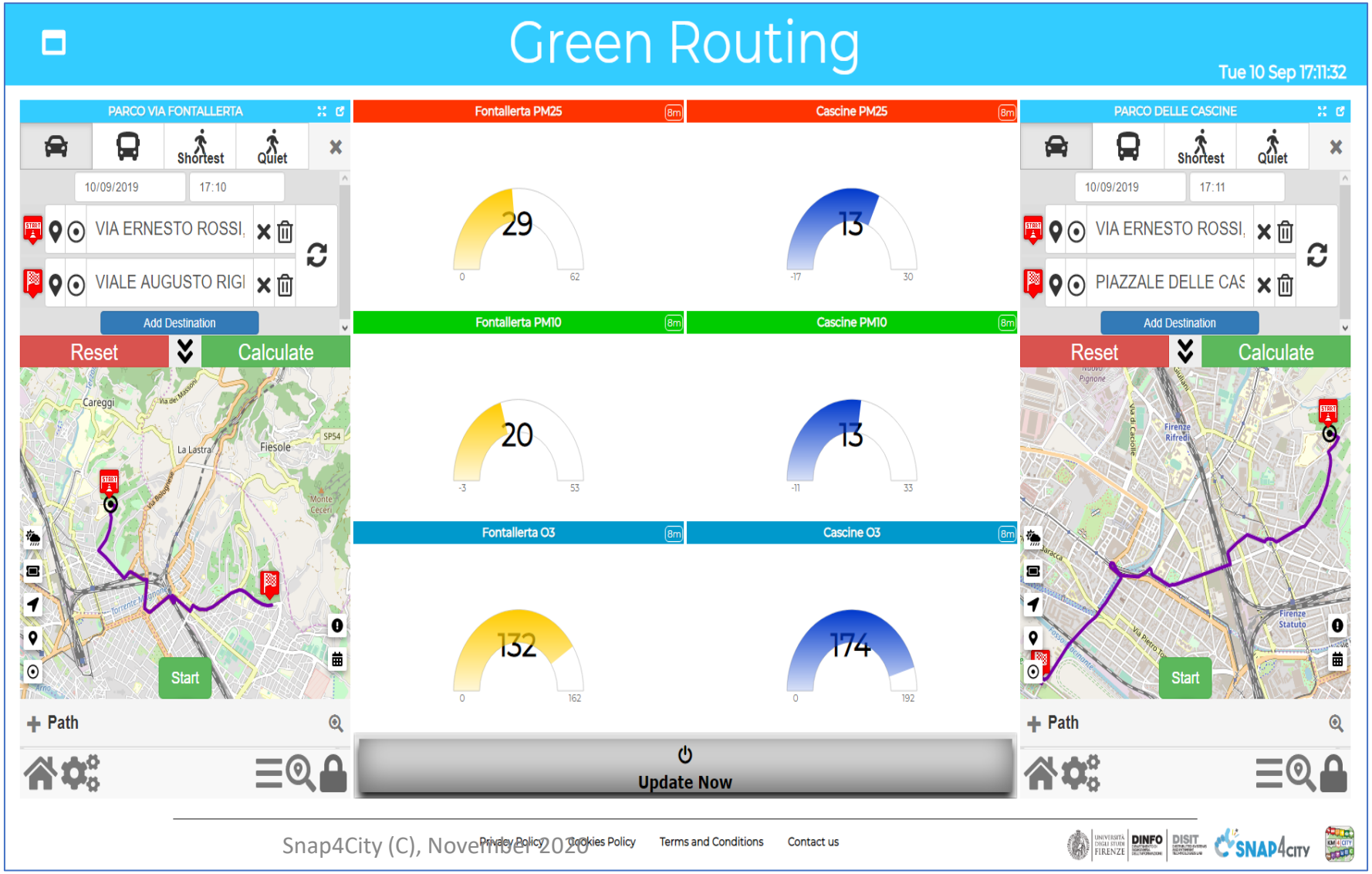
The notification includes the reporting in real time of the **event position, the number of involved people** and the **gravity** of the event.



Check which route is less polluted

In this example, microServices **retrieve information** from **the Smart City storage** and info to create a dashboard that tells the user which is the less polluted path at a precise moment to go jogging.

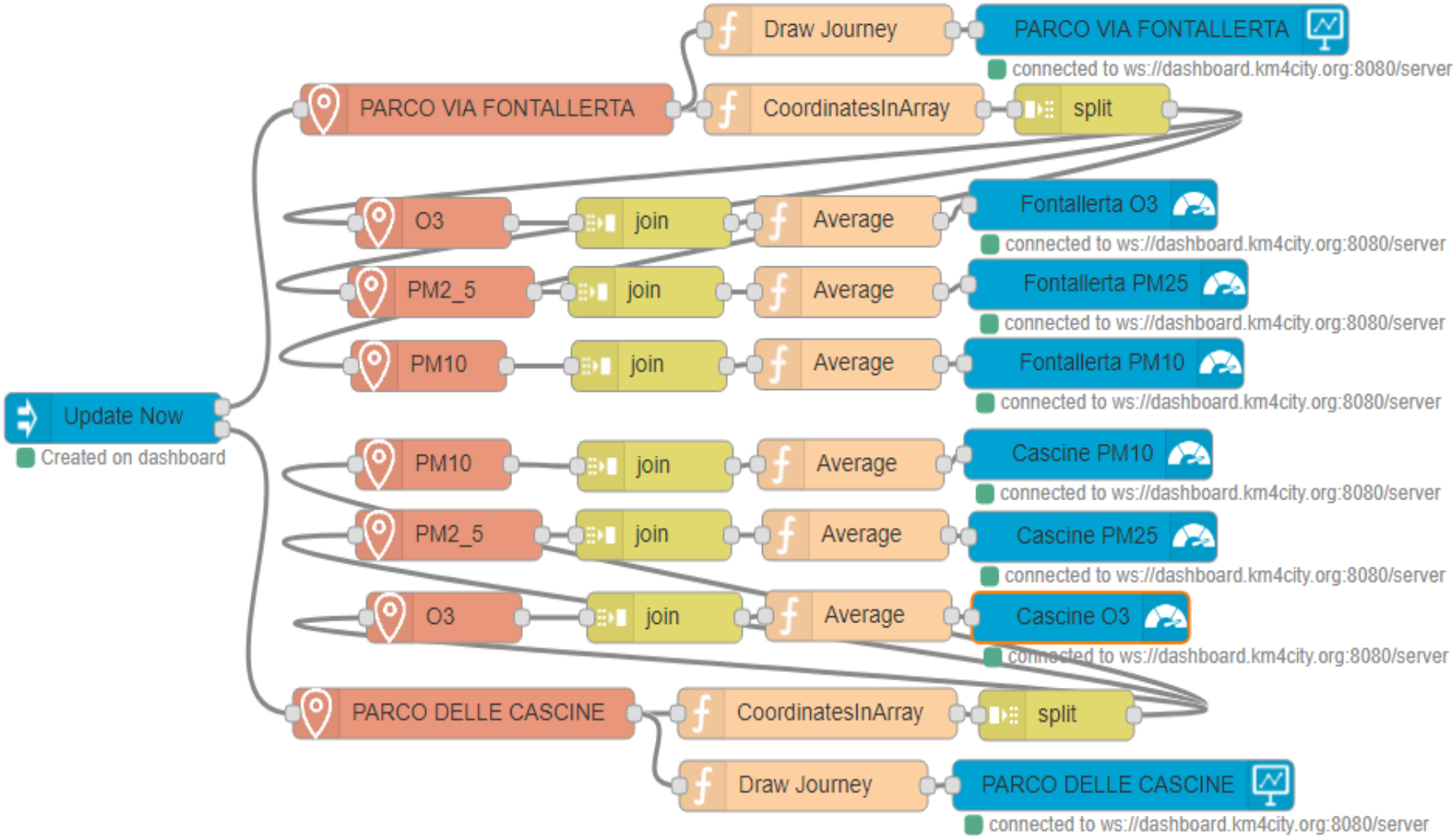
If predictive data are available, it can work on **predictions**



Check which route is less polluted

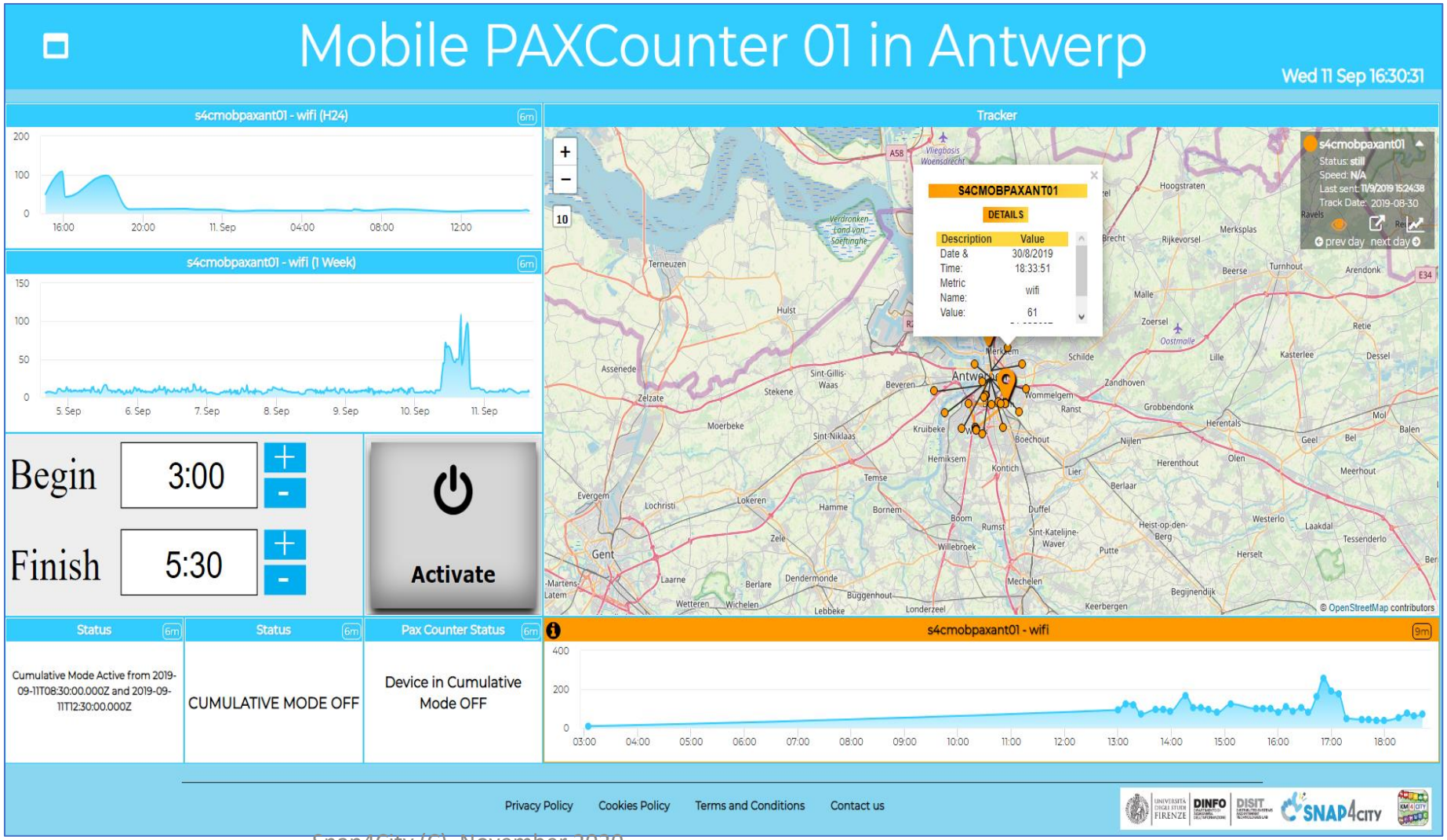
In this example, microServices **retrieve information** from **the Smart City storage** and info to create a dashboard that tells the user which is the less polluted path at a precise moment to go jogging.

If predictive data are available, it can work on **predictions**



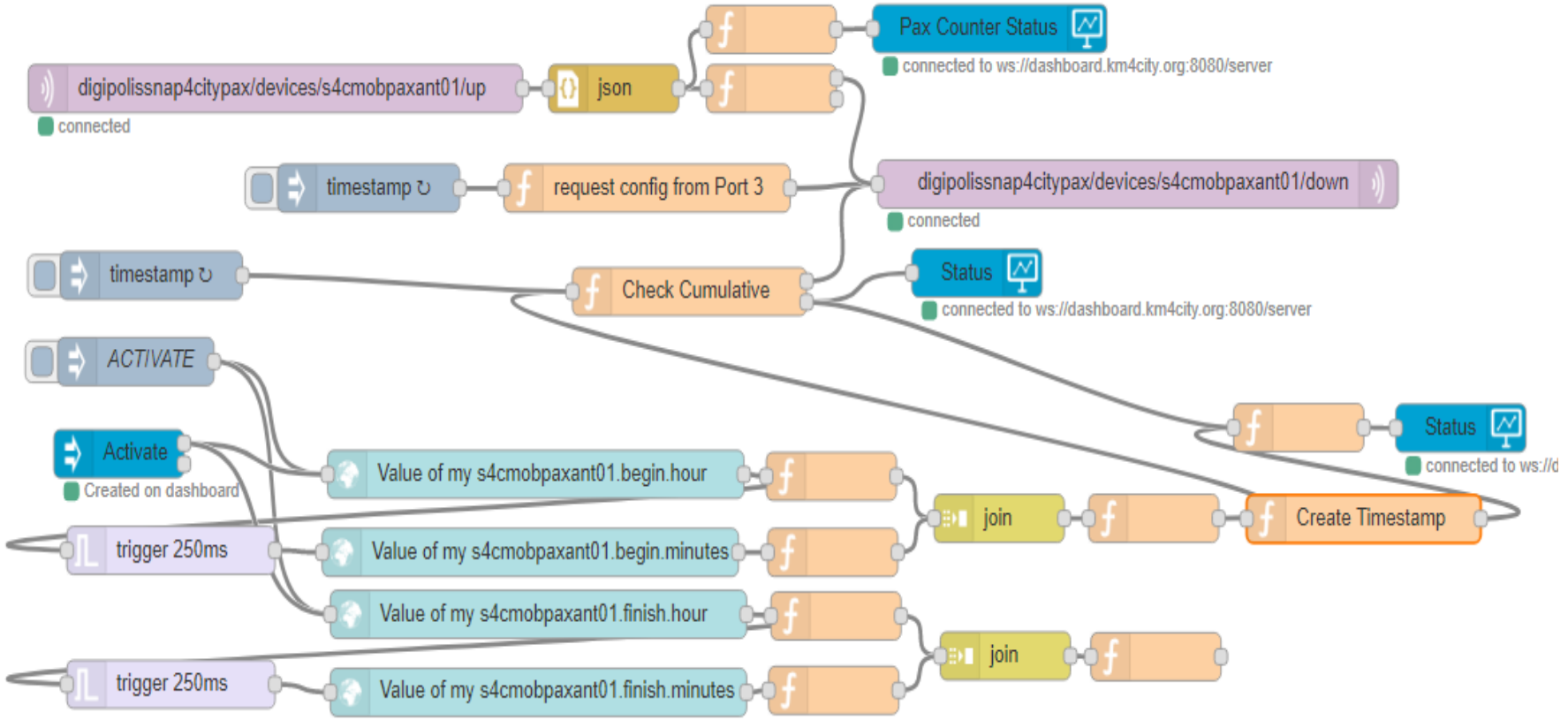
Controlling Personal Mobile PAX Counter

In this example, the **interaction with IOT Devices** counting people by using Wi-Fi and Bluetooth sniffing in its vicinity (according to GDPR)



Controlling Personal Mobile PAX Counter

In this example, the **interaction with IOT Devices** counting people by using Wi-Fi and Bluetooth sniffing in its vicinity (according to GDPR)



What we are going to do now!

- Create a Simple IOT Application (Demo)
- Production of IOT Application (Exercitation)
- Data Processing with IOT Application (Demo)
- Processing Data with IOT Applications (Exercitation)

Start DEMO Section 3

Start DEMO Section 3

Create a Simple IOT Application (DEMO)



Demo of Simple IOT Application

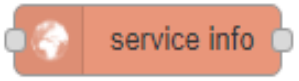
In this demo let's create an IOT Application that:

- reads a realtime value of a service and
- publishes it on a dashboard
- sends email to someone

Nodes for flow



Generates an input for the other nodes. It can be repeated at predefined intervals, entered manually and of various types (string, number, Boolean, json etc.)



Requests detailed information for a specific service on the platform (such as a car park, hotel, etc.)



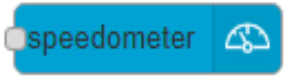
Executes a Javascript code once the input message is received



Transforms the incoming message into a JSON





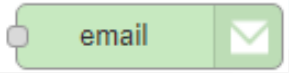




Display values in different modalities on a Dashboard (or on different Dash)

The node called single content accepts strings, numbers and html.

The others only accept numbers.

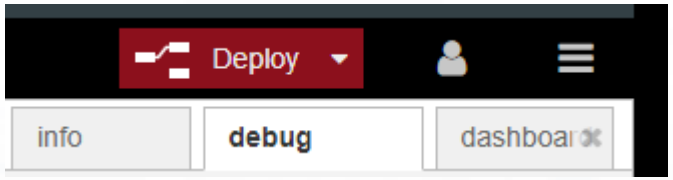


Send an email to the desired recipient. You must enter the username and password of an active email.

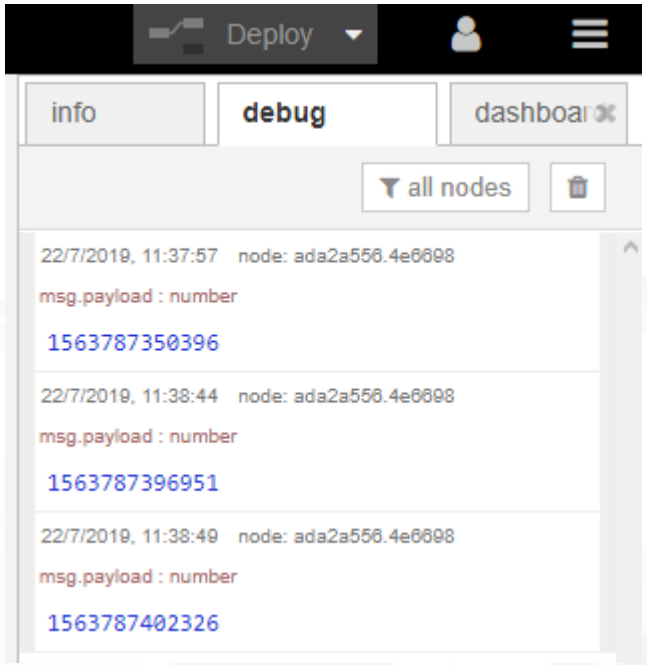
Step 1



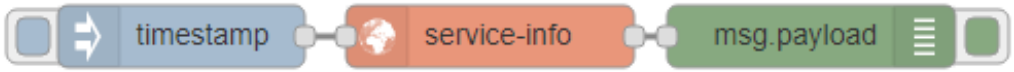
- Inject and Debug 

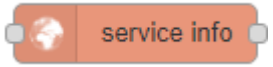
- Connect
- Deploy 

- Click and Observe
- Play with results



Step 2



- Service Info 

- Connect
- Configure

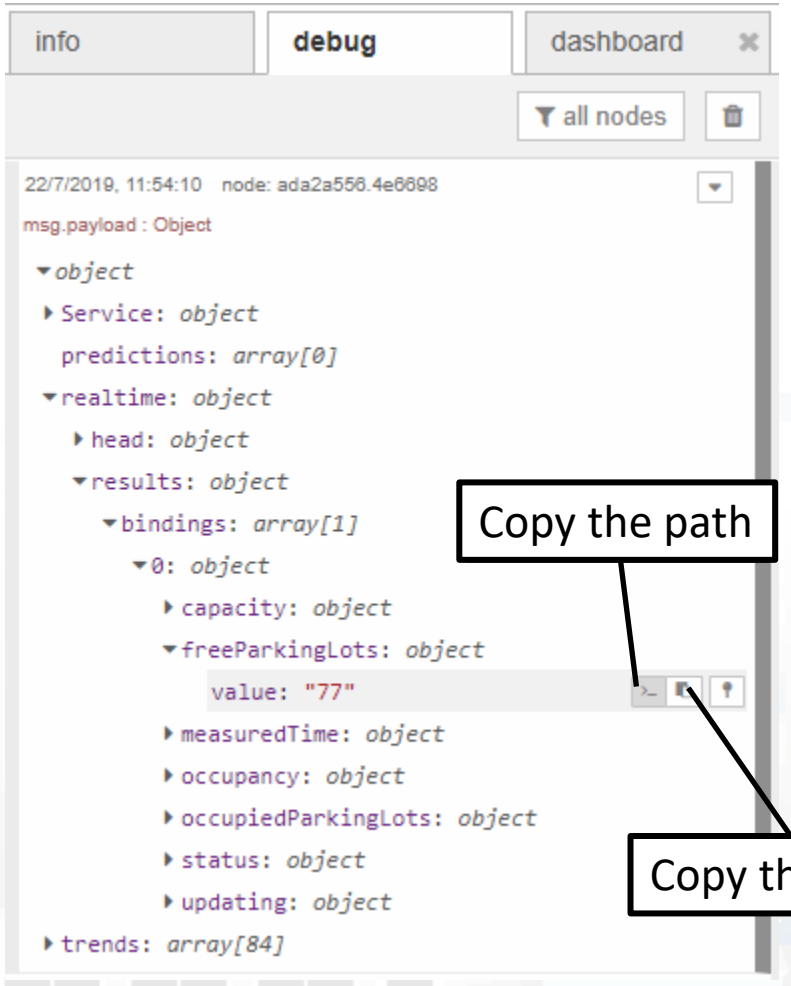
Name: Name

ServiceUri: http://www.disit.org/km4city/resource/CarParkPieracciniMeyer

Language: Italian

http://www.disit.org/km4city/resource/CarParkPieracciniMeyer

- Deploy
- Click and Observe
- Play with results

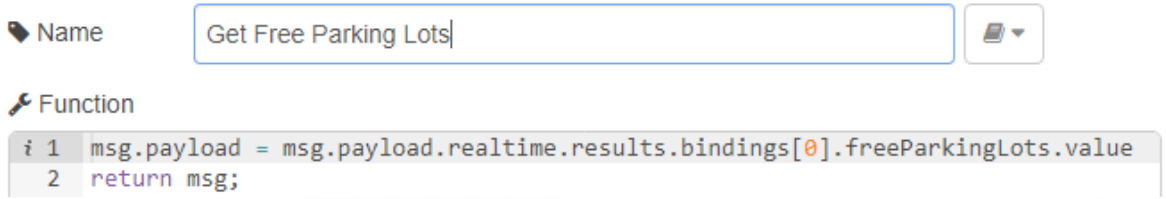


Step 3



- Function
- Connect
- Configure
- Deploy
- Click and Observe
- Play with results





msg.payload = msg.payload.realtime.results.bindings[0].freeParkingLots.value



Step 4

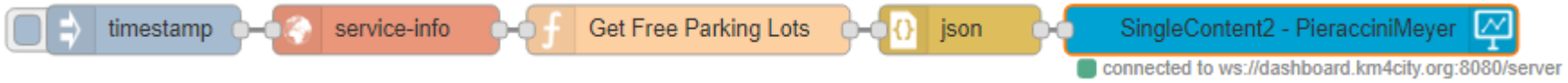


- JSON 

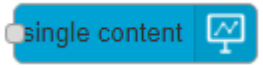
- Connect
- Deploy
- Click and Observe
- Play with results

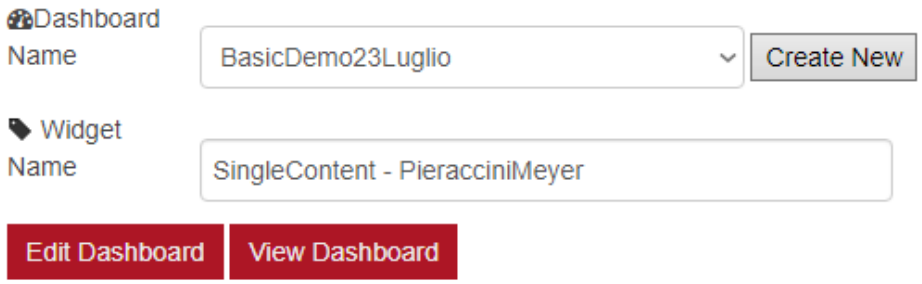


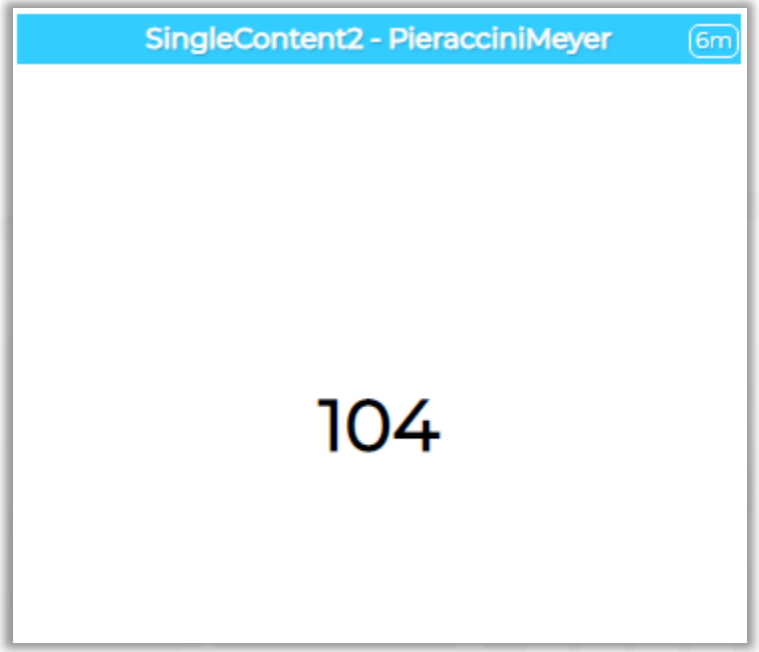
Step 5



- Single content
- Connect
- Configure
- Deploy
- Click and Observe
- Play with results





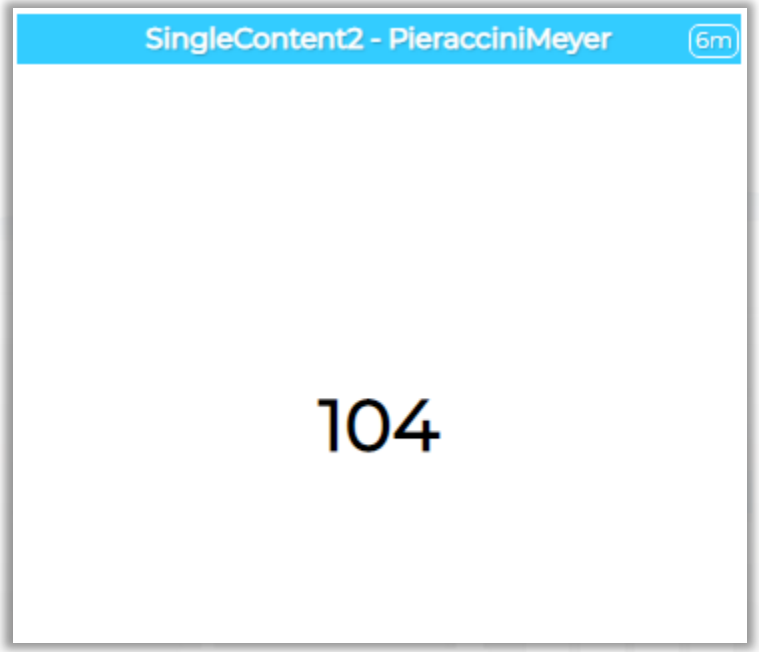


Step 6

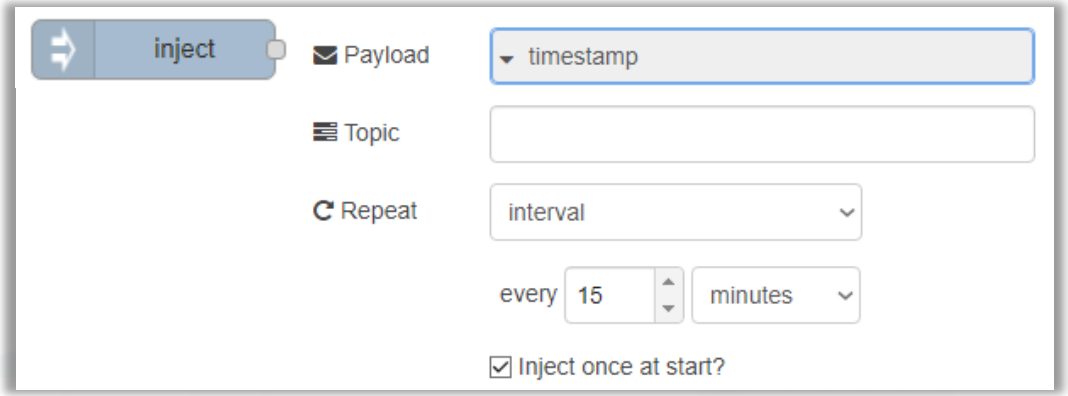


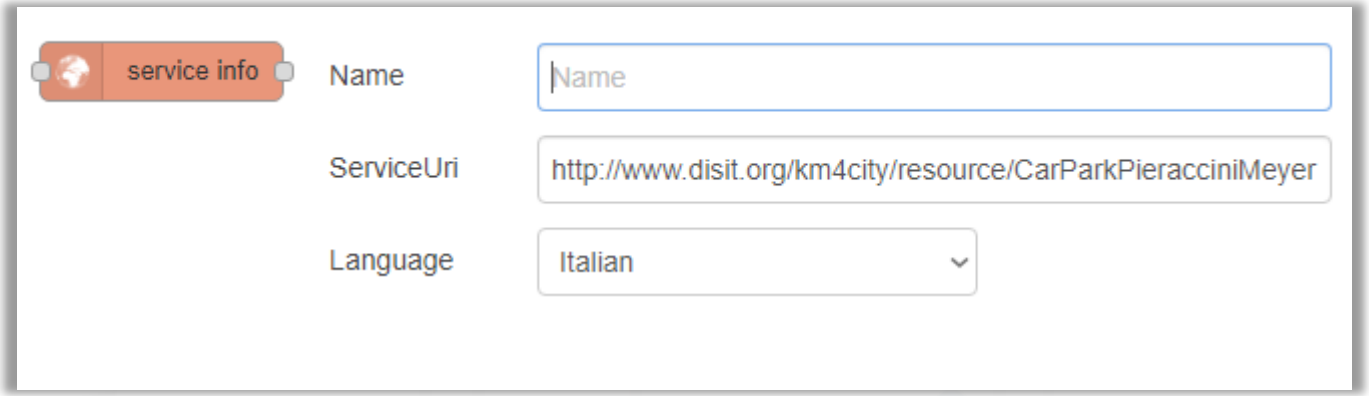
- Email
- Connect
- Configure
- Deploy
- Click and Observe
- Play with results

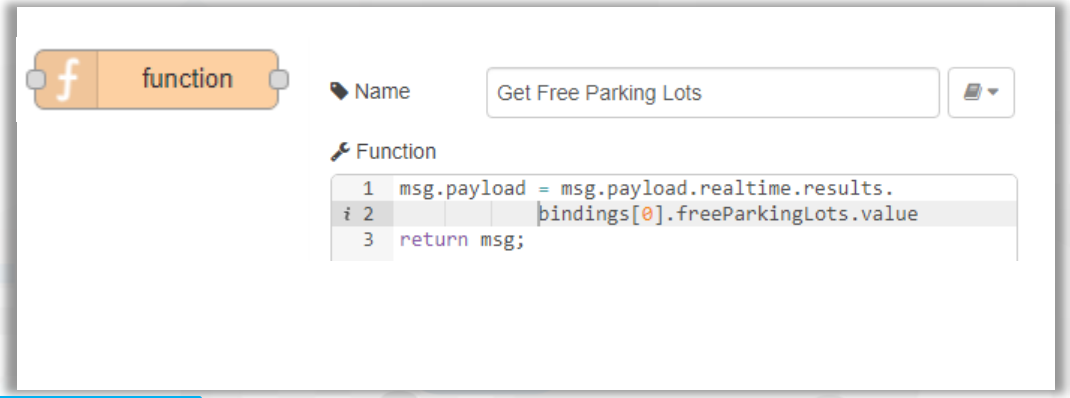


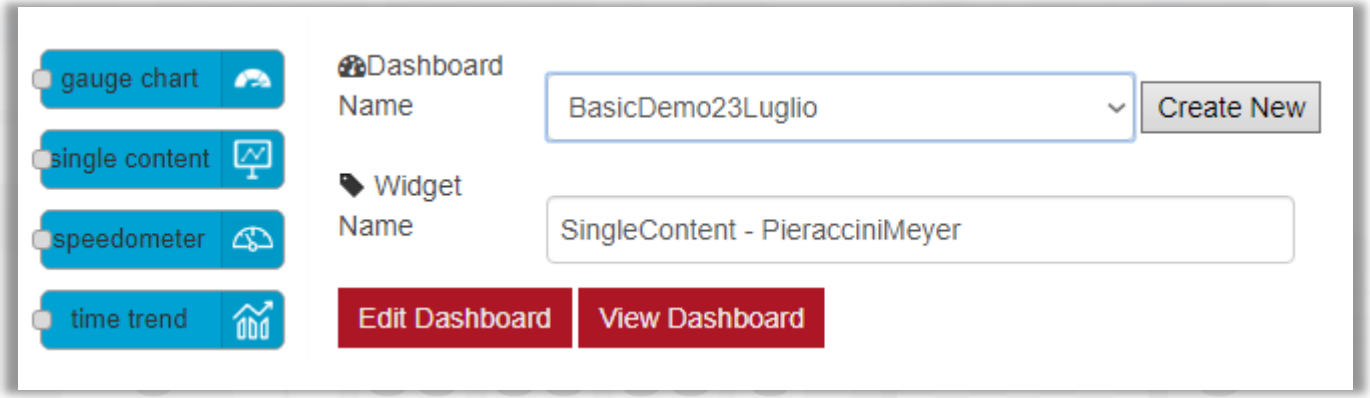


Nodes configuration









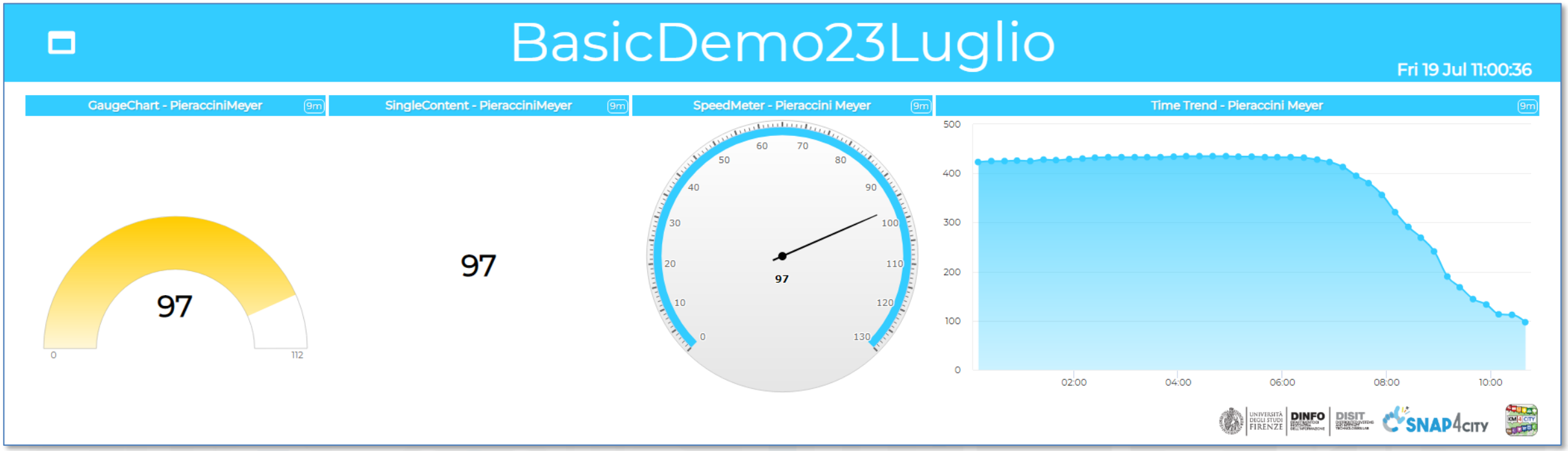
Nodes connections



Explaining: IOT Application Flow

- On Click or Every 15 minutes the ***timestamp*** node sends a message to the ***service-info*** node.
- When the message arrives, a request is sent to get details of the service URI entered in the configuration, in this case the ***Pieraccini Meyer car park***.
- The details are sent to the node named ”***Get Free Parking Lots”***, which recovers the value of the current free places and ignores all the other data received in response.
  - The values in output of node ***Get Free Parking Lots*** is a string.
- THUS ! node ***json*** may transform it into a number (for those who know JavaScript could be used function *parseInt()* inside the function node). Then a number has been obtained!
- The Number can be sent to Different kinds of nodes to show it on Dashboards Widgets.

Resulting Dashboard



https://main.snap4city.org/view/index.php?iddasboard=MTk1OQ==

end DEMO Section 3

Production of IOT Applications Exercitation



IOT Application Exercitation

Goal:

Create an IOT App (flow) that reads a value from a service (for example the parking lot seen in the previous demo)

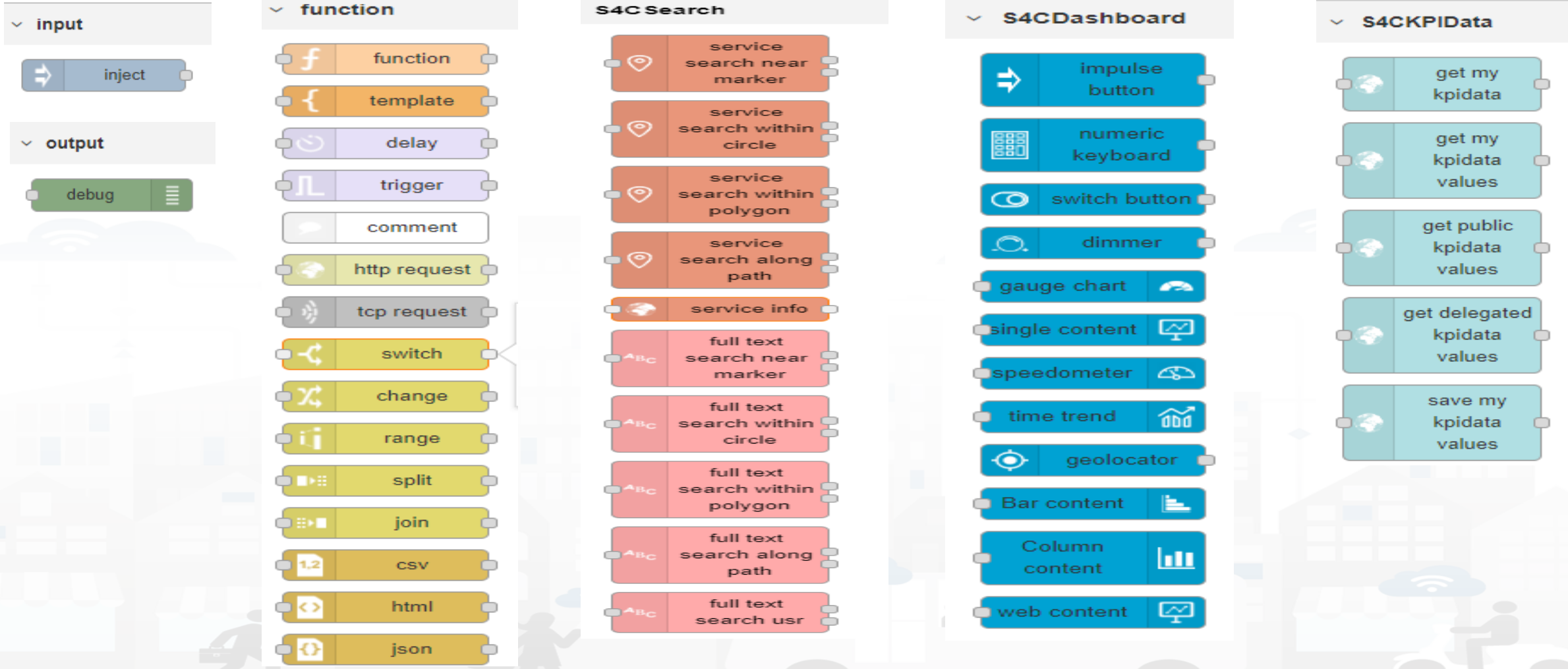
serviceUri: http://www.disit.org/km4city/resource/CarParkPieraccinMeyer

and:

based on a certain threshold sends a different message on the dashboard. For example, **Almost Full Parking or Free Parking**.
OR Send to **you an email** ☺ !

You have 15 Minutes!

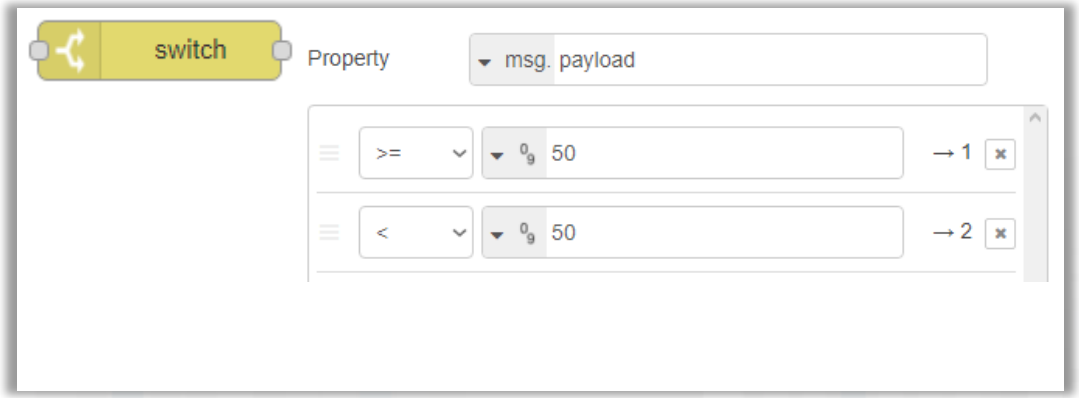
Ex1: Your NickName: ……



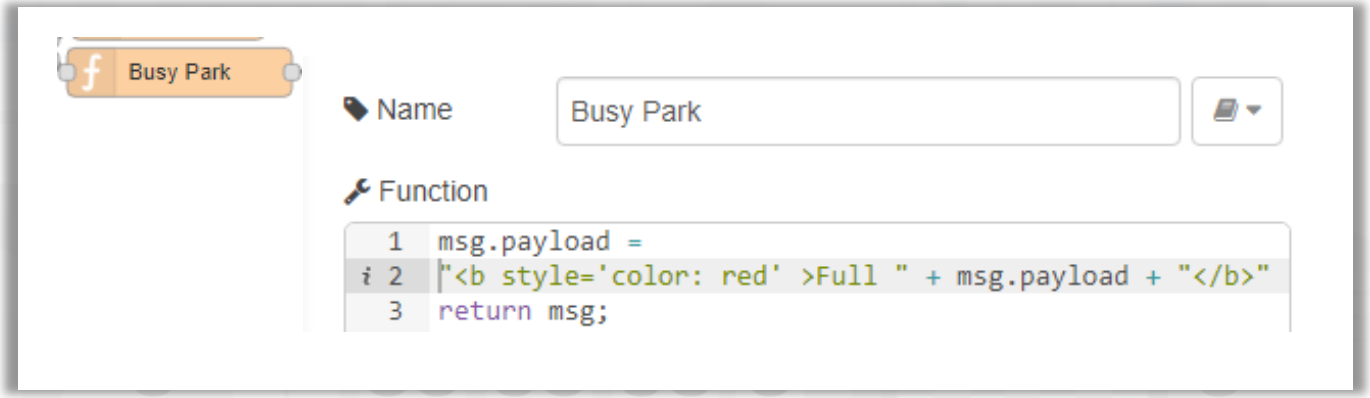
One Possible Solution



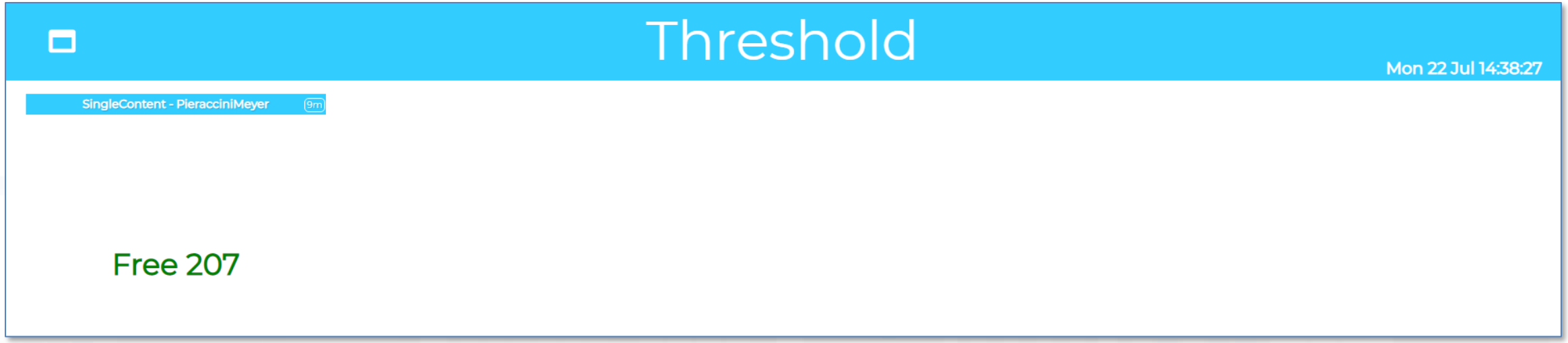
Nodes configuration







Resulting Dashboard



https://main.snap4city.org/view/index.php?iddasboard=MTk2MQ==

Start DEMO
Section 4

Start DEMO Section 4

Data Processing with IOT Application (DEMO)



Example of more Complex IOT Application

In this demo let's create an IoT Application that:

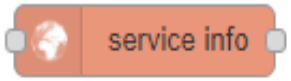
- reads a realtime values from a list of services,
- makes the sum of the value and
- publish the result on a dashboard



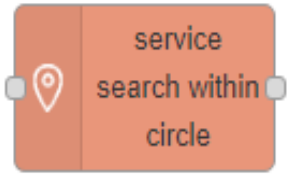
Nodes for flow 1/2



Generates an input for the other nodes. It can be repeated at predefined intervals, entered manually and of various types (string, number, Boolean, json etc)



Requests detailed information for a specific service on the platform (such as a car park, hotel, etc.)

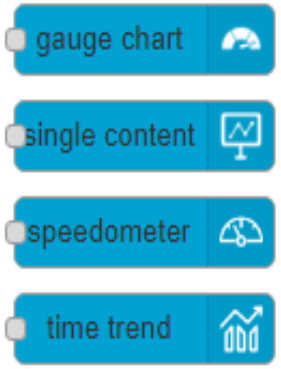


Search in around a certain point of the indicated service. It returns:

- servicesUri of all the services found,
- a GeoJSON containing a minimum of information about the services found, including the coordinates and the name of the service.



Executes Javascript code. For example, exploiting data arrived on input message and producing an output message in JSON

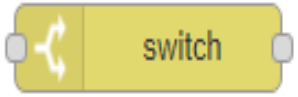


Display values in different modes on a dashboard. The node called single content accepts strings, numbers and html. The others only accept numbers.

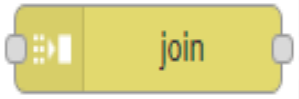
Nodes for flow 2/2



Divides the input message into multiple messages as indicated in the configuration.
If you have an array at the input, you can configure it to send each element of the array individually at the output.

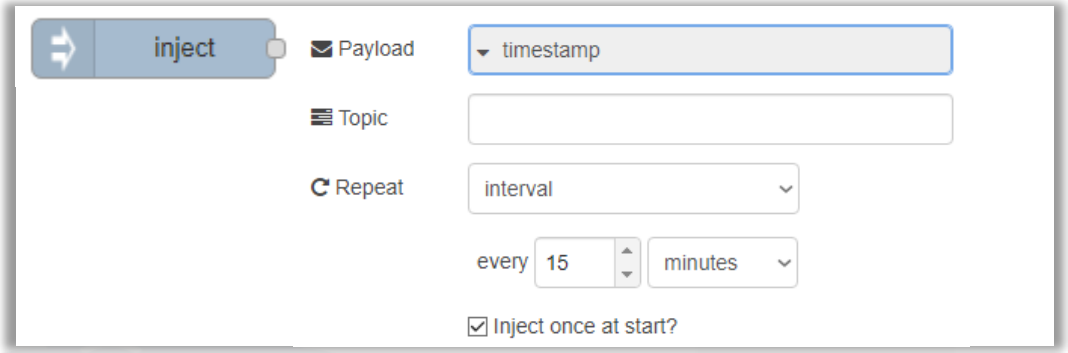


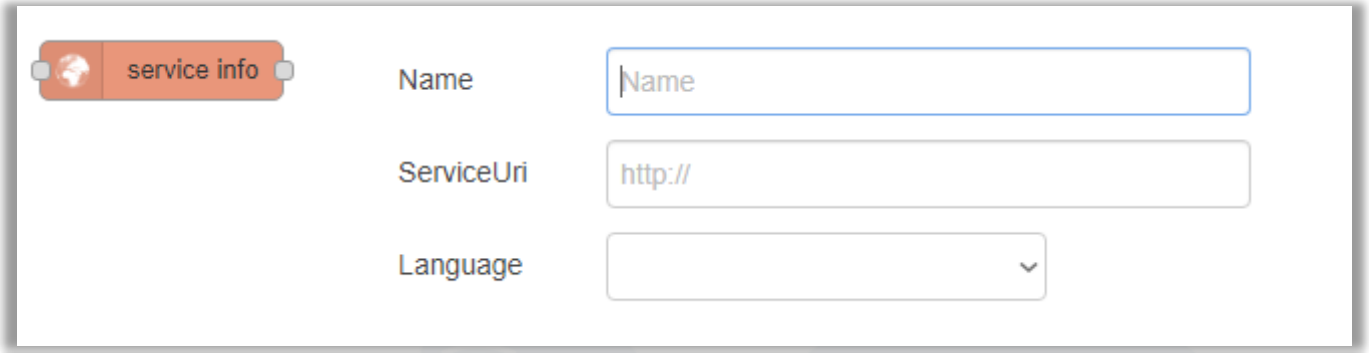
Treads the input message on possible different outputs based on a comparison made on the input message.

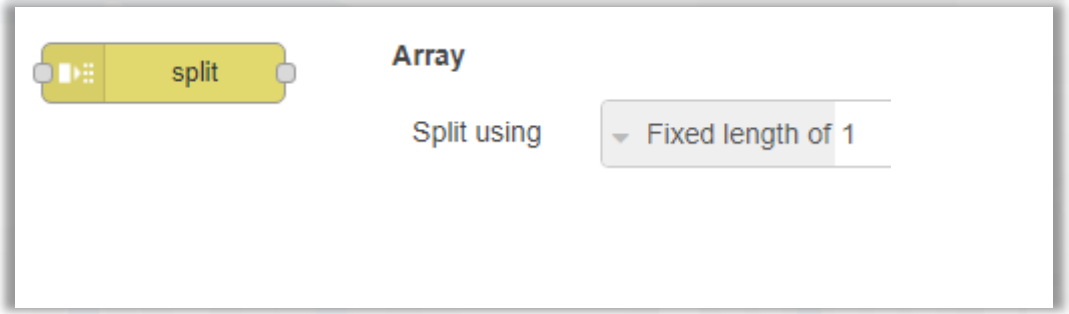


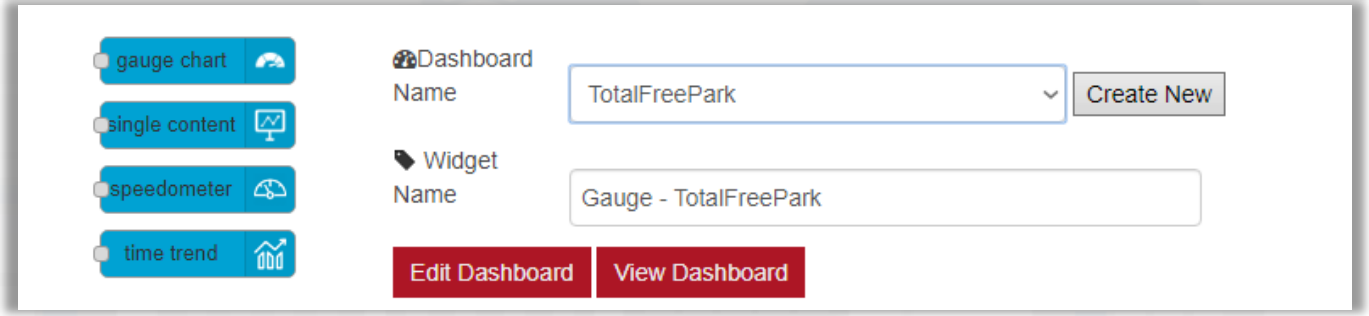
Operates in reverse order to the split. Joins the incoming messages in the mode indicated in the configuration.

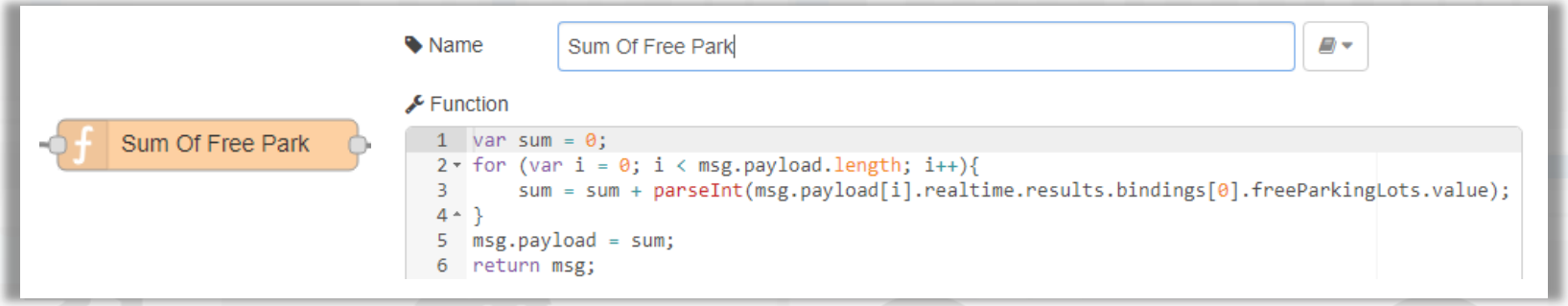
Nodes configuration 1/2



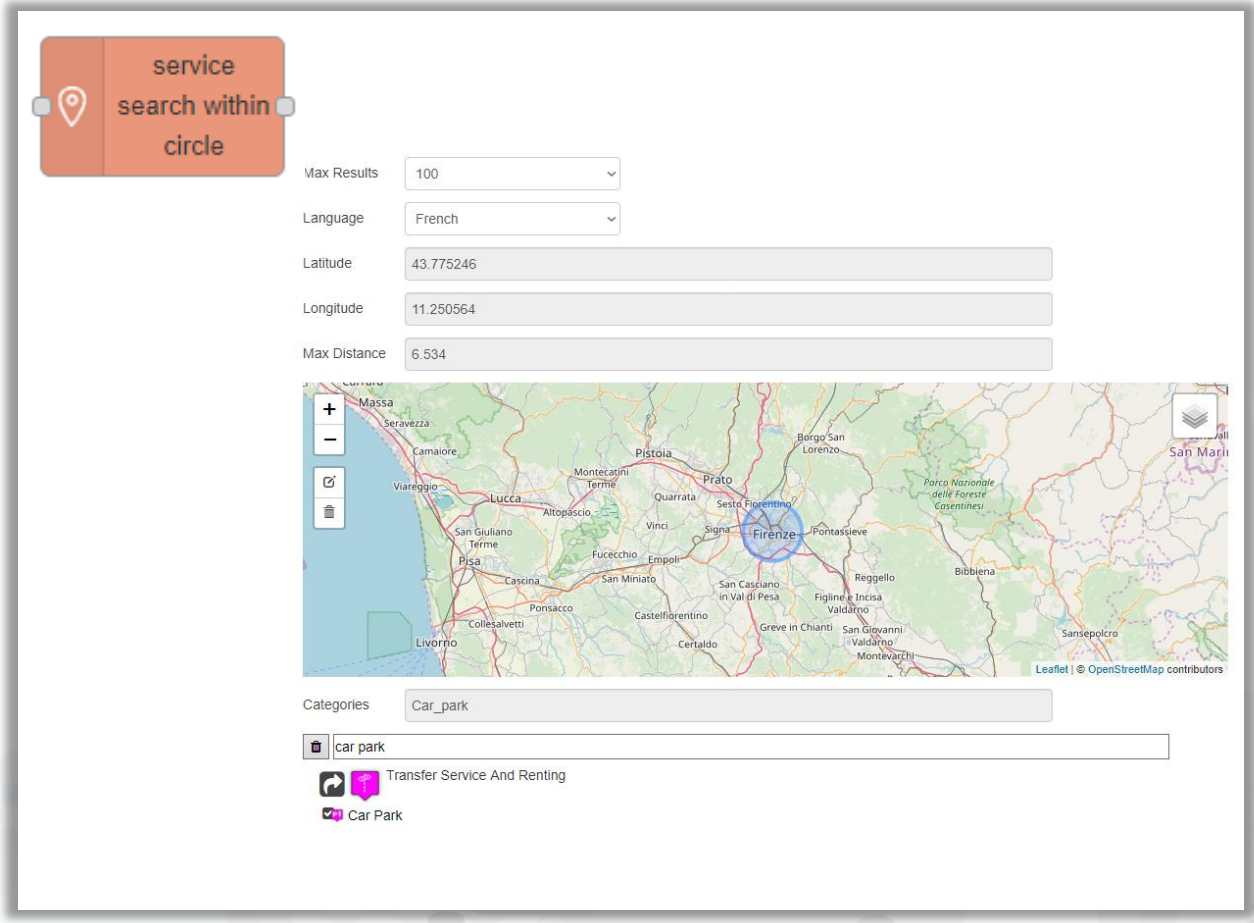


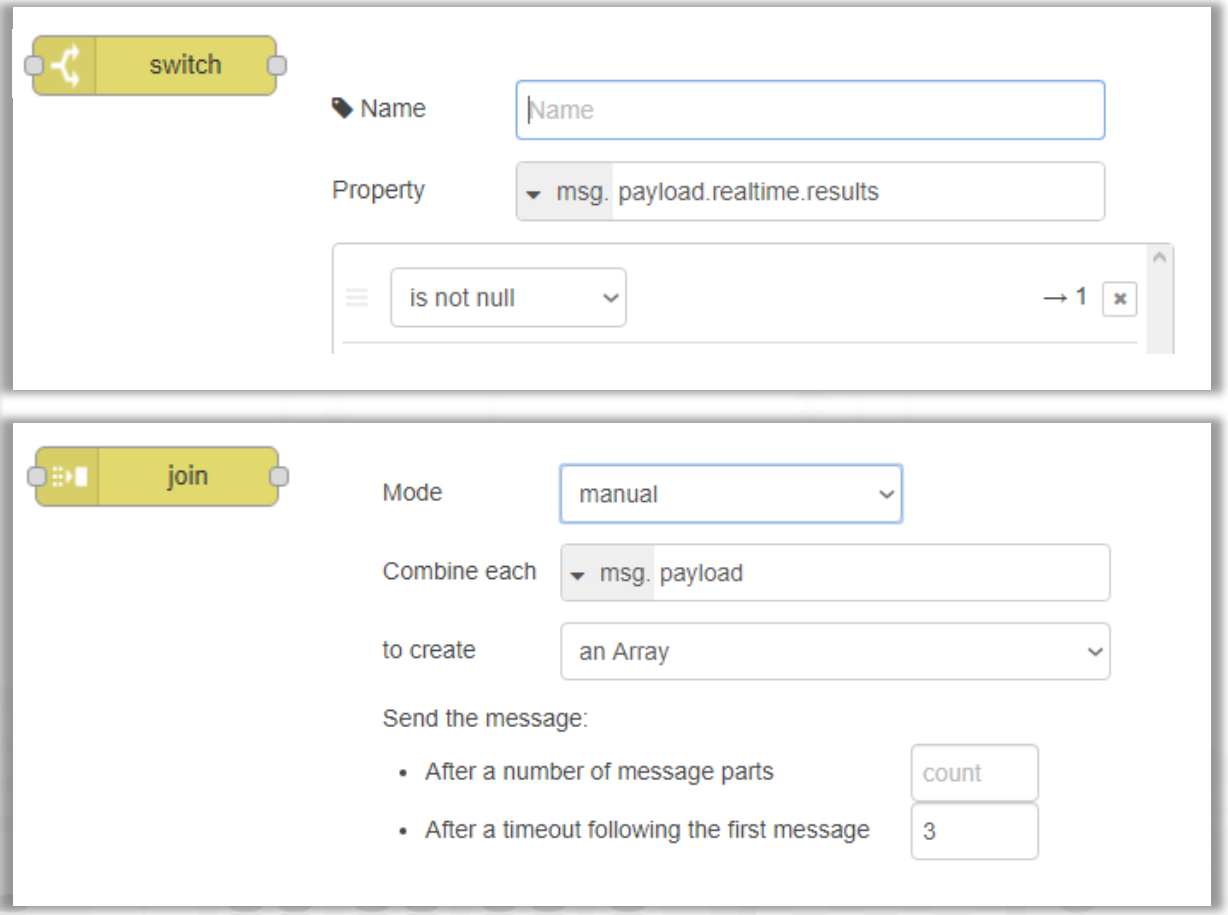




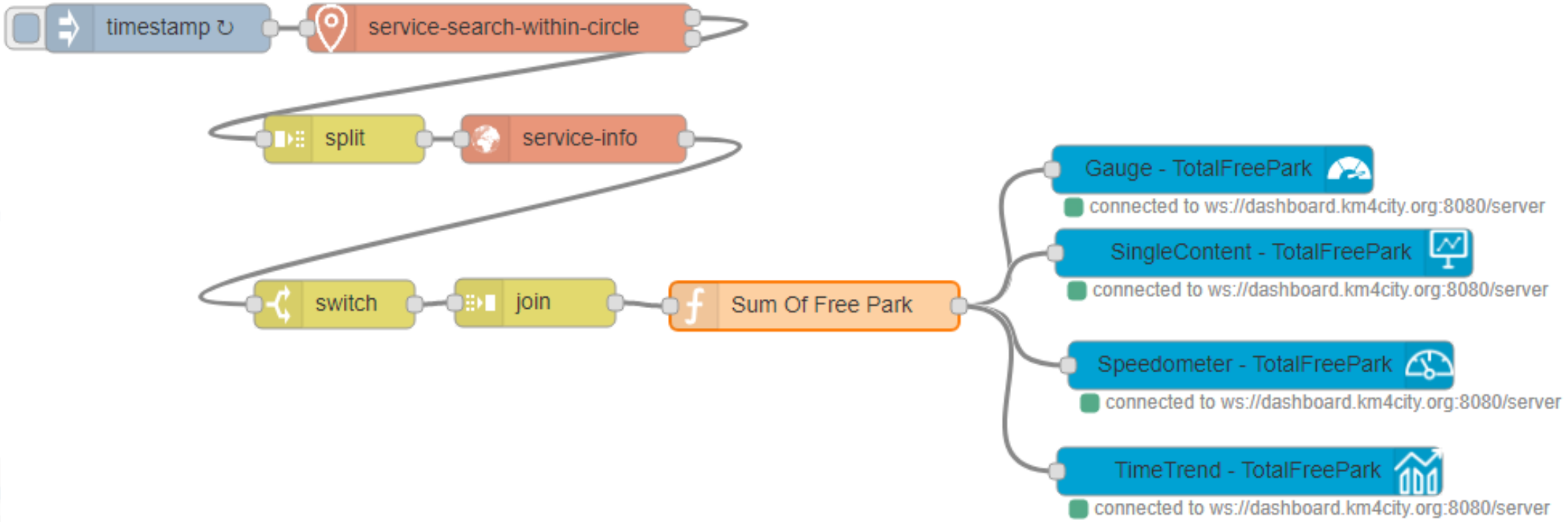


Nodes configuration 2/2

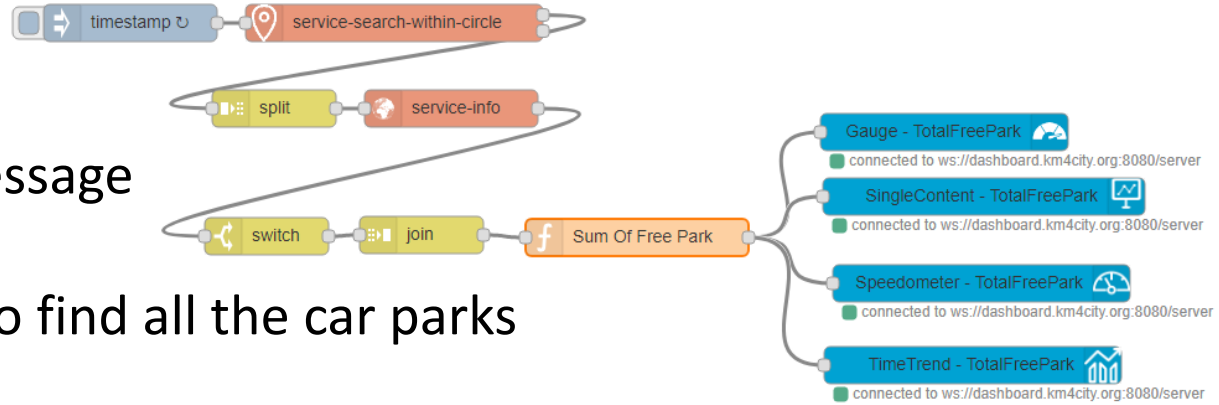




Nodes connections

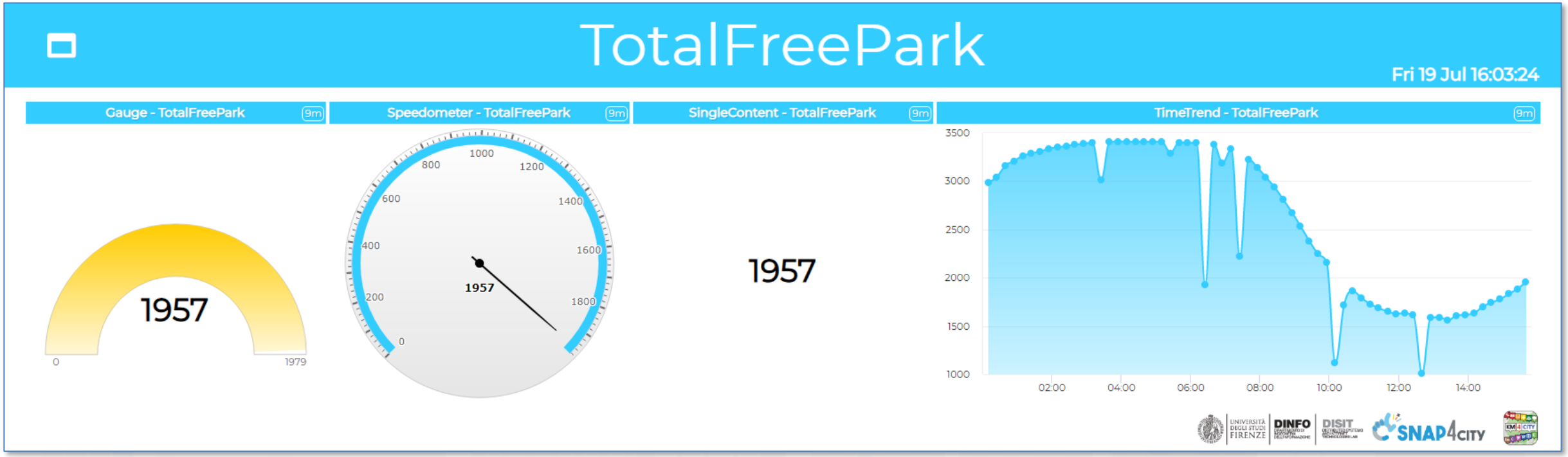


Nodes explanation 1/2



- Every 15 minutes the ***timestamp*** node sends a message to the ***service-search-within-circle node.***
  - When this message arrives, a request is sent to find all the car parks in the search area entered in configuration
- The first output of the ***service-search-within-circle*** node returns an array containing all the uri services of the car parks found. On such array we effect a ***split*** so that in input to ***service-info*** all the services uri arrive as distinct messages in a sequence.
- The configuration of the ***service-info*** node has not been filled because the URI service comes from the incoming message and is considered that URI service for retrieving service details.
- The ***switch*** and ***join*** nodes are used respectively to filter the results eliminating those parking lots that have no value in realtime (because for example that parking lot has no sensor) and bring together the various messages in a single array.
- On this array, node ***Sum of Free Park*** the perform the sum of the free places of all Florence parking and sent to the value to nodes representing Dashboard Widgets.
-

Result



https://main.snap4city.org/view/index.php?iddasboard=MTk2MA==

end DEMO Section 4

Processing data with IOT Applications (Exercitation)



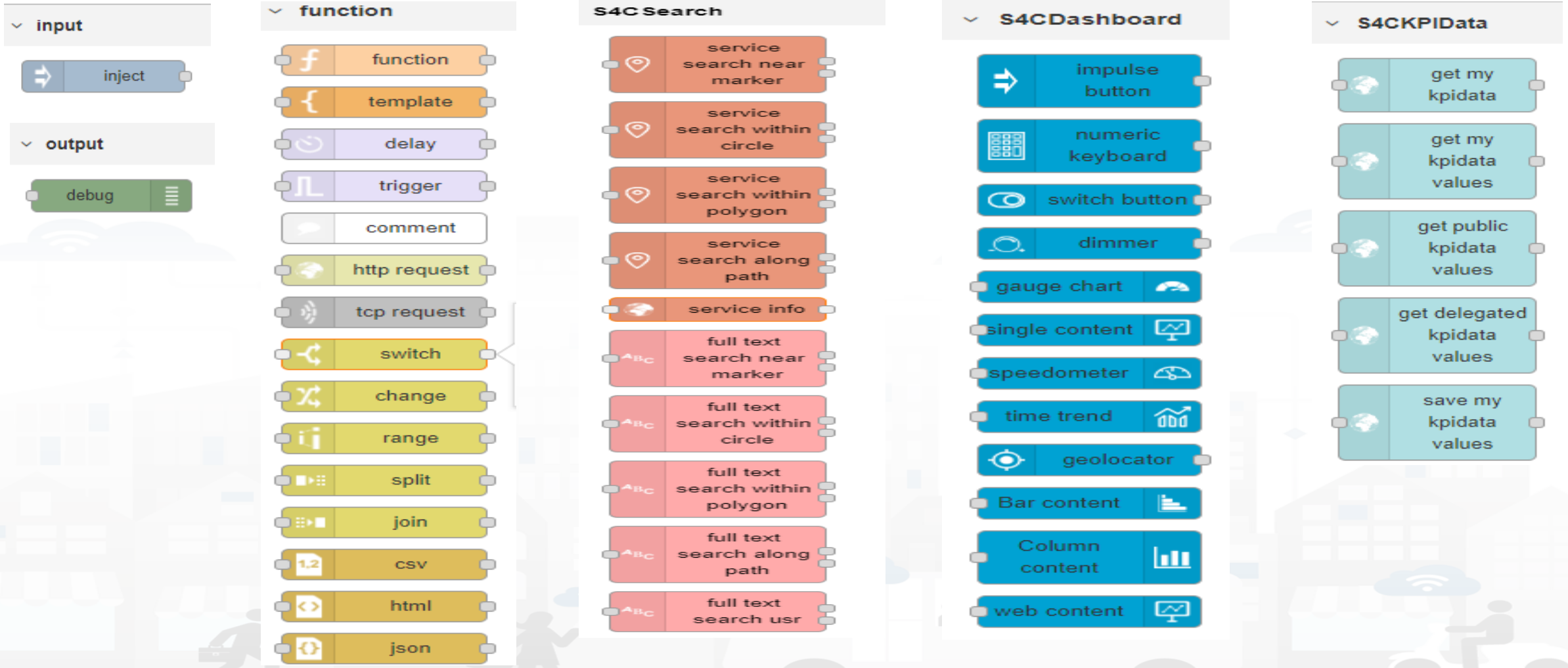
Average IoT Application

Create an IOT Application / flow that:

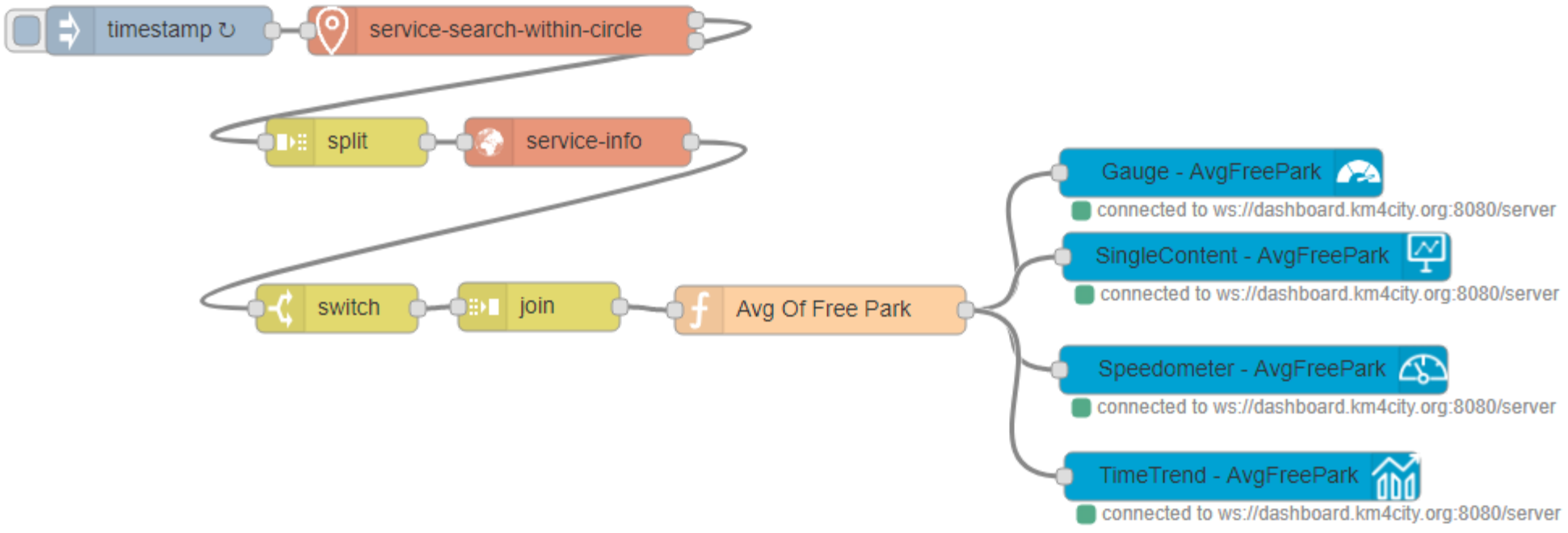
- reads a value from a list of service, for example the car parks in the Florence City Area, as seen in previous demo and
- calculates the average of Free Parking Lots and
- sends the value on a dashboard with the four possible nodes seen in the demo.

Execution Time: 20 Minutes

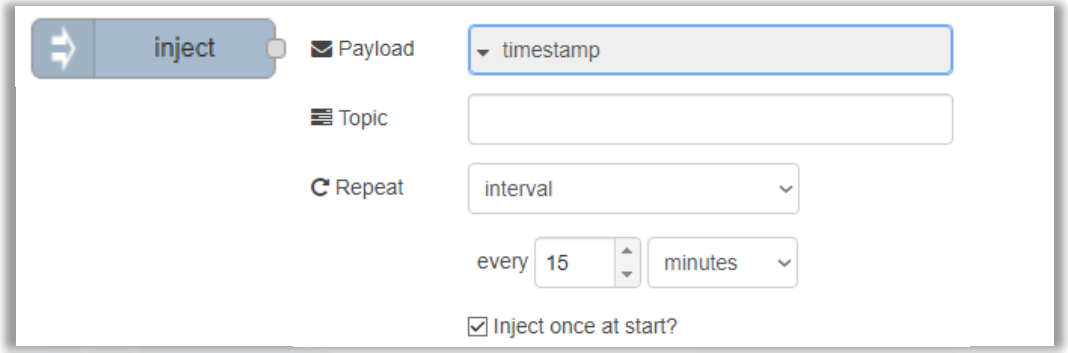
Ex2: Your NickName: ……

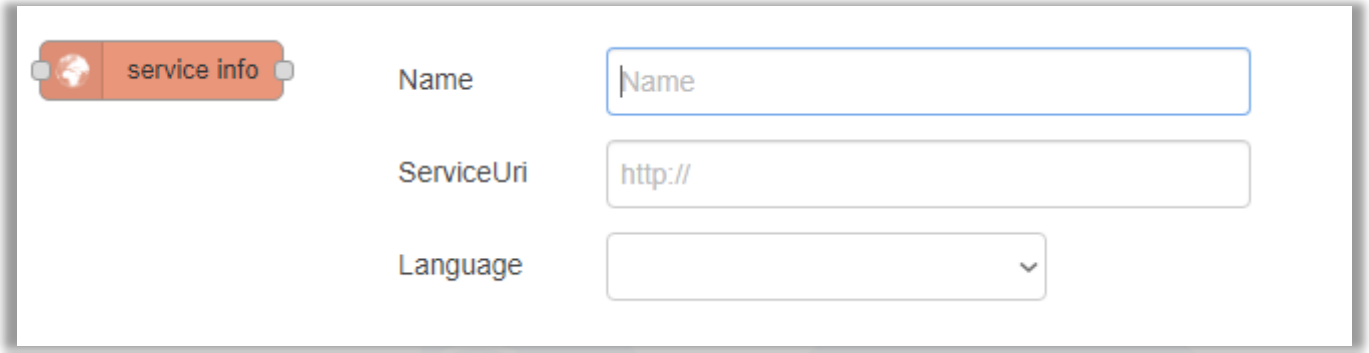


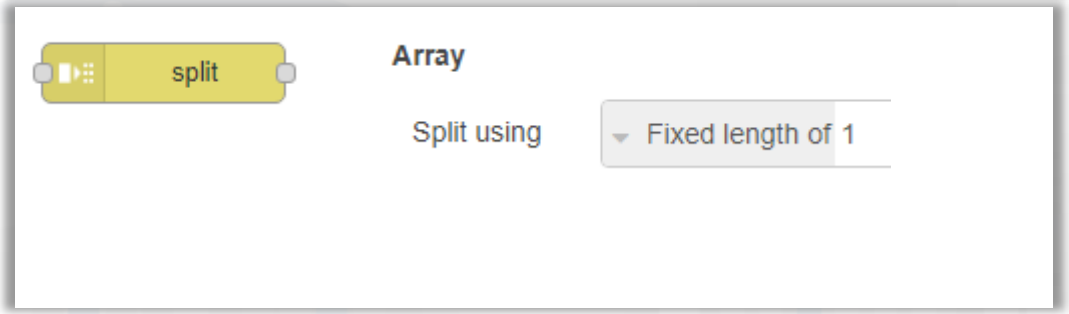
One Possible Solution

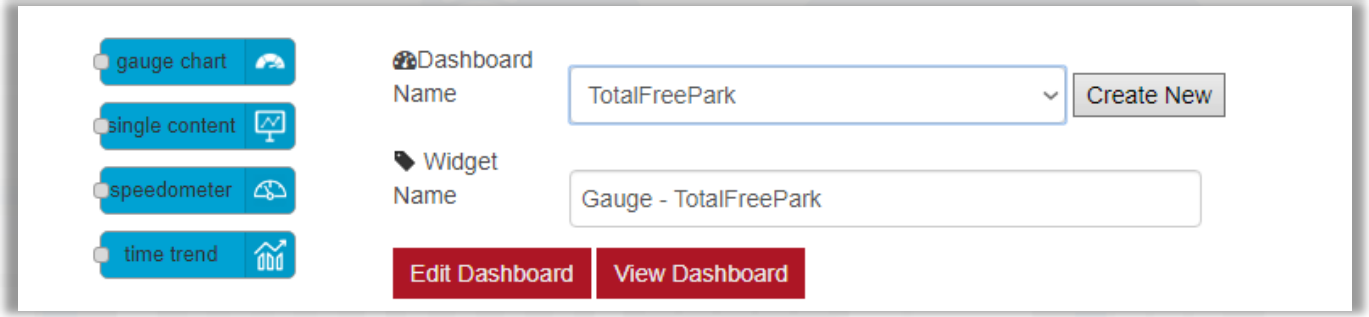


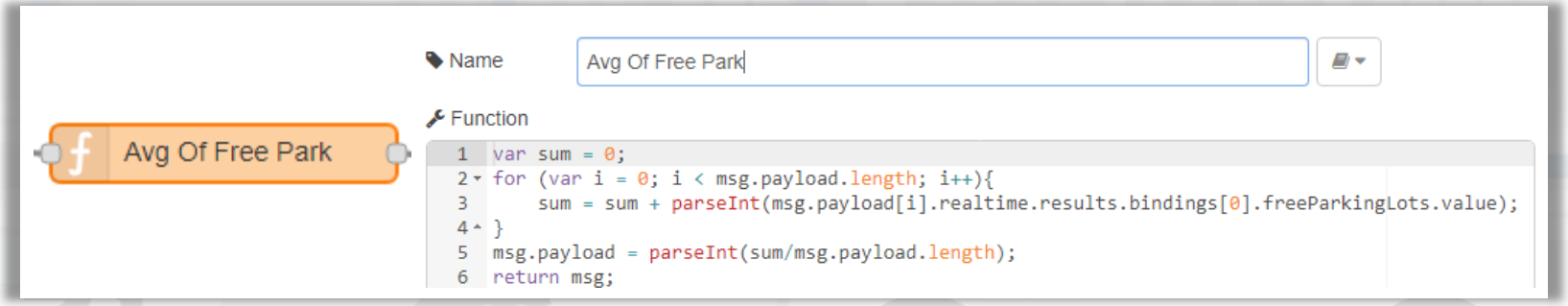
Nodes configuration 1/2



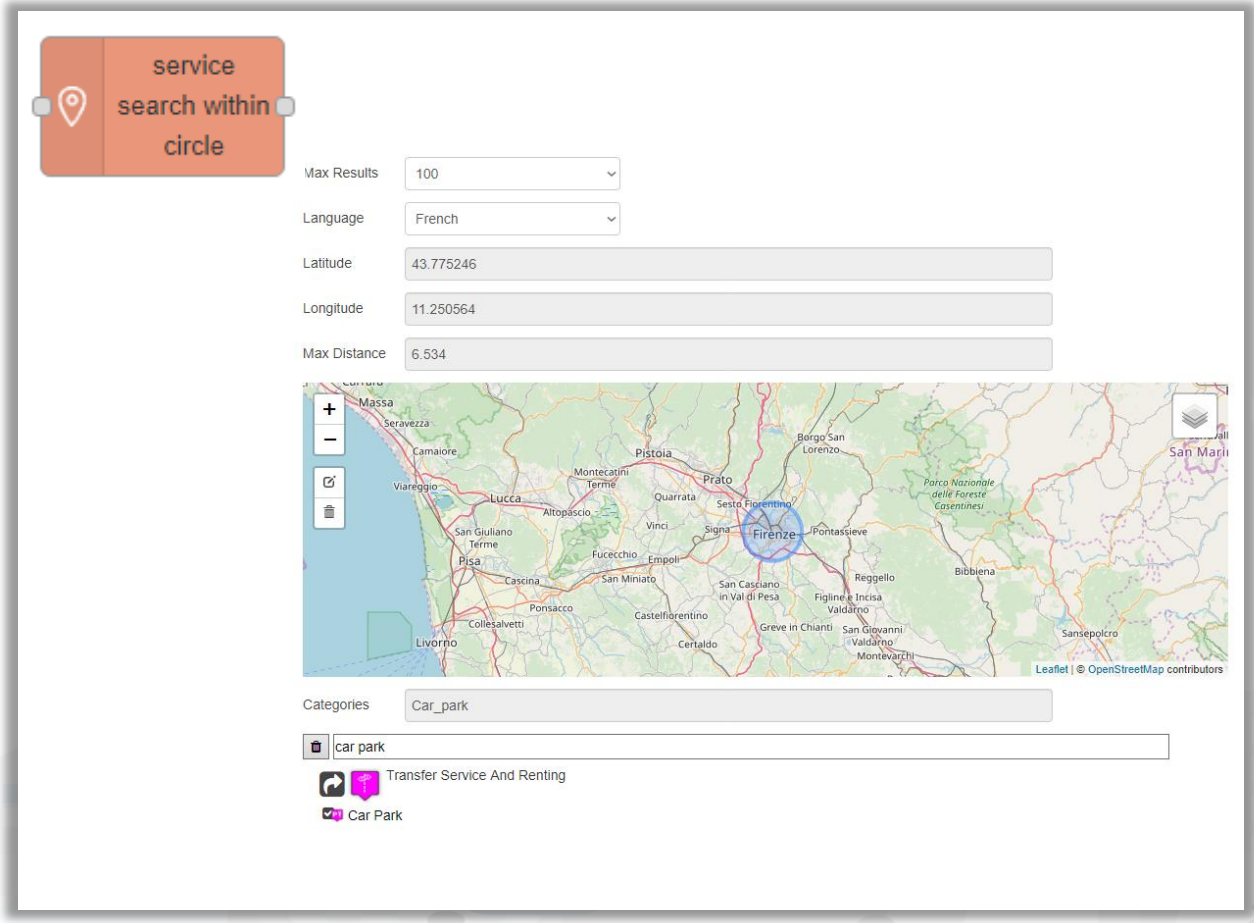


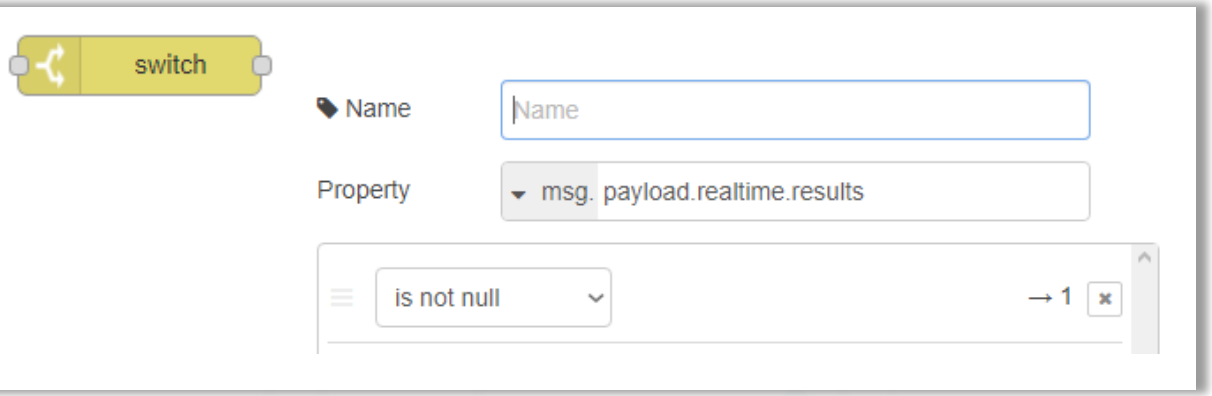


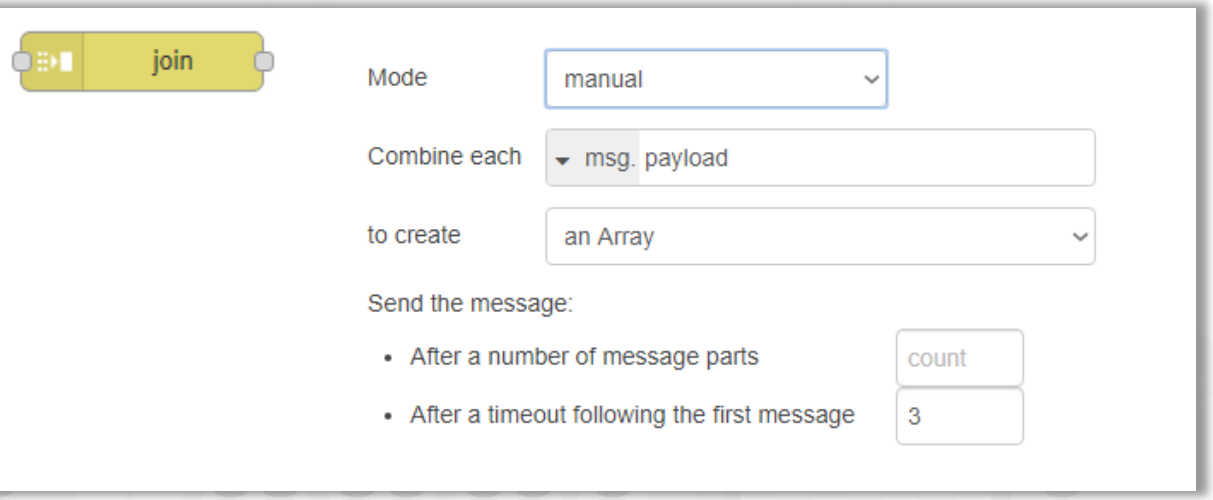




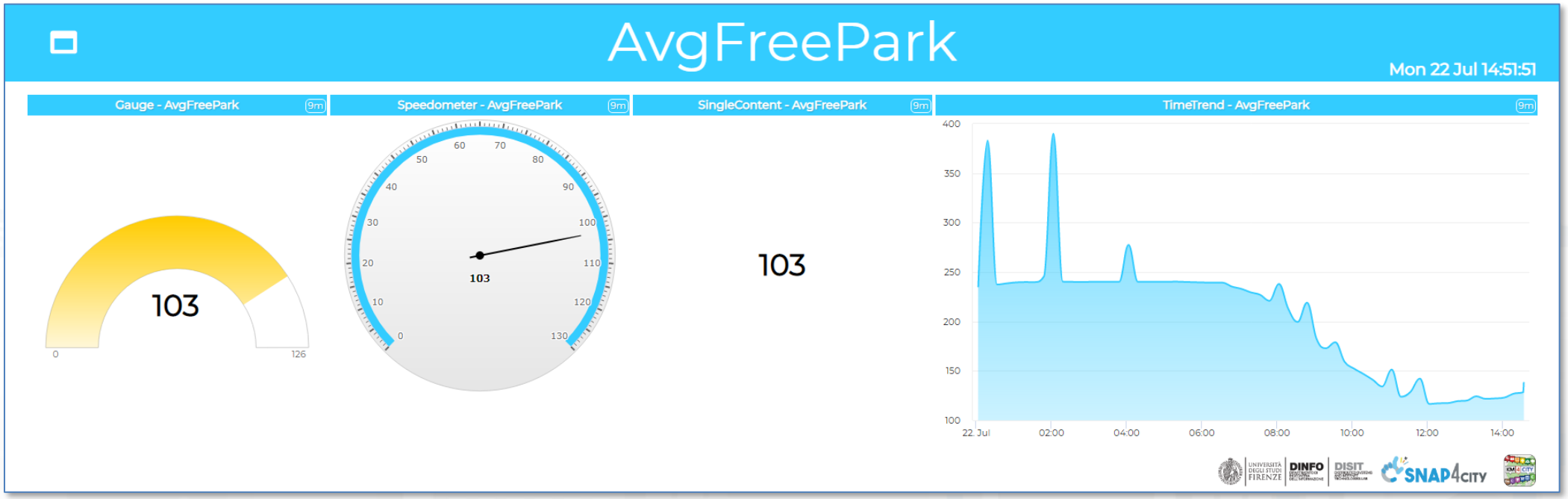
Nodes configuration 2/2







Resulting Dashboard



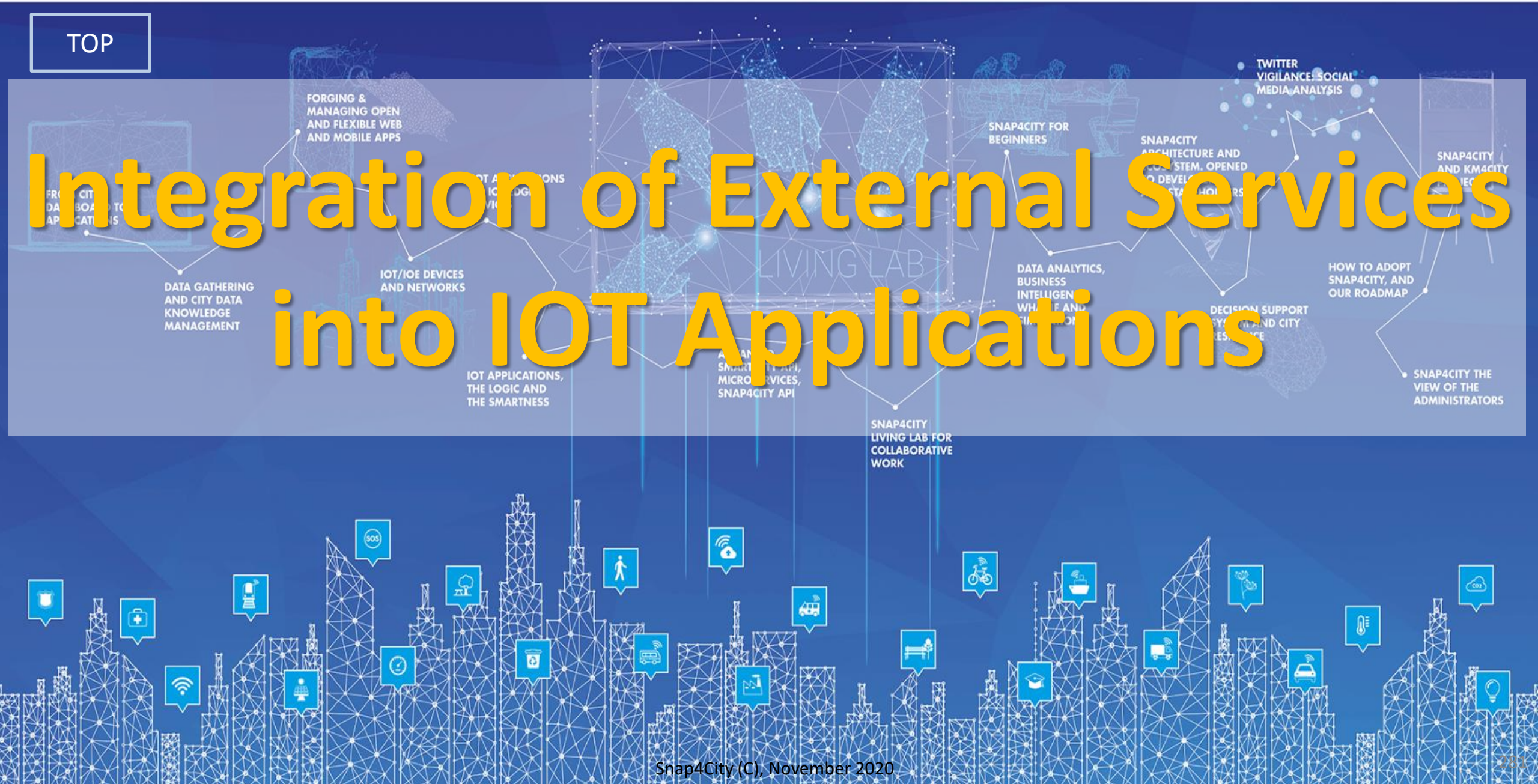
https://main.snap4city.org/view/index.php?iddasboard=MTk2Mg==

Self Training articles

- C. Badii, P. Bellini, A. Difino, P. Nesi, "Smart City IoT Platform Respecting GDPR Privacy and Security Aspects", accepted for publication on IEEE Access, 2020. 10.1109/ACCESS.2020.2968741 https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8966344
- C. Badii, P. Bellini, A. Difino, P. Nesi, G. Pantaleo, M. Paolucci, MicroServices Suite for Smart City Applications, Sensors, MDPI, 2019. https://doi.org/10.3390/s19214798
- C. Badii, P. Bellini, A. Difino, P. Nesi, "Sii-Mobility: an IOT/IOE architecture to enhance smart city services of mobility and transportation", Sensors, MDPI, 2019. https://doi.org/10.3390/s19010001 https://www.mdpi.com/1424-8220/19/1/1/pdf
- See also courses in ITALIANO: https://www.snap4city.org/485

TOP

Integration of External Services into IOT Applications



External Services

- **Any service having some API** (rest, WS, web services, etc..) can be called and exploited from IOT App. Also for example machine learning tool on AWS, MicroSoft cloud, Google Cloud. They can access to Snap4City Data by using Smart City API, ASCAPI

- Any REST Call can be transformed into a Block/Node for Node-red and thus exploited in IOT Apps.

- **Example of integration of IOT App Snap4City with:**
  - Workflow management system for Ticketing and Maintenance
  - Twitter Vigiliance, data collection
  - CKAN for Open Data gathering and publication
  - Video Wall for Control Room
  - Copernicus, etc.

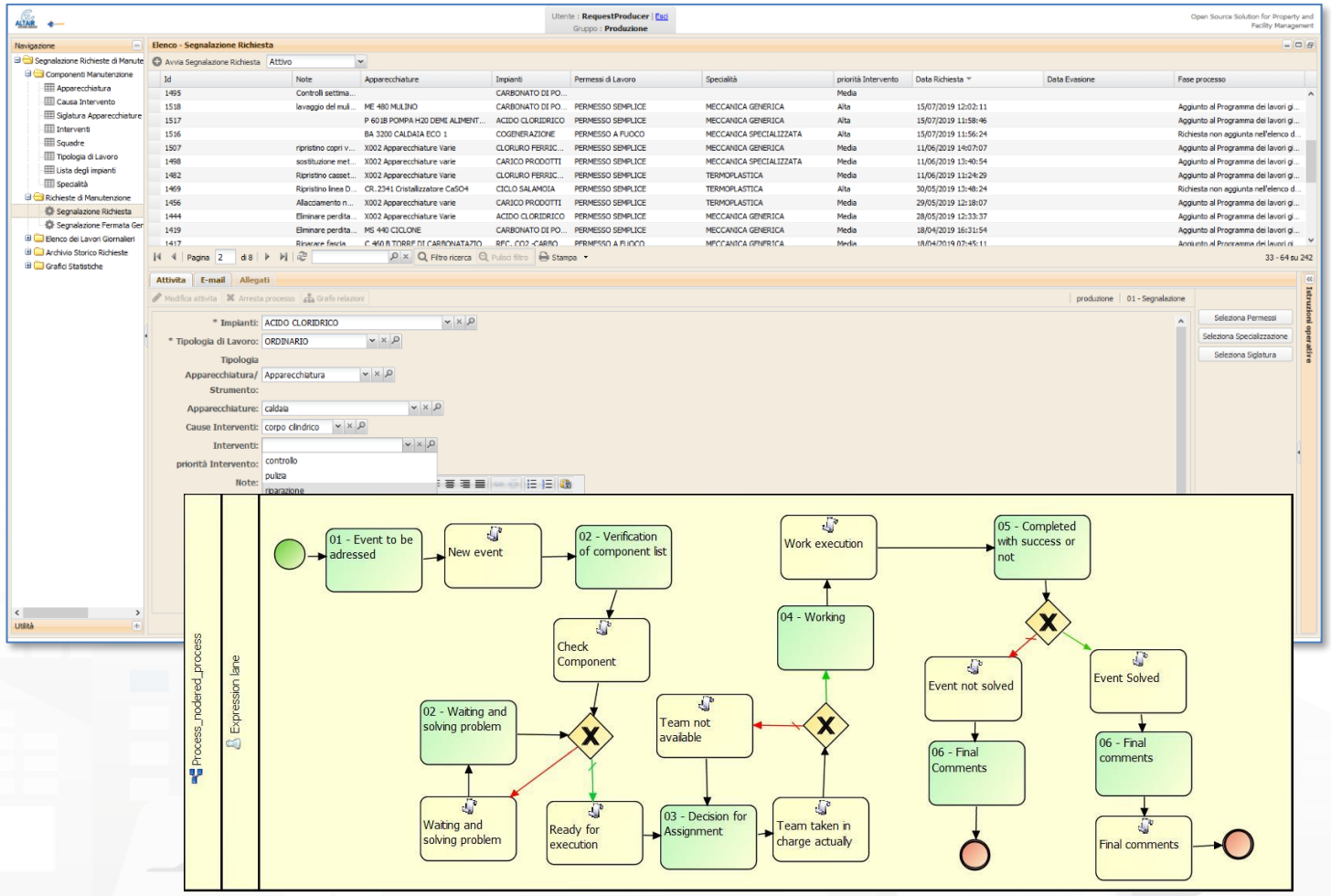
Integration with Ticketing Systems Workflows



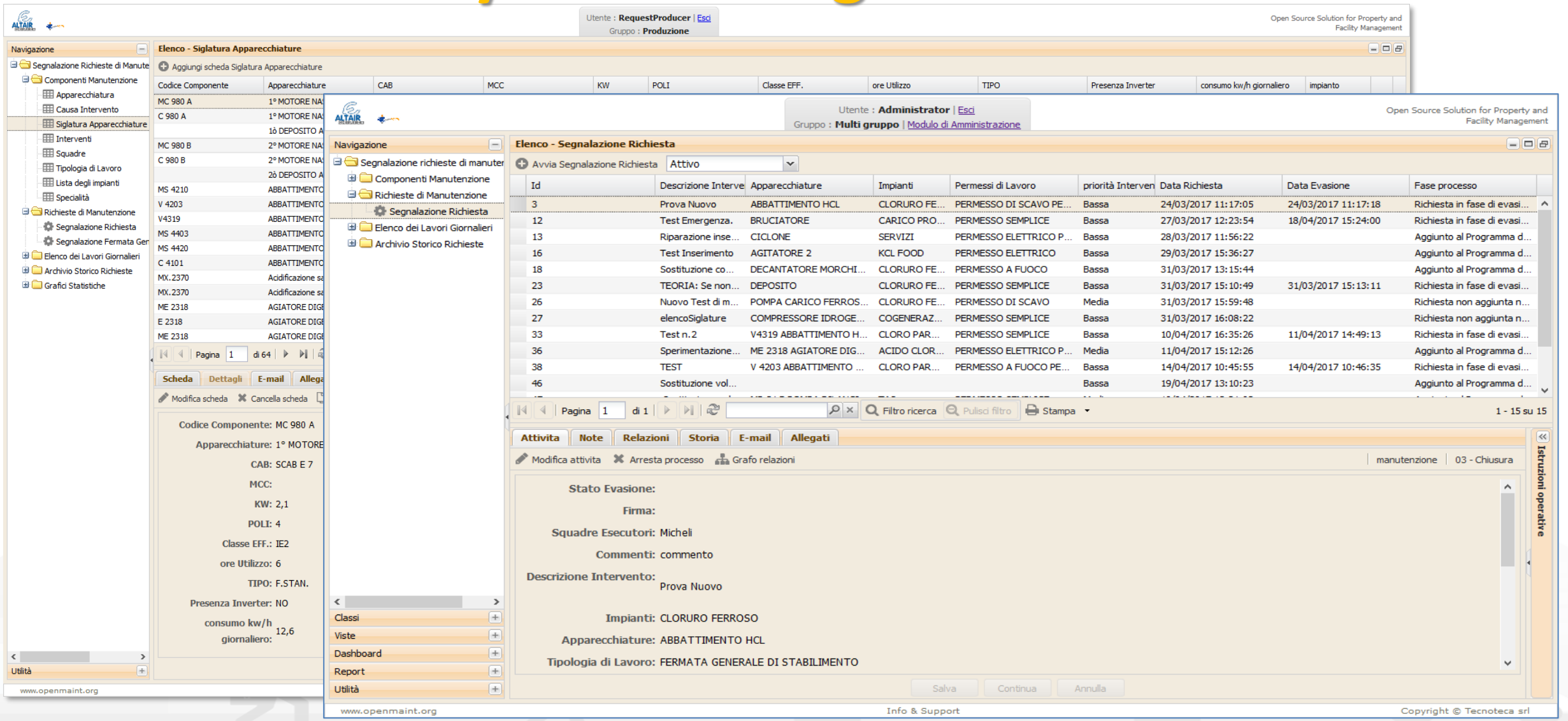
Integration with Ticketing Systems Workflow

- **Snap4City is integrated with OpenMaint Ticketing system. An Open Source solution for ticketing and workflow management.**

- Any ticketing systems can be integrated with Snap4City, by means of IOT Applications and Dashboards
- https://www.snap4city.org/597



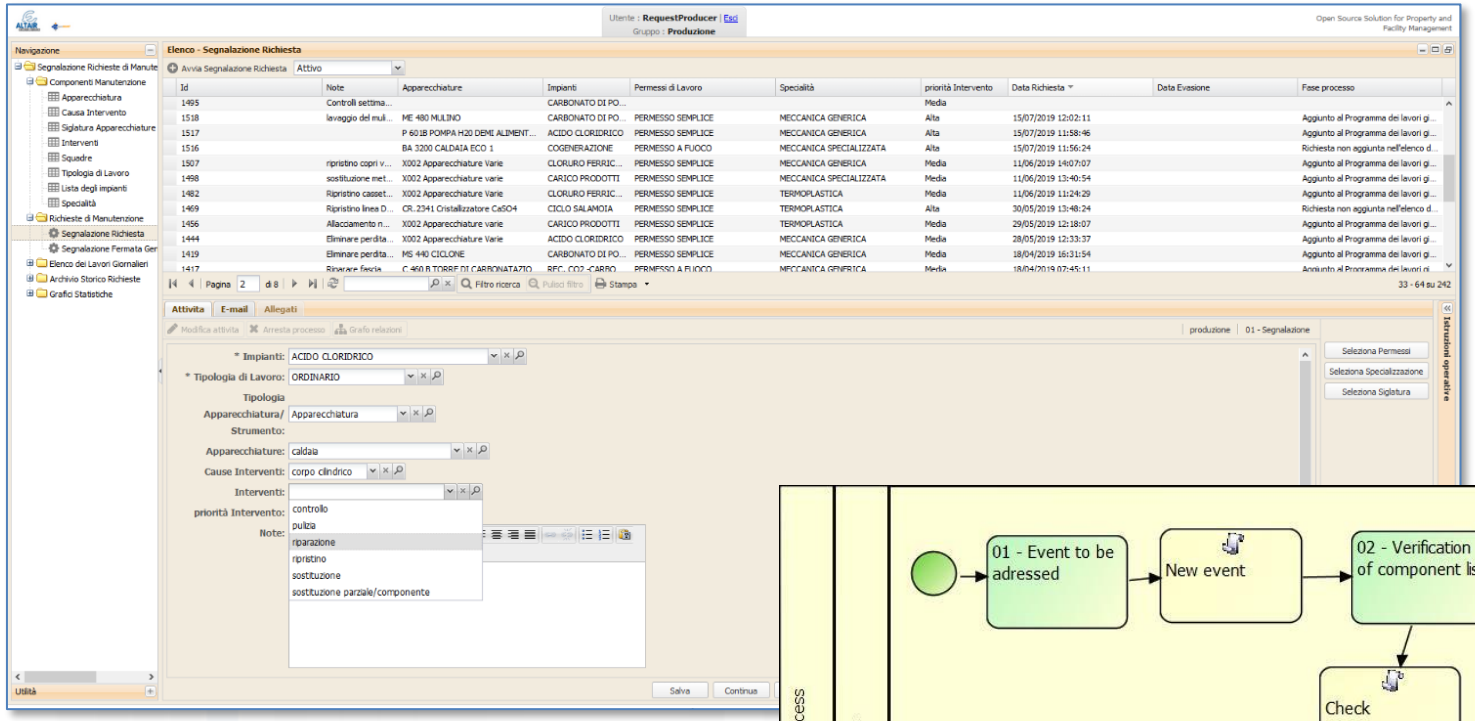
Assets inventory and management



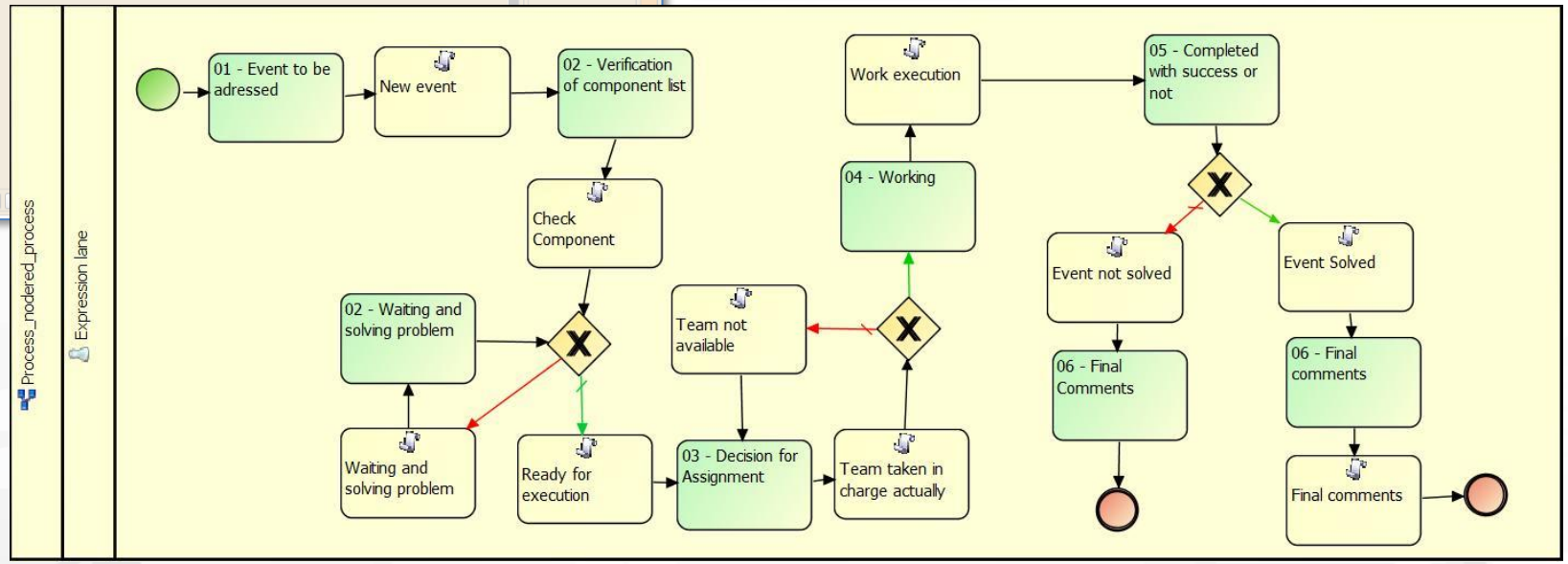
Solution for Asset Management and Maintenance

- Inventory of industry assets (movable, logistics, equipments, etc.)
- Tickets management for corrective maintenance
- Reports and Dashboards
- Predictive maintenance and Early Warning support via analytics
- Business Intelligence support
- User management with different levels of access

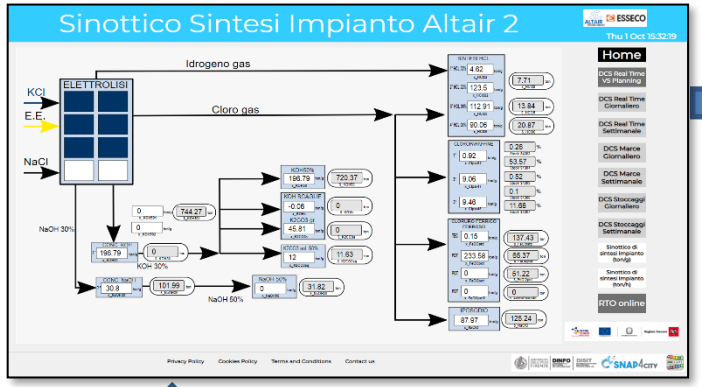
OpenMaint Ticketing System



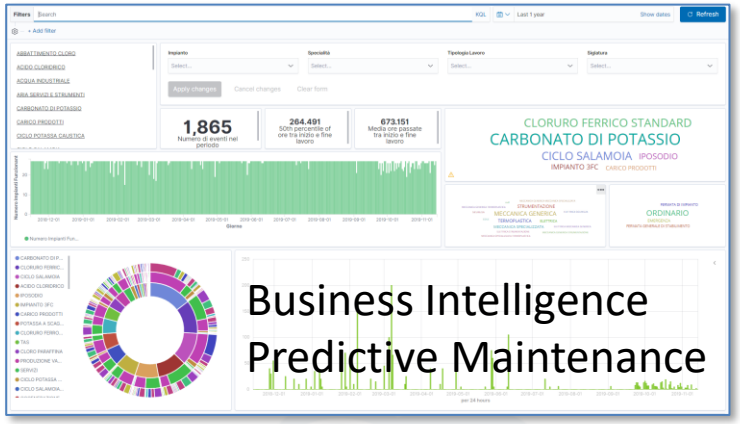
- Activate events
- Manage steps, workflow
- Collecting feedbacks and results from teams



Example of Integrated workflow

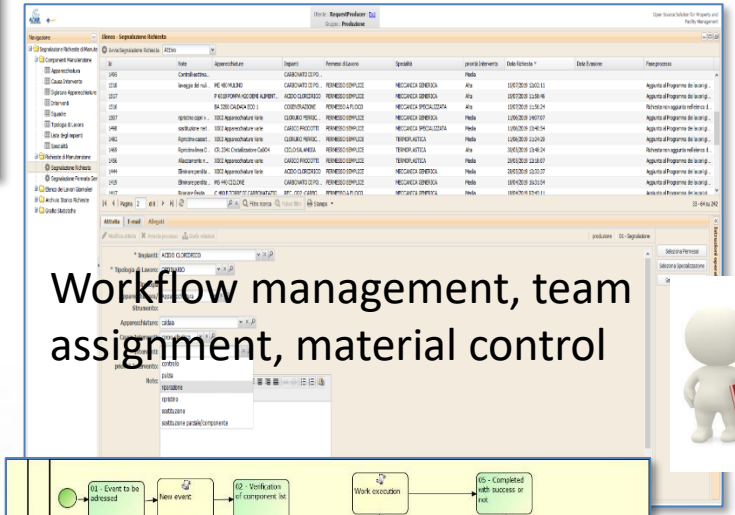


Consumptions/productions

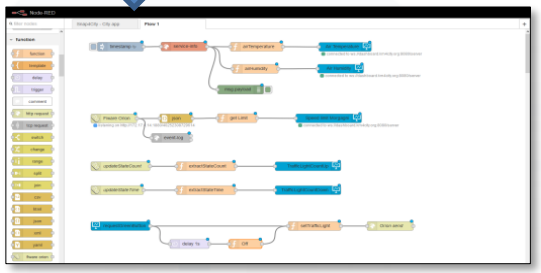


Events/actions

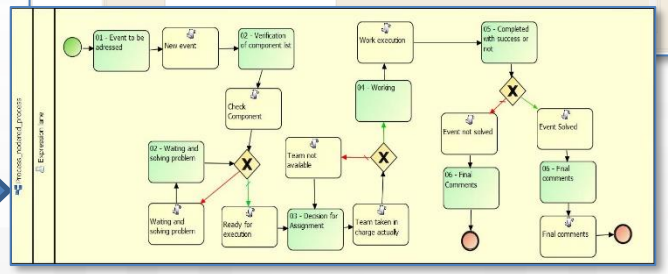
Business Intelligence
Predictive Maintenance

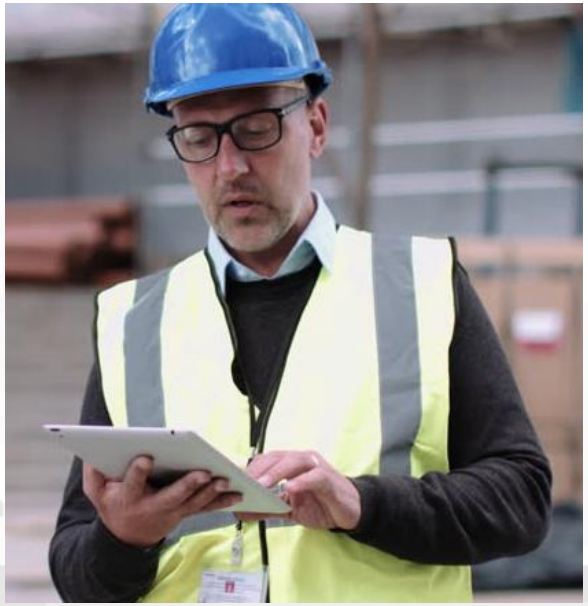


Workflow management, team assignment, material control

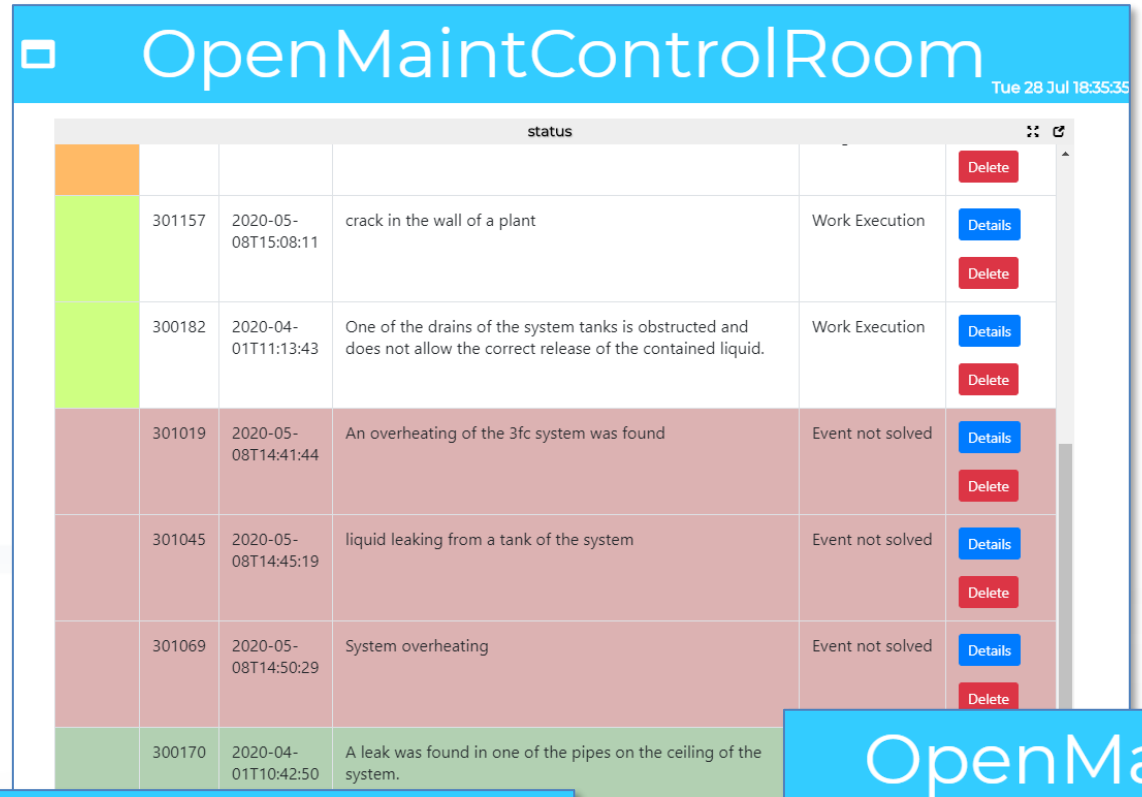


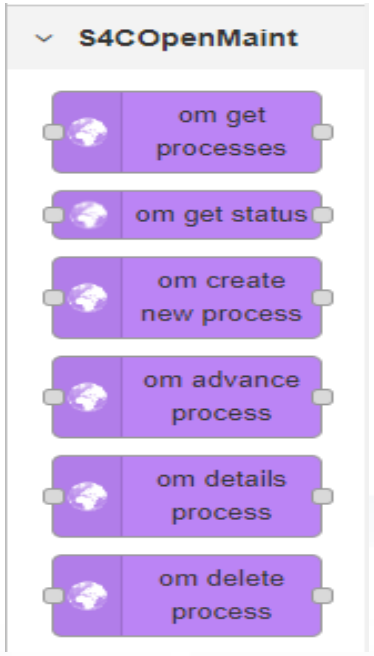
Event detection and firing
Critical event management

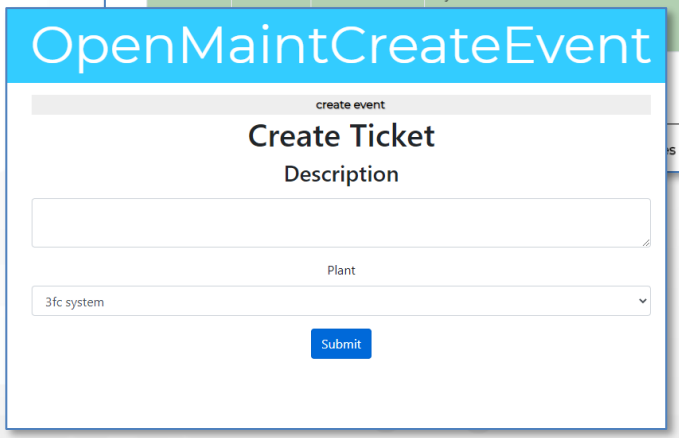


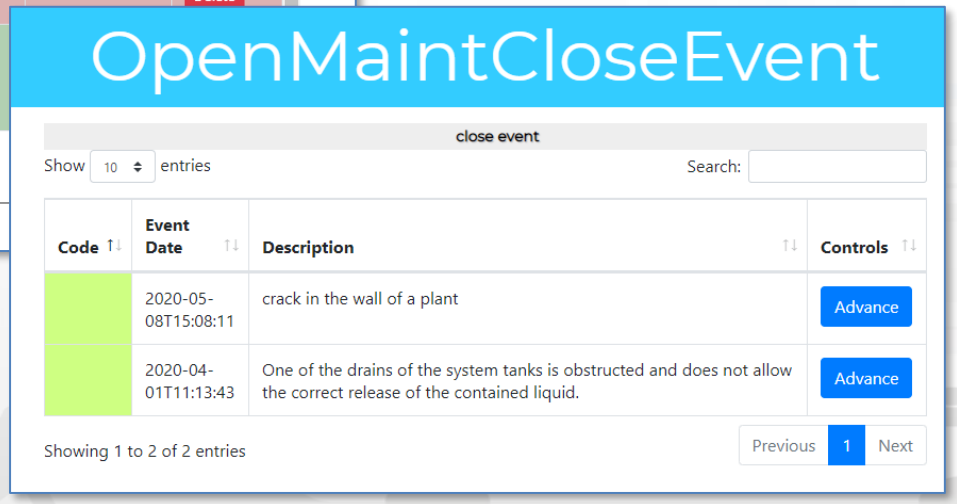


Dashboards









- **Snap4City can**
  - Create new tickets
  - Manage steps, workflow
  - Collecting feedbacks and results from teams
  - Manage all phases of the workflow on the fields via IOT Apps and logics
  - The integration if via API and MicroServices into IOT App.

Filters | Search | KQL | Last 1 year | Show dates | Refresh

+ Add filter

- ABBATTIMENTO CLORO
- ACIDO CLORIDRICO
- ACQUA INDUSTRIALE
- ARIA SERVIZI E STRUMENTI
- CARBONATO DI POTASSIO
- CARICO PRODOTTI
- CICLO POTASSA CAUSTICA

Impianto: Select...

Specialità: Select...

Tipologia Lavoro: Select...

Siglatura: Select...

Apply changes | Cancel changes | Clear form

1,865
Numero di eventi nel periodo

264.491
50th percentile of ore tra inizio e fine lavoro

673.151
Media ore passate tra inizio e fine lavoro

CLORURO FERRICO STANDARD
CARBONATO DI POTASSIO
CICLO SALAMOIA
IPOSODIO
IMPIANTO 3FC
CARICO PRODOTTI

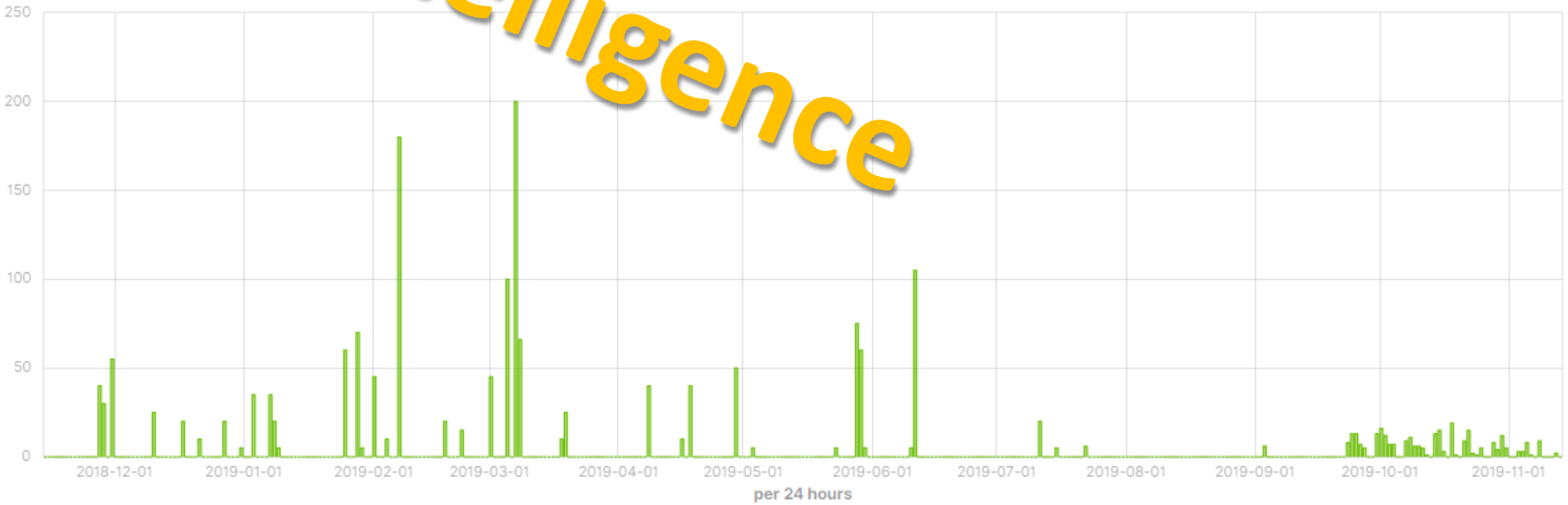




FERMATA DI IMPIANTO
ORDINARIO
EMERGENZA
FERMATA GENERALE DI STABILIMENTO

- CARBONATO DI P...
- CLORURO FERRIC...
- CICLO SALAMOIA
- ACIDO CLORIDRICO
- IPOSODIO
- IMPIANTO 3FC
- CARICO PRODOTTI
- POTASSA A SCAG...
- CLORURO FERRO...
- TAS
- CLORO PARAFFINA
- PRODUZIONE VA...
- SERVIZI
- CICLO POTASSA ...
- CICLO SALAMOIA...





Maintenance Intelligence

Integration with

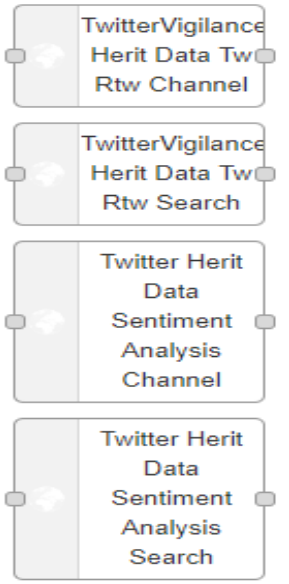
Twitter Vigilance



Twitter Vigilance

- A separate tool of DISIT lab: provided in different versions
  - **Described into Data Analytic part of the training course**
- It can be used to:
  - Collect and Monitor Twitter data
  - Perform Multilingual processing: English, France, Italia, etc.
  - Estimated in real time metrics: volume, sentiment, ratio,
  - Provide data into the smart city and thus alerting and firing
  - Compute predictions
  - Set up Early Warning systems

- **Snap4City** integration is done via API and MicroServices into IOT App.

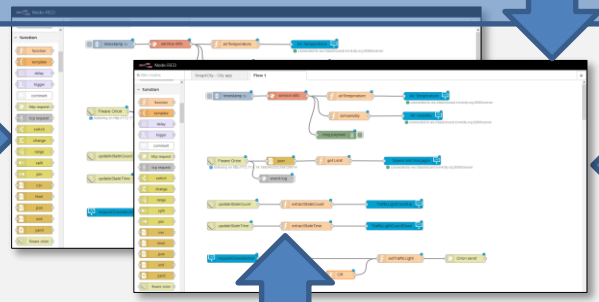


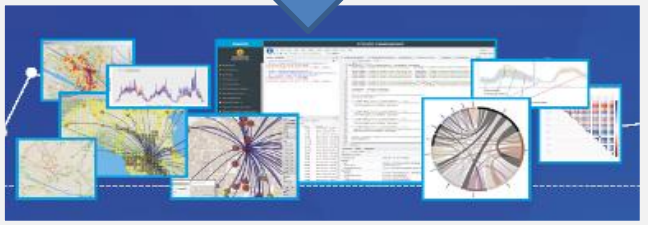
Twitter Vigilance

IOT and data World



IOT Applications





Big Data Analytics, Artificial Intelligence

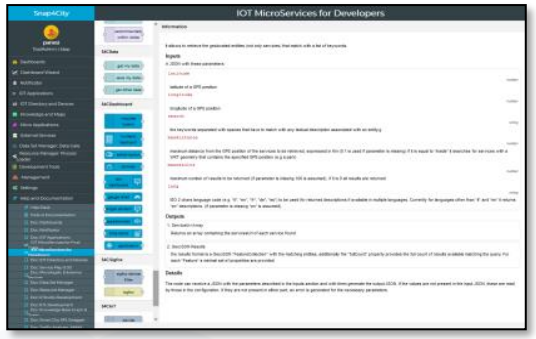
Dashboards and Apps



IOT Applications Development

IOT Discovering

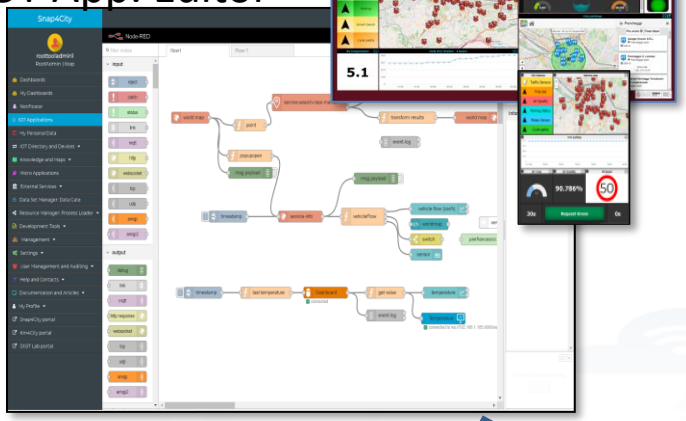
MicroServices collections



My IOT Applications

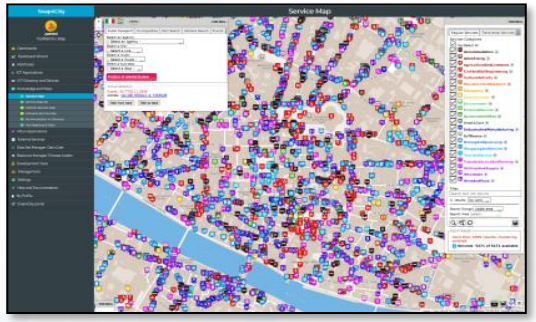


IOT App. Editor

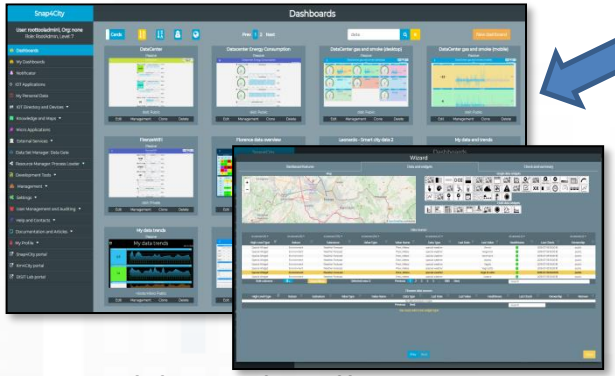


Generating IOT App
With Dashboard





ServiceMap Discovery



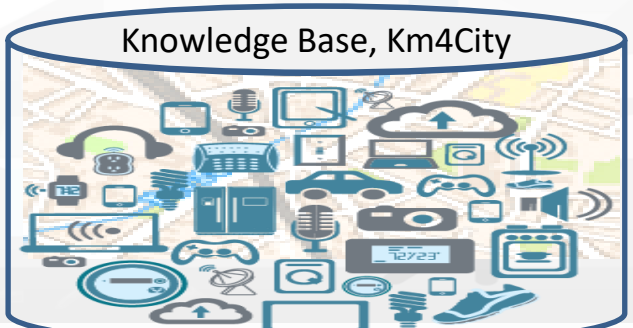
Dashboard Collection,
Editor and Wizard

Sharing/saving
reusing IOT App

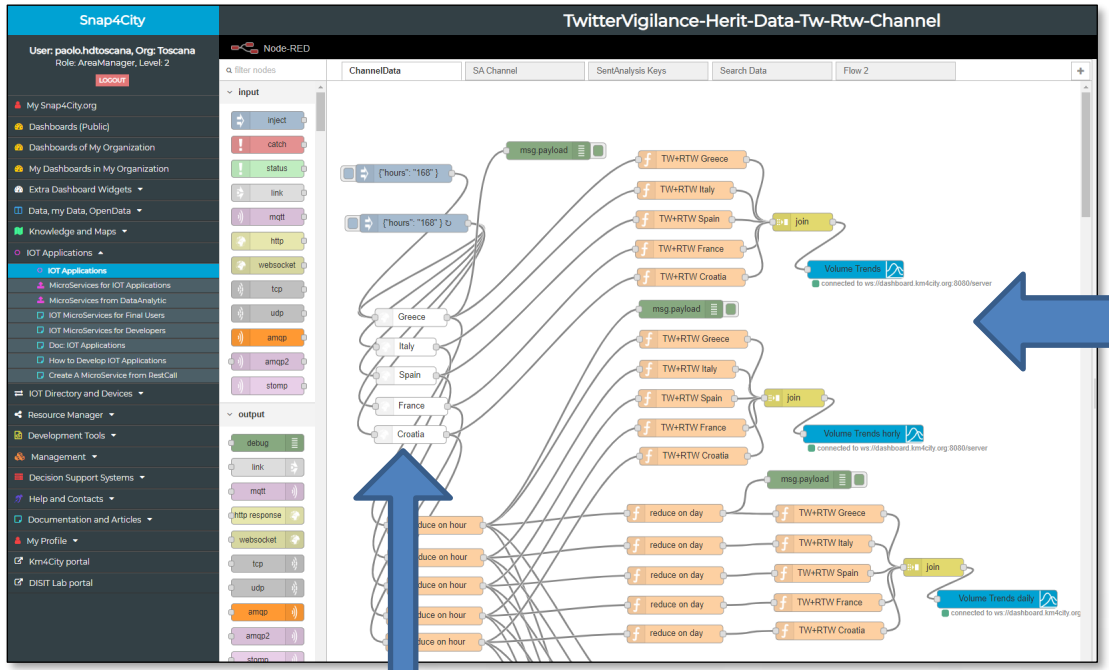


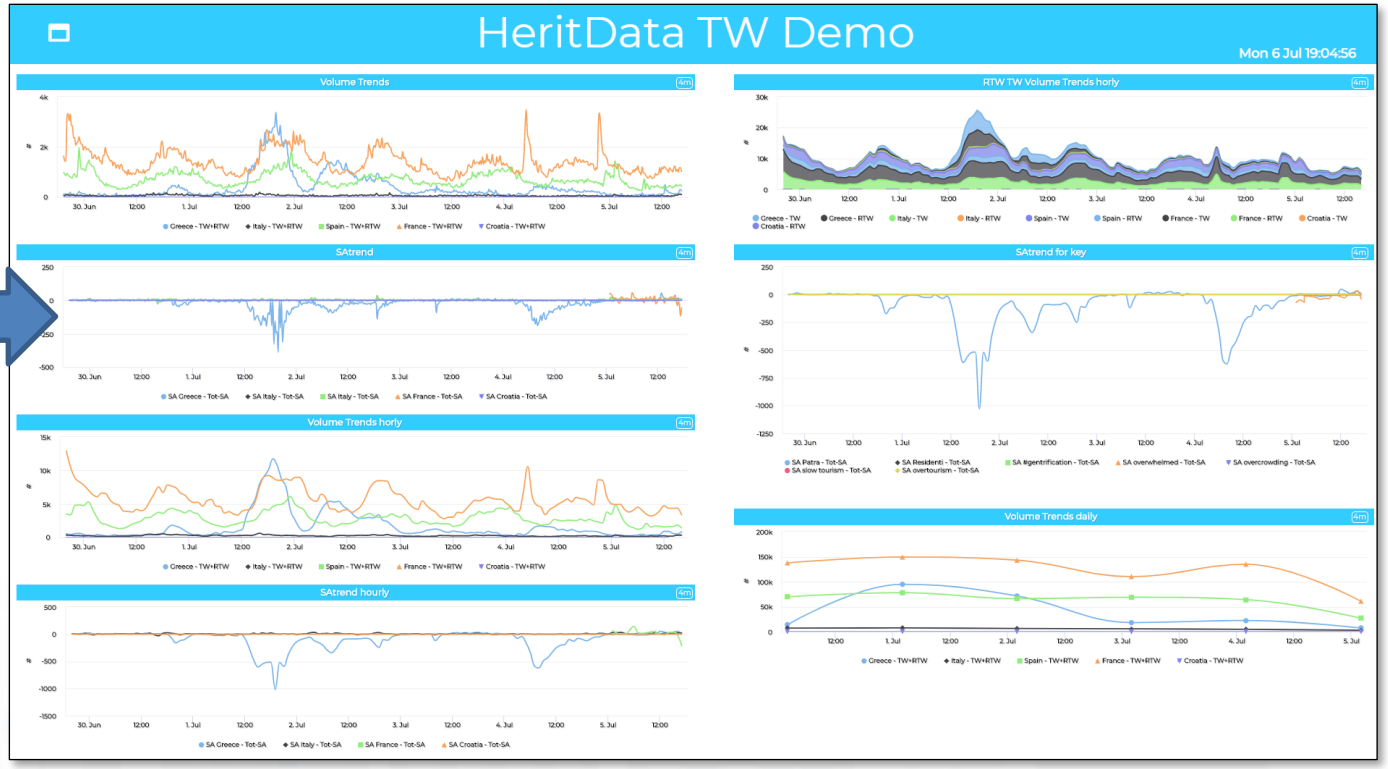
Resource Manager

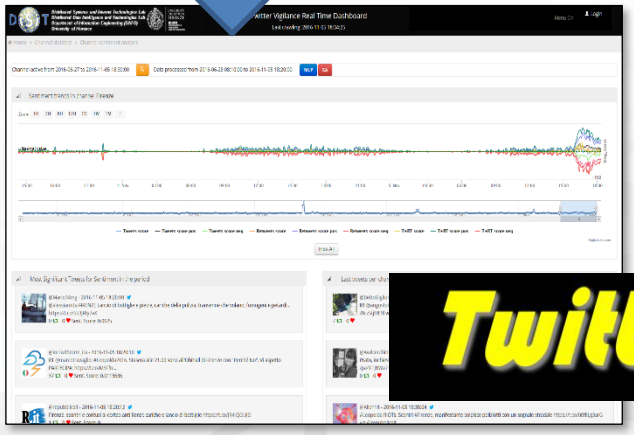




Overview







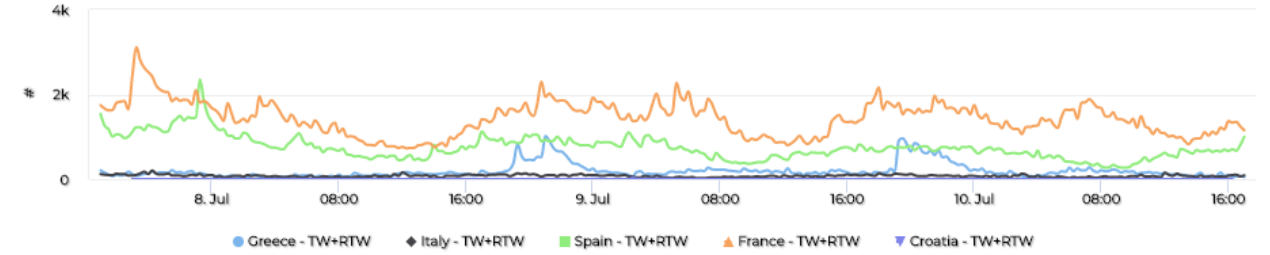
Twitter Vigilance

https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=Mjc3NQ==

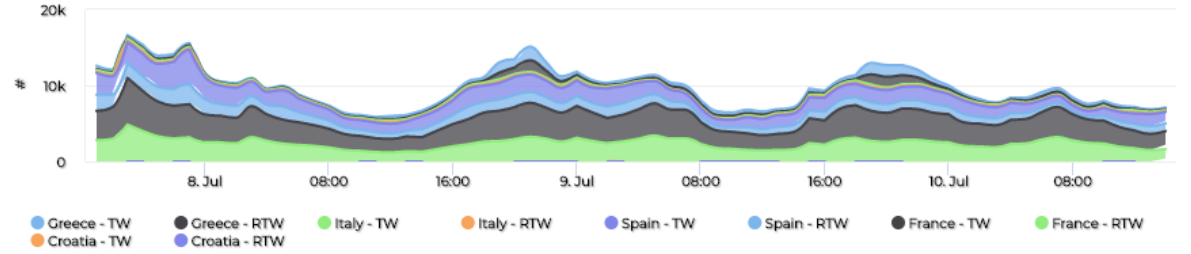
HeritData TW Demo

Tue 14 Jul 16:54:43

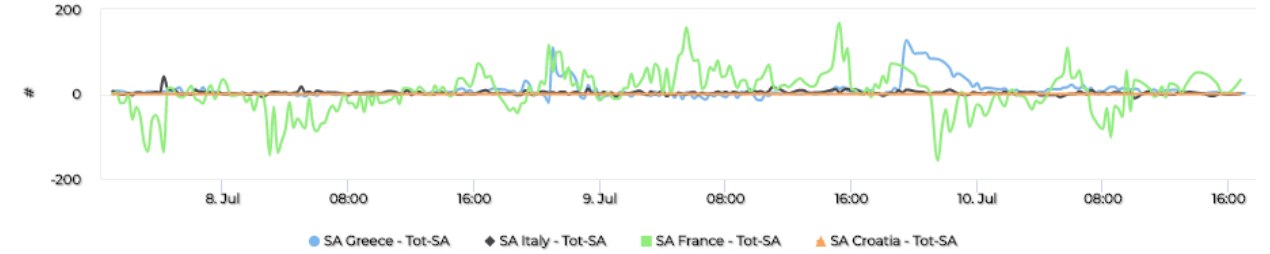
Volume Trends



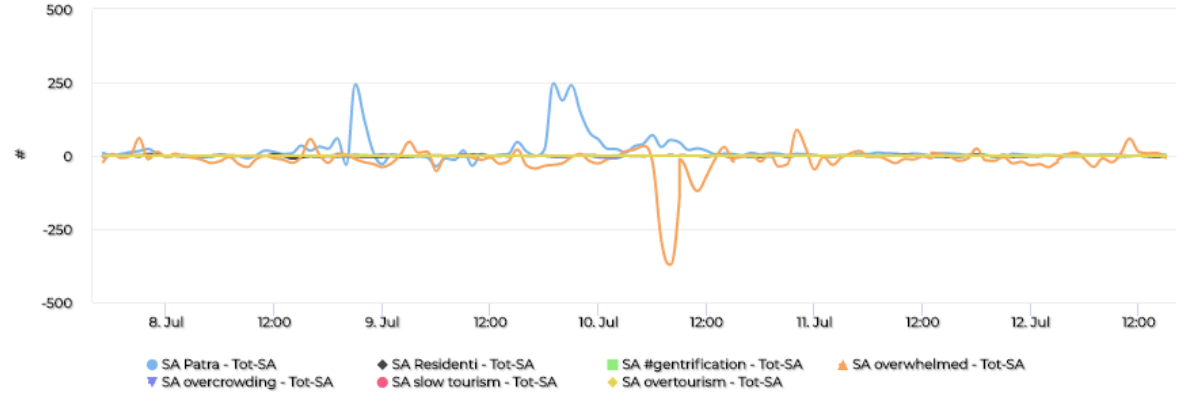
RTW TW Volume Trends horly



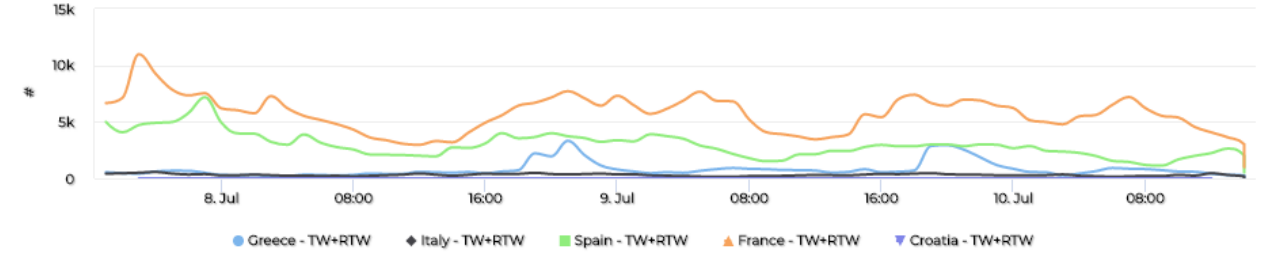
SAtrend



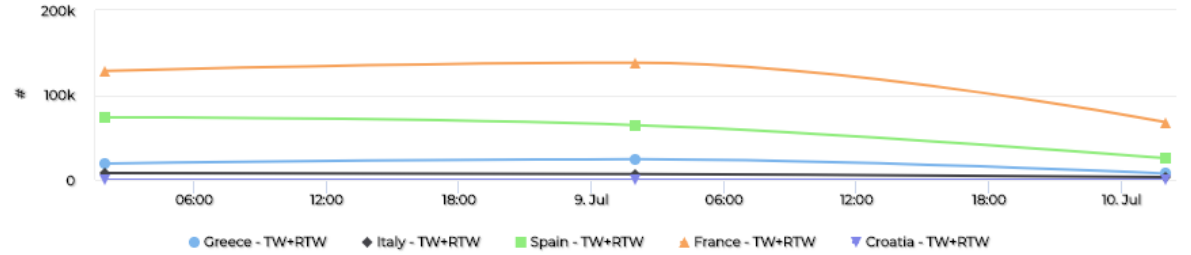
SAtrend for key



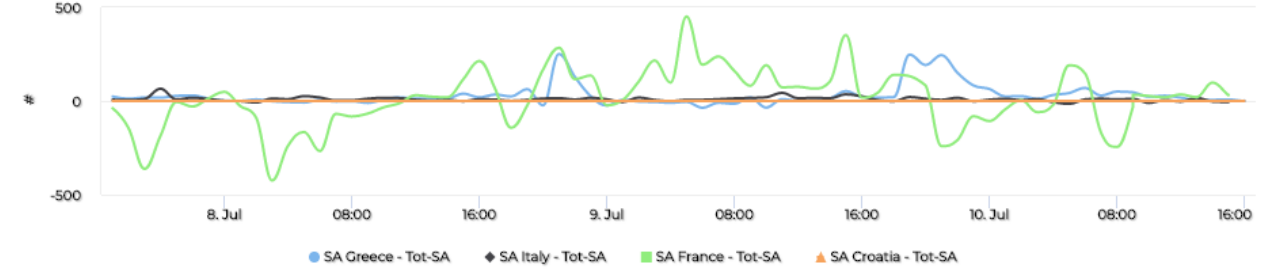
Volume Trends horly

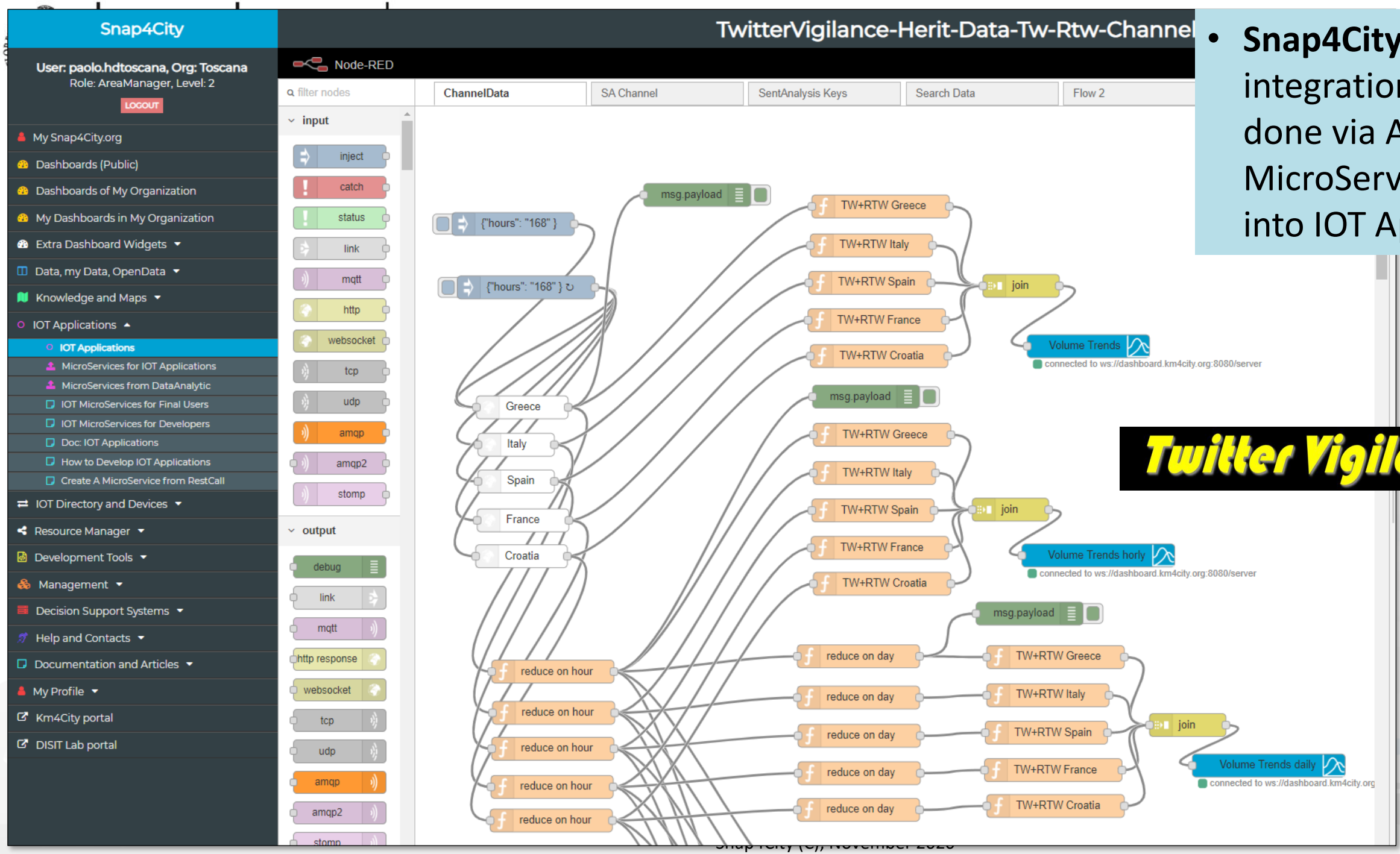


Volume Trends daily



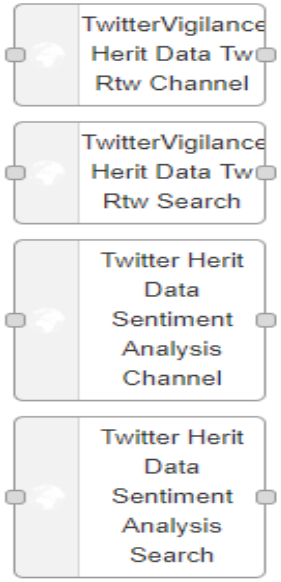
SAtrend hourly

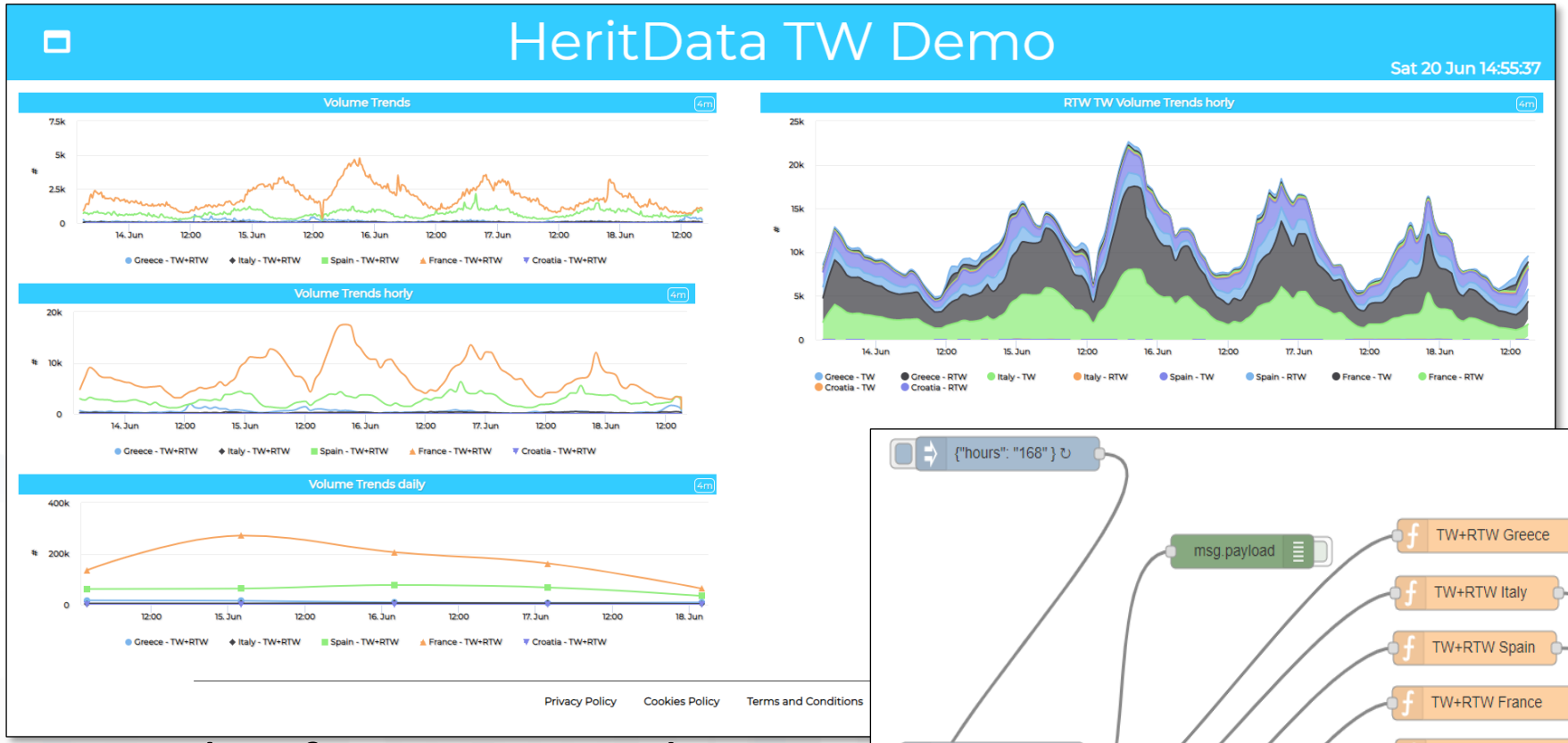


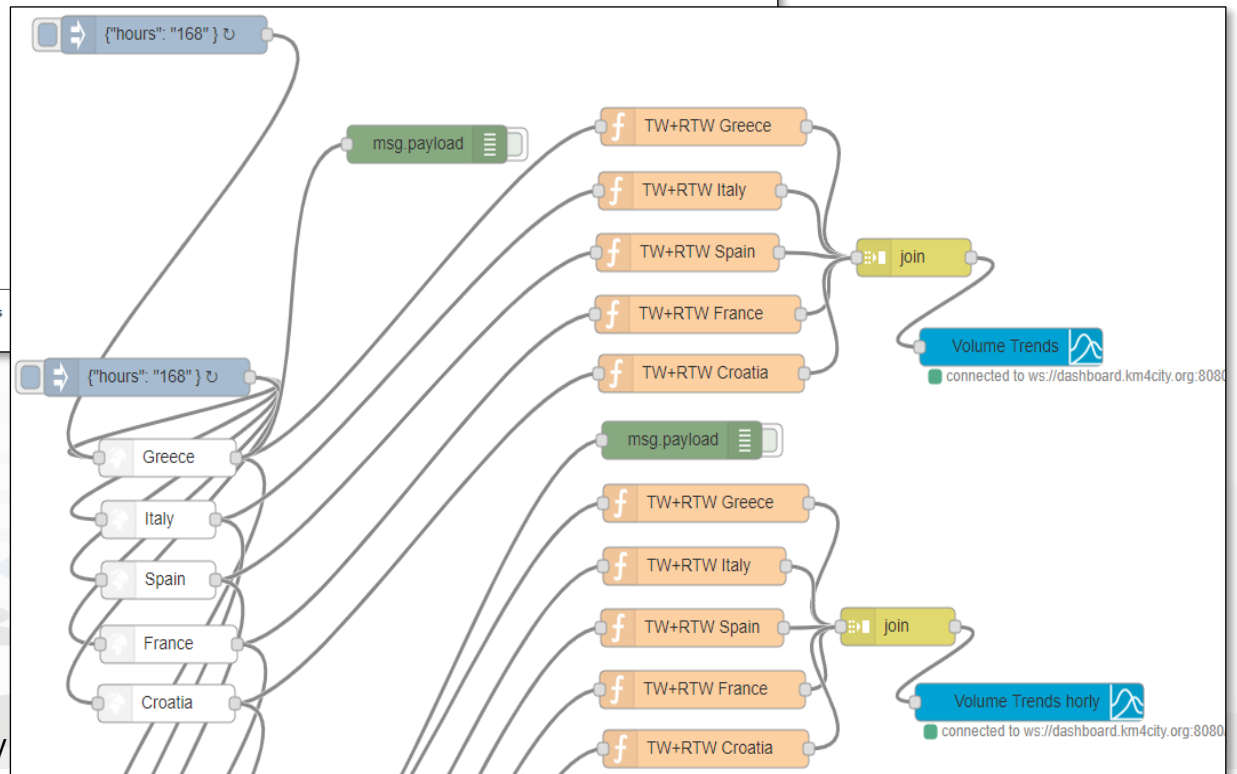


- **Snap4City** integration is done via API and MicroServices into IOT App.

Twitter Vigilance



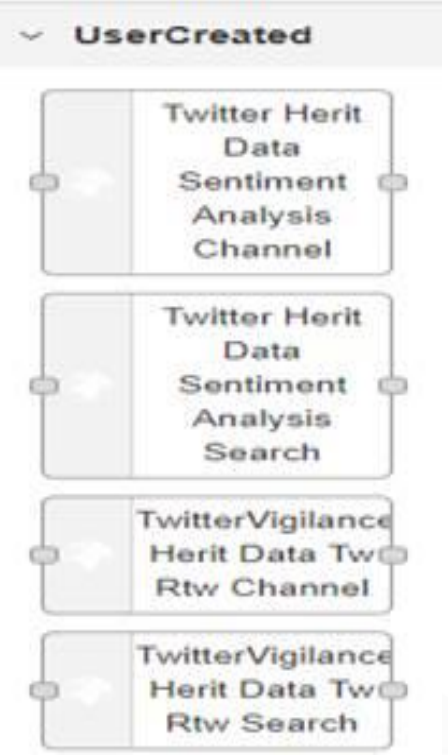




- This is an example of Dynamic Widget Data production for MultiSeries
- Each country block produce a vector and the vectors are joined and sent to the Multiserie Widget

Further reading on Social Media

- TC2.22- Exploiting Twitter Vigilance as External Service, in Dashboard, and as RestCall as MicroService in IOT applications
- TC2.21- IOT Applications with Social Media Actions, and cultural scenarious
- External Services
- TC2.21- IOT Applications with Social Media Actions, and cultural scenarious



Integration with CKAN: automated data set (i) ingestion and (ii) production via IOT App



UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

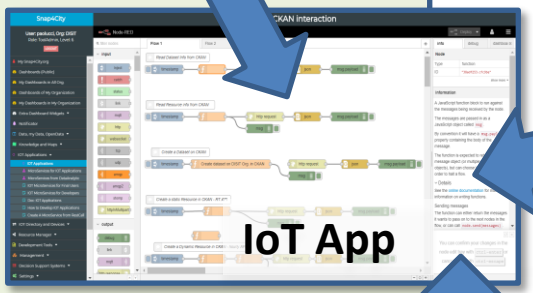
SNAP4CITY | KM4CITY

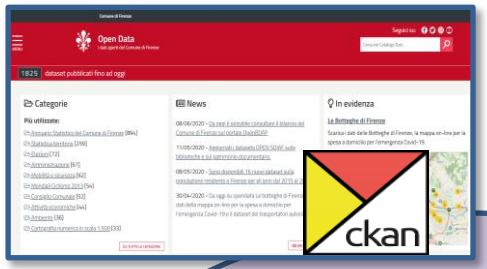
Snap4City Portal and Integrated tools

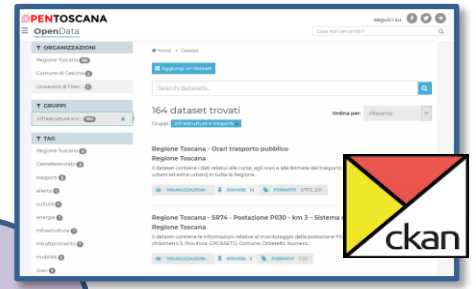


IoT App – Automatize:

- Import data from CKAN to Snap4City
- Upload Public Data from Snap4City to CKAN
- Data Harvesting
- Dashboards and Mobile/Web Apps creation







Open or Private External CKAN Data Portals



Advanced Snap4City APIs and MicroServices



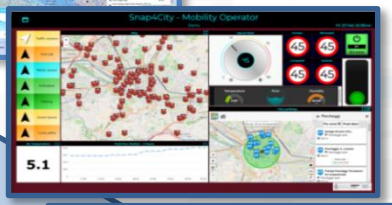
Knowledge and Storage Data from the Field and City

Real Time Data





Heatmaps
What-IF Analysis



Remote Control

Snap4City Dashboards Mob & Web Apps

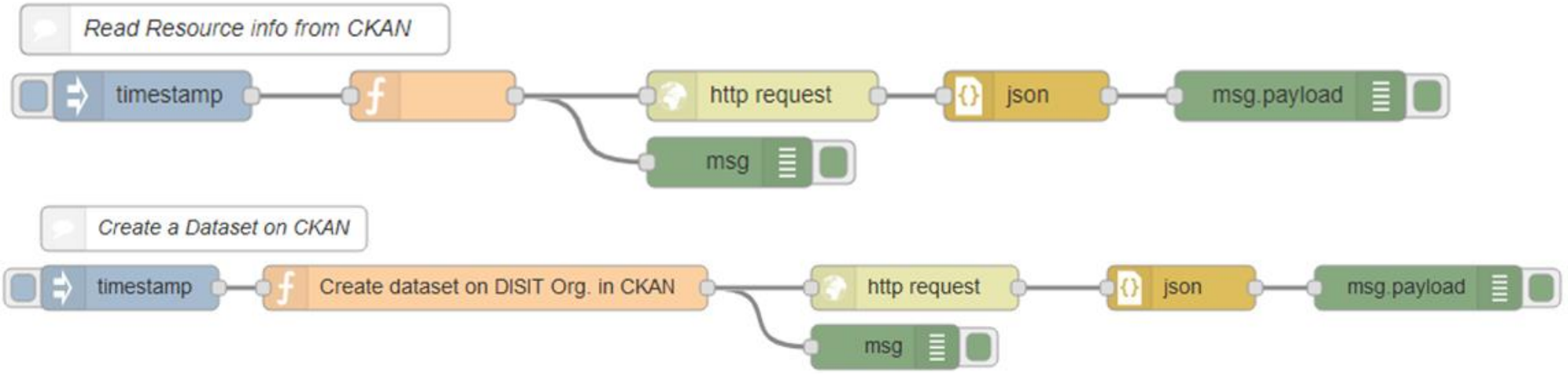




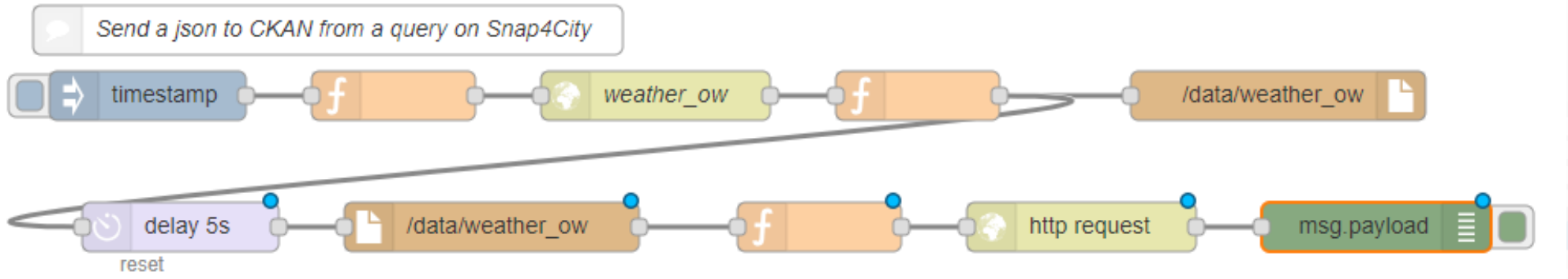


Historical Data

Some IOT App segements



Almost all the calls to CKAN are quite similar



Read more on

- **TC9.17 – CKAN vs Snap4City Integration and Interaction**
  - automating the *Read of a Dataset Info from CKAN*
  - automating the *Read of a Resource info from CKAN*
  - automating the *Creation of a Dataset on CKAN*
  - automating the *Creation of a static Resource in CKAN*
  - automating the *Creation of a dynamic Resource in CKAN*
  - automating the *Sending of a json to CKAN from a query to Snap4City to perform any other action on the Smart City*
- **Data Set Manager: Data Gate / CKAN federated**

Automated production of MicroService for IOT App from External REST CALL API



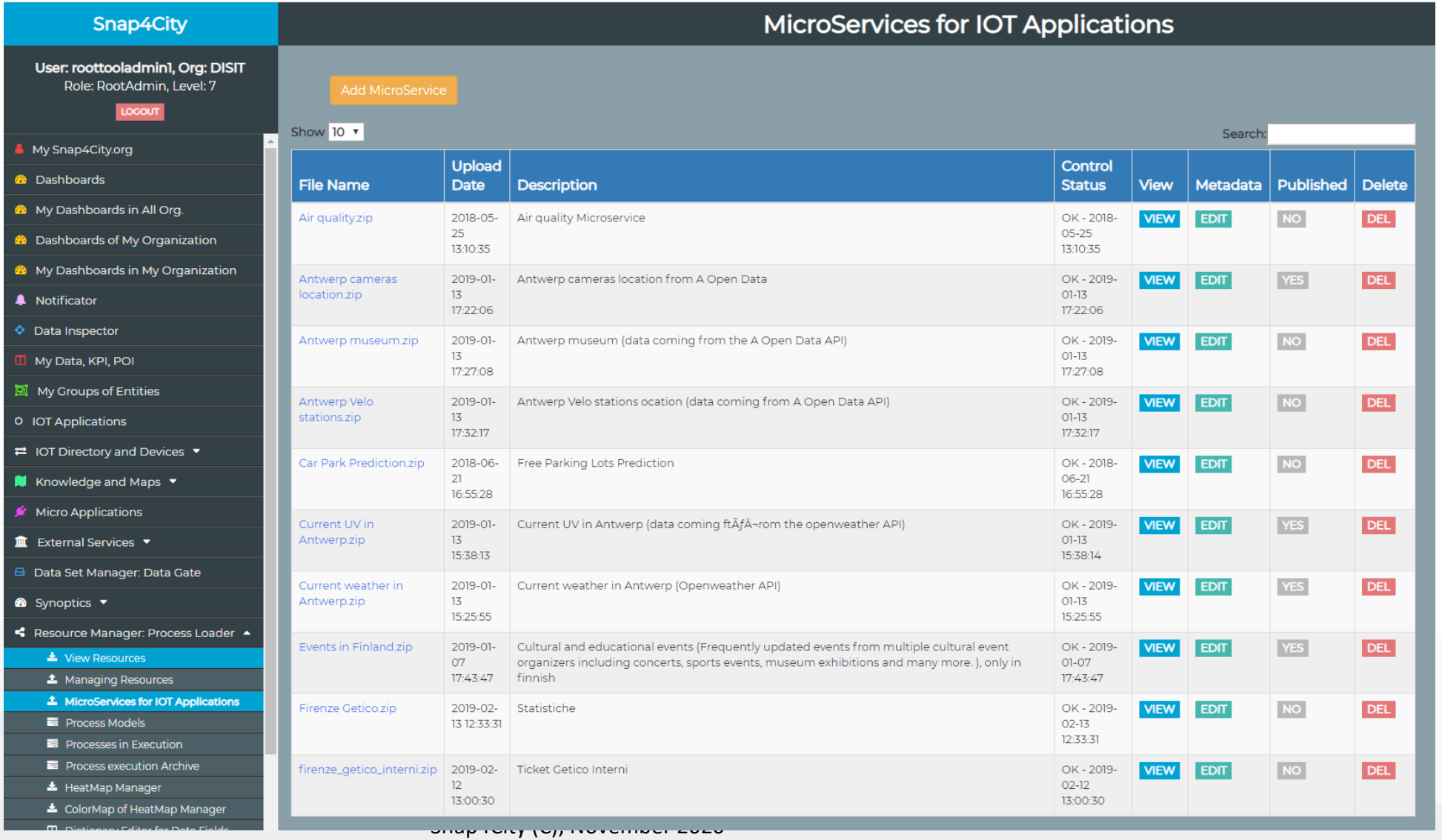
General solution, bring data from API to Dashboards

- You can **save/consolidate your rest API** transforming it in a MicroService usable for many colleagues into IOT Applications:
  - TC2.25- Registering external MicroService calling RestCall services, using it on IOT applications https://www.snap4city.org/129

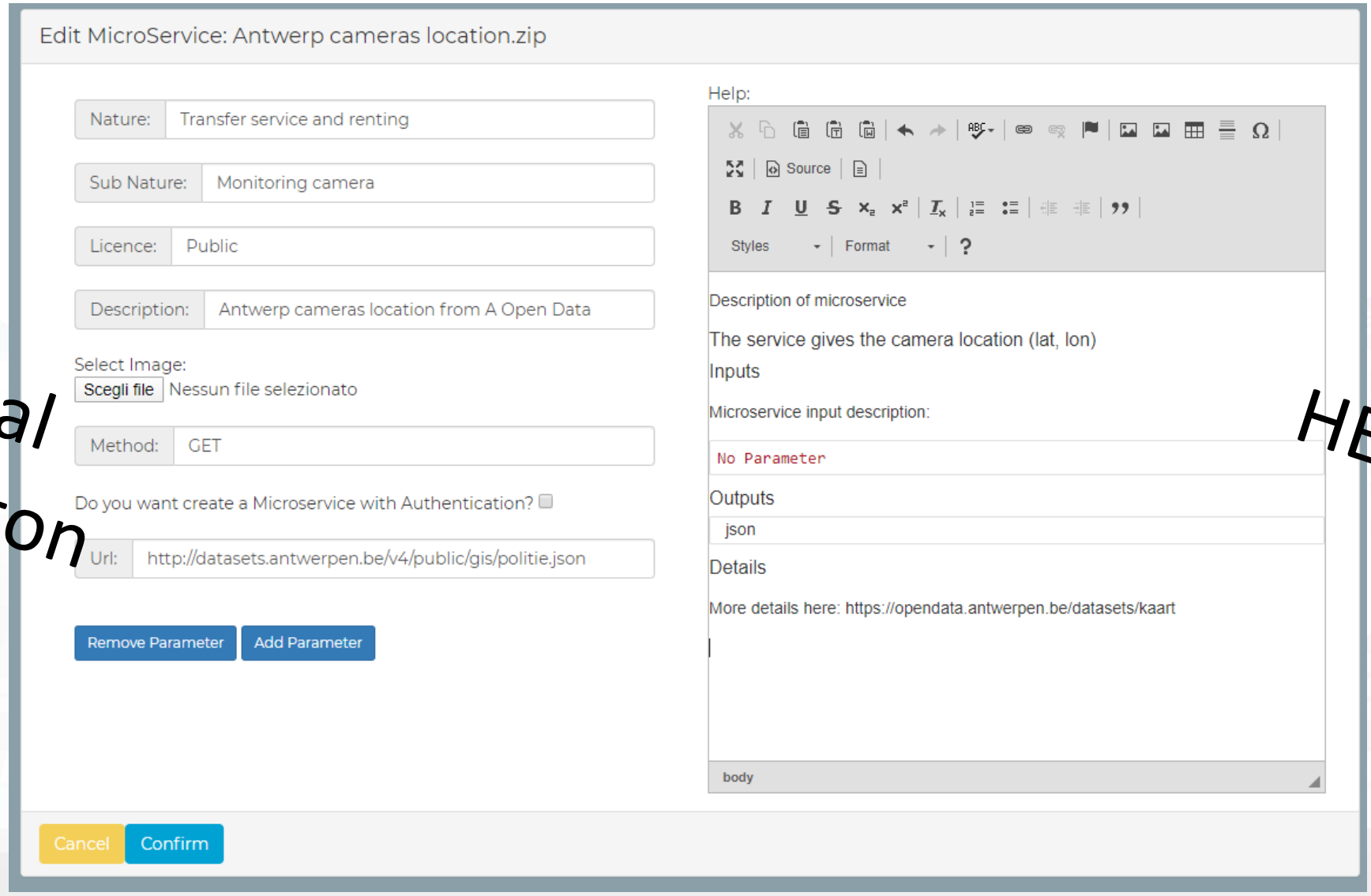
- IF your REST API is going to use credentials as username and password, we suggest you to save them into MyPersonalData of Snap4City
  - so that the code will not provide clear credentials and you can update from user interface on your personal data profile.
  - The IOT App can retrieve the Username and Password at the moment in which they are used with the security shield of Snap4City

External REST Call API vs MicroServices

- Each Rest Call API can be automaticaly transformed into e MicroService for the IOT Applications



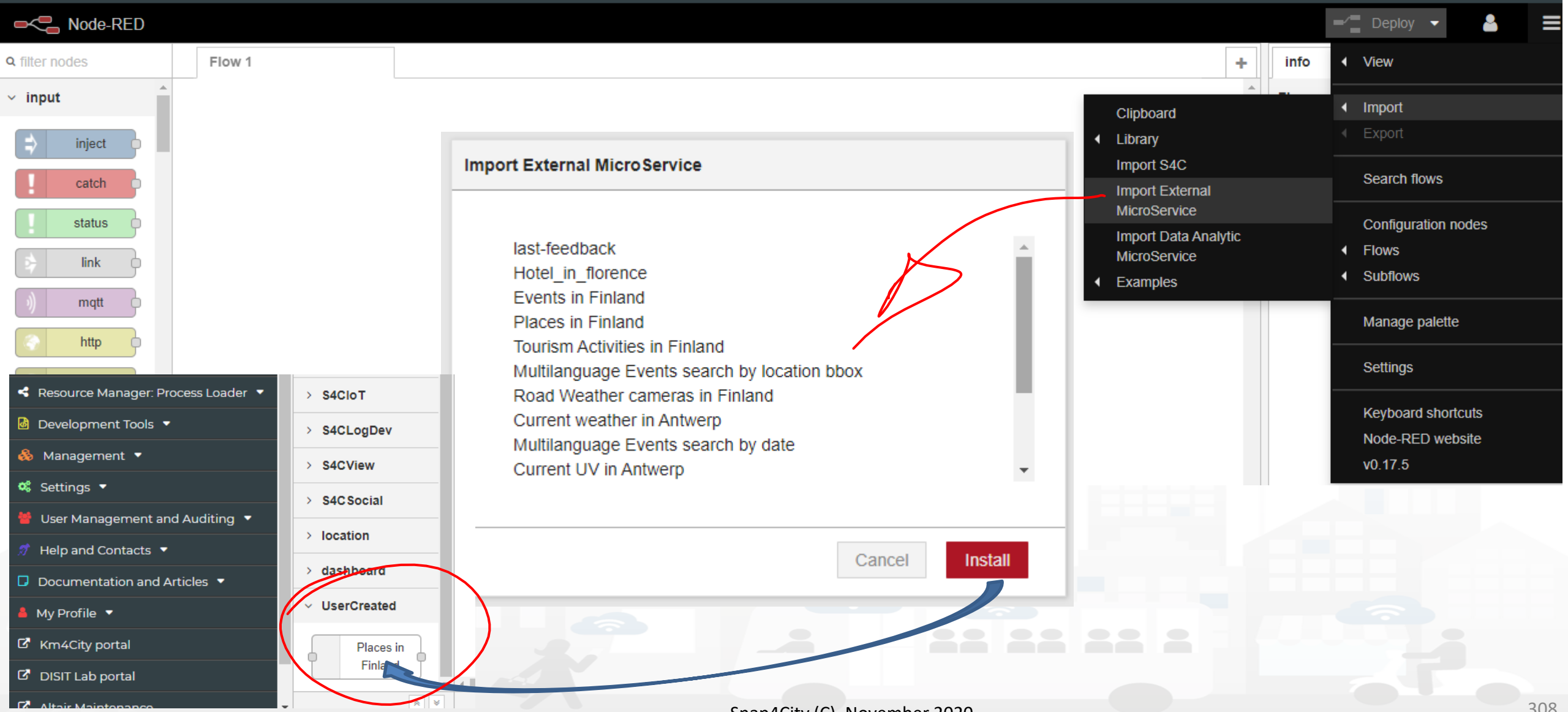
Edit MicroService Call: call and help editing



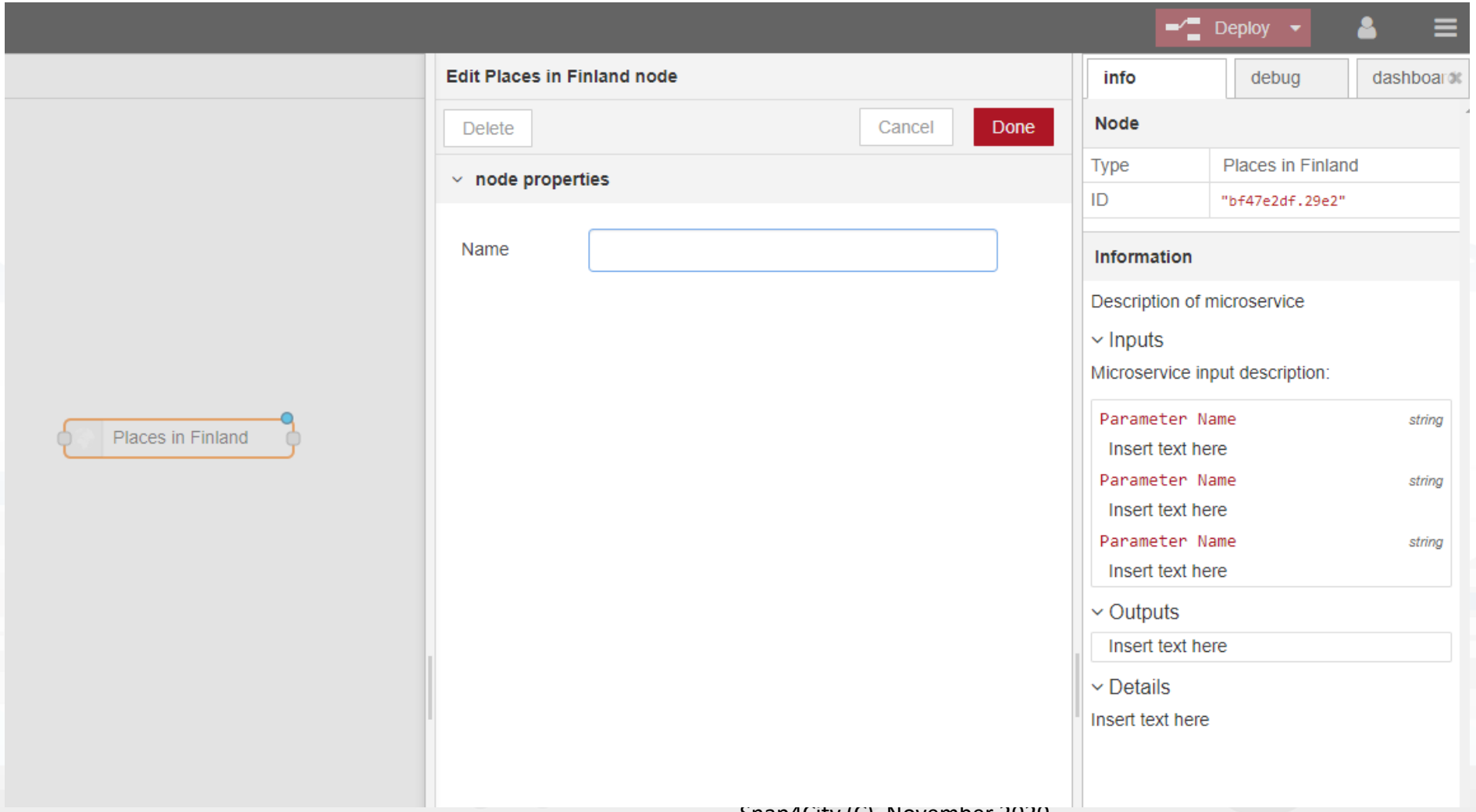
Formal definiton

HELP

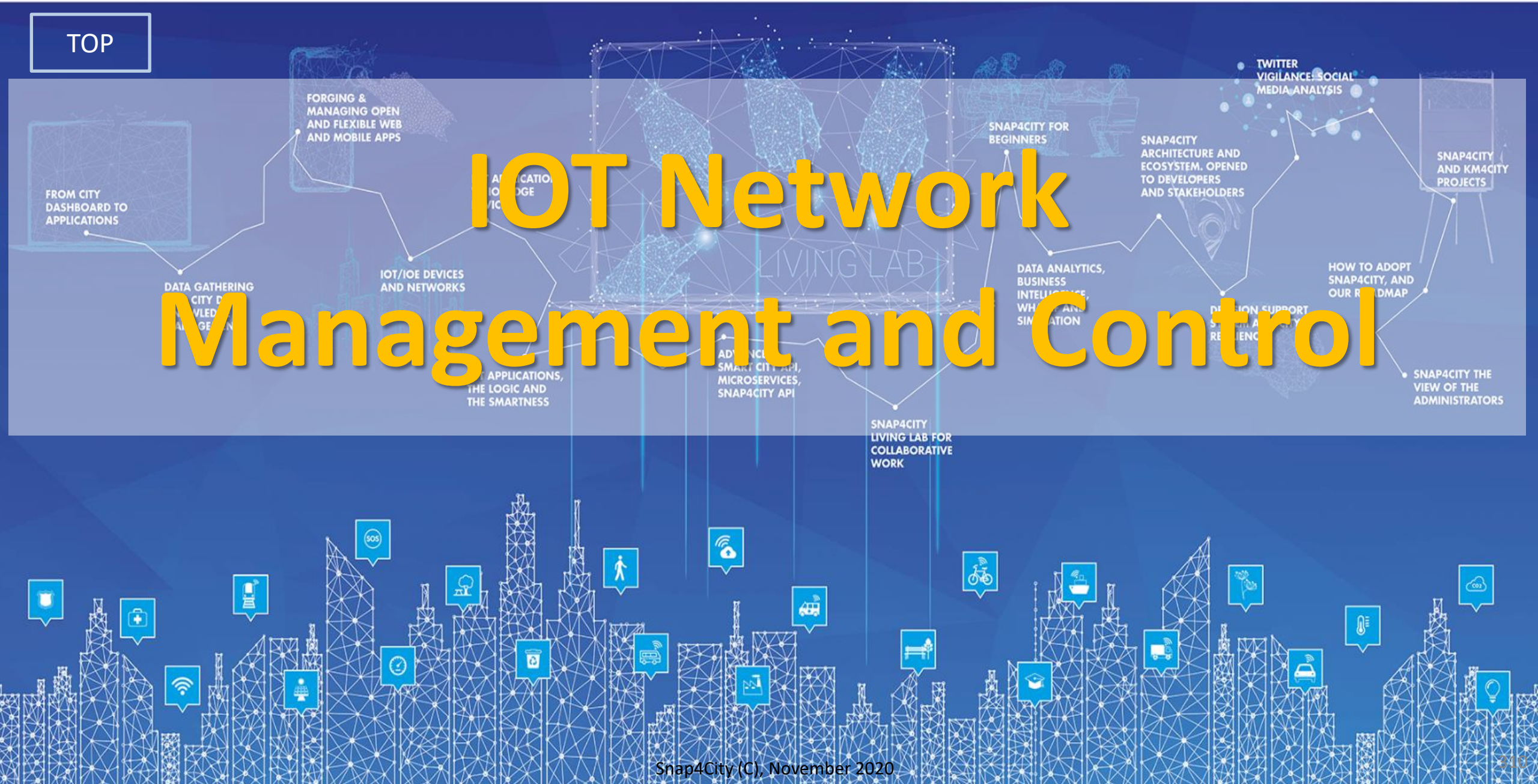
a


Node-RED
Deploy
filter nodes
Flow 1
info
input
inject
catch
status
link
mqtt
http
View
Import
Export
Search flows
Configuration nodes
Flows
Subflows
Manage palette
Settings
Keyboard shortcuts
Node-RED website
v0.17.5
Clipboard
Library
Import S4C
Import External MicroService
Import Data Analytic MicroService
Examples
Import External MicroService
last-feedback
Hotel_in_florence
Events in Finland
Places in Finland
Tourism Activities in Finland
Multilanguage Events search by location bbox
Road Weather cameras in Finland
Current weather in Antwerp
Multilanguage Events search by date
Current UV in Antwerp
Cancel
Install
Resource Manager: Process Loader
Development Tools
Management
Settings
User Management and Auditing
Help and Contacts
Documentation and Articles
My Profile
Km4City portal
DISIT Lab portal
S4CIoT
S4CLogDev
S4CView
S4CSocial
location
dashboard
UserCreated
Places in Finland

Usage of the MicroService from IOT App

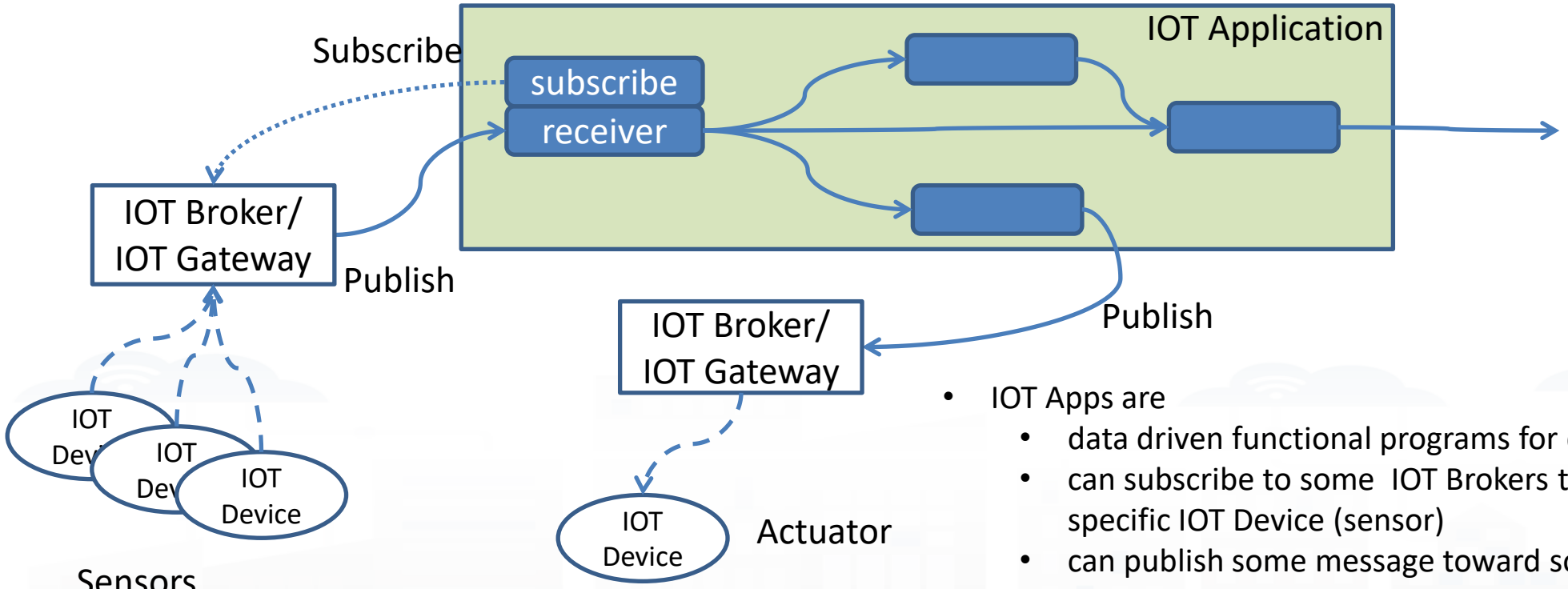


TOP



IOT Network Management and Control

IOT Basic



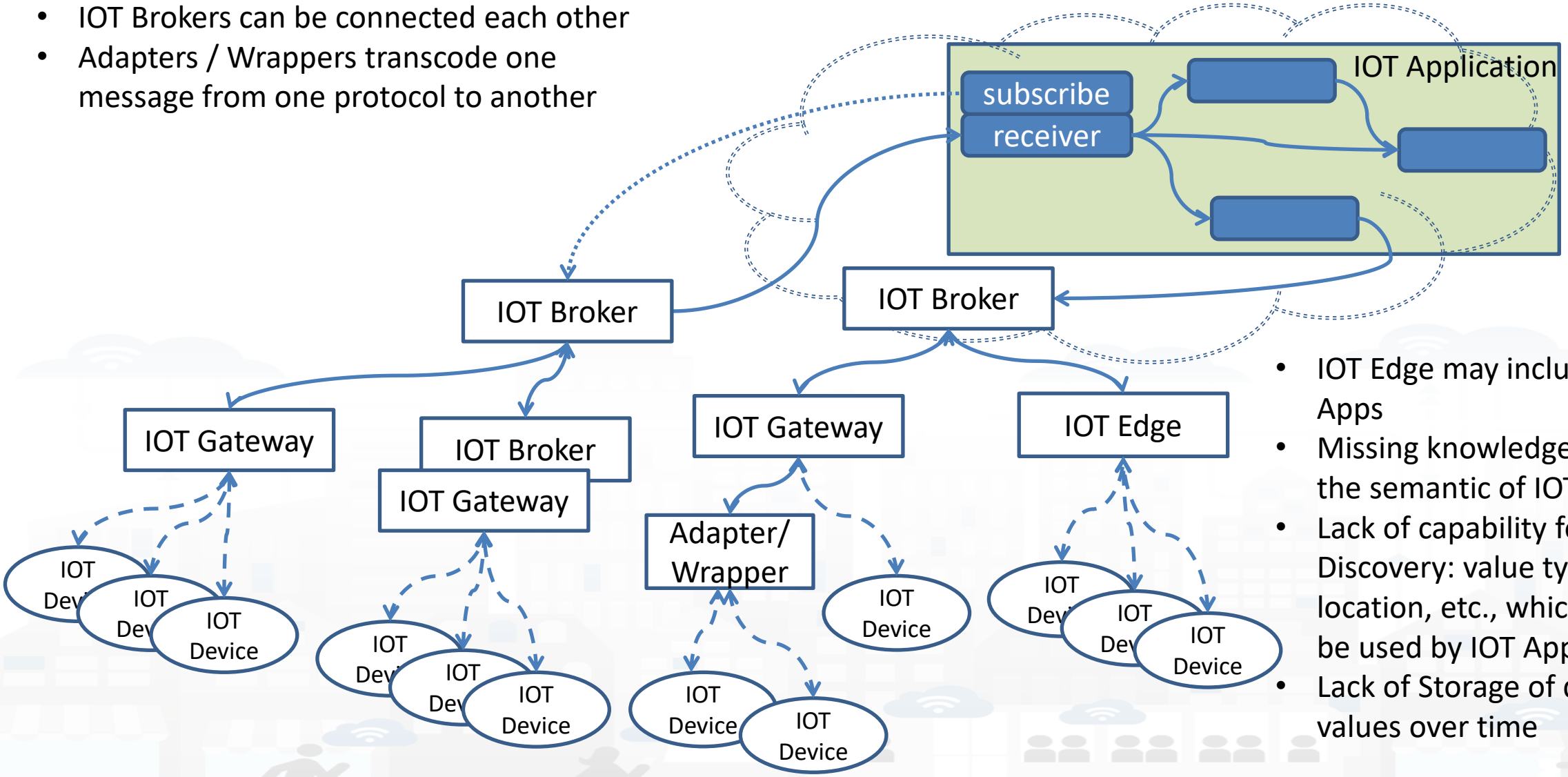
Sensors

- Sensors are programmed to send data (i) periodically, or (ii) when a relevant change occurs in the sensor value, or (iii) when events occur (for example a change of status of something), etc.
- Actuator perform some action on the field: change of status, reset, turn on something, change setting value, etc.

- IOT Apps are
  - data driven functional programs for data transformation.
  - can subscribe to some IOT Brokers to receive data in Push from a specific IOT Device (sensor)
  - can publish some message toward some IOT Device (Actuator), passing via an IOT Broker.
  - Can be used to create ADAPTERs of any kind
- Continuous lines are messages via TCP/IP
- Dashed lines are message via some radio channel (Lora, BT, Wi-Fi, …)
- IOT Brokers and IOT Gateway can be distinct servers
- IOT Brokers can be on cloud
- IOT Gateway performs the SW update, the business management, access in Push and Pull

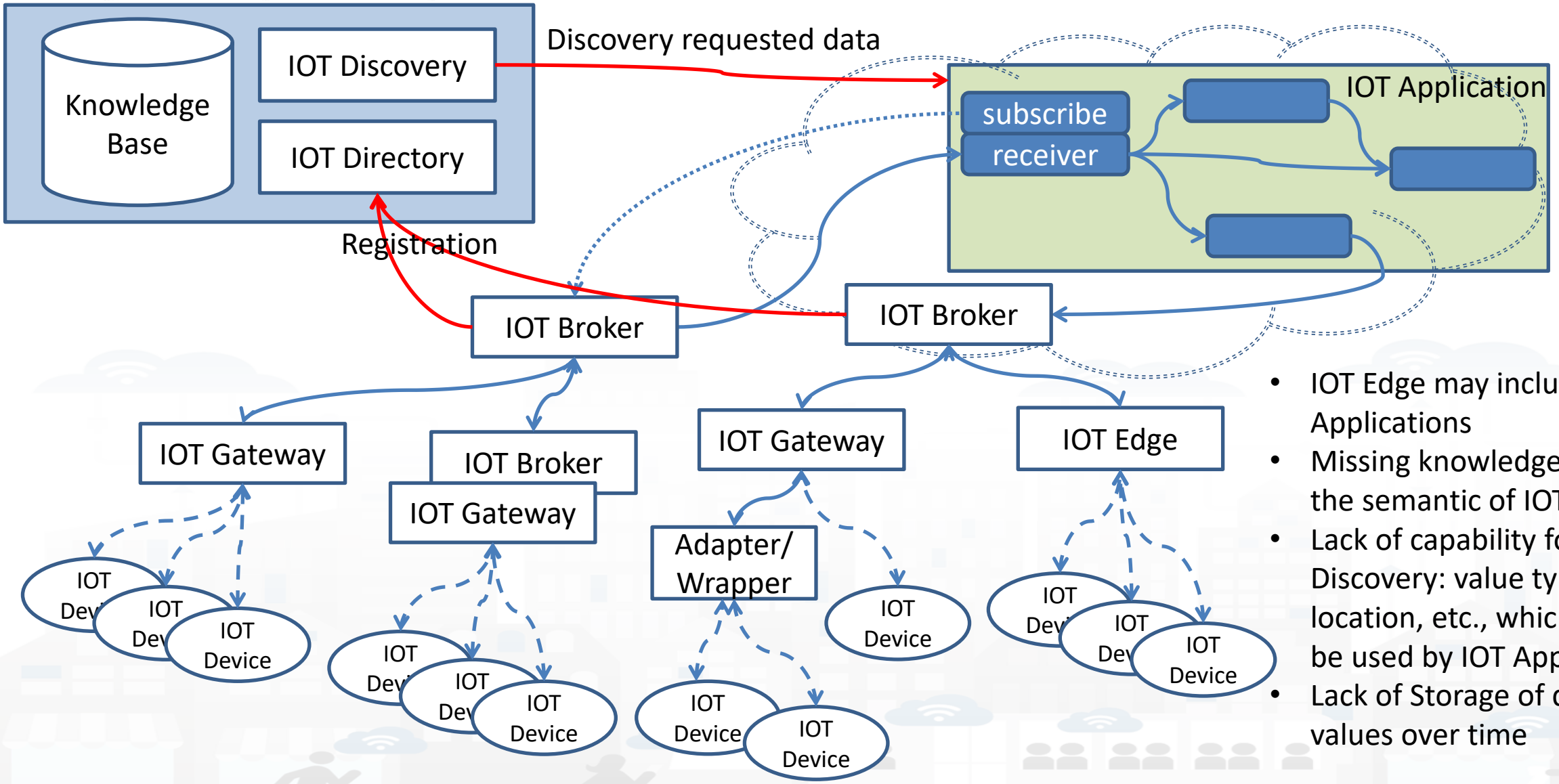
Definitions

- IOT Brokers can be connected each other
- Adapters / Wrappers transcode one message from one protocol to another



- IOT Edge may include IOT Apps
- Missing knowledge about the semantic of IOT devices
- Lack of capability for IOT Discovery: value type, location, etc., which could be used by IOT App
- Lack of Storage of data values over time

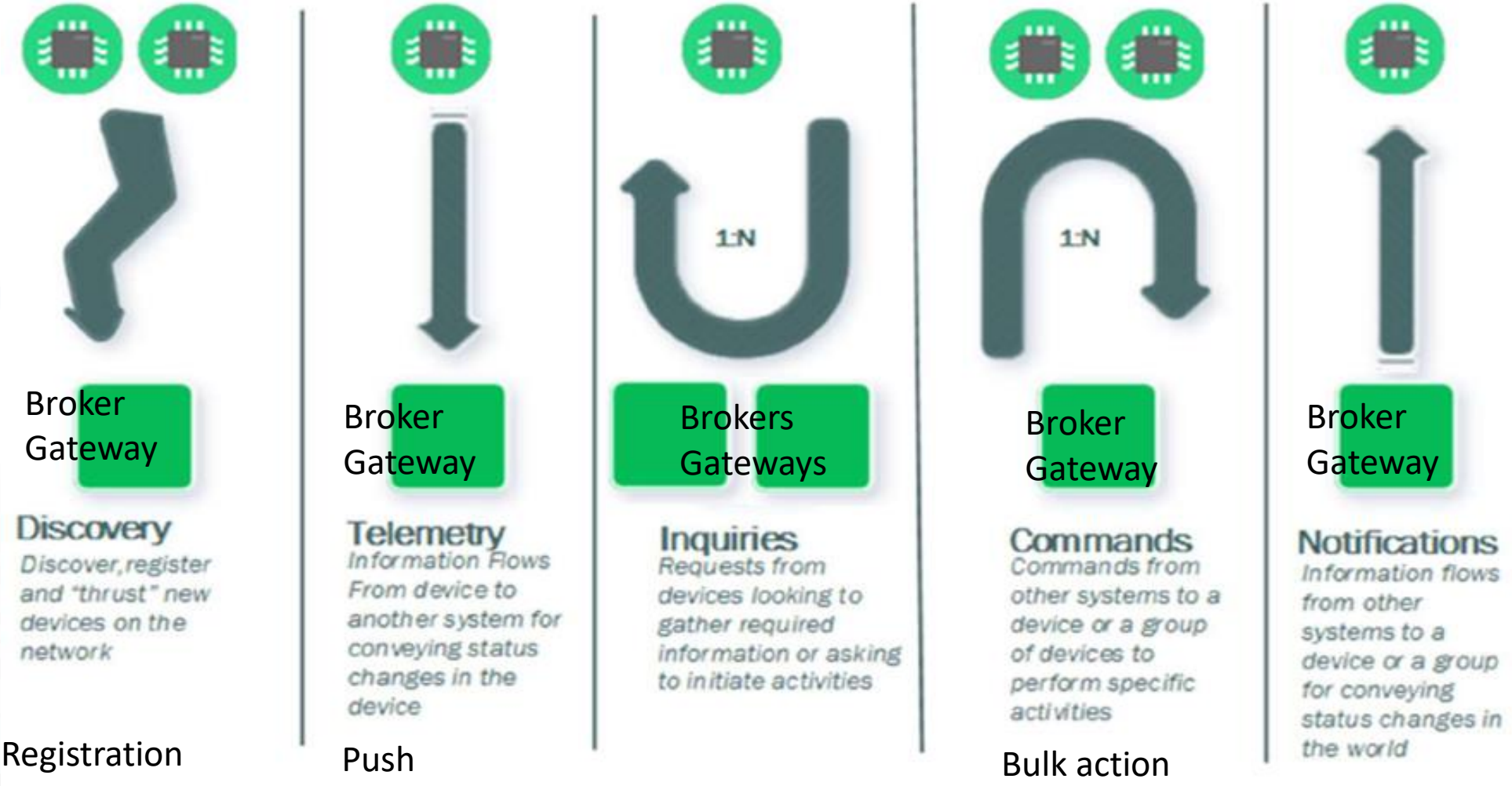
Definitions



- IOT Edge may include IOT Applications
- Missing knowledge about the semantic of IOT devices
- Lack of capability for IOT Discovery: value type, location, etc., which could be used by IOT App
- Lack of Storage of data values over time

IOT/IOE Protocols

Communication Patterns



- MQTT
- HTTP(s)
- AMQP
- COAP
- NGSI
- OneM2M
- WebSockets
- ……..
- Etc.

Note on Communication patterns

- Not all Communication Patterns are supported by all Protocols
- Protocols implement Patters, + formats, + sequences, etc.
- They are referred at level of communications
  - IOT Device ← → IOT Gateway ←→IOT Broker
- IOT Protocols mostly used at level of IP are:
  - NGSI V1/2, MQTT, COAP, AMQP, OneM2M, WS, ModBUS,
- Radio protocols are: Lora, ZigBee, 3G, Wi-Fi, etc.
- Formats: JSON, Geo-JSON, Linked Data, XML, CSV,

IOT Networks on Snap4City



SNAP4CITY

KM4CITY

URBAN PLATFORM: SMART CITY IOT AS A SERVICE AND ON PREMISE

100% OPEN SOURCE

APPLIANCES CONTAINERS



- LOCAL GOVERN
- STAKEHOLDERS
- CITY USERS
- IN-HOUSE
- ENERGY OPERATORS
- MOBILITY OPERATORS
- COMMERCIAL OPERATORS
- SECURITY OPERATORS
- INDUSTRIES
- RESEARCHERS
- START-UPS
- ASSOCIATIONS

- GDPR
- SECURITY
- PRIVACY
- ASSESSMENT
- AUDITING
- PENTESTED

- OPEN IOT DEVICES
- IOT EDGE
- IOT GATEWAY
- PAX COUNTERS
- IOT BUTTONS

- TEST CASES, SCENARIOUS, VIDEOS, HACKATHONS
- OPEN SOURCES, COMMUNITY OF CITIES
- TRAINING TUTORIALS, COMMUNITY MANAGEMENT

IOT APPLICATIONS - INSTANT APPS



DATA DRIVEN APPLICATIONS • REAL TIME PROCESSING • BATCH PROCESSING • ANY PROTOCOL & FORMAT

DASHBOARDS & APPLICATIONS



CONTROL ROOM • SITUATION ROOM • OPERATOR DASHBORDS • BUSINESS INTELLIGENCE • WHAT-IF ANALYSIS • DECISION SUPPORT • SIMULATIONS • RISK ANALYSIS • RESILIENCE ANALYSIS

MOBILE & WEB APPLICATIONS



DEVELOPMENT KIT • SUGGESTIONS • MOBILE APPS • MONITORING PANELS • PLATFORM UTILITIES • READY TO USE SMART APPLICATIONS

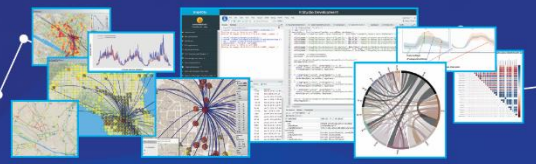
MICROSERVICES & ADVANCED SMART CITY API

LIVING LAB - DEV TOOLS - COWORKING

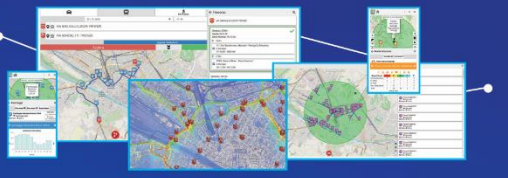


IOT DIRECTORY • SERVICE MAP • RESOURCE MANAGER • DATA GATE • R STUDIO • ETL

BIG DATA - DATA ANALYTICS



DATA ANALYTICS TOOLS - MICRO-APPLICATIONS



PREDICTIONS • ANOMALY DETECTION • WHAT-IF ANALYSIS • TRAFFIC FLOW RECONSTRUCTION • ORIGIN-DESTINATION MATRICES • SOCIAL MEDIA ANALYSIS • OFFER VS DEMAND ANALYSIS • ENVIRONMENTAL DATA ANALYSIS AND PREDICTIONS • REAL TIME HEATMAPS • ROUTING • ALERTING • EARLY WARNING • PERSONAL AND VIRTUAL ASSISSTANTS • SMART SOLUTIONS • SMART SHARING • PARTECIPATORY

KM4CITY DATA AGGREGAT KNOWLEDGE BASE - EXPERT SYSTEM OF THE CITY – BIG DATA STORE

IOT MNG - DATA MNG - DATA INSPECTOR – PROCESS MNG – USER ENGAGEMENT – GDPR MNG ...

GIS	CITY UTILITIES	OPEN DATA	LEGACY & EXTERNAL SERVICES	PERSONAL DATA	IOT / IOE	BROKERS	KPI	INDUSTRY 4.0	SOCIAL MEDIA



SNAP4CITY

KM4CITY

URBAN PLATFORM: SMART CITY IOT AS A SERVICE AND ON PREMISE

100% OPEN SOURCE

APPLIANCES CONTAINERS



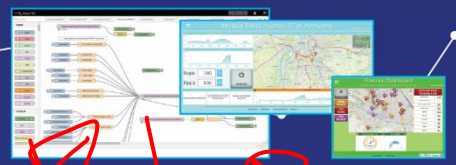
- LOCAL GOVERN
- STAKEHOLDERS
- CITY USERS
- IN-HOUSE
- ENERGY OPERATORS
- MOBILITY OPERATORS
- COMMERCIAL OPERATORS
- SECURITY OPERATORS
- INDUSTRIES
- RESEARCHERS
- START-UPS
- ASSOCIATIONS

- GDPR
- SECURITY
- PRIVACY
- ASSESSMENT
- AUDITING
- PENTESTED

- OPEN IOT DEVICES
- IOT EDGE
- IOT GATEWAY
- PAX COUNTERS
- IOT BUTTONS

- TEST CASES, SCENARIOUS, VIDEOS, HACKATHONS
- OPEN SOURCES, COMMUNITY OF CITIES
- TRAINING TUTORIALS, COMMUNITY MANAGEMENT

IOT APPLICATIONS - INSTANT APPS



DATA DRIVEN APPLICATIONS • REAL TIME PROCESSING • BATCH PROCESSING • ANY PROTOCOL & FORMAT

DASHBOARDS & APPLICATIONS



CONTROL ROOM • SITUATION ROOM • OPERATOR DASHBORDS • BUSINESS INTELLIGENCE • WHAT-IF ANALYSIS • DECISION SUPPORT • SIMULATIONS • RISK ANALYSIS • RESILIENCE ANALYSIS

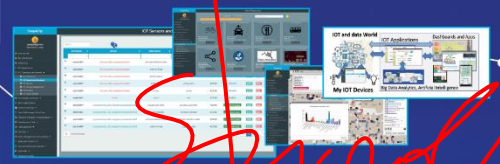
MOBILE & WEB APPLICATIONS



DEVELOPMENT KIT • SUGGESTIONS • MOBILE APPS • MONITORING PANELS • PLATFORM UTILITIES • READY TO USE SMART APPLICATIONS

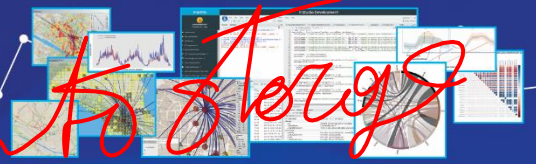
MICROSERVICES & ADVANCED SMART CITY API

LIVING LAB - DEV TOOLS - COWORKING

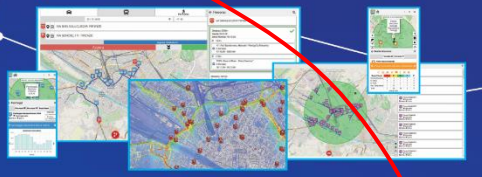


IOT DIRECTORY • SERVICE MAP • RESOURCE MANAGER • DATA GATE • R STUDIO • ETL

BIG DATA - DATA ANALYTICS



DATA ANALYTICS TOOLS - MICRO-APPLICATIONS



PREDICTIONS • ANOMALY DETECTION • WHAT-IF ANALYSIS • TRAFFIC FLOW RECONSTRUCTION • ORIGIN-DESTINATION MATRICES • SOCIAL MEDIA ANALYSIS • OFFER VS DEMAND ANALYSIS • ENVIRONMENTAL DATA ANALYSIS AND PREDICTIONS • REAL TIME HEATMAPS • ROUTING • ALERTING • EARLY WARNING • PERSONAL AND VIRTUAL ASSISTANTS • SMART SOLUTIONS • SMART SHARING • PARTECIPATORY

KM4CITY DATA AGGREGAT KNOWLEDGE BASE - EXPERT SYSTEM OF THE CITY - BIG DATA STORE

IOT MNG - DATA MNG - DATA INSPECTOR – PROCESS MNG – USER ENGAGEMENT – GDPR MNG ...

GIS	CITY UTILITIES	OPEN DATA	LEGACY & EXTERNAL SERVICES	PERSONAL DATA	IOT / IOE	BROKERS	KPI	INDUSTRY 4.0	SOCIAL MEDIA



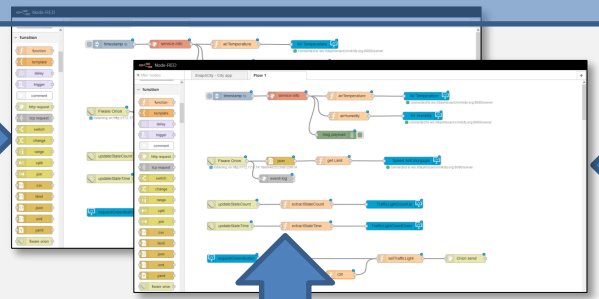
Snap4City: Builder of Sentient Cities Solutions

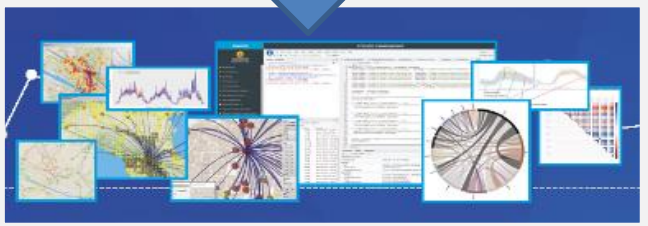
Dashboards with data driven IOT Applications enforcing intelligence

IOT and data World

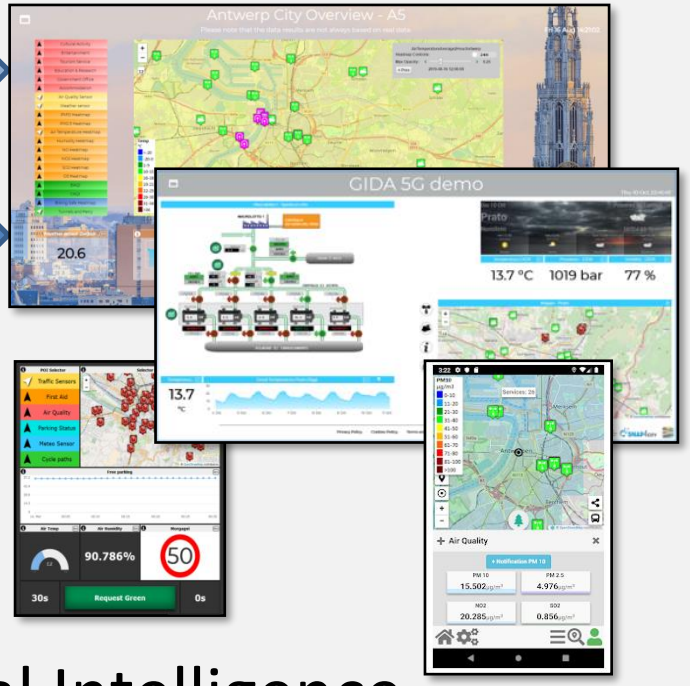


IOT Applications



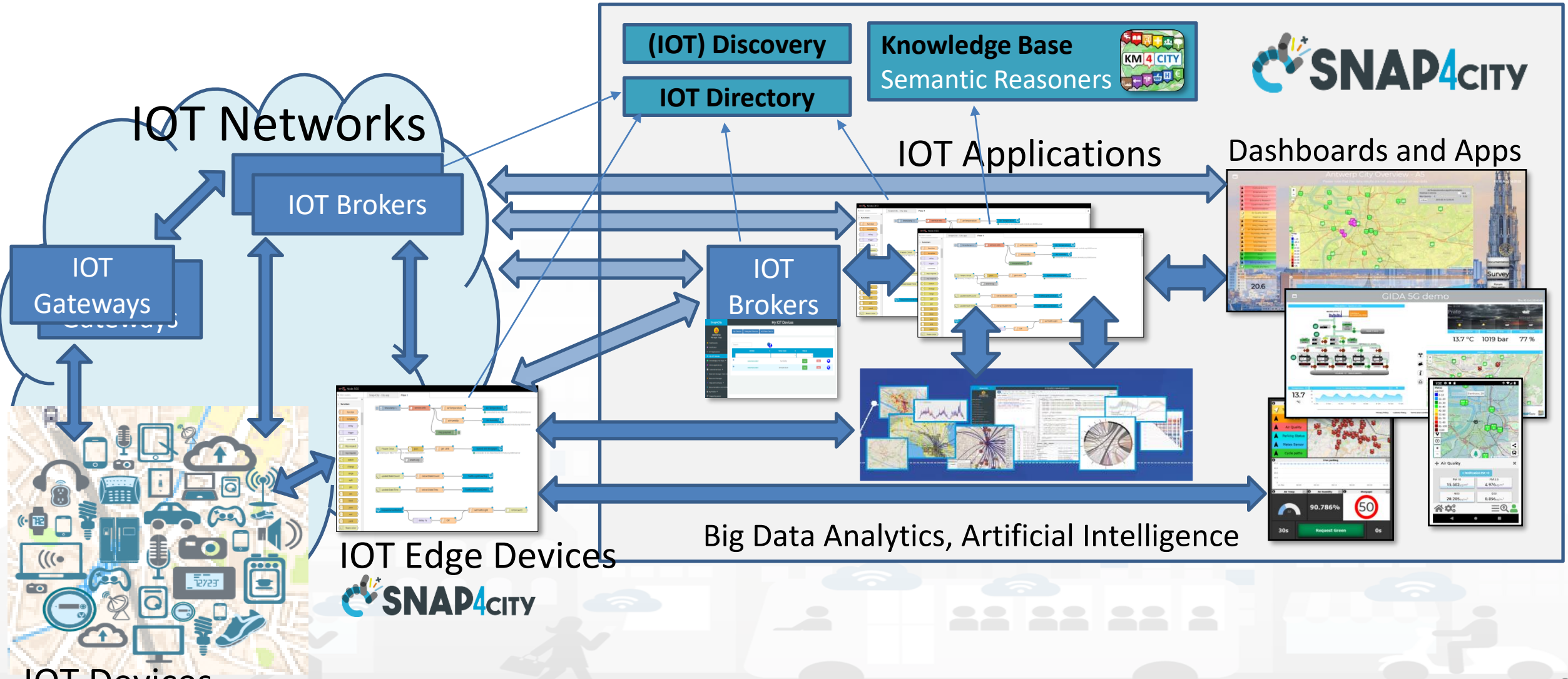


Dashboards and Apps



Big Data Analytics, Artificial Intelligence

Snap4City: IOT Directory and data/device Discovery



Standards and Interoperability

Compliant with: AMQP, COAP, MQTT, OneM2M, HTTP, HTTPS, TLS, Rest Call, SMTP, TCP, UDP, NGSI, LoRa, LoRaWan, TheThingsNetwork, SigFOX, DATEX II, SOAP, WSDL, Twitter, FaceBook, Telegram, SMS, OLAP, MySQL, Mongo, HBASE, SOLR, SPARQL, EMAIL, FTP, FTPS, WebSocket, WebSocket Secure, ModBUS, OPC, GML, RS485, RS232, WFS, WMS, ODBC, JDBC, Elastic Search, Phoenix, XML, JSON, CSV, db, GeoJSON, Enfuser FMI, Android, Raspberry Pi, Local File System, ESP32, Libelium, IBIMET/IBE, OBD2, SVG, XLS, XLSX, TXT, HTML, CSS, KNX, Enocean, Zigbee, DALI, ISEMC, Alexa, Sonoff, HUE Philips, Tplink, etc.

https://www.snap4city.org/65

Snap4City vs Formats

- Snap4City is capable to ingest and work with any format:
  - Data **exchange**: JSON, GeoJSON, XML, HTML, HTML5, DATEX, GTFS, binary, etc.
  - **Table**: CSV, XLSX, XLS, database, …
  - Any **archive** file format: zip, rar, 7z, tgz, …
  - Any **image** format: png, gif, tiff, ico, jpg, …
  - Any **video** format: mp4, avi, mov, …
- Search the format you need to cope on the search box of Snap4City portal!

IOT Network Manager vs Final User

Network of IOT Brokers

IOT Broker

External

Internal

IOT Directory

Registering

Discovering

Knowledge Base, Km4City

Knowledge and Storage
Data from the Field and City

Browsing

ServiceMap
Knowledge Base

IOT Network Manager

Discovering

Prepare
Deep Manage

Exploit

My IOT Device

Register

IOT Application

Final user Manager

Dashboard Wizard

Main Features of the Snap4City IOT Directory:

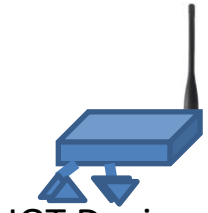
- **IOT Directory is a technology of Snap4City ONLY**
- **Register IOT Brokers**
  - Different kind of Brokers, different kinds of authentications and protocols
  - Registered IOT Orion Brokers can be queried for collecting their managed devices (typically for External IOT Brokers), so that those IOT Devices are registered
  - IOT Brokers/Gateways are registered on NIFI to send messages into the Data Shadow, automatically
- **Register IOT Devices**: singularly or at groups (in Bulk)
  - Registration can be custom or based on IOT Device Model
  - IOT Edge are registered as IOT Devices as well
  - Registered IOT Devices are saved into local DataBase and into the Knowledge Base
- **Provide support for security aspects**:
  - Generation of Certificates, Keys, etc., according to the model
  - Collection of keys when IOT devices are on some IOT Gateway or Second Level IOT Broker.
- **Manage Ownership and Delegation** for
  - IOT Brokers, IOT Devices, IOT Device Values also called Sensors/actuators, IOT Device Models

IOT Directory Features vs Users Roles

Entities	what	By using IOT Directory and:	Manager	AreaManager	ToolAdmin/ RootAdmin	IOT App microservices
IOT Sensor/Actuator	Browse, use	Several Tools	X	X	X	Yes
	Delegate	API, ..	X	X	X	
	Discovery	KB, API, ..	X	X	X	Yes
IOT Devices	Browse, use	Several Tools	X	X	X	Yes (use)
	Create, change, delete	API, ..	X	X	X	
	Register in Bulk	API, ..		X	X	
	Delegate, Change Owner	API, ..	X	X	X	Yes
	Discovery	KB, API, ..	X	X	X	Yes
IOT Device Model	Browse, Use		X	X	X	
	Create, change, delete			X	X	
	delegate, change ownership			X	X	
IOT Broker	Browse, use		use	Browse, use	X	Yes (use)
	Register/change/Delete				X	
	Delegate				X	
	Periodic Update				X	

In which case you are?

https://www.snap4city.org/drupal/node/474

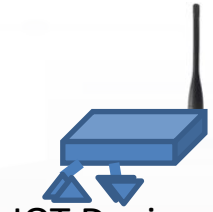


IOT Broker — SNAP4CITY — Case B2



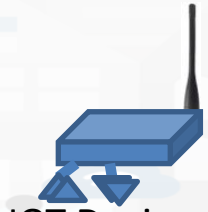
IOT Broker → SNAP4CITY — Case B1

i) Registered IOT Broker on Snap4City



a) Registered IOT Device on Broker

IOT Broker — SNAP4CITY — Case A1.2



a) Registered IOT Device on Broker

IOT Broker → SNAP4CITY — Case A2

i) Registered IOT Broker on Snap
ii) Registered IOT Device on Snap

A. **Already registered on an IOT Broker** of your/city organization. In this case, who provided you the IOT Device may have provided also at least: an IOT Device Identifier, a description of the data produced by the Device, the protocol kind and the IOT Broker in which it is registered, etc. **For example**: device ID: es286481295, temperature and humidity, NGSI, the "orionFinland" IOT Broker on https://ngsi.fvh.fi or the "Antwerp" IOT Broker on https://ext-api-gw-p.antwerpen.be. **In order to** exploit the data of your IOT Device in the Platform, you

1. have an **IOT Device which is registered** on an **IOT Broker** (for example, you received with the IOT Device the name of the IOT Broker) that is registered to Snap4City. You have two cases:
   1. **the IOT Device has been **already registered** on Snap4City by the organization/city**. This case can be verified by using the steps described in:
      - See in this case: HOW TO: verify if an IOT device is registered and accessible for me.
      - if the IOT device is registered please note that you are in case A3, if not, go at case A1.2
   2. **the IOT Device is **not yet registered** on Snap4City** (for example when a IOT Broker is managed by a third organization for security aspects, for example the FHV or DIGIPOLIS, or IMEC, ...), they given to your the device to test on different platforms.
      - See in this case: HOW TO: Add an IOT Device on Snap4City platform that is already registered on an external IOT Broker
   3. need to access IOT Device data **without registering the IOT Device in the Snap4City platform and Broker**, you need to know some configuration parameter of the IOT Broker and IOT Device, and Snap4City IOT App can get data directly from the IOT broker of your device without the need of having the Broker officially registered on snap4City.
      - See in this case: HOW TO: add IOT Device data source from external broker to the platform.
2. have an **IOT Device which is already registered to an internal Snap4City IOT Broker** (a IOT Broker managed by Snap4City for security aspects). In this case, the IOT Device and corresponding data are immediately accessible, and you can find them into the list of your data in the Data Inspector view, for Dashboards, etc., go in the Data Inspector to search your data by GPS location, name, nature as you like.

A. **Already registered on an IOT Broker** … see above

B. **Not registered to an IOT Broker**. In this case, you need to know, at least, how the IOT Device works and how it can be internally configured to communicate with an IOT Broker: to authenticate, register, etc. So that you need to know: an IOT Device Identifier, a description of the data produced by the device, the protocol, etc. For example: device ID: 286481295, temperature and humidity, NGSI with basic authentication,

1. **In this case,** the first step is to register the IOT Device to an IOT Broker. Snap4City offers you a number of **Snap4City IOT Brokers** compliant with different protocols to which you can connect your device. To this end, please follow this tutorial:
   - See in this case: HOW TO: Add an IOT Device on Snap4City platform by registering it on an Internal Snap4City IOT Broker
2. in alternative you can find some other brokers in your area according to the protocol of your device.
3. **Once registered the IOT Device to an IOT Broker please restart from case (A); if you registered with a Snap4City IOT Broker it will be easy an (A2).**

How to setup and IOT Data Stream

Managers/AreaManagers:

1. Register the IOT Broker you want to use.
   - If you do not have one, you can ask one to Snap4City
2. Register the IOT Device you want to use.
   - If it is only one Device to reg, you can do it manually,
   - if they are many, we suggest you to create an IOT Device Model, then register the device (only AreaManagers)
3. Use IT

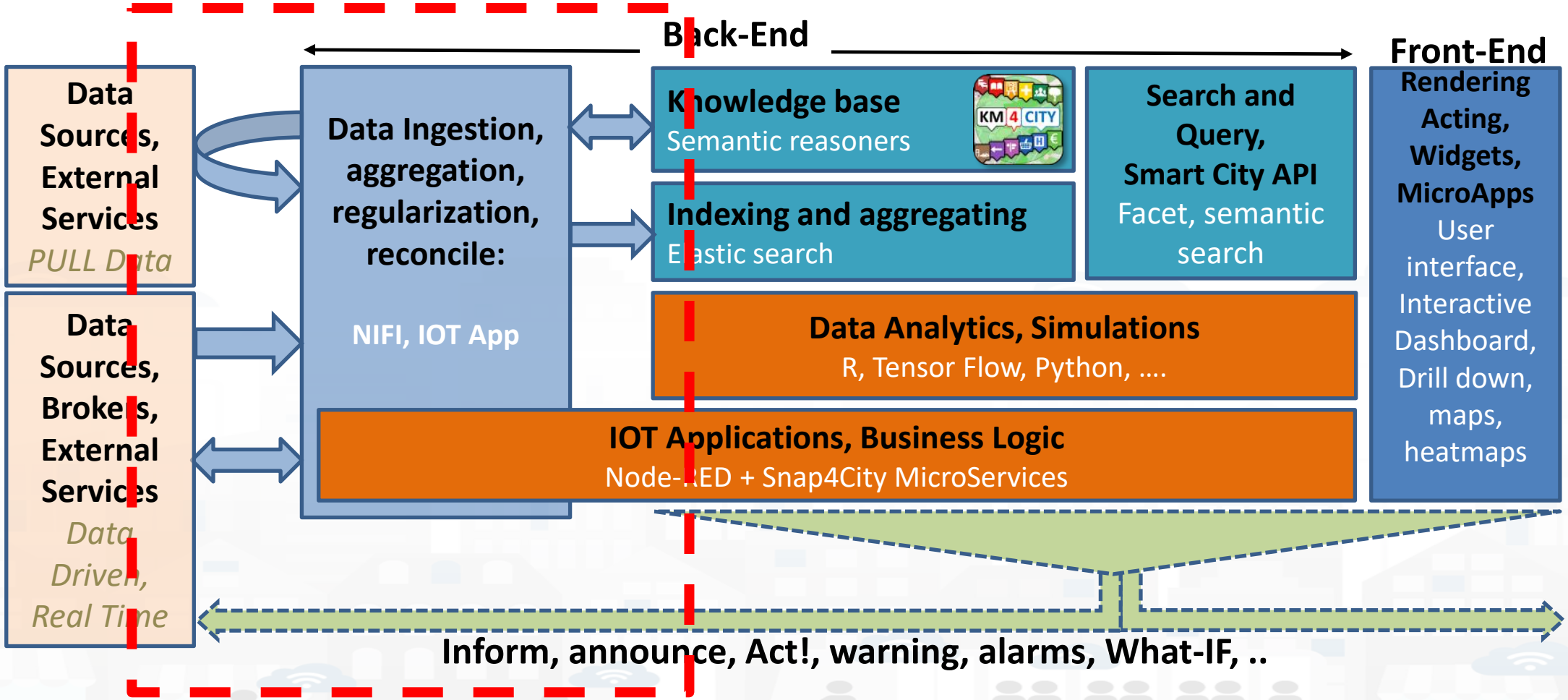
Administrators:

1. Register the IOT Broker you want to use, or use one already registered.
   - If the IOT Orion Broker has IOT Devices registered in you can use the procedure for automated registration (from your Broker to the IOT Directory and KB), with rule for transformation, etc.
   - If not see points 2 and/or 3
2. Register a single IOT Device manually
3. Register a group of IOT Devices
   - create a IOT Device Model
   - Create a CVS file for Registering devices in Bulk
4. Use IT

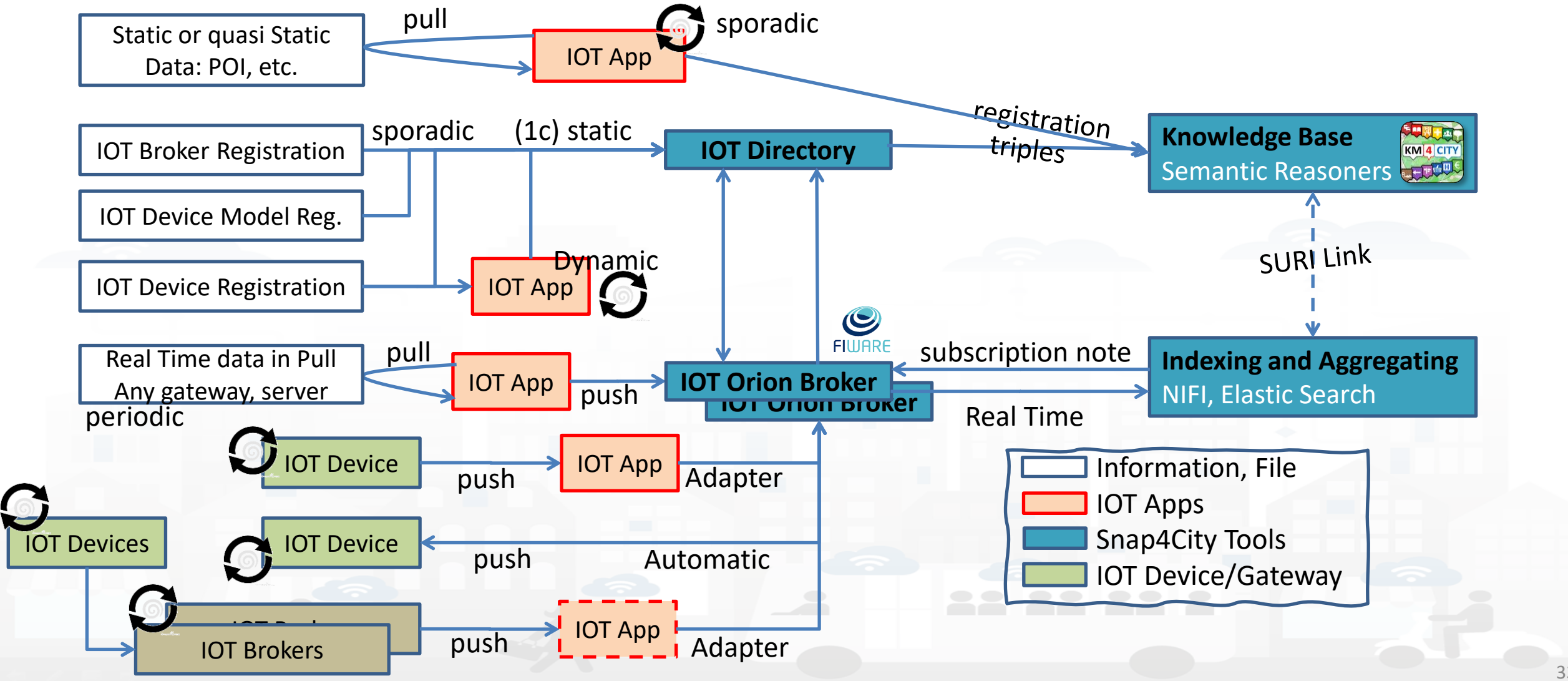
Data Ingestion Strategy



Snap4City Architecture vs Data Ingestion

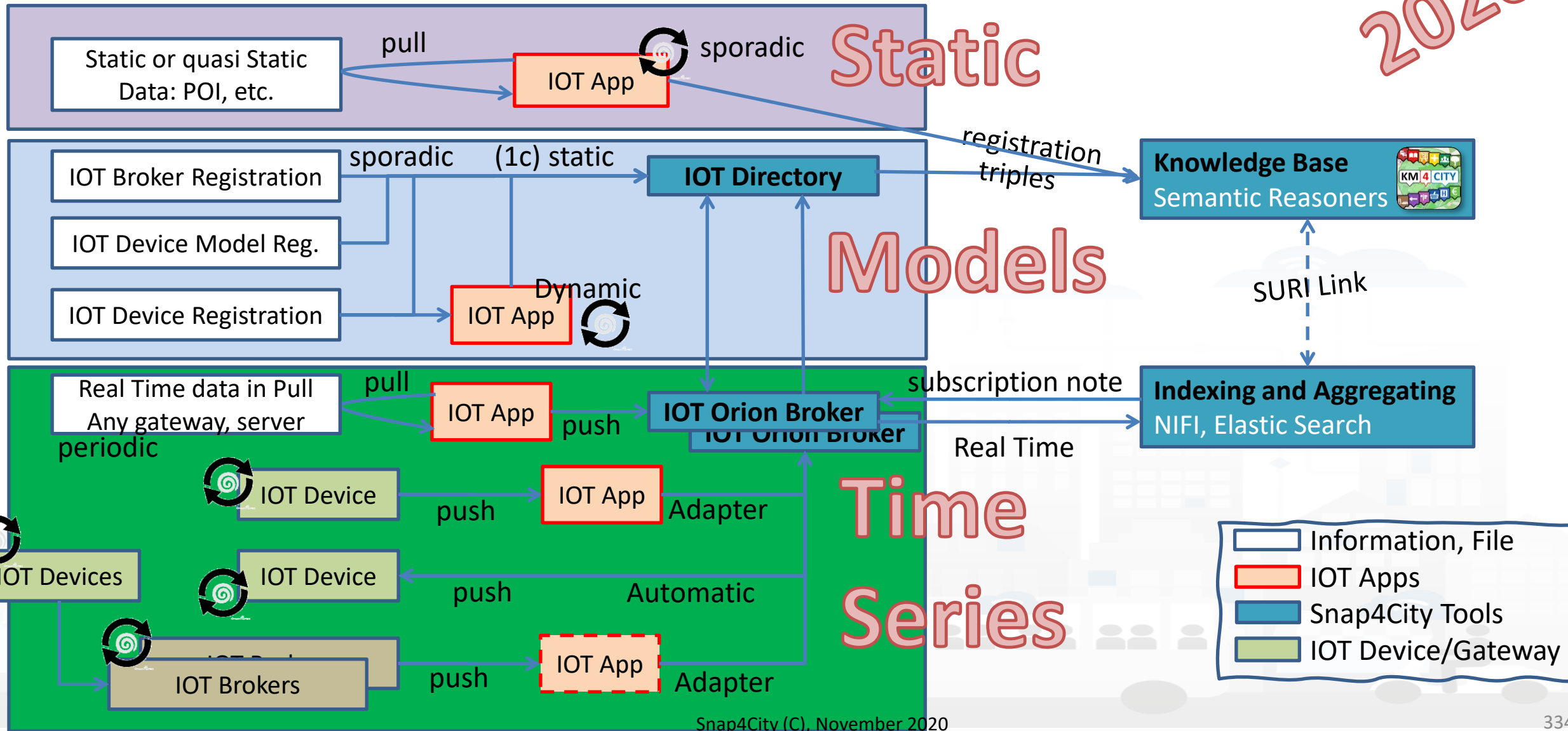


Snap4city Data Ingestion Flow Diagram



Snap4city Data Ingestion Flow Diagram

2020



On the Flow Diagram

- **Static** (unified model for multidomain indexing)
  - Geodata, Open Data as POI, Data Bases, records, etc.
  - They change over time sporadically
- **Models** (Registration of IOT Device Models, IOT Devices, Brokers)
  - Registration of entities with their metadata and data structures
    - Brokers, Devices, structures of real time data, machine models for IOT Industry 4.0, sensor models/structure, etc.
- **Time Series (dynamic data)**
  - Any instance of IOT Devices over time and space with any kind of entity relationship each other and with other city entities

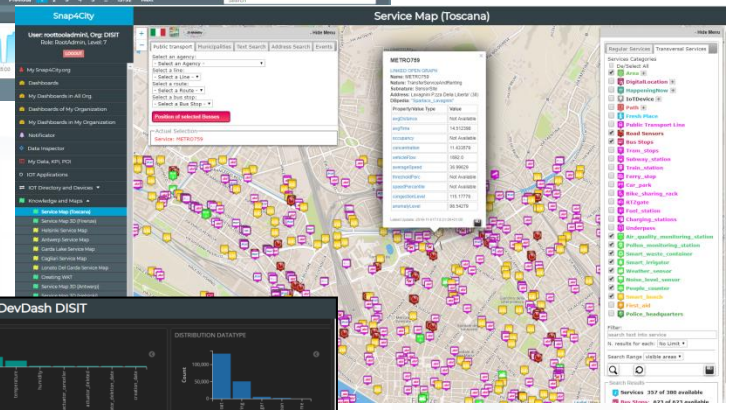
Checking data ingestion results

Knowledge base
Semantic reasoners

- Data Inspector
- ServiceMap, ASCAPI
  - LOG / LOD viewer
  - Super Service Map
- IOT Broker
- ACAPI: Swagger
- IOT Directory



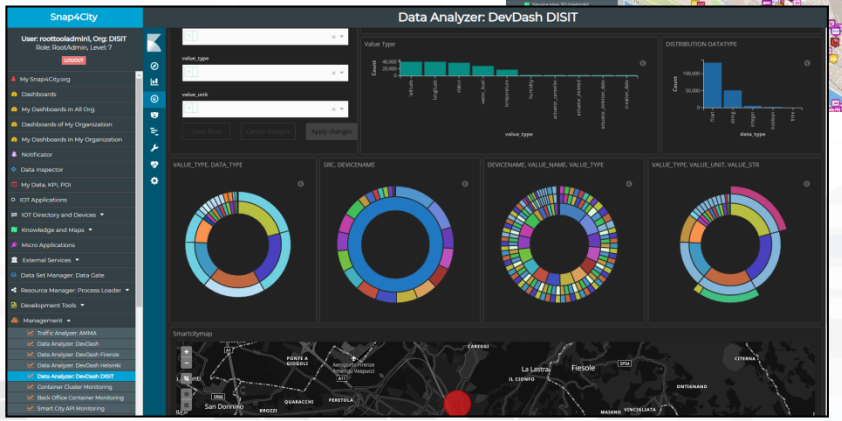
Data Inspector



ServiceMap

Indexing and aggregating
Elastic search

- Data Inspector
- ServiceMap, ASCAPI
- DevDash (Kibana)
- Elastic Search



DevDash

Some functionalities are limited to certain roles

Additional Notes

- The **IOT Orion Broker** can be feed by means of
  - **IOT App** of Snap4City (to implement Agents and/or Adapters)
  - **IOT Agents** and/or **NGSI Adapters** by FiWare for different protocols
  - IOT brokers of any kind

- **IOT App, IOT Agents, Adapters and IOT Brokers** can
  - be on IOT Edge
  - be on other clouds and services
  - work on a large range of different protocols and kinds
  - have or not Snap4City tools and libraries installed

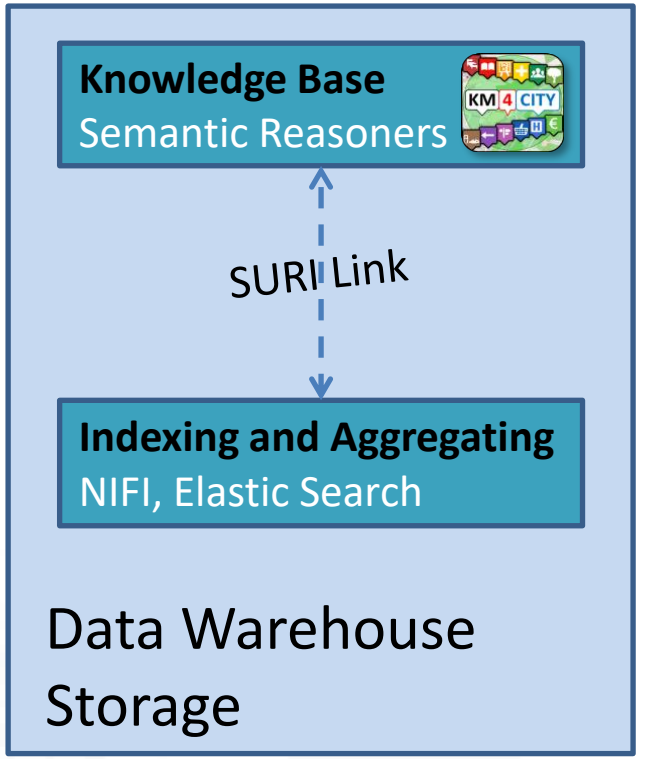
Additional Notes

- The **Internal IOT Orion Brokers** at Snap4City IOT Orion Broker are used as gate for data ingestion and actuations. Since they are
  - connected with the IOT Directory and discovery of the Knowledge Base to make easy the production of Dashboards by wizard, Data Inspector;
  - Synchronized automatically with NIFI/ElasticSearch for the Automated Data Shadow and Indexing
  - Ready to be used by IOT App to subscribe for creating even driven IOT Apps, on IOT Edge and Cloud, etc...
  - Compatible and harmonized with FiWare networks

Direct Ingestion

- **Direct Data Ingestion is also possible**

- **Data Storage:** KB, and I&A, reported on right side can be acted via API REST Call
  - for direct feeding data into store and retrieval,
  - which can be exploited by:
    - IOT Applications
    - applications in Python, R Studio, Java



IOT Broker Registration



IOT Brokers

Snap4City

User: roottooladmin1, Org: DISIT
Role: RootAdmin, Level: 7
LOGOUT

- My Snap4City.org
- Dashboards
- My Dashboards in All Org.
- Dashboards of My Organization
- My Dashboards in My Organization
- Notificator
- Data Inspector
- My Data, KPI, POI
- My Groups of Entities
- IOT Applications
- IOT Directory and Devices
  - My IOT Devices
  - IOT Sensors and Actuators
  - IOT Devices
  - IOT Devices Management
  - IOT Brokers
  - IOT Device Models
  - IOT Devices Bulk Registration
  - IOT Broker Periodic Update setting
  - IOT Orion Broker Mapping Rules
- Knowledge and Maps
- Micro Applications
- External Services
- Data Set Manager: Data Gate
- Synoptics
- Resource Manager: Process Loader
- Development Tools
- Management

IOT Brokers

New IOT Broker

Show 10 entries

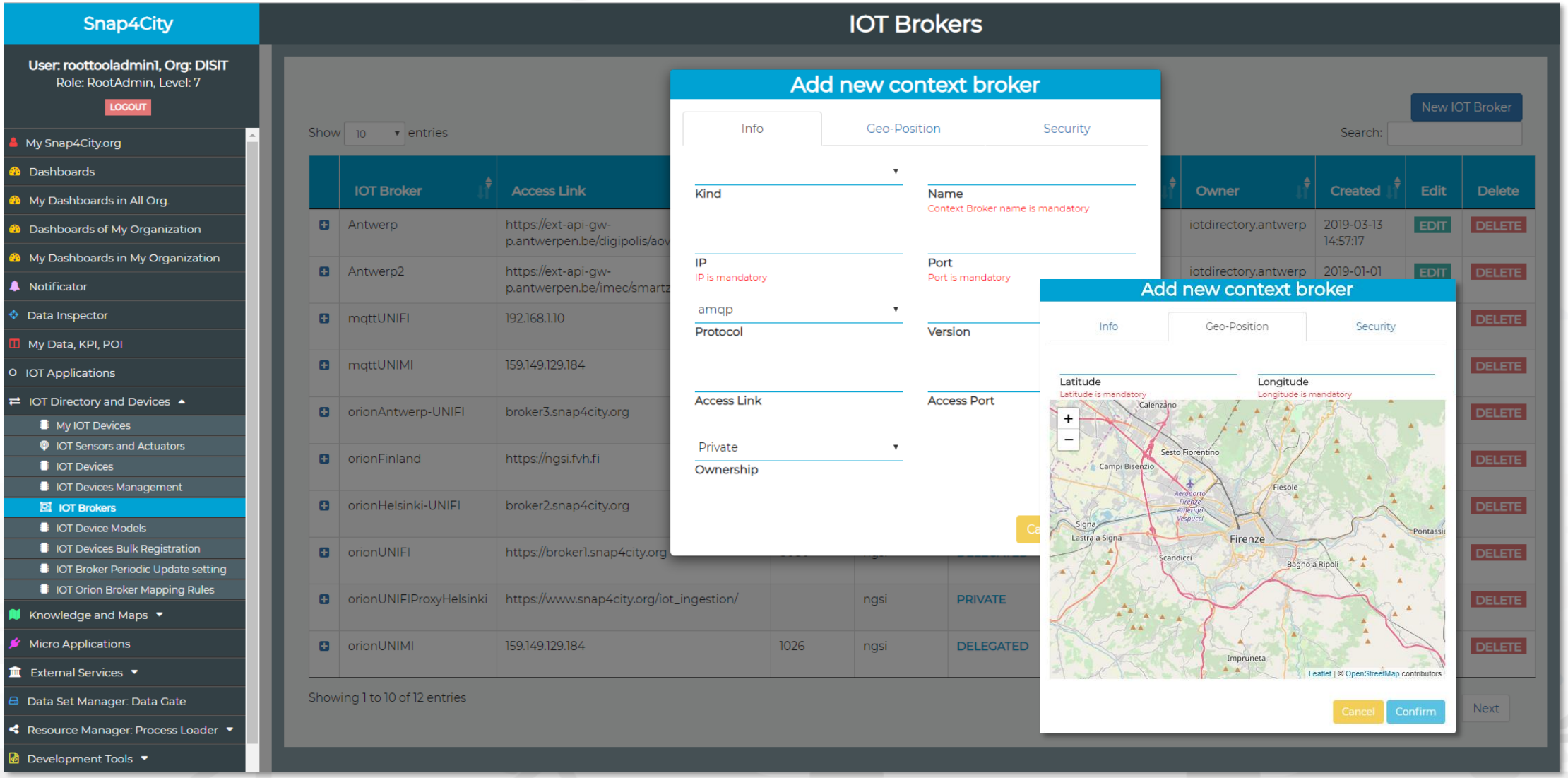
Search:

	IOT Broker	Access Link	Access Port	Protocol	Ownership	Organization	Owner	Created	Edit	Delete
+	Antwerp	https://ext-api-gw-p.antwerpen.be/digipolis/aovmma/v1/entities		ngsi	DELEGATED	Antwerp	iotdirectory.antwerp	2019-03-13 14:57:17	EDIT	DELETE
+	Antwerp2	https://ext-api-gw-p.antwerpen.be/imec/smartzone/v1		ngsi	DELEGATED	Antwerp	iotdirectory.antwerp	2019-01-01 00:00:00	EDIT	DELETE
+	iotobsf-smartbed	192.168.1.47	8443	ngsi	DELEGATED	SmartBed	angelo.difino	2019-11-29 15:31:51	EDIT	DELETE
+	mqttUNIFI	192.168.1.10	1883	mqtt	DELEGATED	DISIT	iotdirectory.disit	2018-02-07 15:14:39	EDIT	DELETE
+	mqttUNIMI	159.149.129.184	1884	mqtt	DELEGATED	DISIT	iotdirectory.disit	2018-04-30 16:49:05	EDIT	DELETE
−	orionAntwerp-UNIFI	broker3.snap4city.org	8080	ngsi	PUBLIC	Antwerp	iotdirectory.antwerp	2019-06-03 14:25:16	EDIT	DELETE
	IP: 192.168.1.12					Port: 1026				
	Latitude: 43.76666					Longitude: 11.26242				
	Login:					Password:				
	SHA:									
+	orionFinland	https://ngsi.fvh.fi		ngsi	PUBLIC	Helsinki	iotdirectory.helsinki	2018-11-21 16:05:24	EDIT	DELETE
+	orionFirenze-UNIFI	192.168.1.17	8443	ngsi	PRIVATE	Firenze	iotdirectory.firenze	2019-10-28 10:01:53	EDIT	DELETE
+	orionHelsinki-UNIFI	broker2.snap4city.org	8080	ngsi	PUBLIC	Helsinki	iotdirectory.helsinki	2019-06-03 10:26:34	EDIT	DELETE
+	orionLonatoDelGarda-UNIFI	192.168.1.46	8443	ngsi	DELEGATED	LonatoDelGarda	iotdirectory.ldg	2019-10-28 10:04:05	EDIT	DELETE

Showing 1 to 10 of 17 entries

Previous 1 2 Next

Register IOT Broker



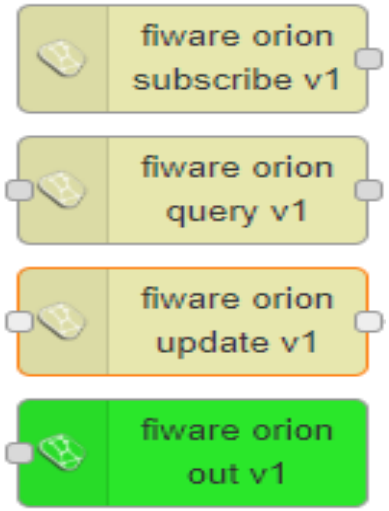
Snap4 technology is broker Agnostic

- **IOT Brokers**
  - are associated with an Organization
    - Each **Organization** has a **Knowledge Base** of reference (KB, ServiceMap)
    - Each **KB** may host multiple **Organizations** and addresses multiple **Geographic areas**
  - can be compliant with
    - **NGSI version**: V1, V2-1, V2, etc…
      - with Snap4City Security or regular NGSI FiWare
    - other protocols as well such as: MQTT, COAP, AMQP, etc.
  - can
    - expose different **authentication methods**: K1/K2, Certificate, etc.
    - be accessible from IOT Devices and IOT App **in Cloud only**
    - be accessible **from Internet to post data from outside, etc.**

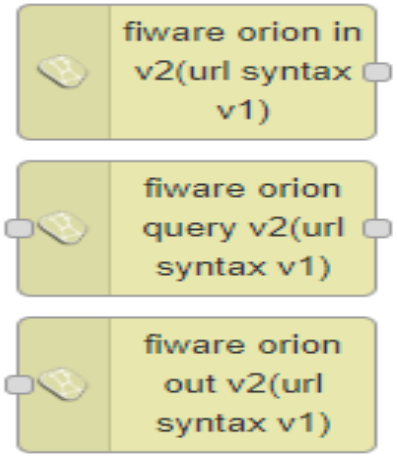
IOT Orion Broker Network: V1 and V2

- **IOT Broker can be Internal (on Snap4City Cloud)**
  - Registration of IOT Devices can be performed by the IOT Directory
  - Authentication is automatic, K1 and K2 are not needed since the security is performed via Access Token, M2M secure communication, on the basis of IOT App ownership
  - The **NIFI Cluster** automatically subscribes to all the entities on the Broker, to post data into the Data Shadow enriched with data of the KB
- **IOT Broker can be External (managed by third party)**
  - Registration of IOT Devices is managed by third parties
  - The registered IOT Devices can be collected and queried from the IOT Directory as well
  - The NIFI Cluster **may** automatically subscribes to all the entities on the Broker, to post data into the Data Shadow enriched with data of the KB
- **IOT Brokers can be networked**
  - Services, Service paths: for managing the IOT Broker network
  - Multi-tenant: more than one user/org on the same IOT Broker

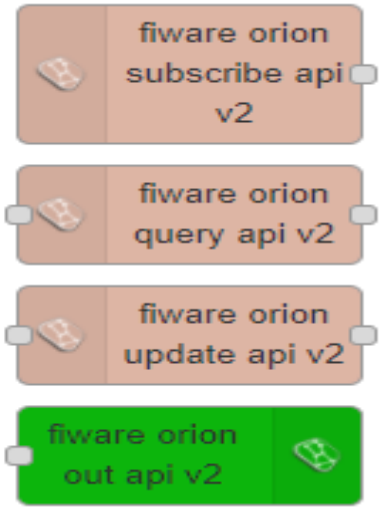
NGSI versions



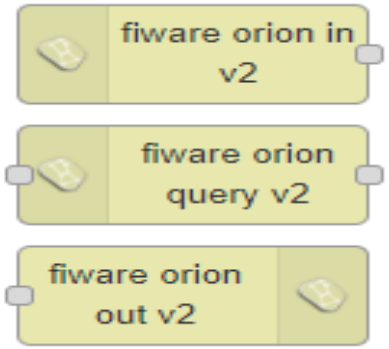
- Orion Broker of V1 with **NGSI syntax of V1** + Secure Filter of Snap4city



- Orion Broker of V2 with NGSI syntax of V1



- Orion Broker of V2 with **NGSI syntax of V2** + Secure Filter of Snap4city



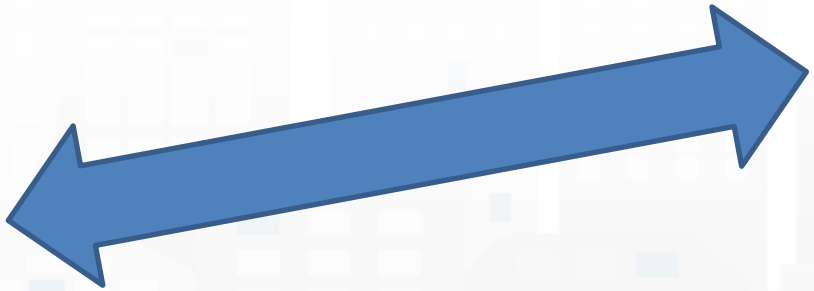
- Orion Broker of V2 with NGSI syntax of V2

IOT Devices / Brokers with Service / Service paths (SP)



IOT Orion Broker: Service and Service Paths

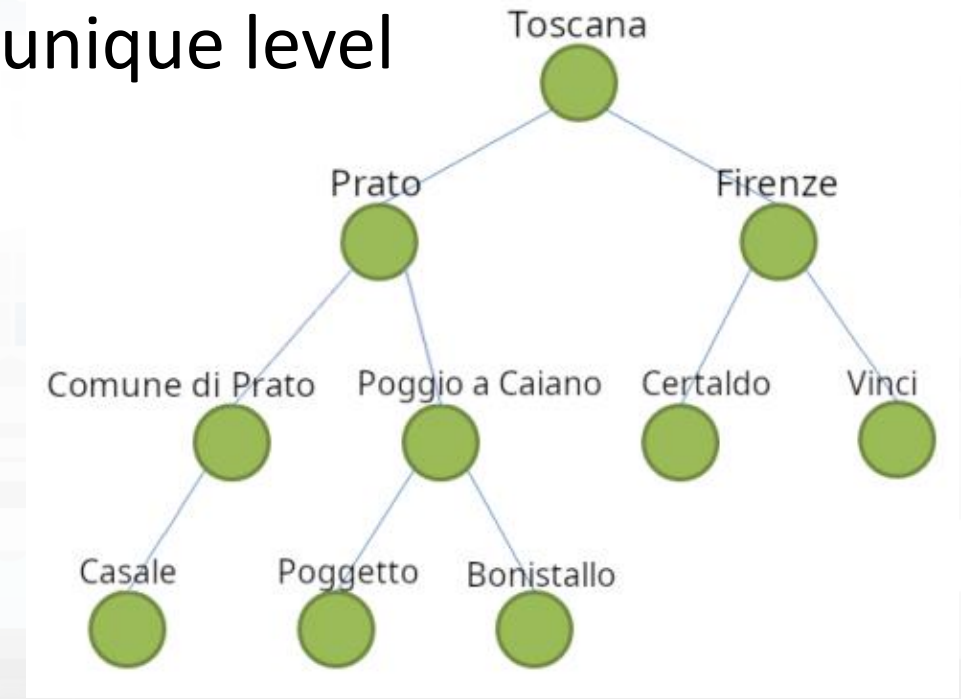
- The concept of Service Path (SP) allows to organize the IOT Devices managed by a Brokers in Directories and SubDirectories.
  - This implies that IOT Devices with identical ID may be located on different paths
  - Path may be used for logica organization:
    - **Quadrant1 of the City**
      - **Lamps: L1, L2, ……**
      - **Waste……**
    - **Quadrant2 of the city**
      - **Lamps: L1, L2, ……**
      - **Waste……**



- Lamps
  - Street 1. d1., d2..
  - street 2, …
  - street 3, ..
- Waste
  - street 1: d1, d2, …
  - street 2
  - street 3

The Paths

- Have to start with /, only absolute
- Max 10 levels
- Max 50 chars for each level
- A Device can belong only to a unique level




```
POST orioncb:1026/v2/entities
Content-Type: application/json
Fiware-ServicePath: /toscana/prato/comune_di_prato/casale
```

```
{
    "id": "Sensor1",
    "type": "MicroController",
    "temperature": {
        "value": 23.4,
        "type": "Float"
    },
    "pressure": {
        "value": 980,
        "type": "Integer"
    }
}
```

IOT Devices / Brokers with Multi Tenant (MT)



IOT Orion Broker: MultiTenant

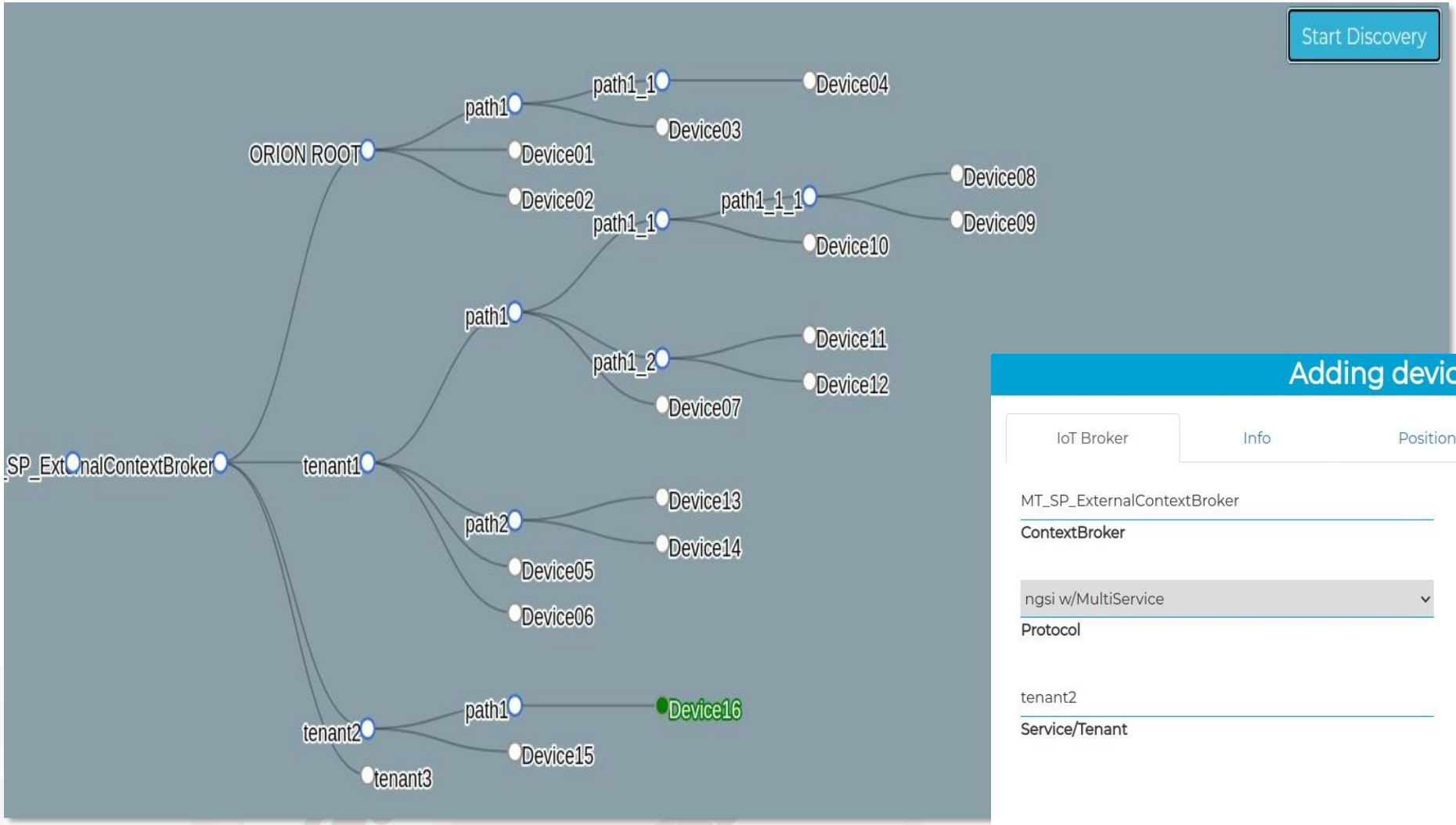
- The usage of **Service** can be used as **Tenant** of the same Broker
- The IOT Orion Broker and NGSI protocol do not allow to brose the whole set of Services, ServicePaths.
  - So that the single user may know its own Service which can be regarded as its own Tenant

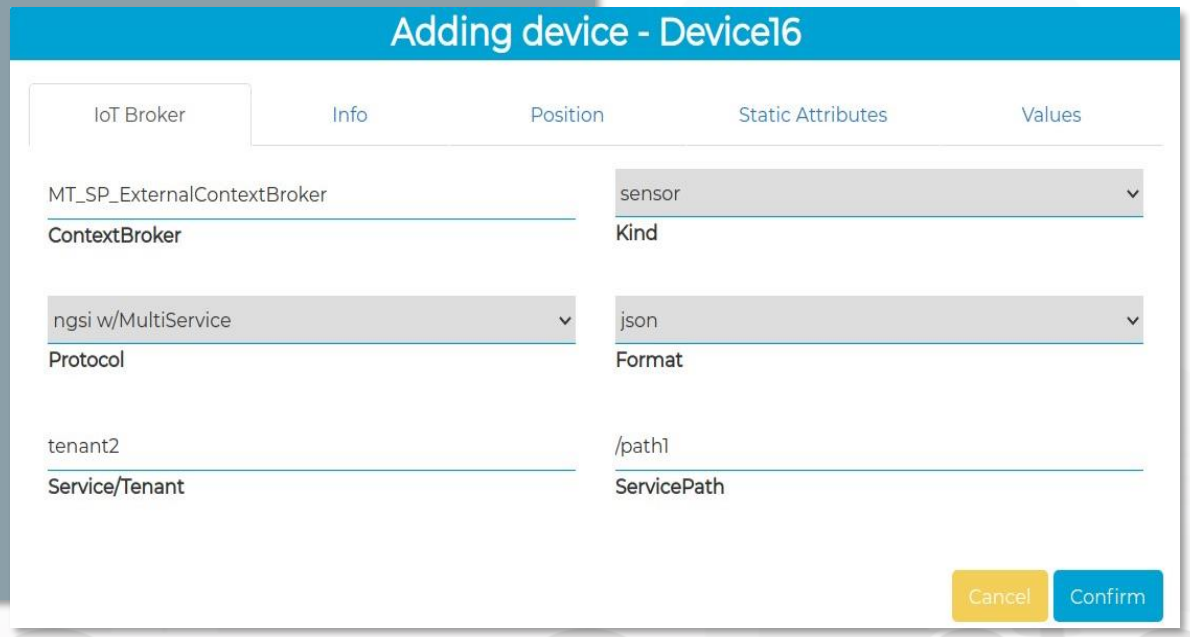
```
Fiware-Service: tenant2

Fiware-ServicePath: /path1
```

Discovery of IOT Devices on External Brokers with SP/MT



Device Discovery MT/SP on External Brokers





IOT Device vs Time Series



What About IOT Devices, time series



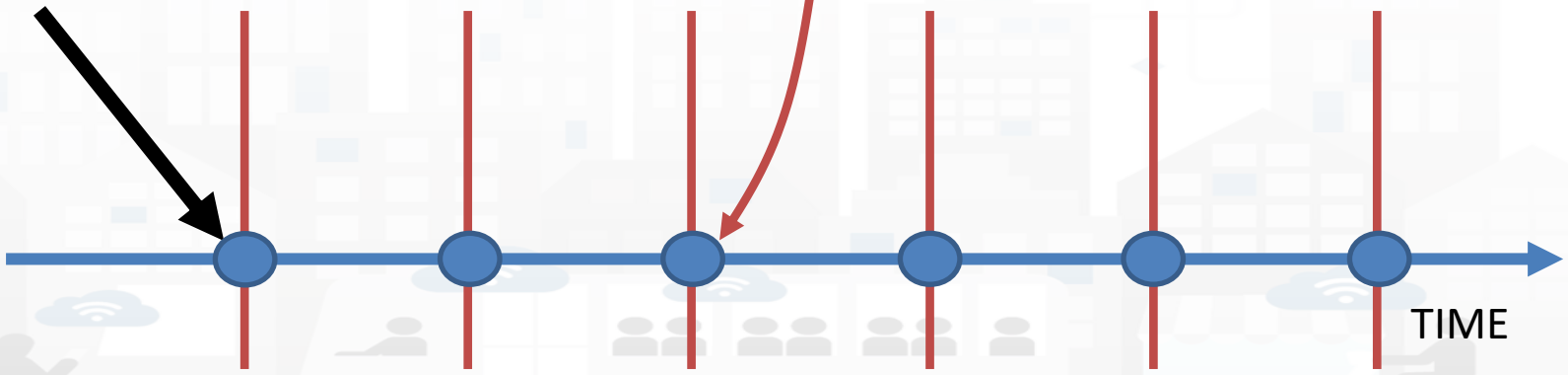
IOT Device

Sends a message

Message (
timestamp: «02-04-2020 at 10:30»,
Temperature: 29.34,
Humidity: 35
)

dateObserved	Temp	Humidity
02-04-2020 10:30	34.5	23
02-04-2020 10:40	36.5	24
02-04-2020 10:50	36.0	22.5

- A set of data coming from an IOT Device with multiple sensor become a time series of values for devices.
- For example taking a new measure every 10 minutes (Red Lines)
- Each new measure in Snap4City is conventionally time located in «dateObserved»

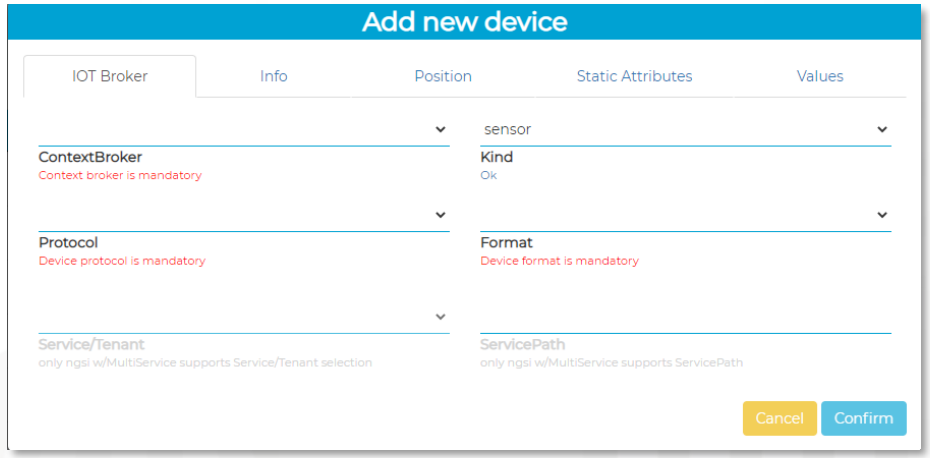


IOT Device Model

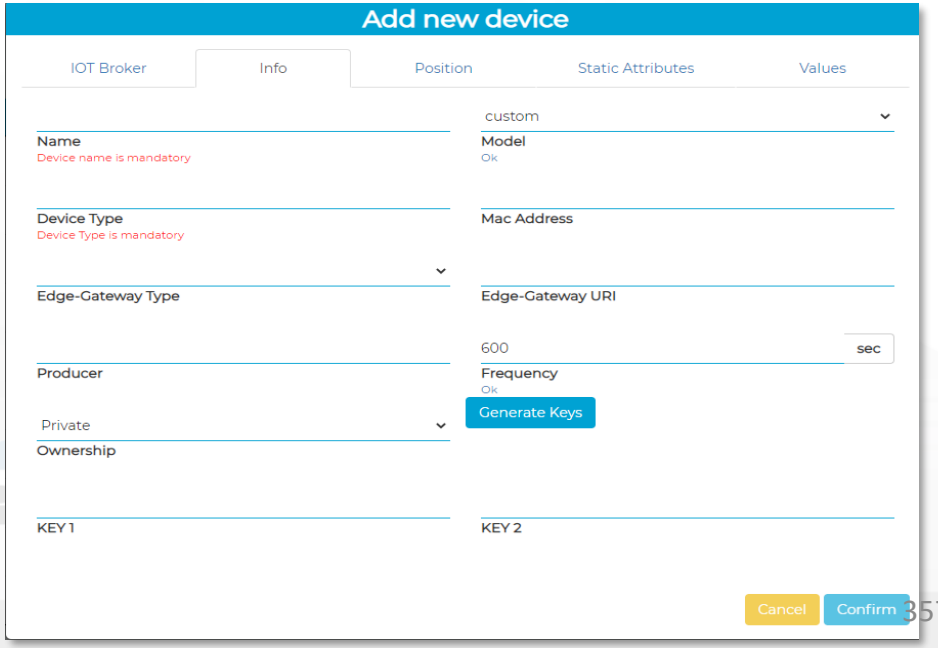


IOT Device Data Model (1)

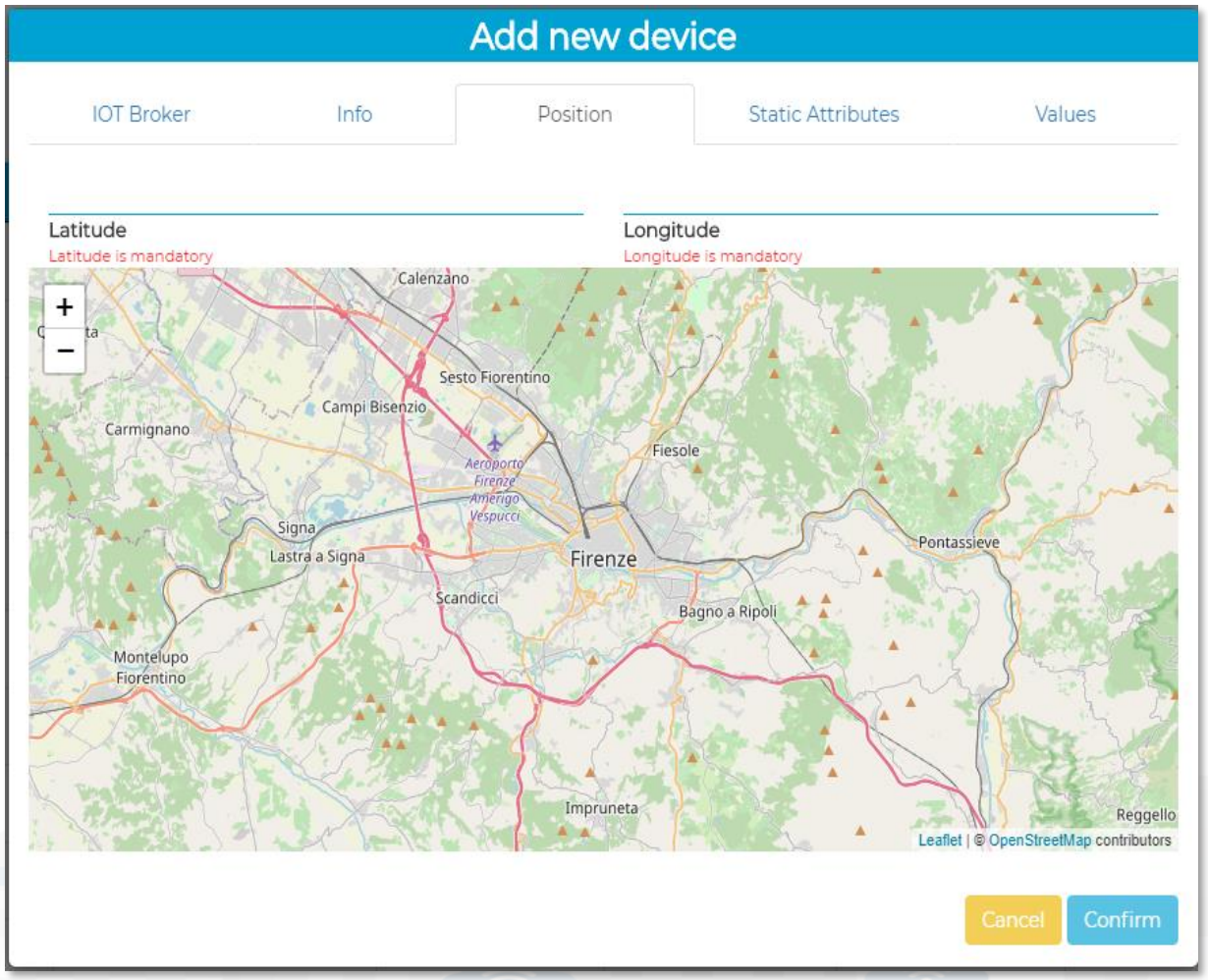
- IOT Broker
  - Name of the Brokers: among those registered
  - Protocol: NGSI, AMQP, MQTT, etc..
  - Format: CSV, JSON, XML.
  - Service/Tenant:……
  - ServicePath:……

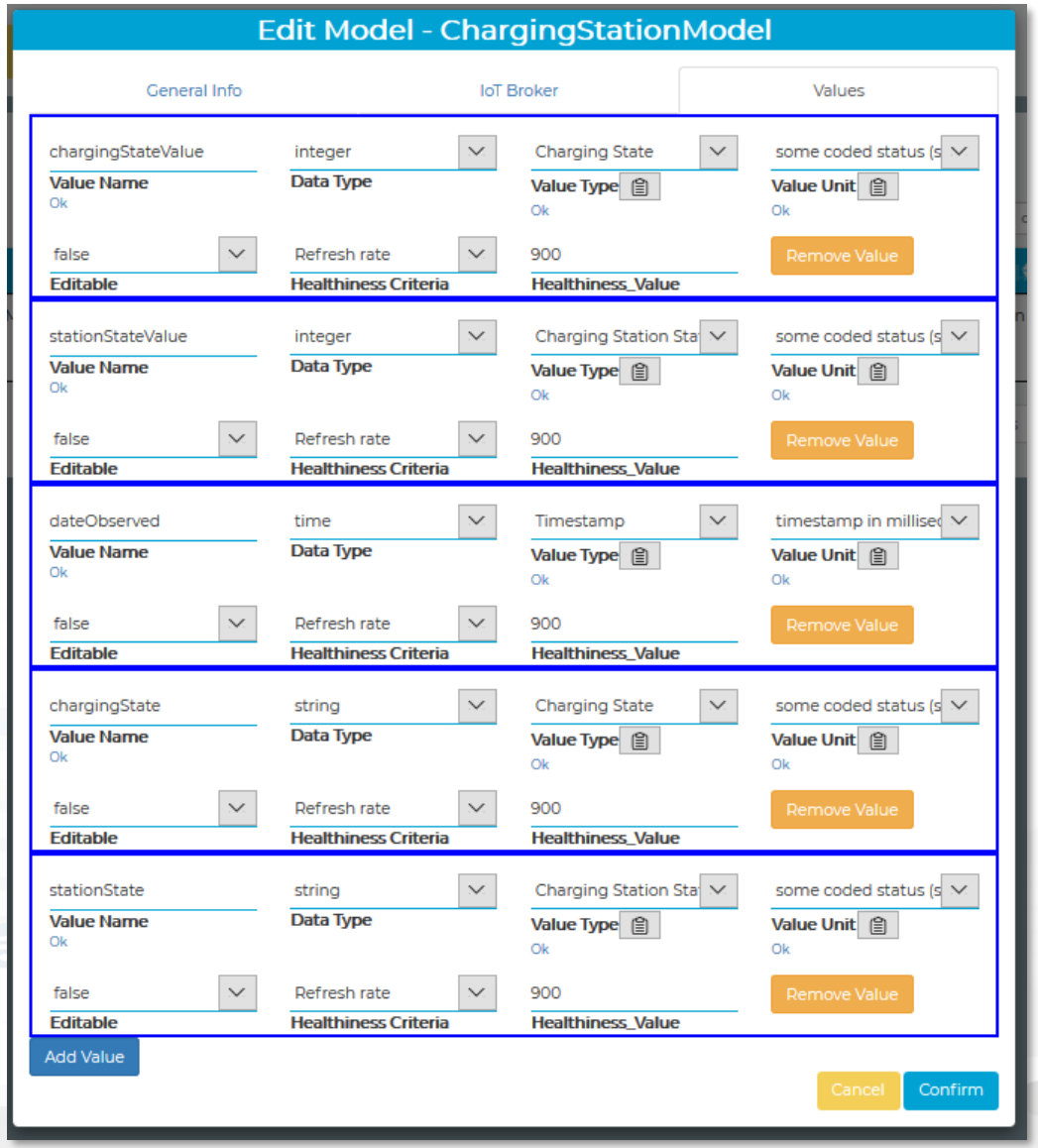


- Info
  - Name (Identifier)
  - Model: Custom or Model ID
  - DeviceType: ..a string..
  - MAC address: …optional…
  - Edge-GW: Raspberry, Android, …
  - Edge-GW: URI
  - Producer
  - Owner
  - Freq: ….. Sec
  - Keys: K1, K2



IOT Device Data Model (2)





IOT Device Data Model (3): Attributes

Where	IOT Device Model	IOT Device	A Temporal Instance
IOT Broker	**Broker: OrionUNIFI**		
IOT Broker	**Protocol: NGSI**		
Info	**ID: string**	ID: "park45"	**park45**
Position	**GPS: lat, long**	GSP Position: 43.12, 11.34	GSP Position: 44.12, 11.12
Static attribute	**Description: string**	Description: "parking massaia"	
Static attribute	**Location: string**	Location: "Via Massaia"	
Static attribute	**Civic Number: string**	Civic Number: 3	
Static attribute	**MaxCapacity: number, cars**	MaxCapacity: 456	
Values	**dateObserved: Timestamp**		23-12-2019T20:13:12…
Values	**FreeSlots: Integer, #**		345
Values	**Humidity: float, %**		25,5
Values	**Temperature: float, celsius**		34.4

Model meaning

- **ID:** is the unique identifier for reconnecting Temporal Instances with registered IOT Devices
- **Static Attributes:**
  - Are typically associated with instances of the IOT Device.
    E.g.:, You have a set of parking areas, each of them is located in a specific street, and has its one name, etc.
  - Different kinds of attributes can be set for each SubNature. Their definition has to be prepared into the Knowledge Base ☺ for automated indexing.
- **Values**: they are time varying variables (temporal values/instances)
  - They change over time, the time stamp of the time serie is conventionally «dateObserved» in Snap4City
  - In new *SensorMobile* HLT, also GPS can be changing over time as in the MyKPI

Using the IOT Device Model notes!!!

- Once performed the IOT Device Model, a number of IOT Devices can be produce **using the model as a Template**



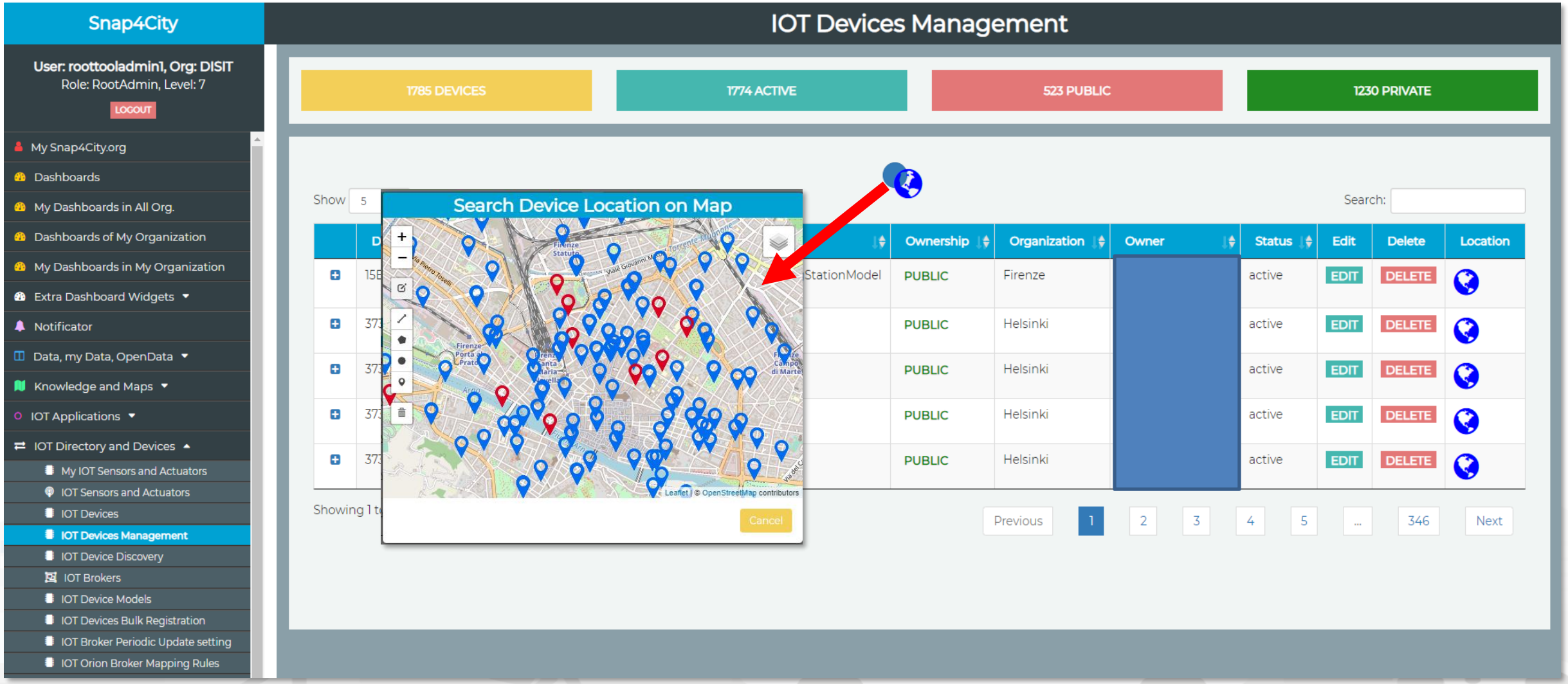
  - **NOTE:** the produced IOT Devices are not going to change if the IOT Device Model is modified.
  - *Your biscuit is not changing if the template is modified after the printout*

IOT Device Management

(only for: RootAdmin and ToolAdmin)



IOT Device Management



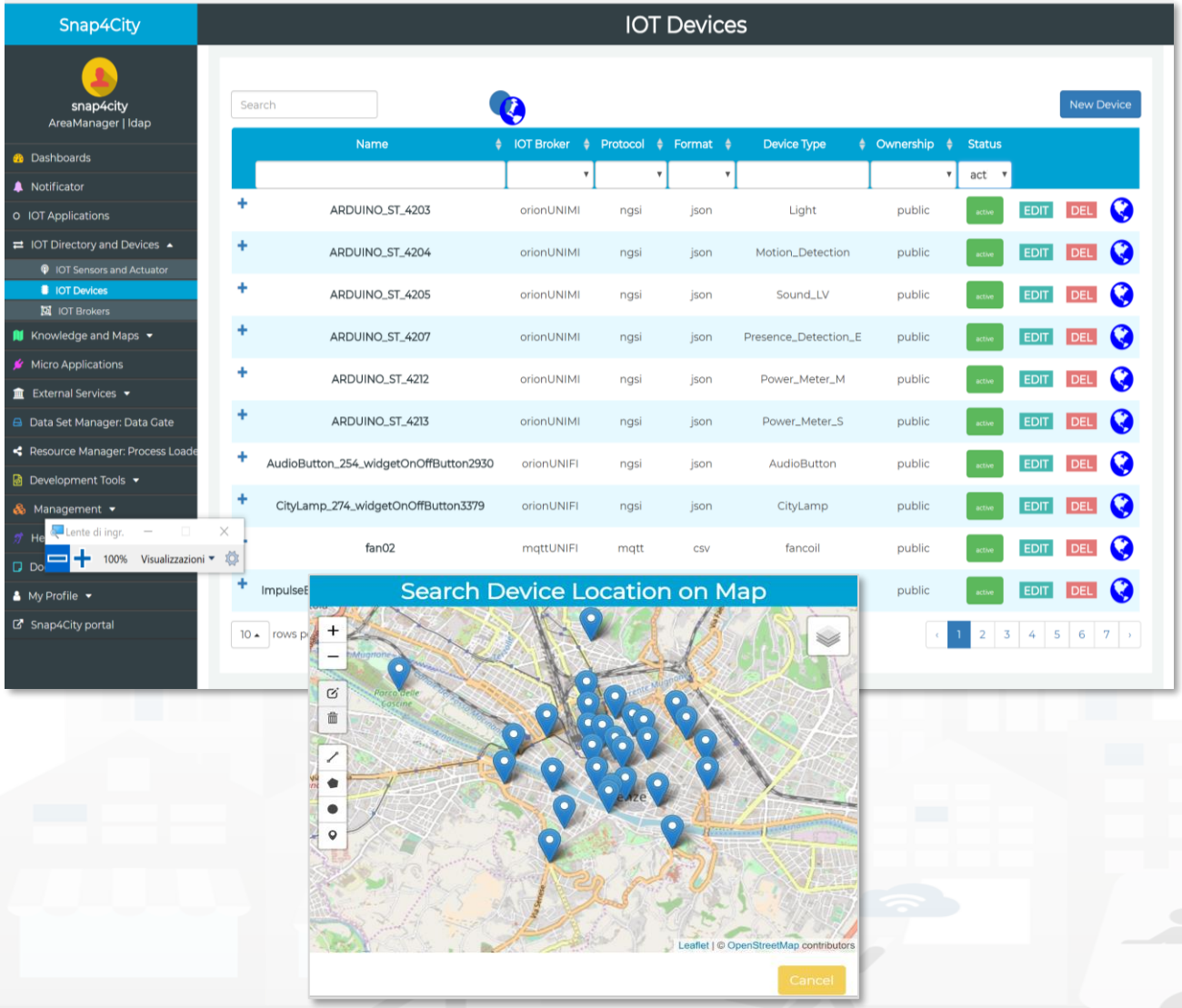
IOT Device Registration

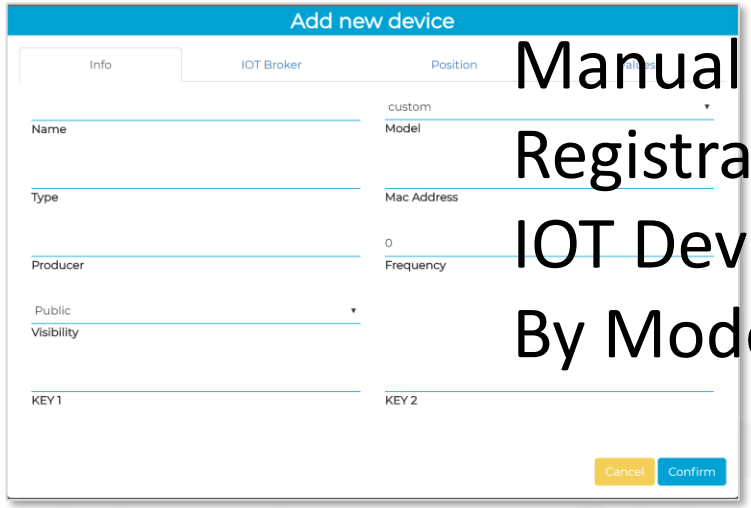


Activities for IOT Device Registration

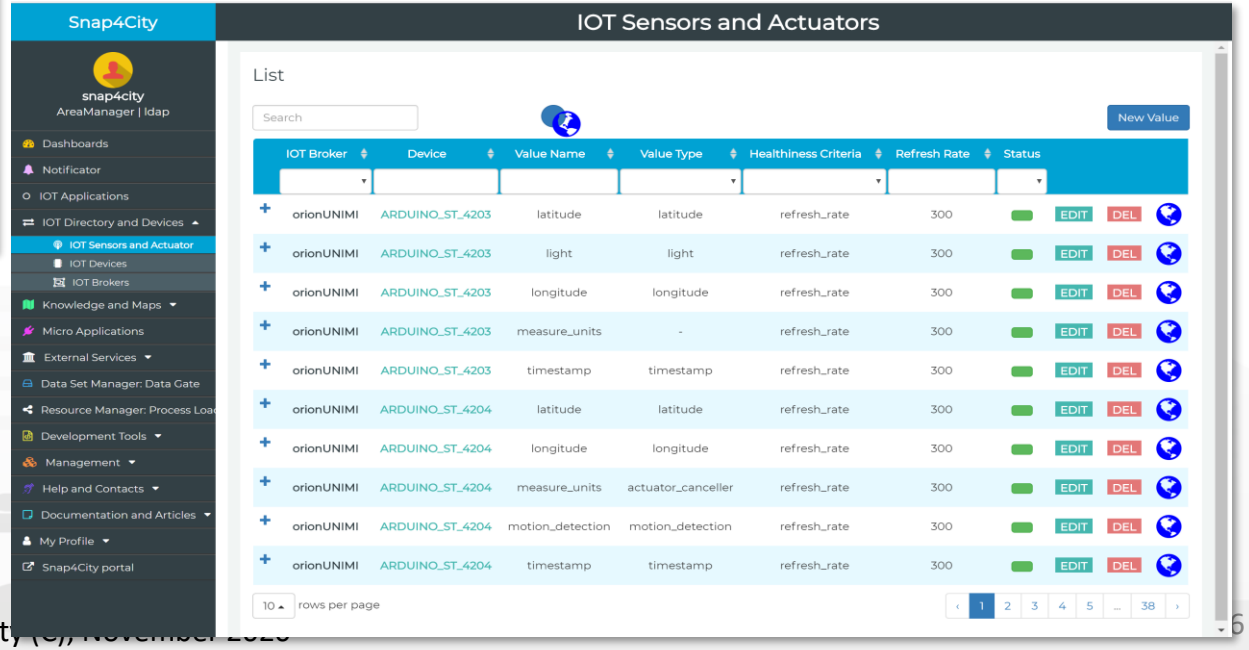
- **Registration of**
  - an IOT Device Manually from Zero or from an IOT Device Model
  - a set of IOT Devices with the same IOT Device Model from IOT App
  - a set of IOT Devices in BULK loading a CSV (with or without a reference IOT Model)
- The IOT Device registration implies the **automated production of the Digital Twin Device into the Knowledge Base**, which implies:
  - Activation of the Storage “DataShadow” for historical data access
  - Activation of all the relationships
  - Activation of Discovery mechanisms via IOT Directory, KB
  - Activation of Dashboard Wizard (after a few minutes), and Data Inspector
  - Etc.

IOT Directory, IOT Device Management

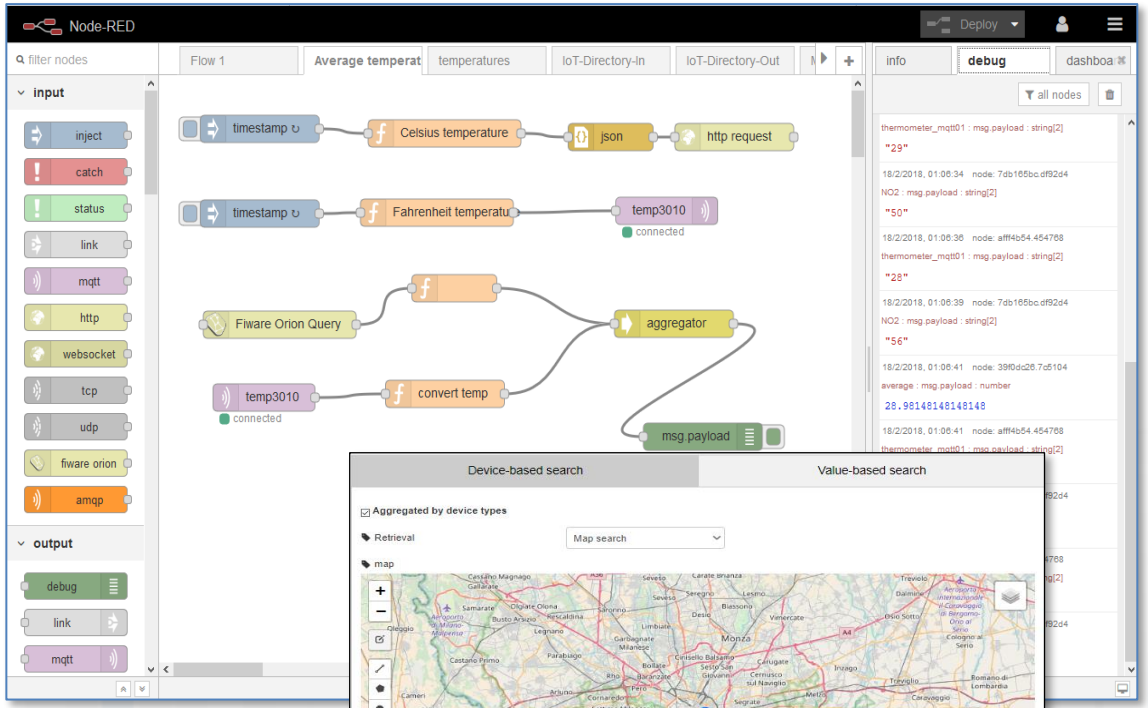


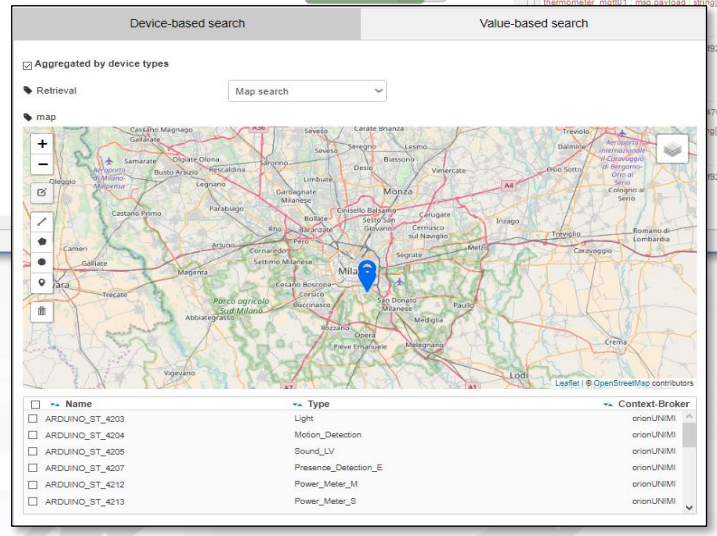


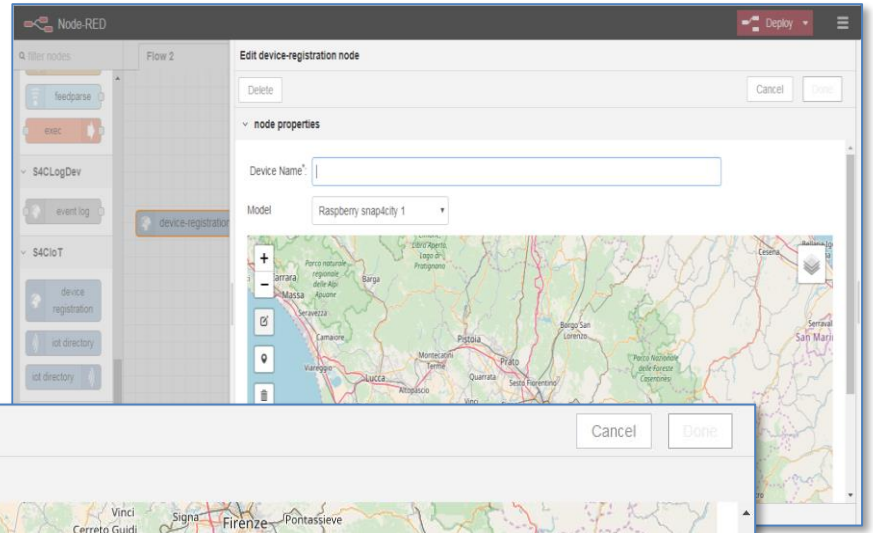
Manual Registration of IOT Device or By Model

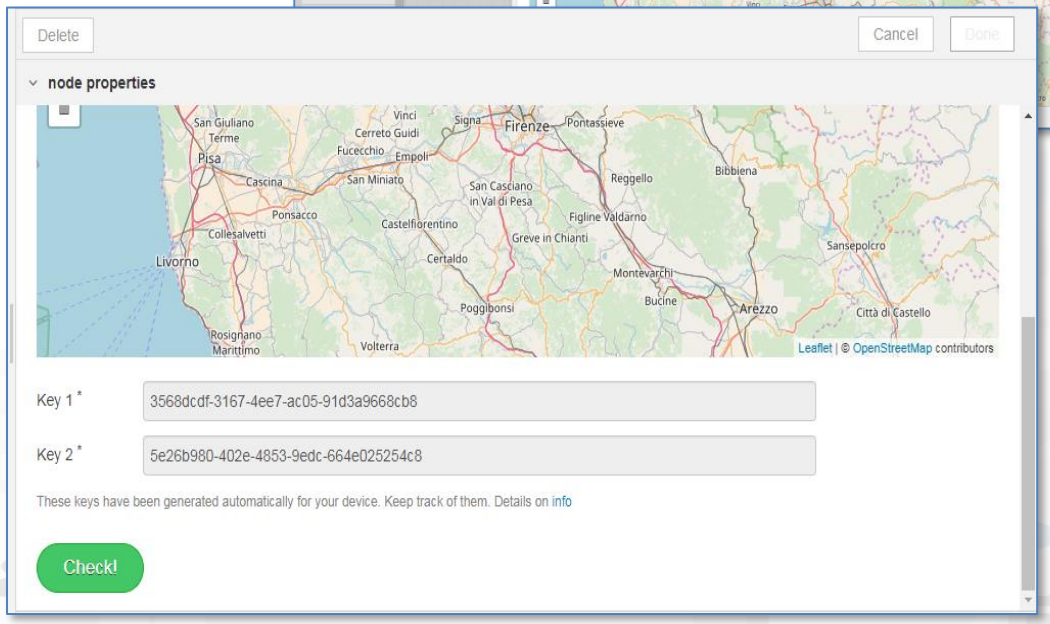


IOT Discovery on IOT Application Development







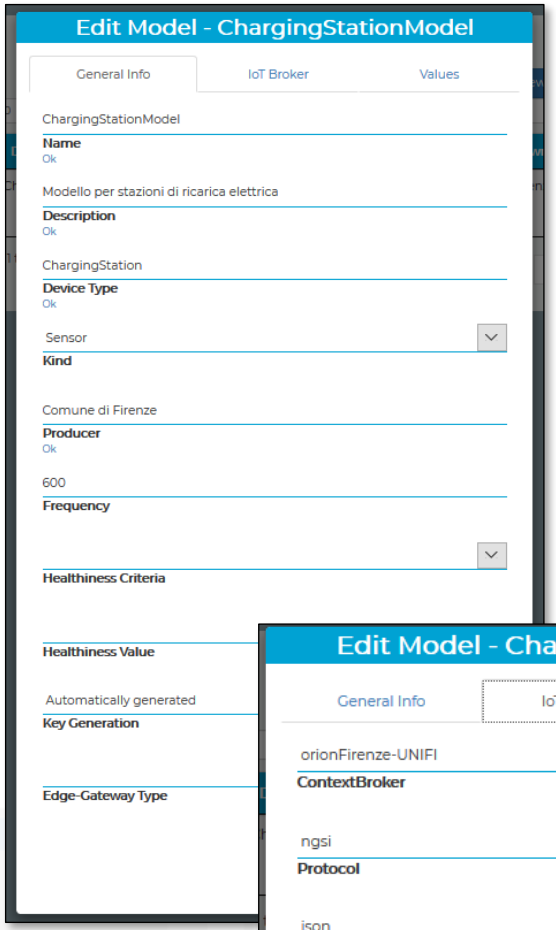


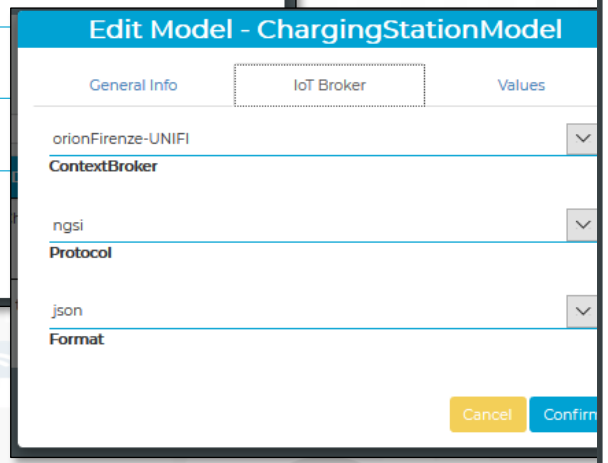
IOT Device Registration via IOT Device Model



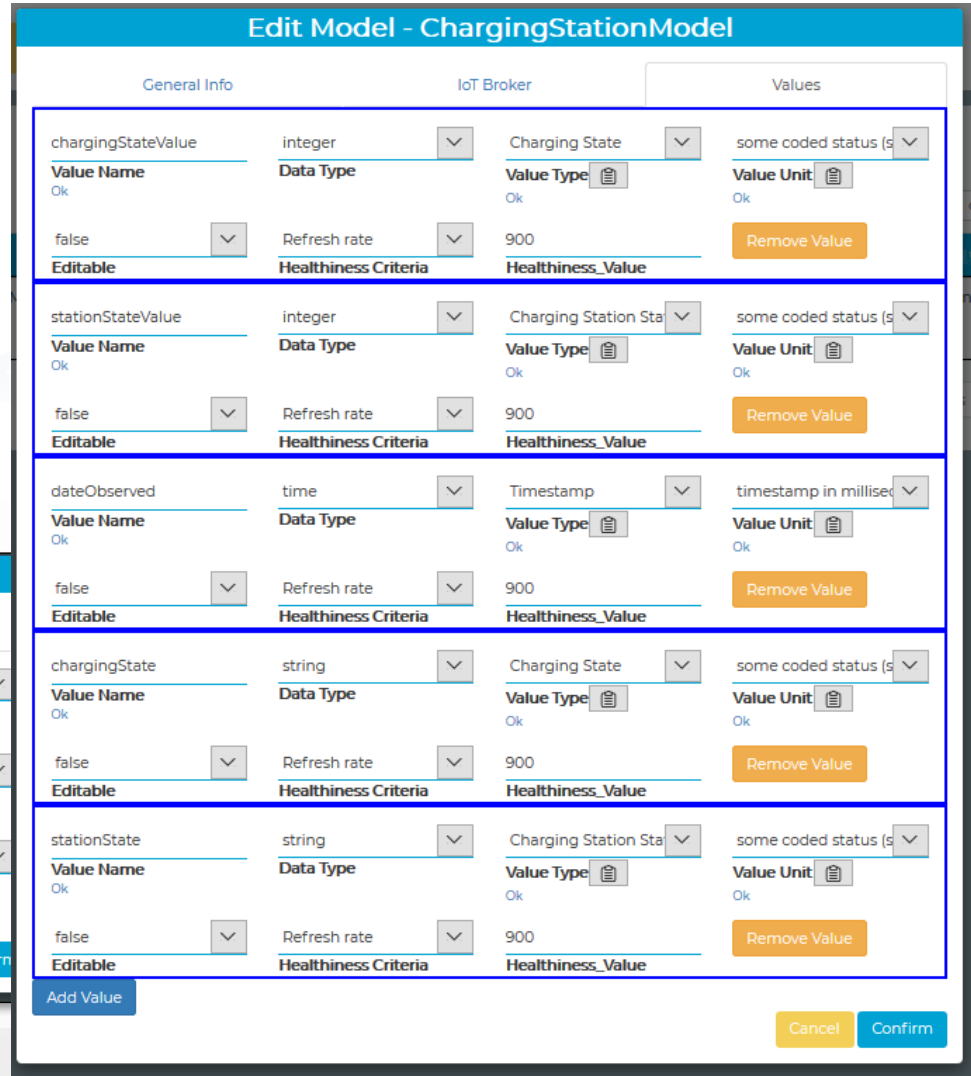
Many IoT Devices? IOT Device Model!!!

- **Prerequirements**: only for AreaManager users
- If you have a set of sensors with the same features,
  - you can create a model and then a set of instances (IoT Devices) in compliance with the model (not time consuming and avoiding errors)
- IoT Directory and Devices > IoT Device Models > ‘New Model’ button

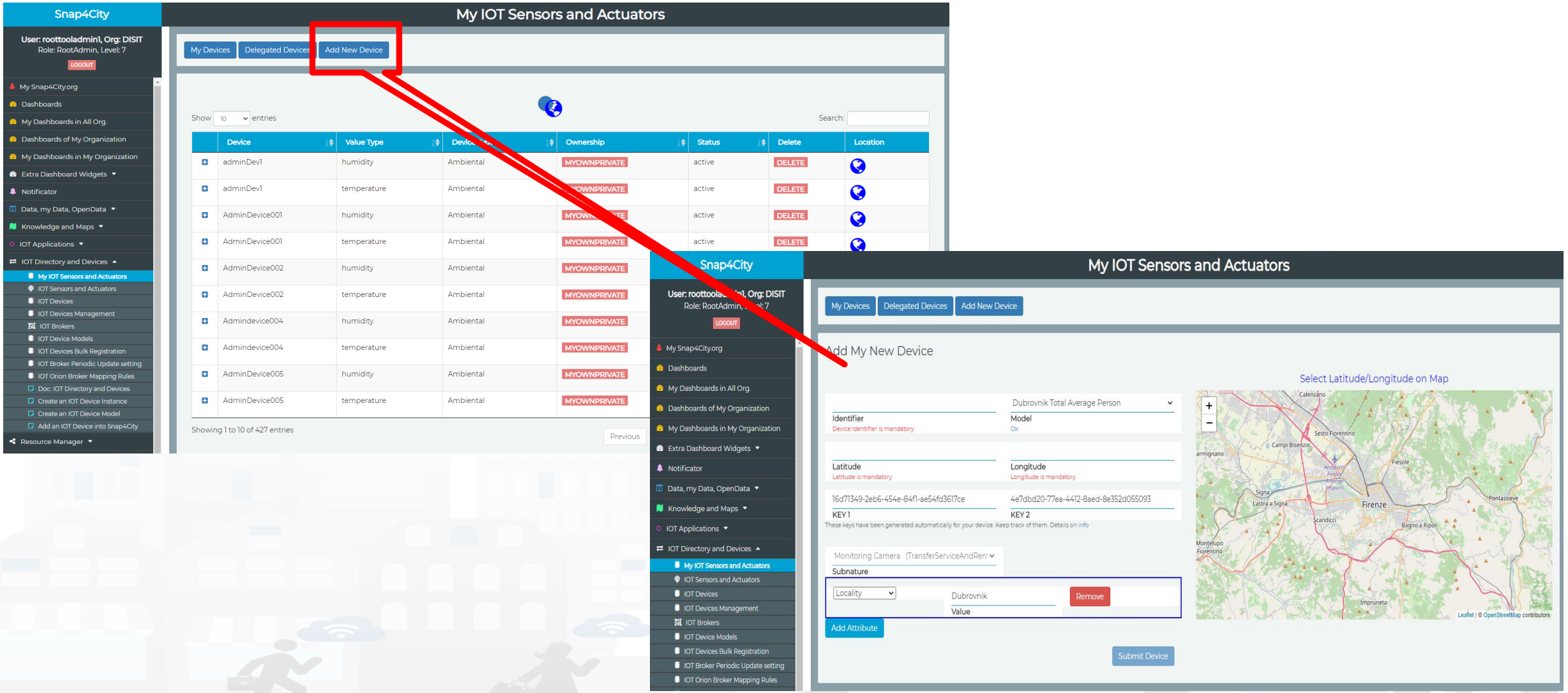




Example: ChargingStationModel



Add IOT/IOE Devices, exploiting an IOT Device Model



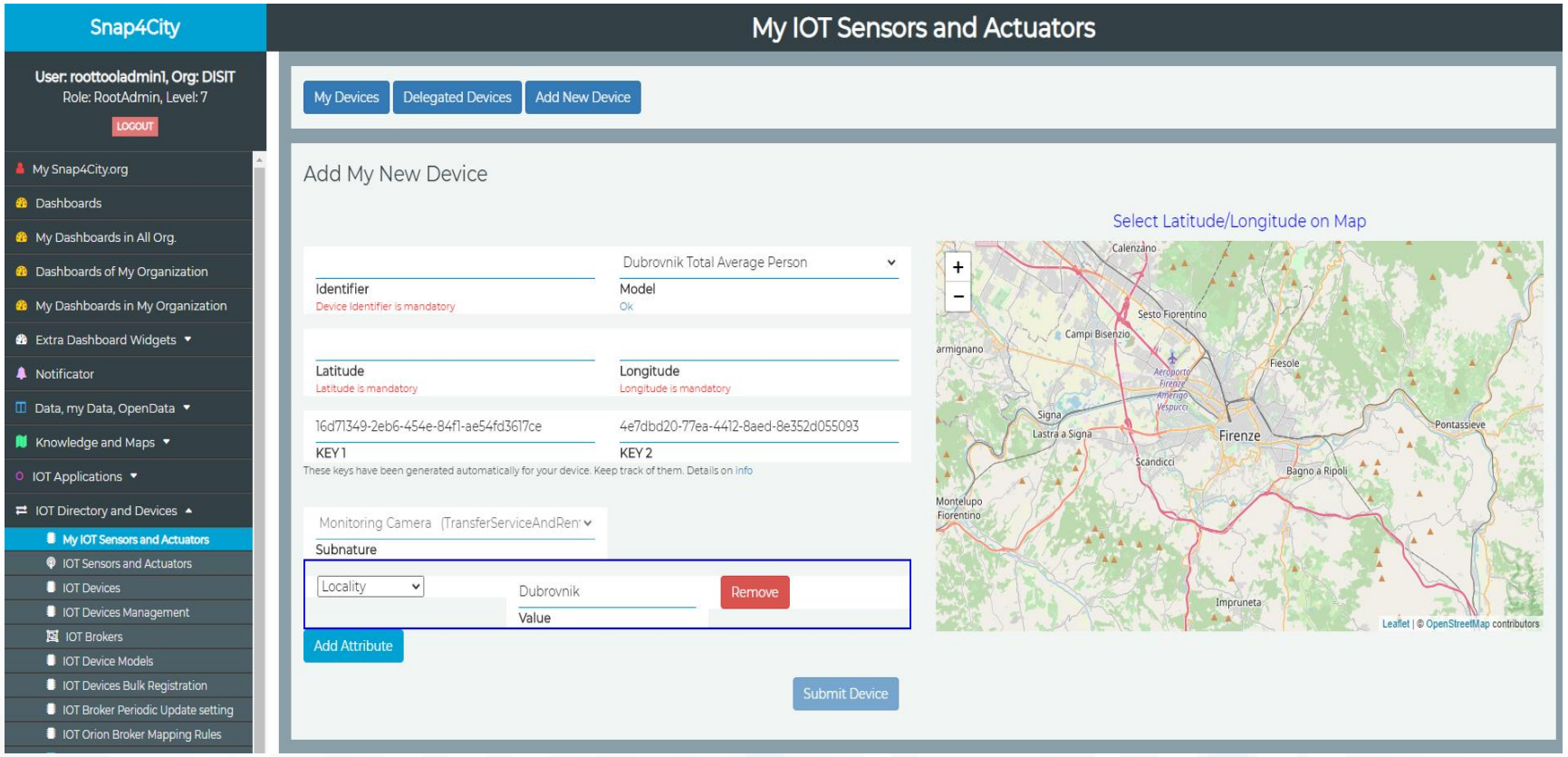
Add IOT/IOE Devices, exploiting an IOT Device Model

Just Buy an IOT Device and register: SigFOX, MQTT, NGSI (FiWare), …

- Attach them by
  - Models
- A range of protocols, formats, approaches

Create your own devices:

- Arduino,
- Raspberry,
- Android,
- LoraWAN + Arduino,
- etc.

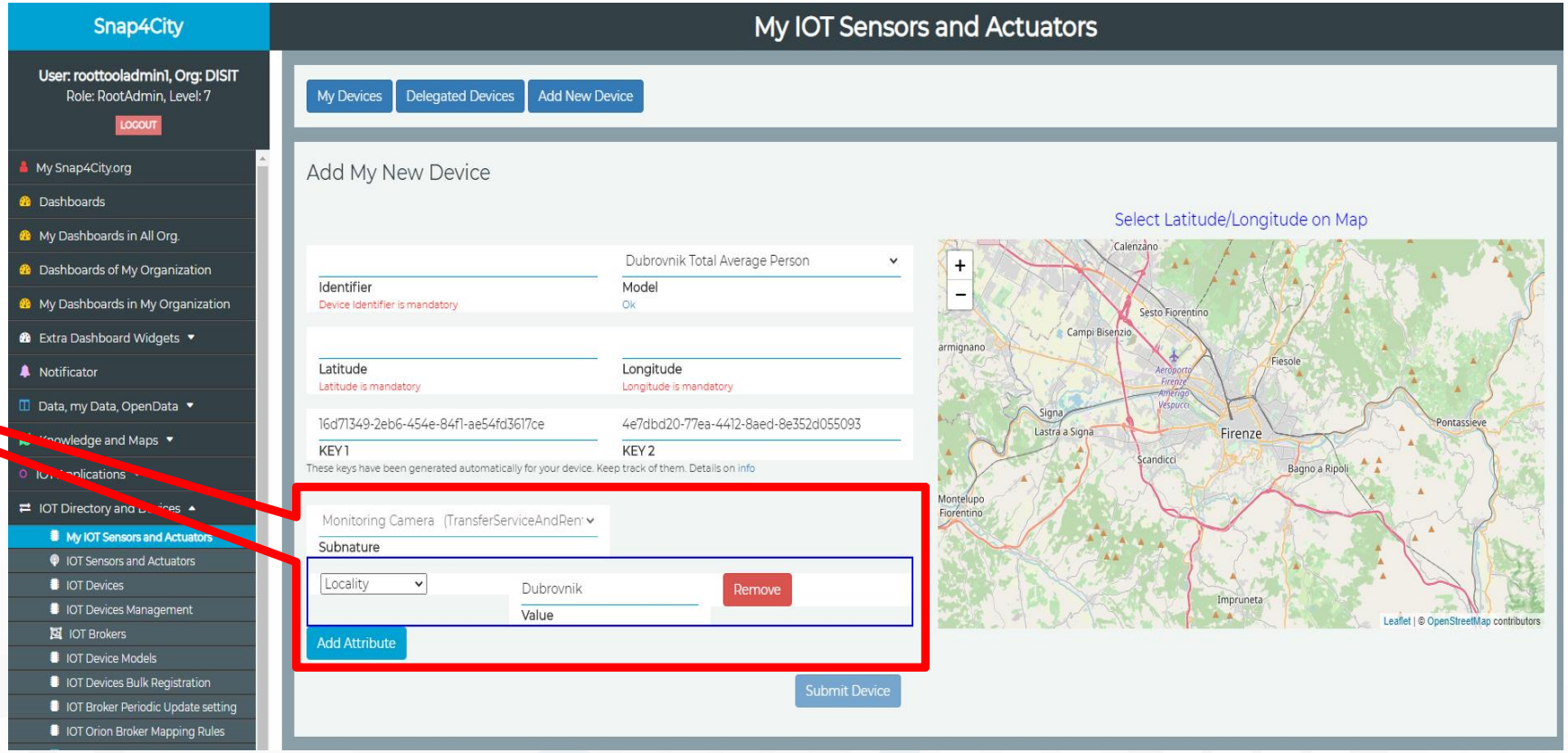


Secure Communication: HTTPS, TLS (K1, K2), Certificates

Add IOT/IOE Devices, exploiting an IOT Device Model

Addition of Static Attributes of the IOT Device

Only if you enabled from model



IOT Device Registration via IOT Device Model in Bulk



From CSV → register IoT Devices in BULK

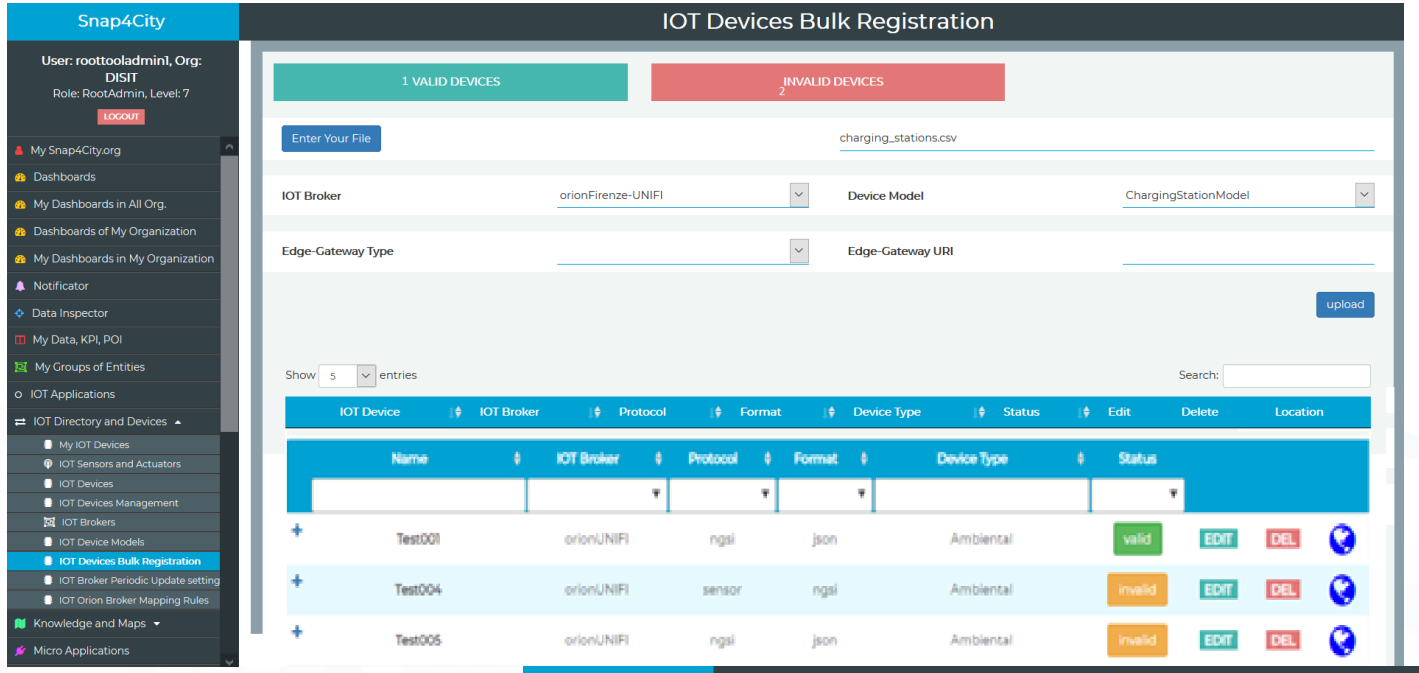
- Create a CSV from the CSV Model provided
- The columns must respect the CSV Model (every field present in the Model)
- One row of the CSV is one new IOT Device
- You have to create to create two keys (called k1, k2) that are necessary to read and write access to the device. They must be different each other.
- Each group of devices, that has the same IoT Model (data set), could/should have the same K1, K2. In this way, it is easier to read or write all the IOT Devices of the same set at the same time.
- These keys are in the UUID v4 format and can be generated online on this website: https://www.uuidgenerator.net

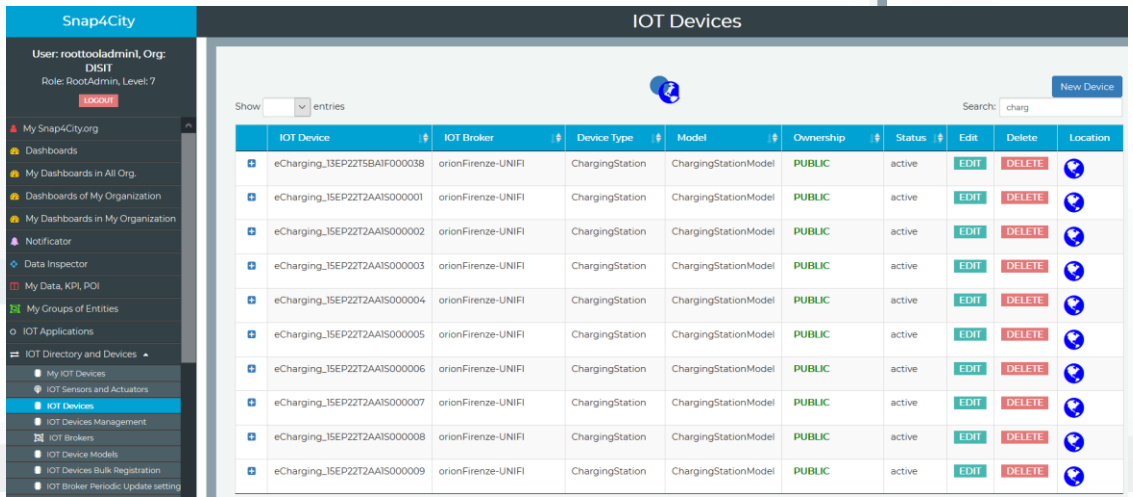
Available example: https://www.snap4city.org/592

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	name	devicetype	mac	frequency	kind	protocol	format	producer	lat	long	valuename	data_type	value_type	editable	value_unit	health_criteria	healthiness_value	K1	k2
2	eCharging_16ZP22T2AA1S000003	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.222	1.125.338	chargingState	string	charging_state	false	-	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38
3	eCharging_16ZP22T2AA1S000003	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.222	1.125.338	chargingStateValue	integer	charging_state	false	#	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38
4	eCharging_16ZP22T2AA1S000003	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.222	1.125.338	stationState	string	charging_station_state	false	-	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38
5	eCharging_16ZP22T2AA1S000003	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.222	1.125.338	stationStateValue	integer	charging_station_state	false	#	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38
6	eCharging_16ZP22T2AA1S000003	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.222	1.125.338	dateObserved	time	timestamp	false	s	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38
7	eCharging_15EP22T2AA1S000051	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.483	1.125.993	chargingState	string	charging_state	false	-	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38
8	eCharging_15EP22T2AA1S000051	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.483	1.125.993	chargingStateValue	integer	charging_state	false	#	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38
9	eCharging_15EP22T2AA1S000051	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.483	1.125.993	stationState	string	charging_station_state	false	-	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38
10	eCharging_15EP22T2AA1S000051	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.483	1.125.993	stationStateValue	integer	charging_station_state	false	#	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38
11	eCharging_15EP22T2AA1S000051	ChargingStations	3D:F2:C9:A6:B3:3F	600	sensor	ngsi	json	Comune di Firenze	4.377.483	1.125.993	dateObserved	time	timestamp	false	s	refresh_rate	900	c6f03a41-880e-46f0-879a-993501ca6b50	80466a36-2b5a-4ac9-a7c3-db12c480da38

Register IoT Devices in BULK

- IoT Directory and Devices > IoT Devices Bulk Registration
- Select: Model, Broker
- Upload the CSV file
- Wait
- Verify the presence of your Devices in:
  - IoT Directory and Devices > IoT Devices
- https://www.snap4city.org/289



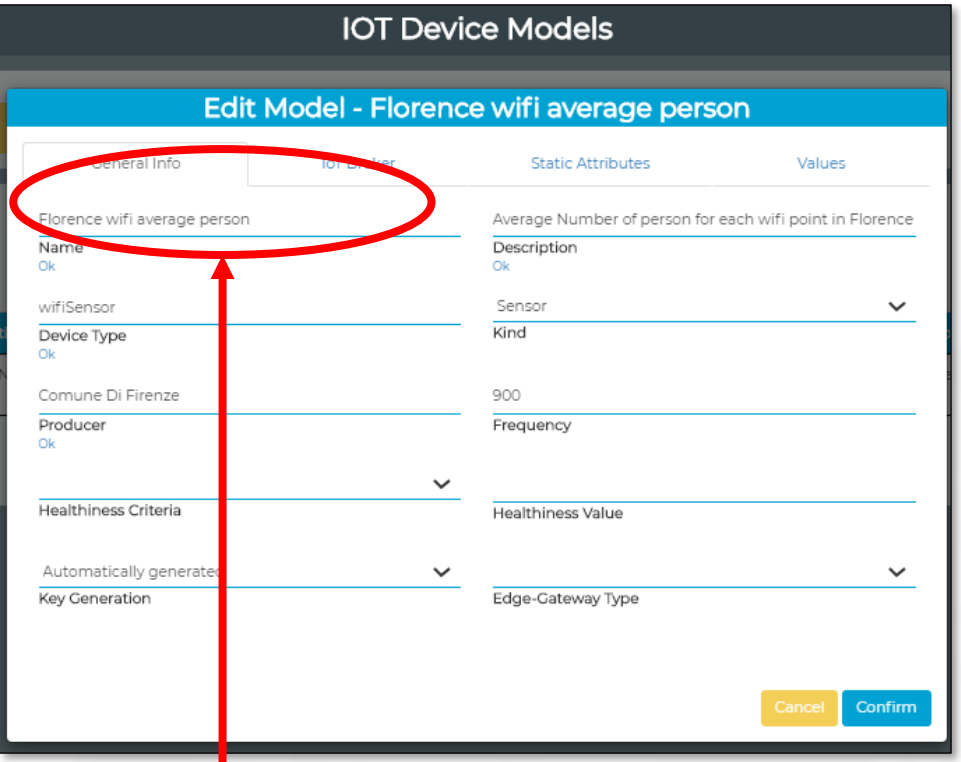


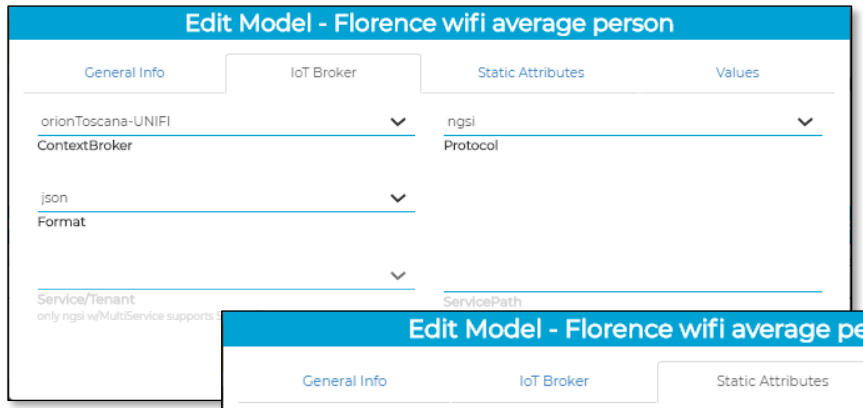
IOT Device Registration from IOT App (automation)

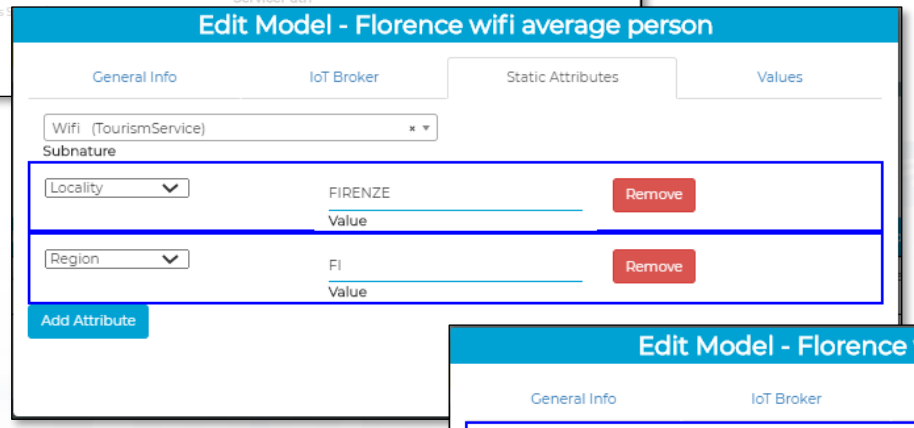


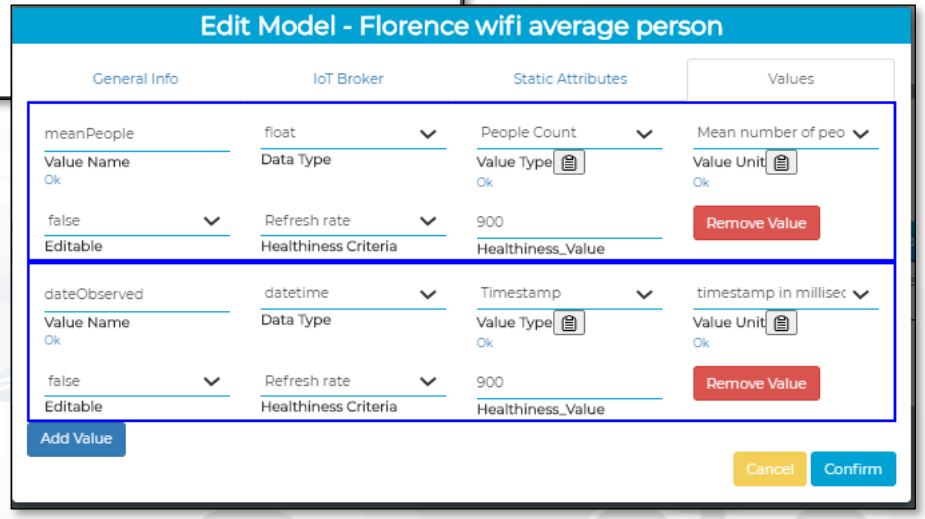
Dataset Creation

1) Model creation



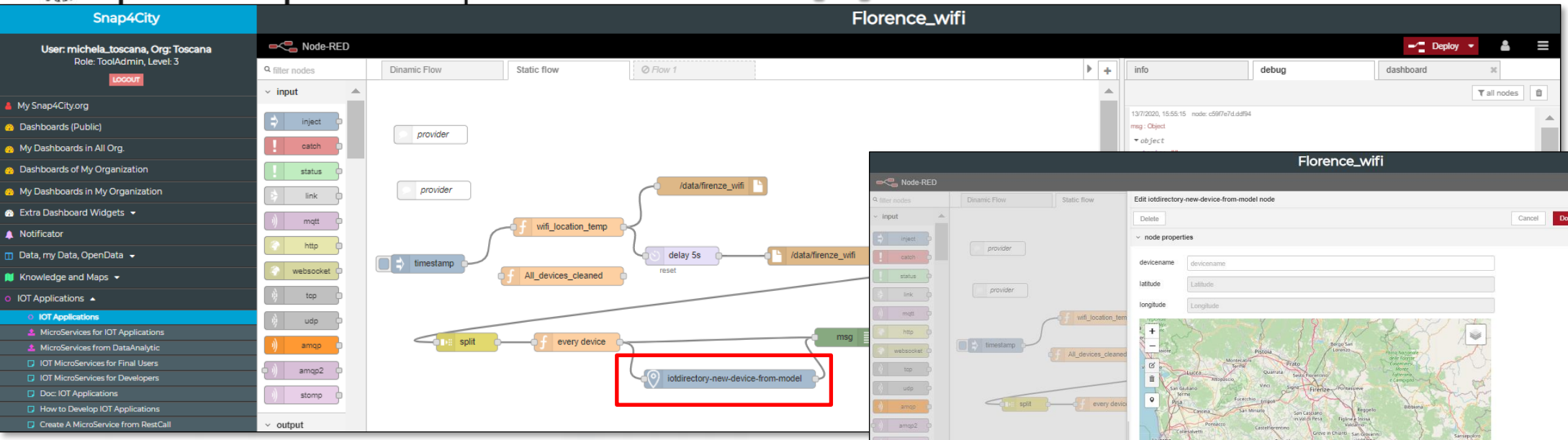






Model name: Florence wifi average person

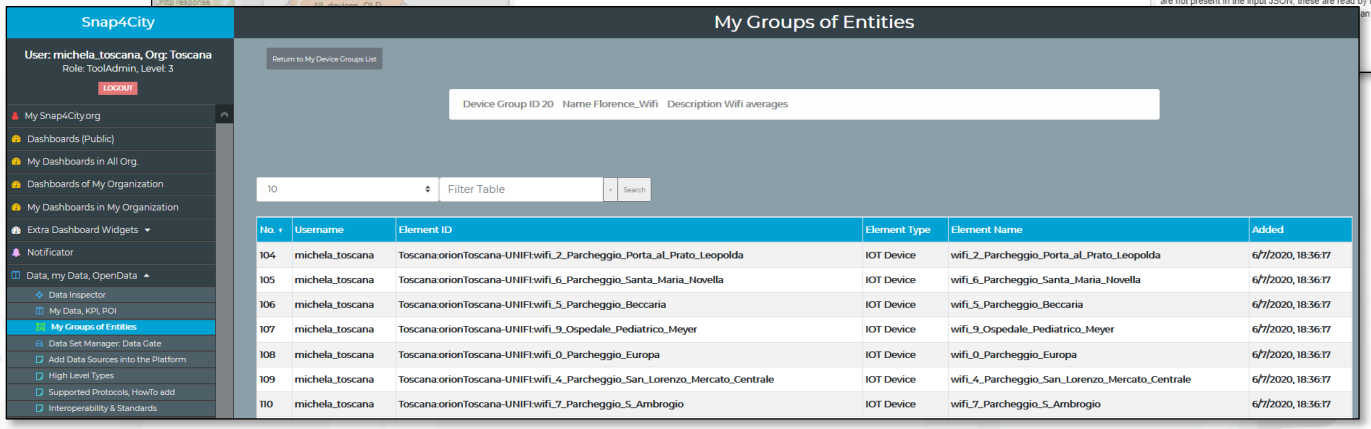
IoTApp static flow



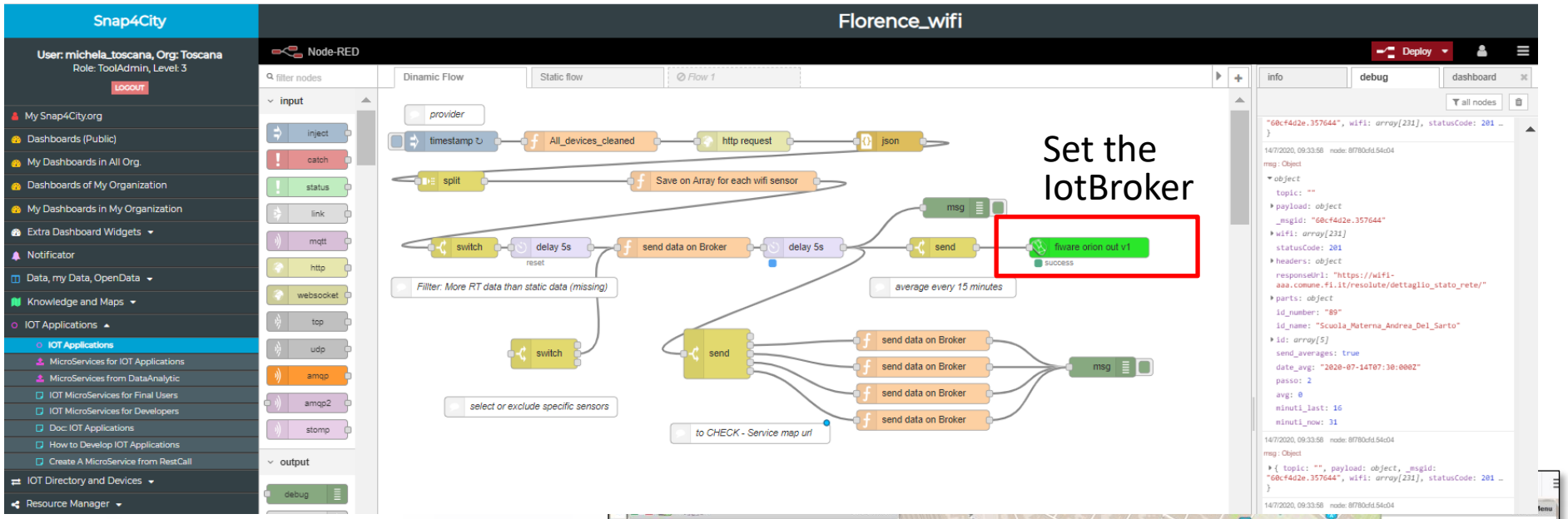
2) IoT Devices Creation

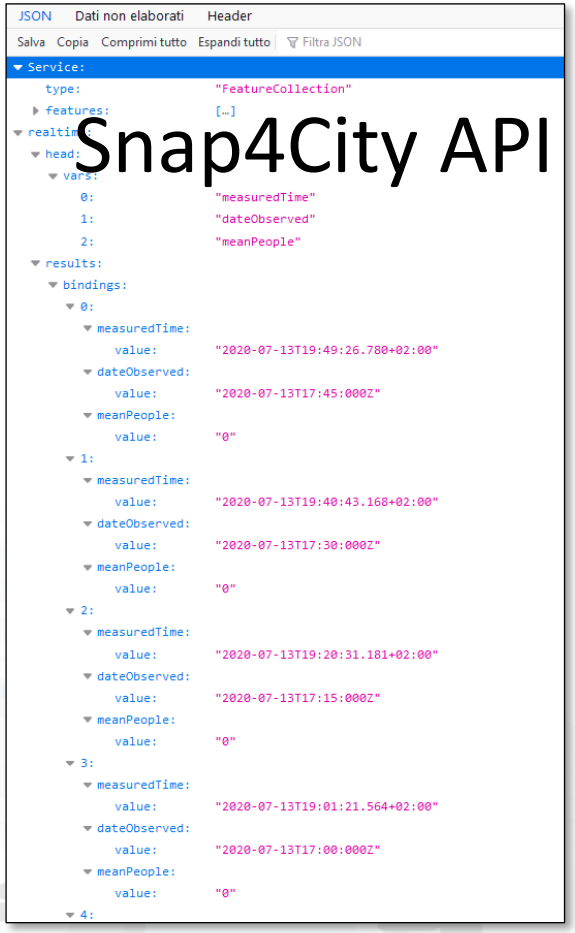
BLOCK: 'IoTDirectory-new-device-from-model'
Model name: Florence wifi average person

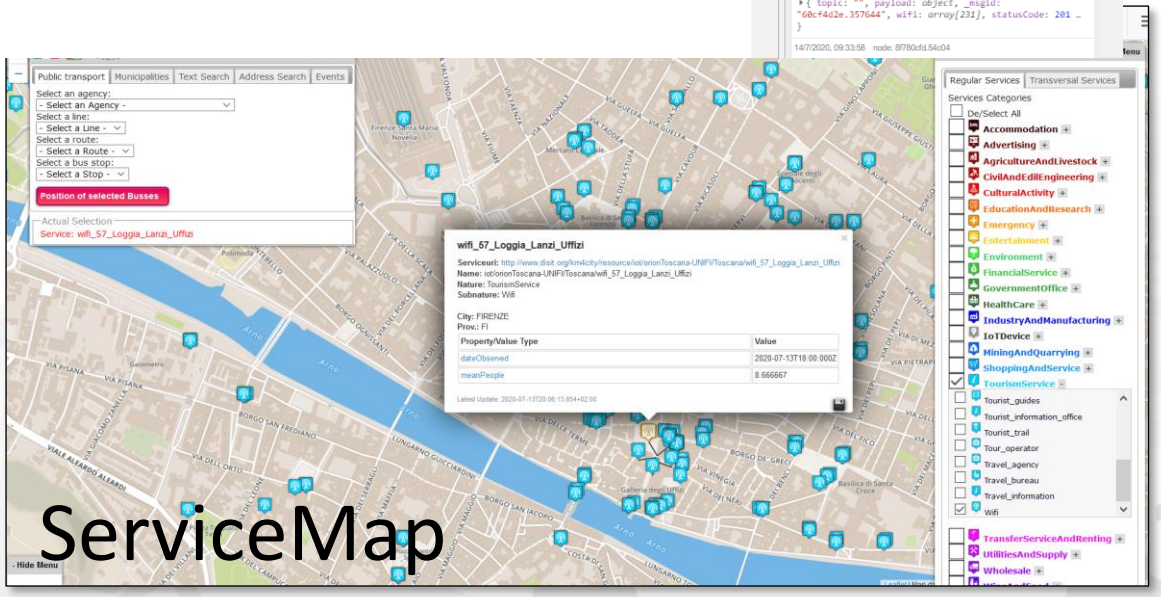
3) Group Creation (more than 200 devices) -> put all the devices in the group and put them as 'public' (or they remain private)



IoTApp Dynamic Flow





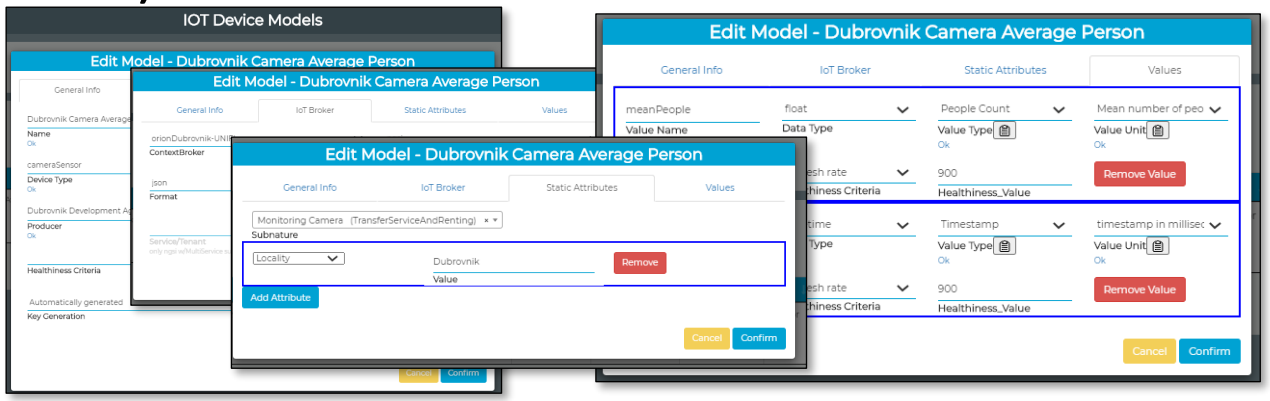


4) Send RT data to the IoTDevices

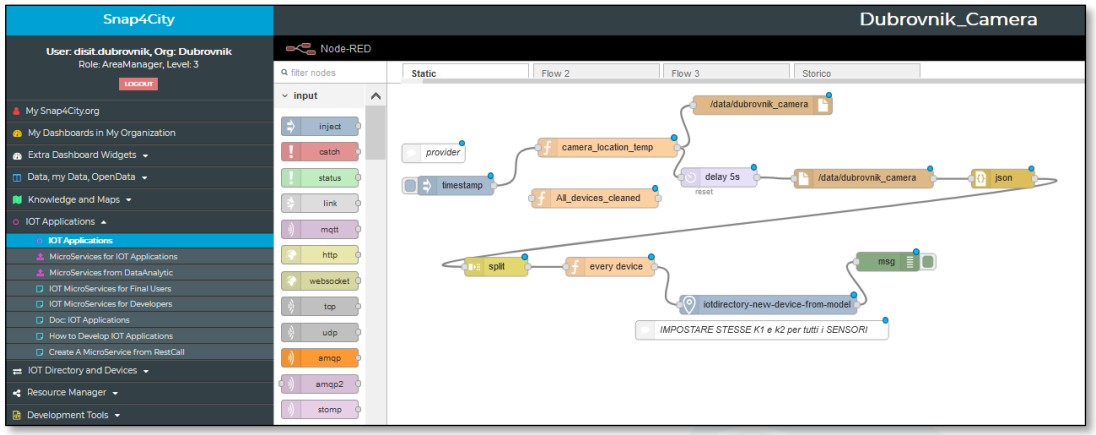
5) Verify RT Data via Snap4City API or via ServiceMap

Work in Progress...

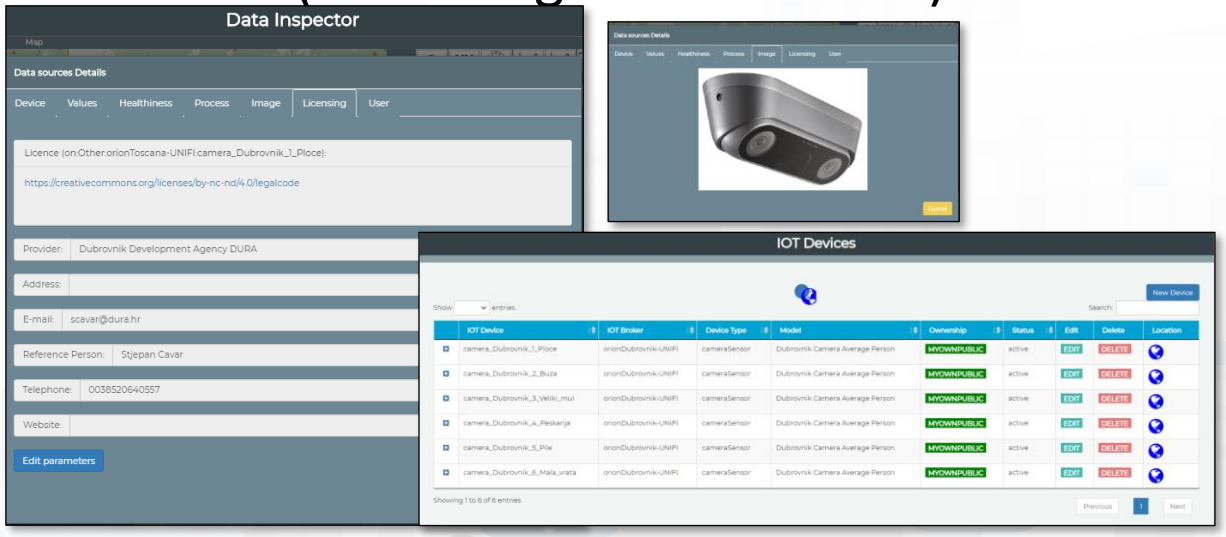
1) IoTModel



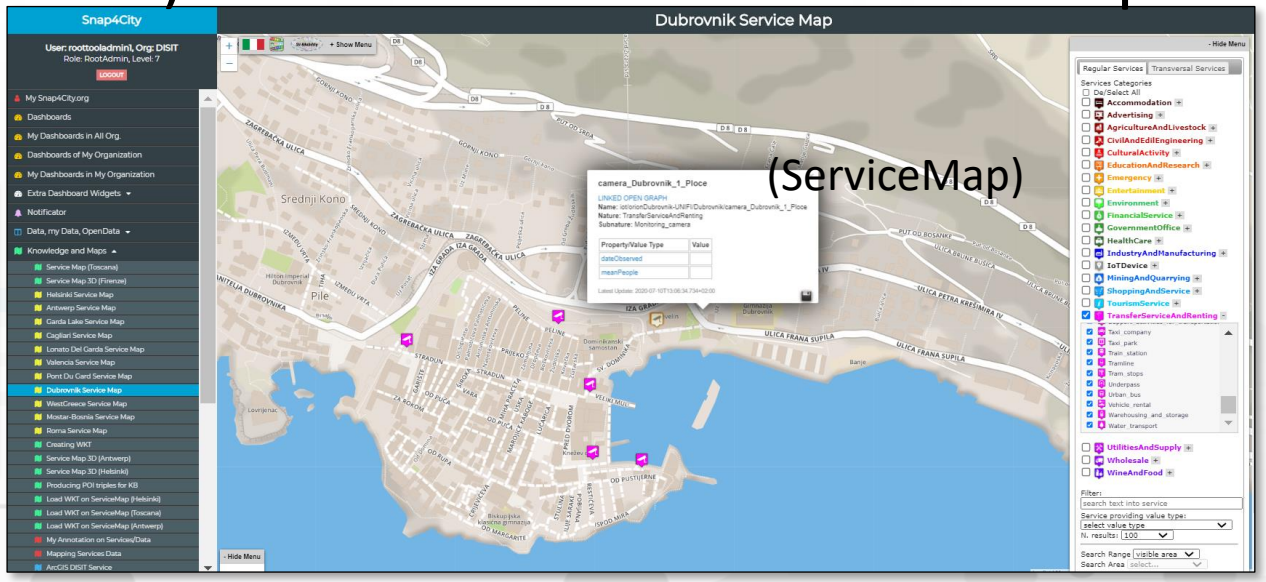
2) Static Flow to create IoTDevices



3) Add the license and Make Public the IoTDevices (according to the license)



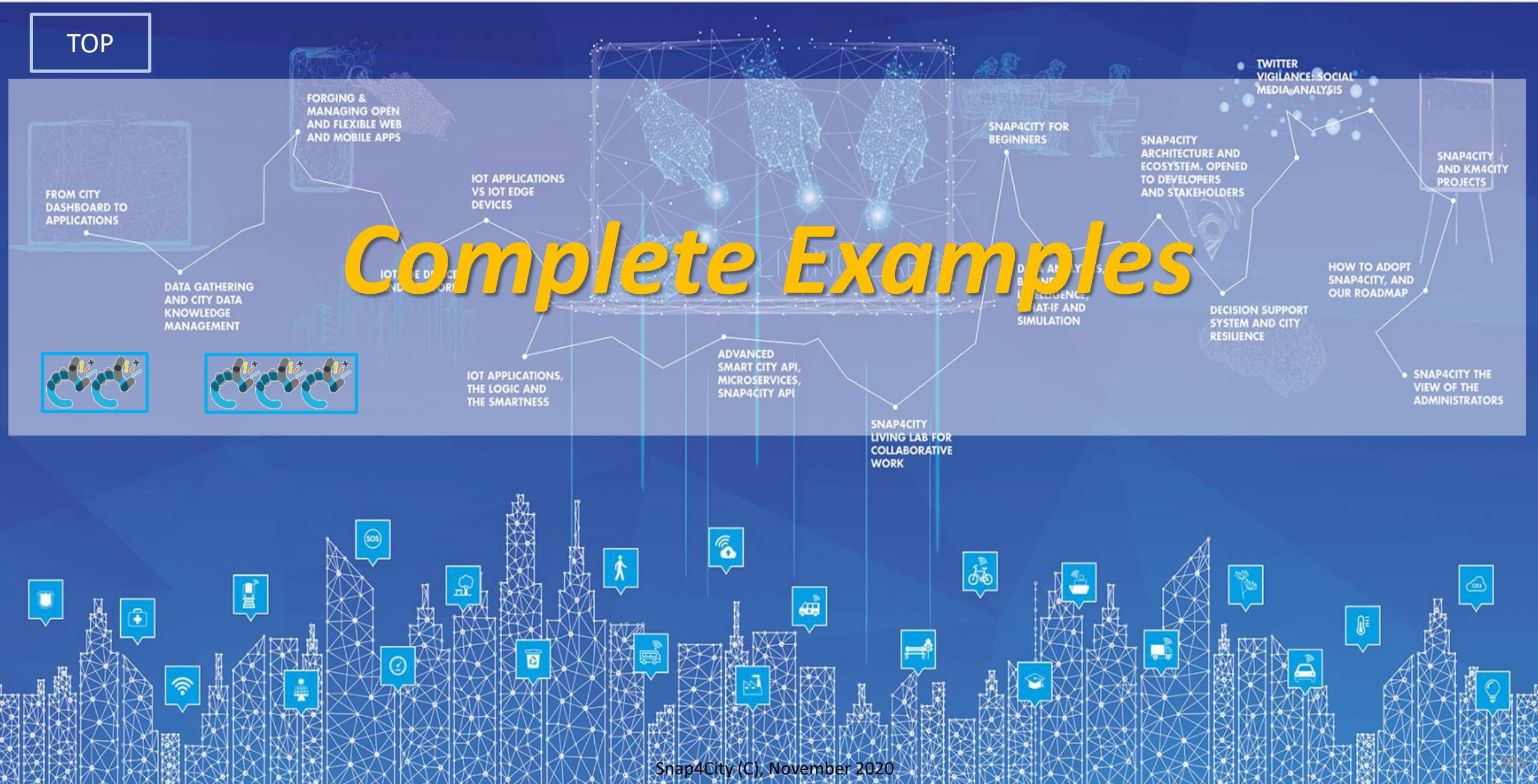
3) Search for the Cameras on Map (ServiceMap)



5) **Working on** Dynamic Flow to save Average #people every 15 minutes for each IoTDevice

TOP

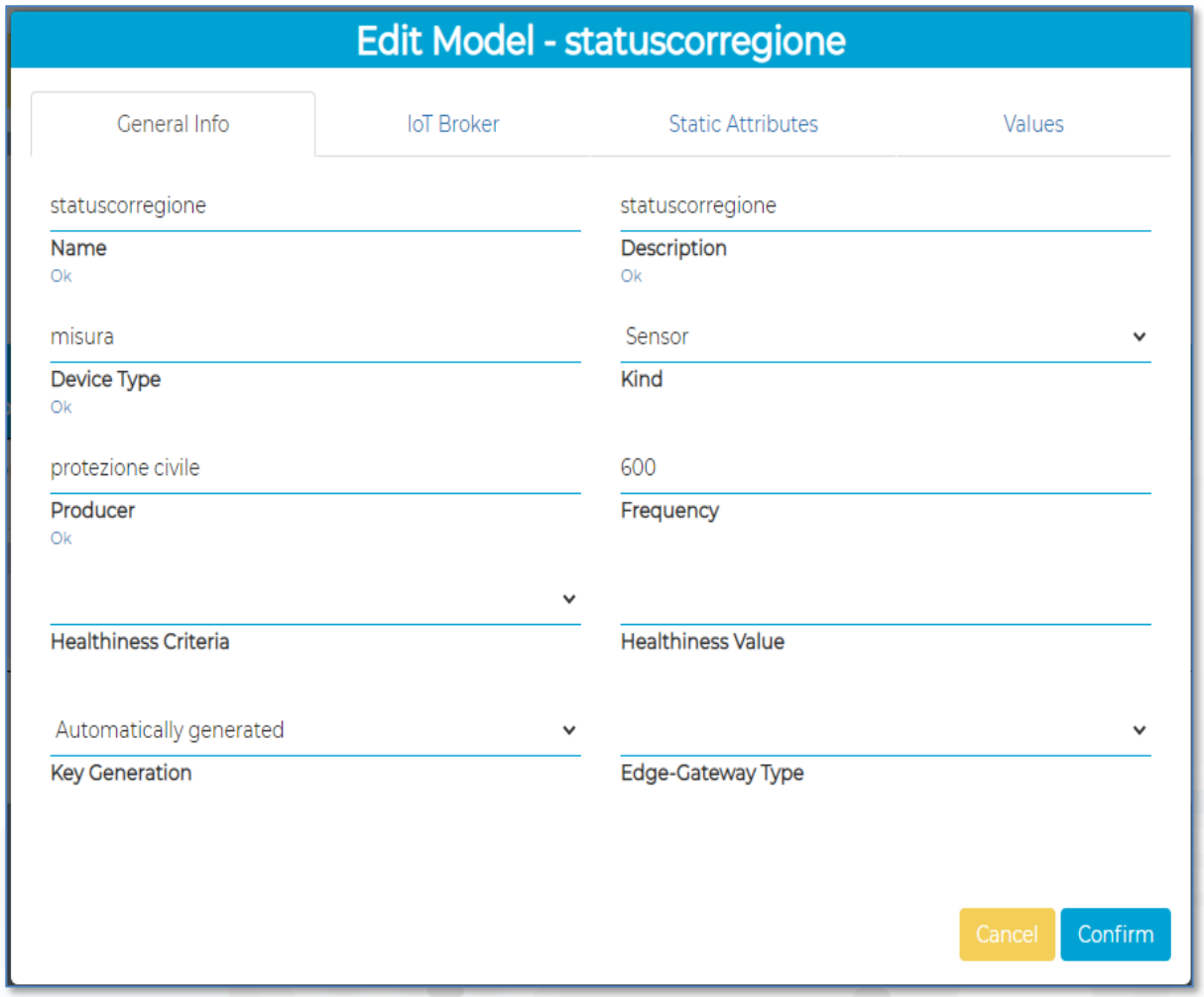
Complete Examples

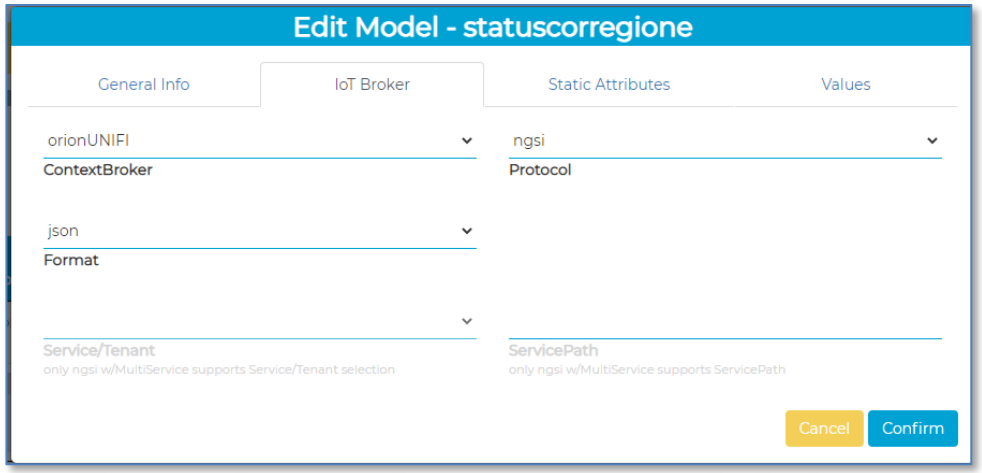


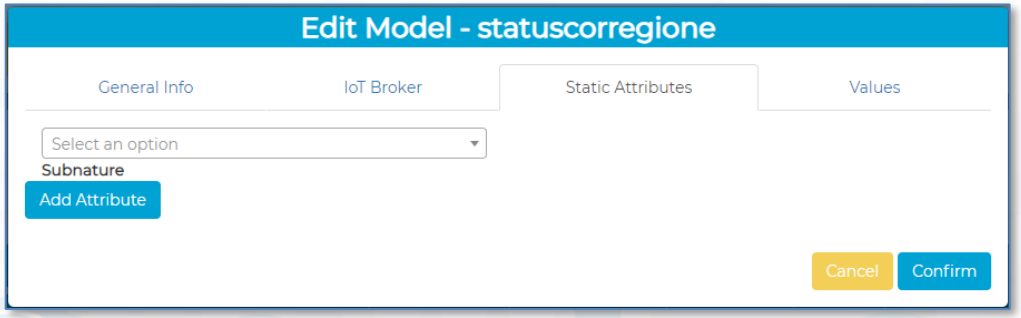
A Complete Example for Time Series: IOT Device Model + IOT Data Ingestion

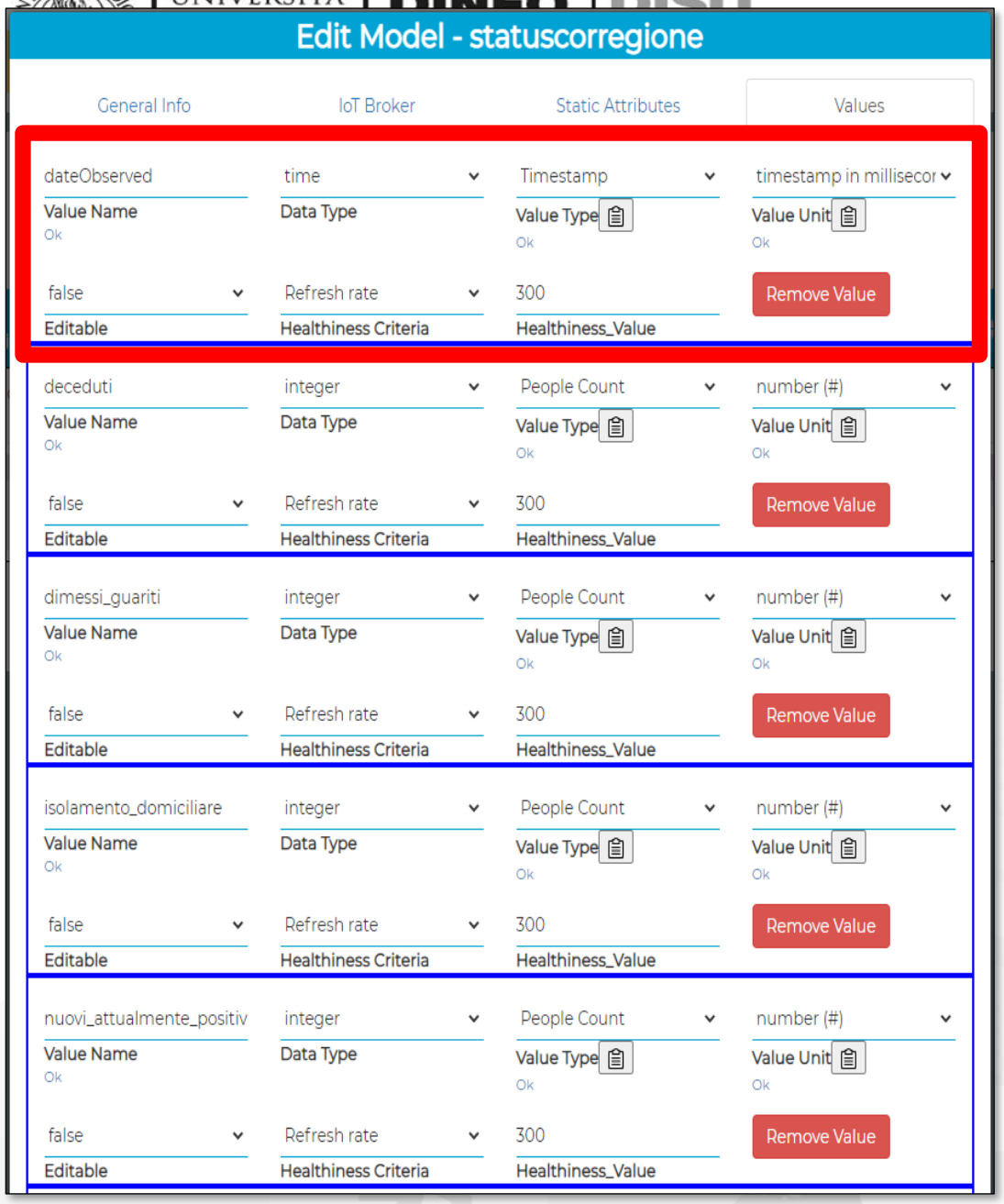


I have created an IOT Device Model as:



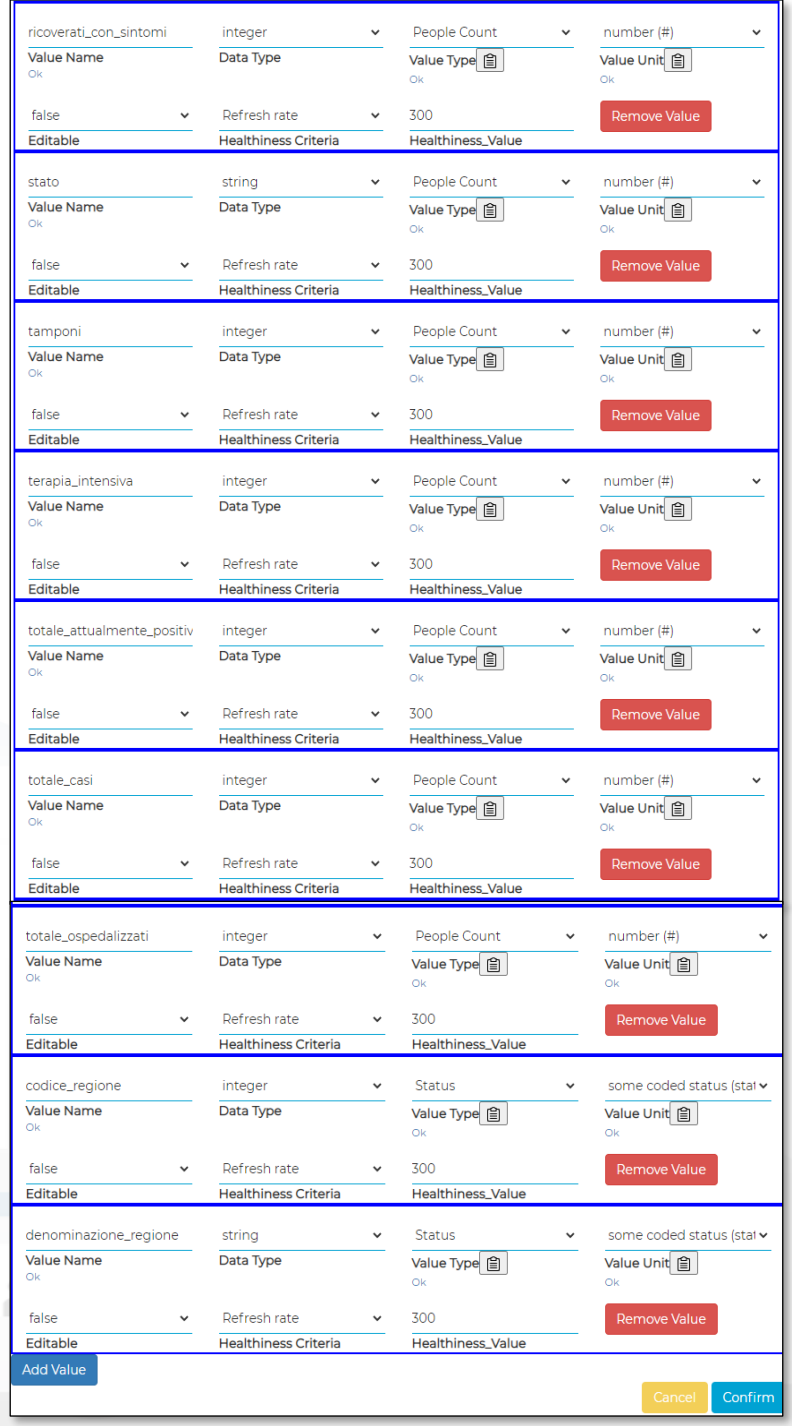




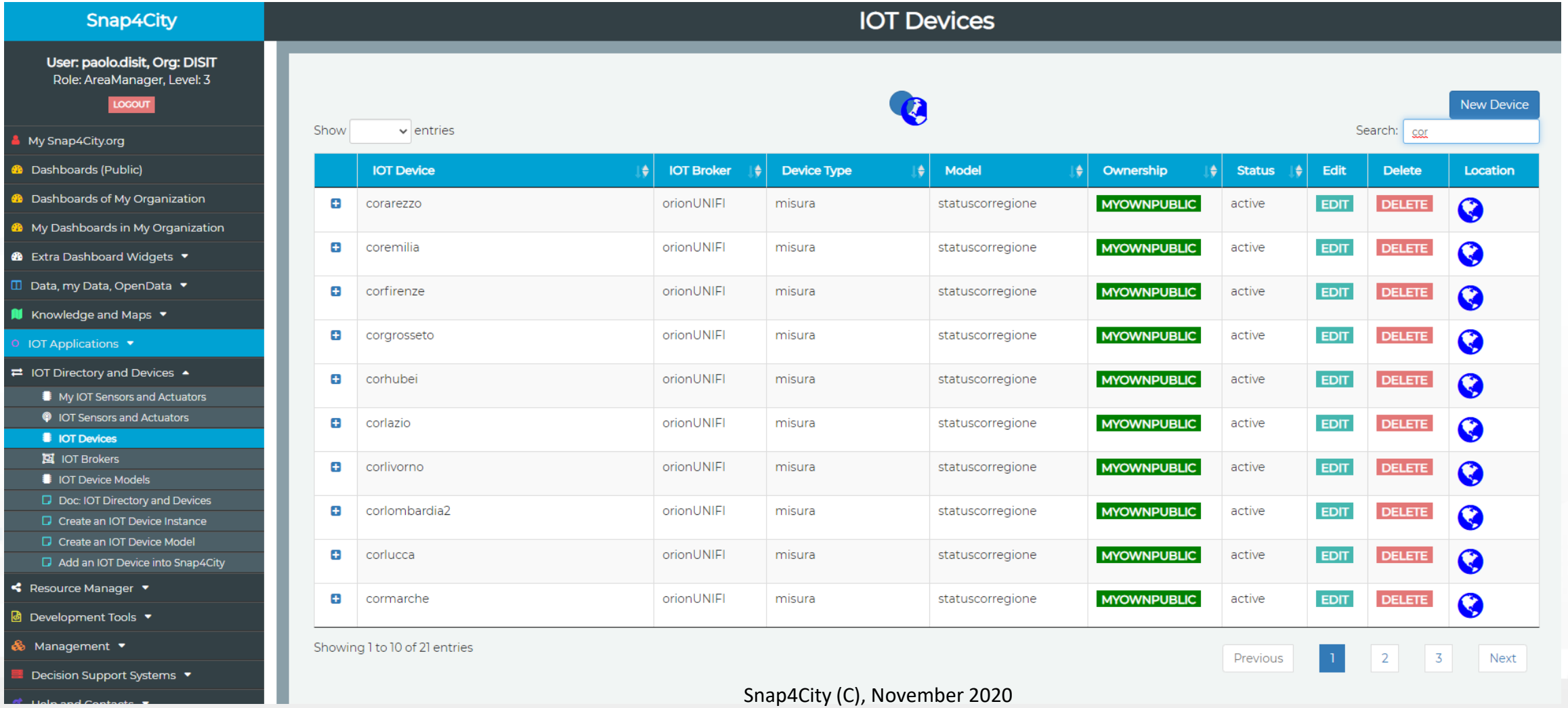


For Time Series

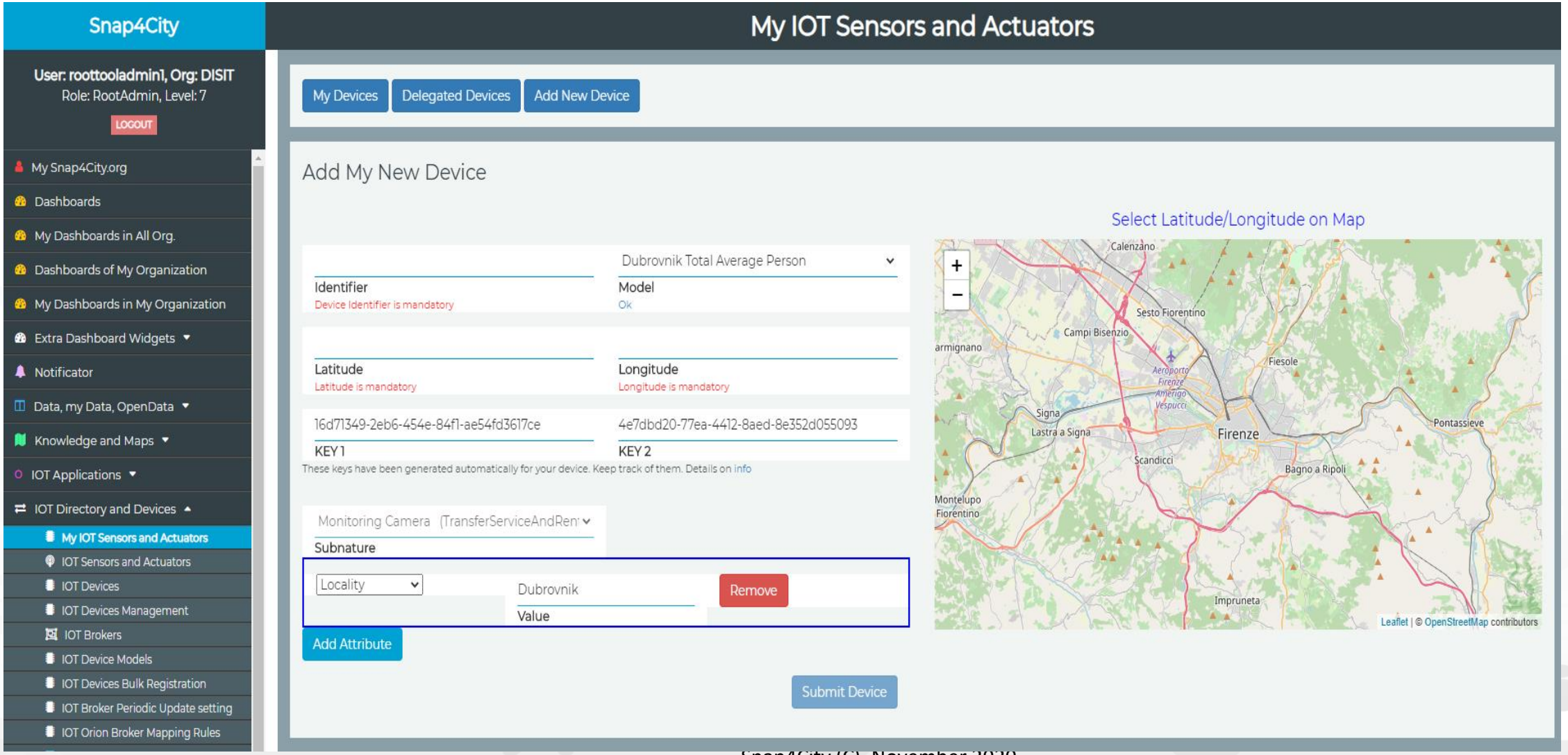
- ValueName: dateObserved
- DataType: time
- ValueType: timestamp
- ValueUnit: timestamp in millisecond
- E.g.: ISO string of the date-time



From IOT Model I have created some instances: the IOT Devices



They have been created by «Add new Device»



IOT Device from IOT Model by Providing:

- **NAME (it has to be unique)**
- Select the IOT Model: «**statuscorregione**»
  - Thus the K1, K2 appears since the model is associated to an Orion Broker that needs to have them, the tool generate them for you but you can impose if you like
  - See in previous slide the ID name of the IOT Broker used
- **Lat** and **Lon**, GPS coordinates you can:
  - pick on the map
  - Write the coordinates manually and see the pin on map

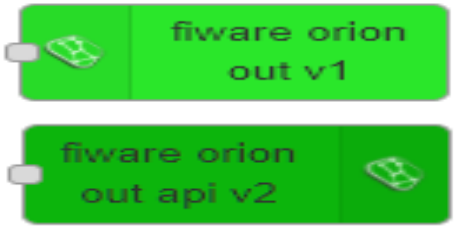
Once created the IOT Device you may send data on it



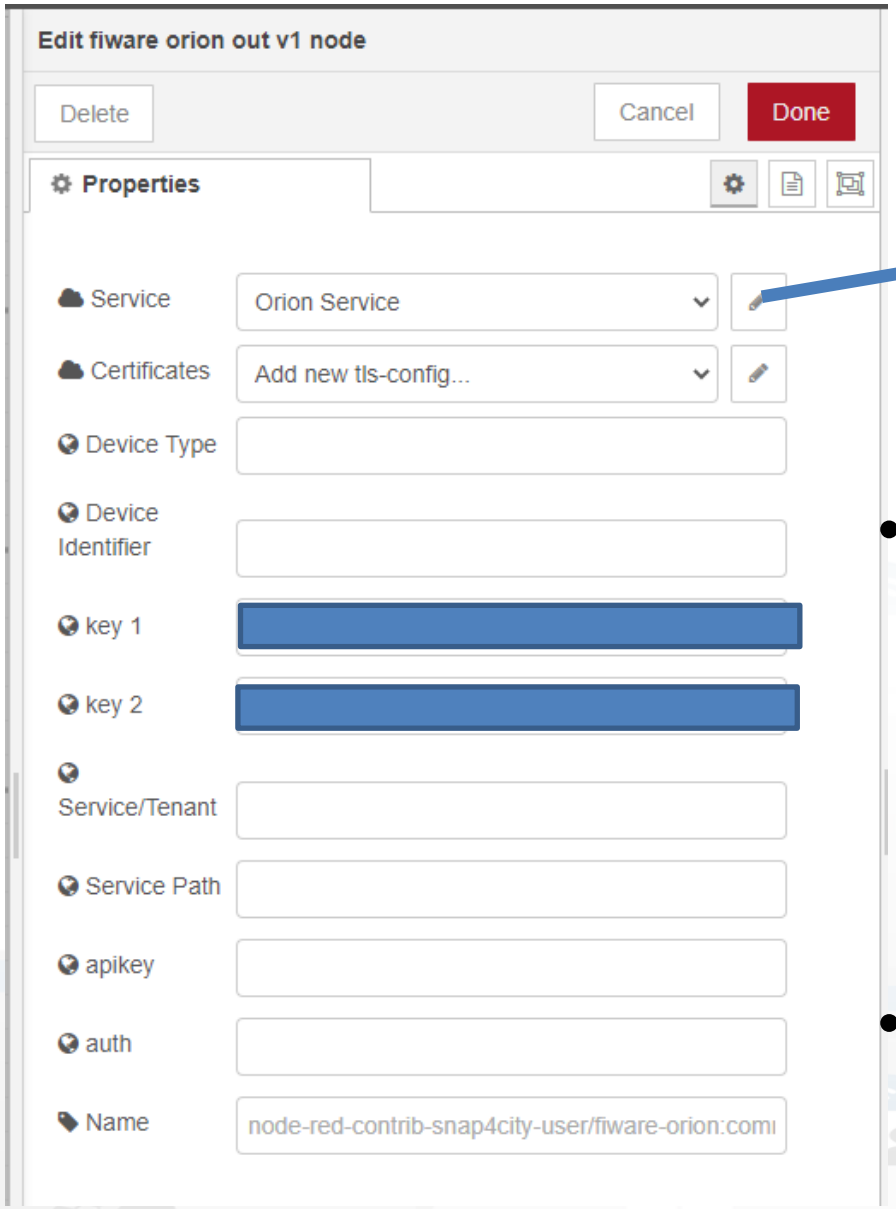


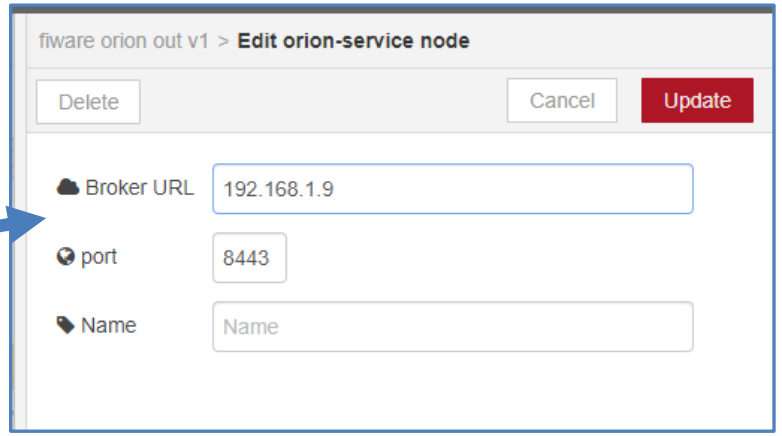
- You may create an IOT App, where:
  - Function: is preparing the JSON package
  - Block «Fi-Ware Orion OUT V1» or V2 is sending the data to the Orion Broker. Namely: «OrionUNIFI»
  - Please note that several version of IOT ORION Brokers and protocols exists:
    - So that you have to know which protocols you need to use for your broker

Settings ?



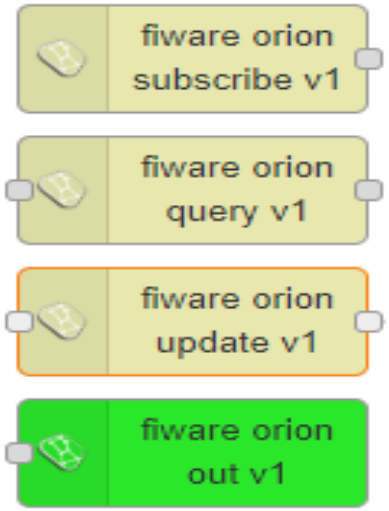
- Certificates are automatically loaded at the first authentication
- Done!!



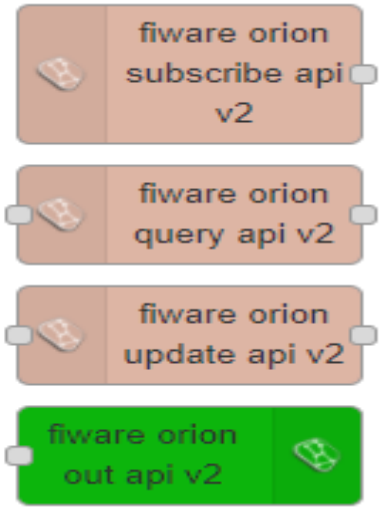


- IP if the Broker is in cloud (internal)
  - List of brokers is automatically provides
  - The K1, K2 is automatically provided if you are authenticated
- Symbolic address of IOT Broker can be taken from IOT Directory

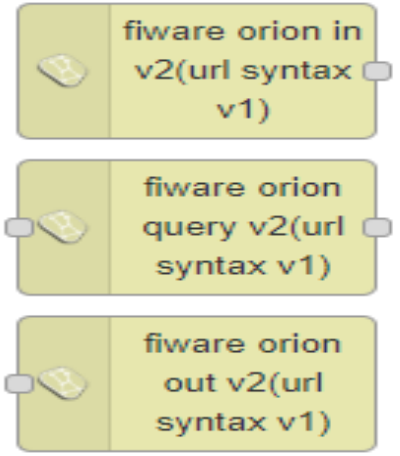
NGSI versions



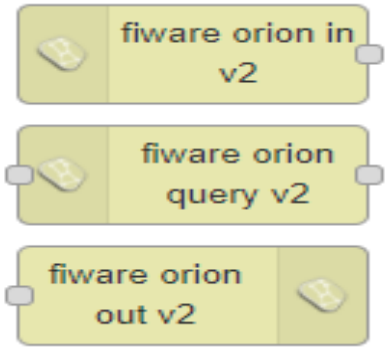
- Orion Broker of V1 with **NGSI syntax of V1** + Secure Filter of Snap4city



- Orion Broker of V2 with **NGSI syntax of V2** + Secure Filter of Snap4city



- Orion Broker of V2 with NGSI syntax of V1



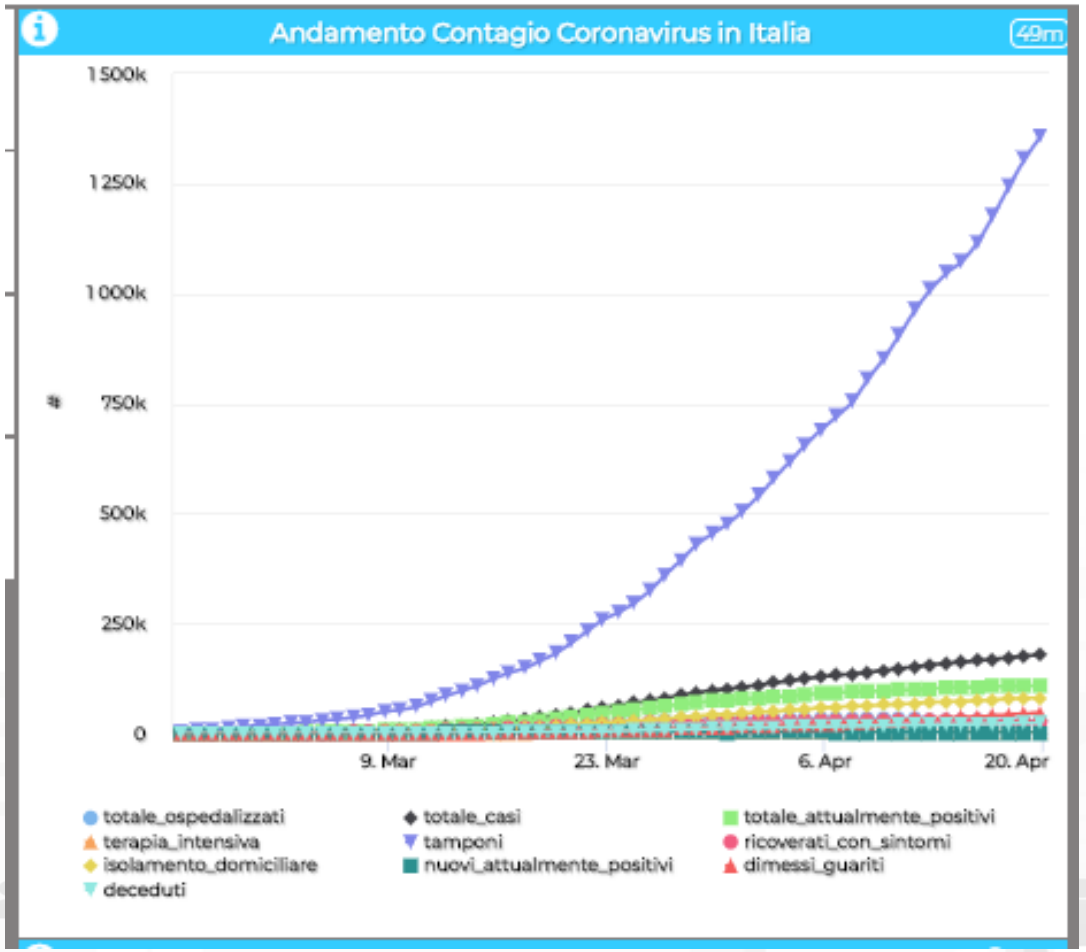
- Orion Broker of V2 with NGSI syntax of V2

- A Json from the IOT App
- **ID**: The Name of the IOT Device: «corveneto»
- **Type** as that defince in the IOT Device when you created
- The **Time stamp**: “dateObserved” to have a time series data
  - “str” is a string with the date and time in standard ISO, such as ,
  - “2020-08-04T04:00:00+02:00”,
  - “2020-08-03T00:00:00.000Z”
- And the **vector** of “**attributes**”

```
msg = { payload : {
 "id": "corveneto",
 "type": "misura",
 "attributes":[
{ "name": "dateObserved",  "value": str,  "type": "time" },
{ "name": "stato",  "value": “active”,  "type": "string" },
{ "name": "ricoverati_con_sintomi",  "value": 12,  "type": "integer" },
{ "name": "terapia_intensiva",  "value": 34,  "type": "integer" },
{ "name": "totale_ospedalizzati", "value": 34,  "type": "integer" },
{ "name": "isolamento_domiciliare",  "value": 334,  "type": "integer" },
{ "name": "totale_attualmente_positivi",  "value": 12,  "type": "integer" },
{ "name": "nuovi_attualmente_positivi", "value": 33,  "type": "integer" },
{ "name": "dimessi_guariti",  "value": 22222,  "type": "integer" },
{ "name": "deceduti",  "value": 2,  "type": "integer" },
{ "name": "totale_casi",  "value": 2222,  "type": "integer" },
{ "name": "tamponi",  "value": 222222344,  "type": "integer" }
]
}
}
return msg;
```

Multi Series Widget coming from the same IOT Device

- Over on the serie label to highlight
- Click on the serie label to on/ok
- Over on the graph to see the values



https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjU2OQ==

Andamenti Nazionali e Regionali infezione COVID-19

Sulla base dei dati della protezione civile, elaborazioni DISITLab

Sun 19 Apr 19:21:39

per evidenziare gli andamenti di vostro interesse: eliminare le curve che non interessano selezionandole in legenda.
Alcuni dati in passato non sono pervenuti alla protezione civile

486 nuovi_att_positivi

108257 Attualmente_positivi

23660 deceduti

47055 dimessi_guariti

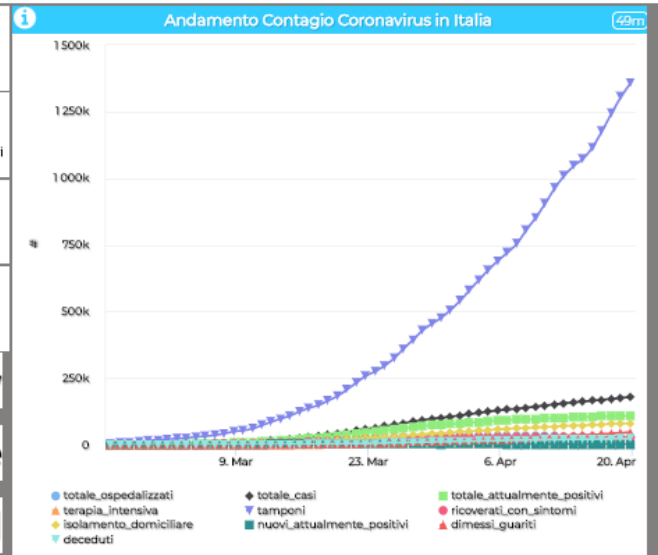
LOG view

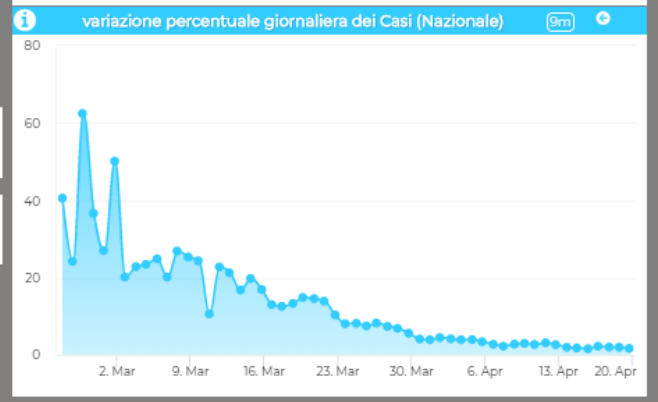
Mobile

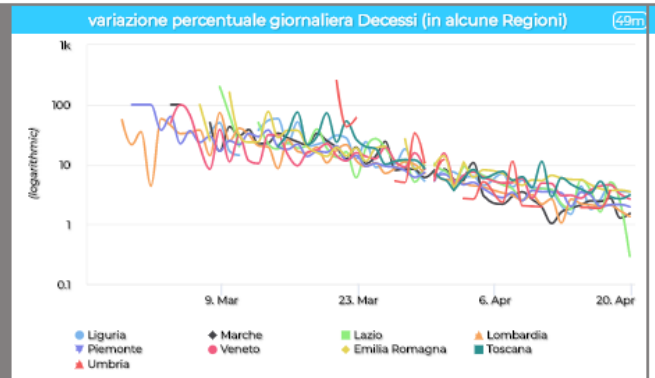
Staked

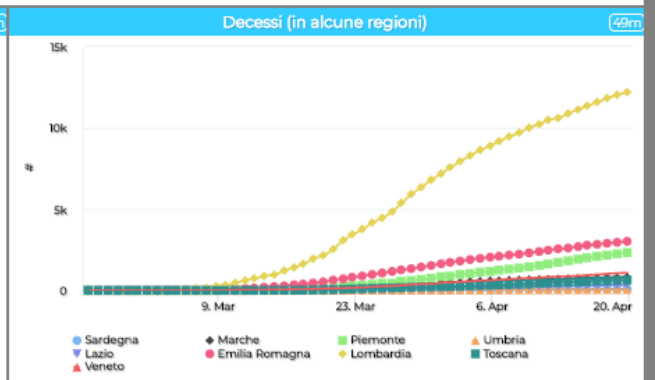
Regione Toscana

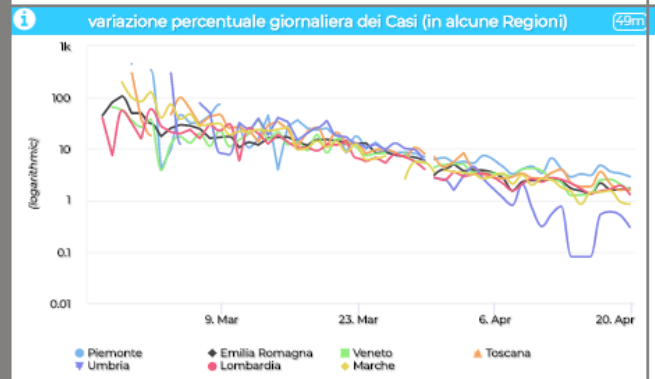
Regione LOG

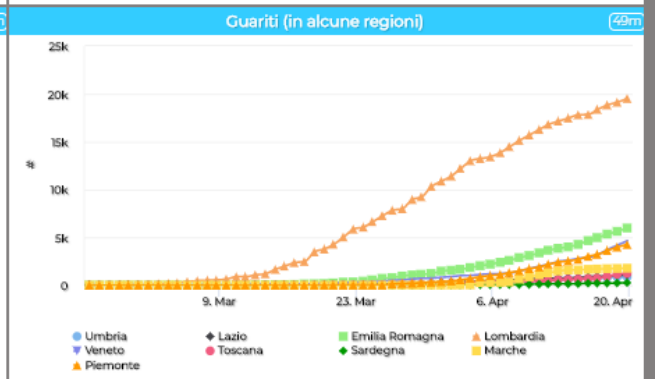


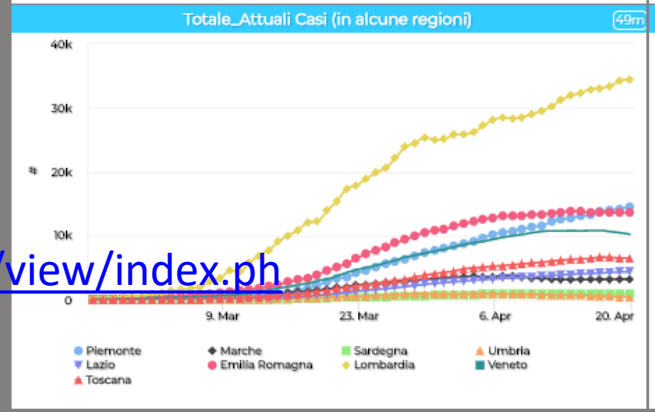


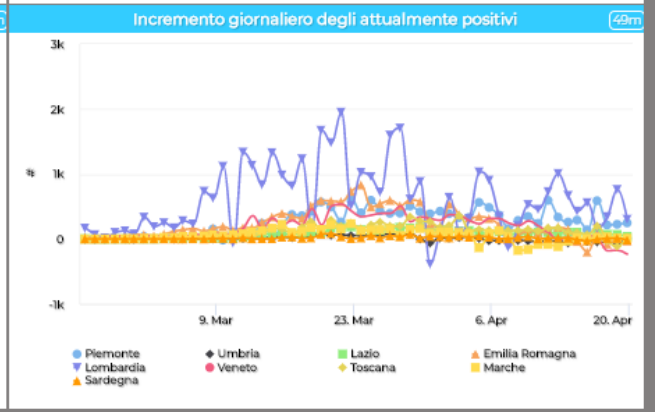










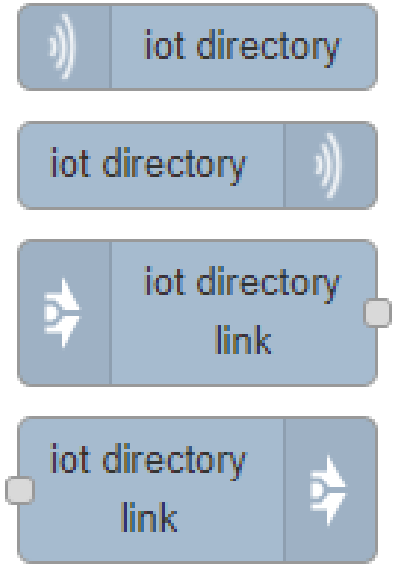


https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MjU2OQ==

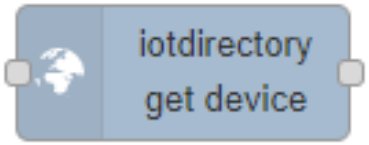
JSON for Authentication as well

- msg.auth= {
- "k1": "1ef0e5e8-yyyy-xxxx-9462-0aa4cfcf5e19",
- "k2": "b2b34425-yyyy-xxxx-818d-2d6cac2314a6",
- "apikey":"apikey",
- "basicAuth": "basicAuthKey"
- };

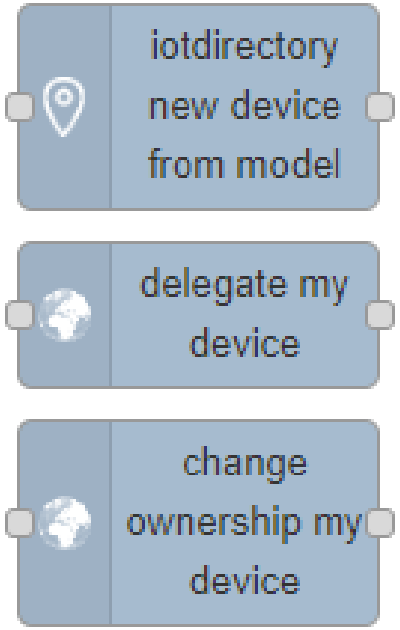
You may use other functions from IOT Directory



- IOT Discovery in an area
- Query on IOT Directory

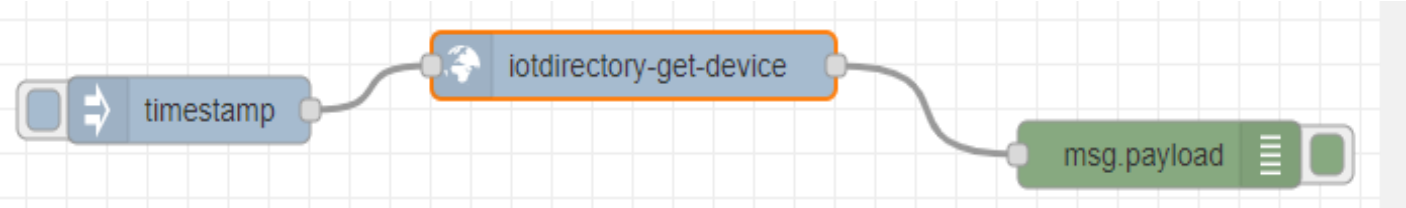


- Get IOT Device Info



- Registering an IOT Device from model
- Delegate an IOT Device
- Change Ownerhip of an IOT Device

Get IOT Device Info



- You can create smart IOT Applications that on t he basis of the list of IOT Devices would request all what you need to load data into **YOUR OWN** IOT Devices including:
  - **Service URI**
  - **K1, K2**
  - Authentication

msg.payload : Object

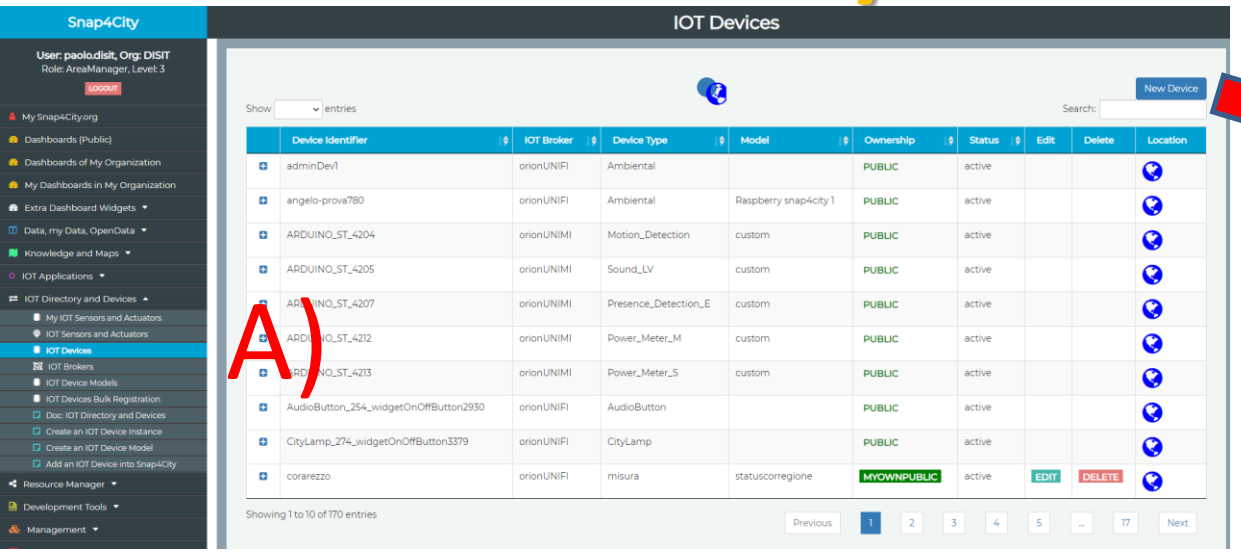
```
object
  status: "ok"
  content: object
    uri:
    "http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT
    /corarezzo"
    devicetype: "misura"
    kind: "sensor"
    status1: "active"
    macaddress: ""
    model: "statuscorregione"
    producer: "protezione civile"
    longitude: "11.88228"
    latitude: "43.46642"
    protocol: "ngsi"
    format: "json"
    visibility: "public"
    frequency: "600"
    created: "2020-03-21 18:34:32"
    privatekey: ""
    certificate: ""
    organization: "DISIT"
    accesslink: "https://broker1.snap4city.org"
    accessport: "8080"
    sha: "C61E32DBFAE7F14C0810177F2D2300843C41C550"
    subnature: null
    static_attributes: null
    k1: "bf739214-f6b4-45fe-85f7-97cd09fe8e57"
    k2: "c19e0b6f-8f98-4130-b135-e7a1dfae9273"
  error_msg: ""
```

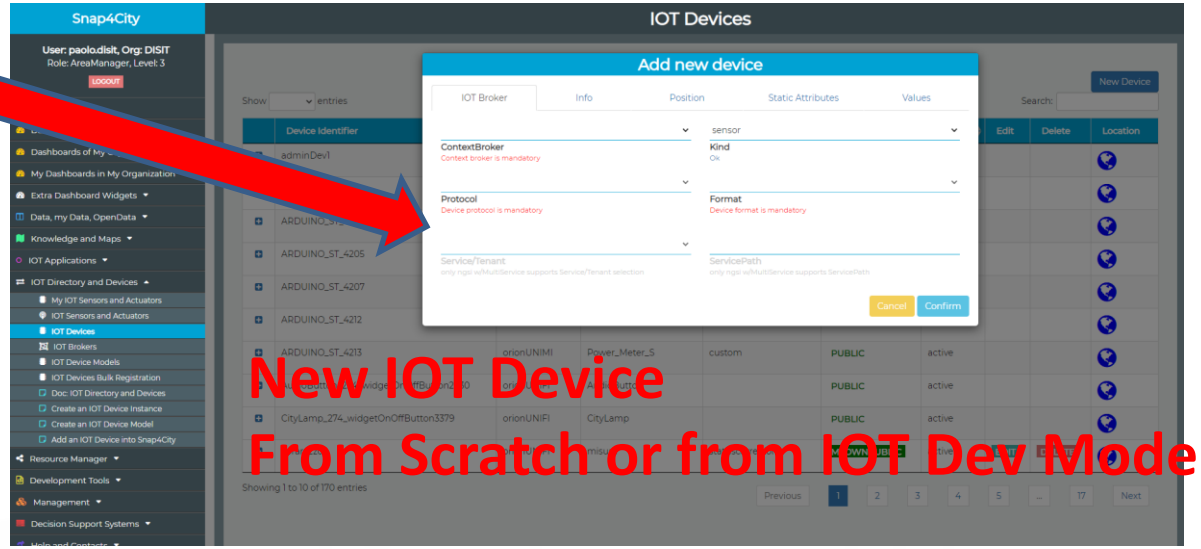
A Complete Example: Time Series, IOT Device Model, IOT Data Ingestion, Dynamic K1,K2 Management



Create/instantiate IOT Devices

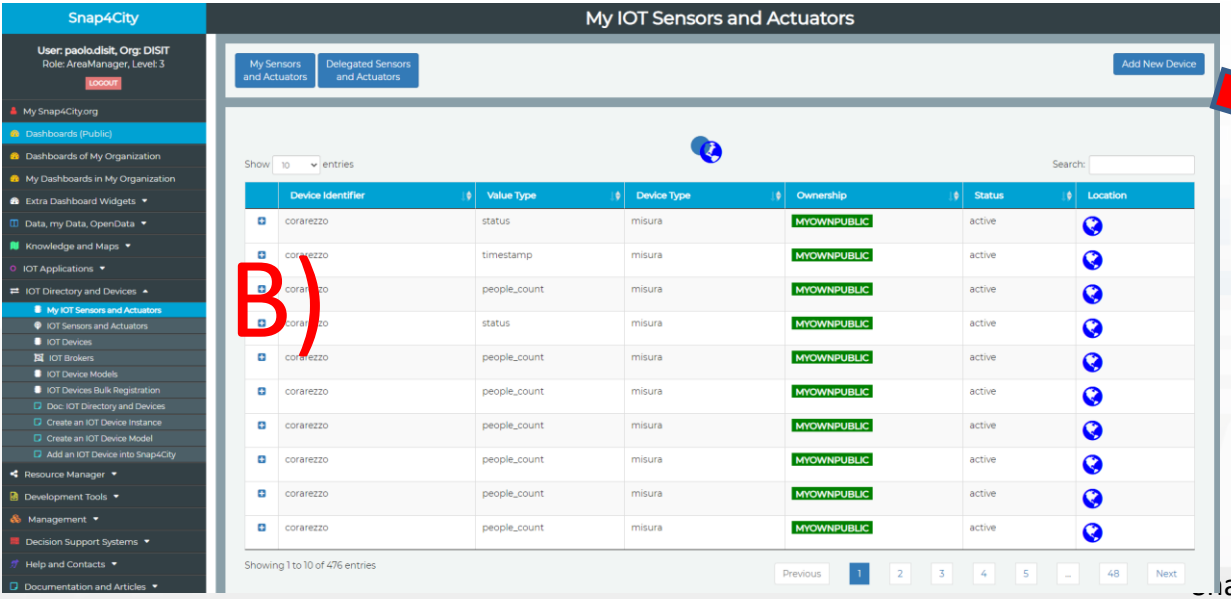
A)

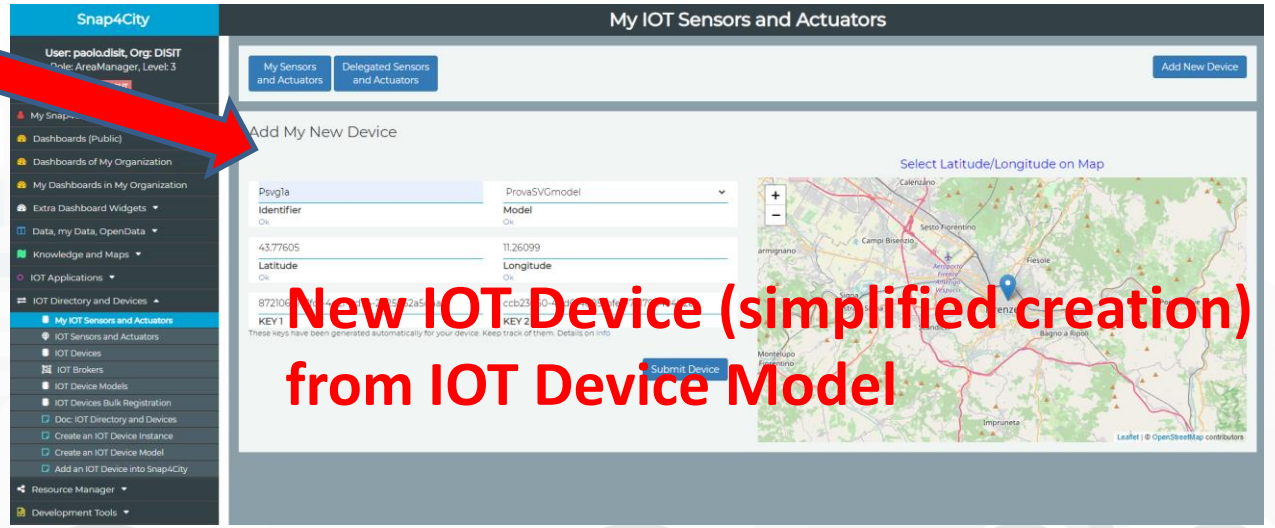




New IOT Device
From Scratch or from IOT Dev Model

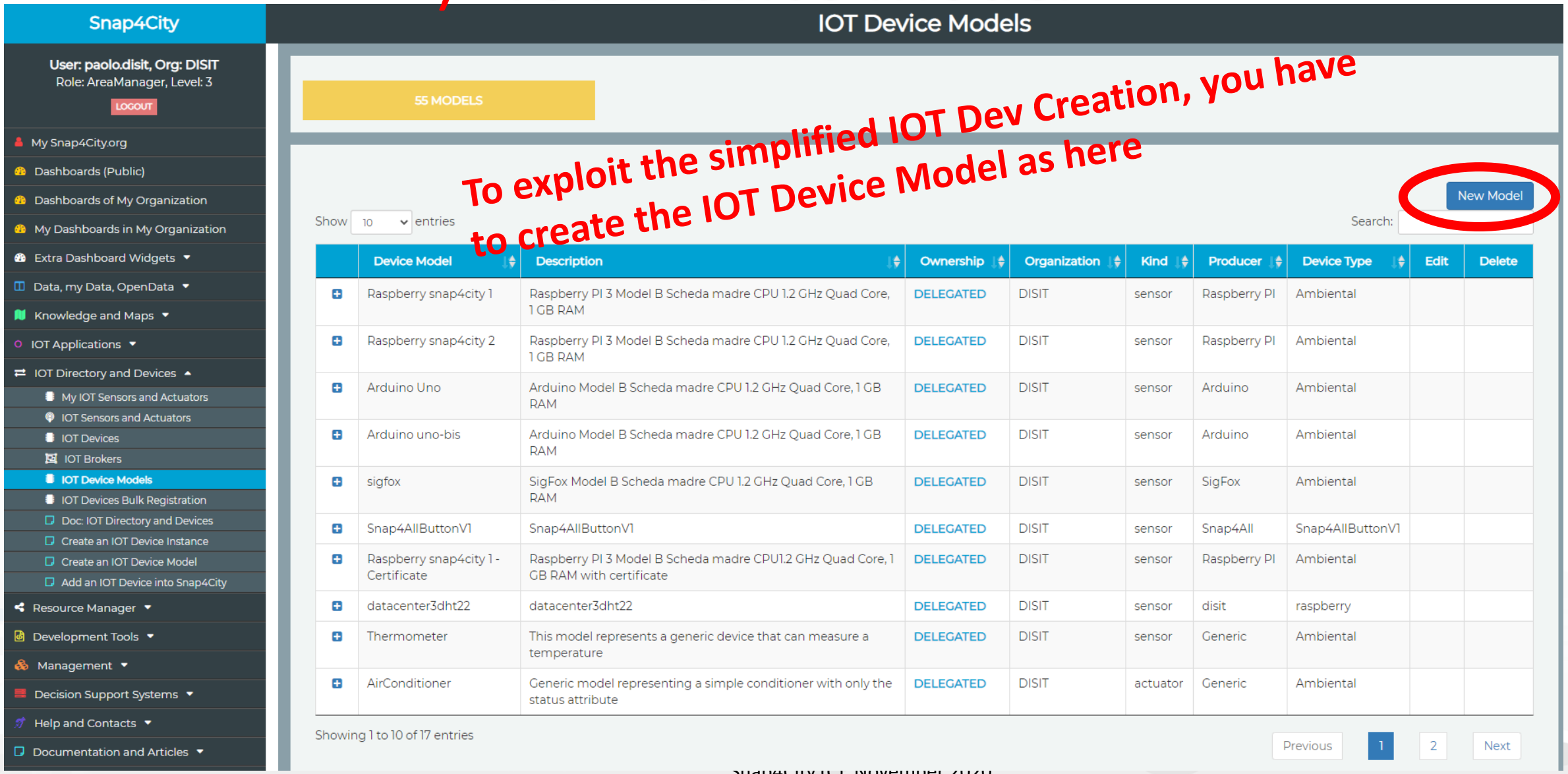
B)



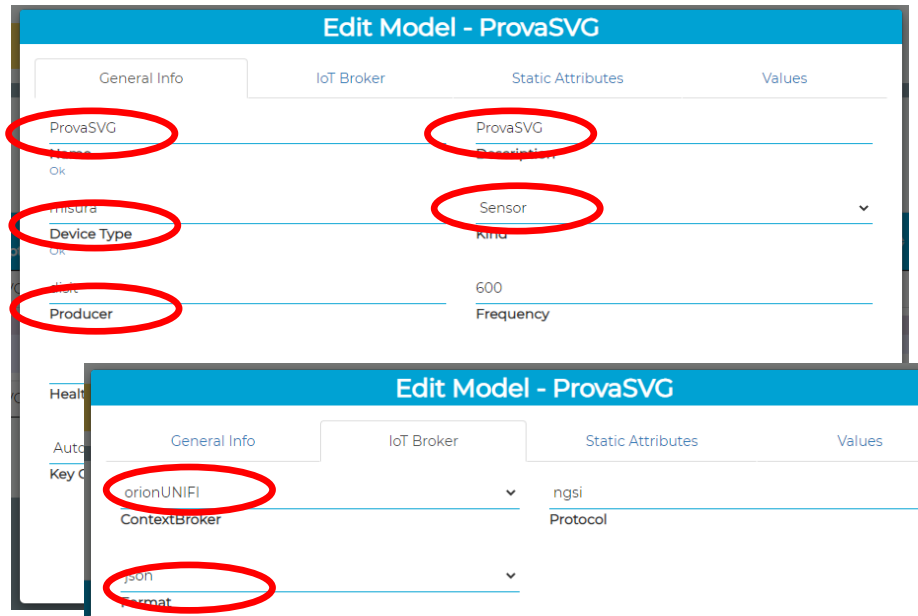


New IOT Device (simplified creation)
from IOT Device Model

B) IOT Device Model

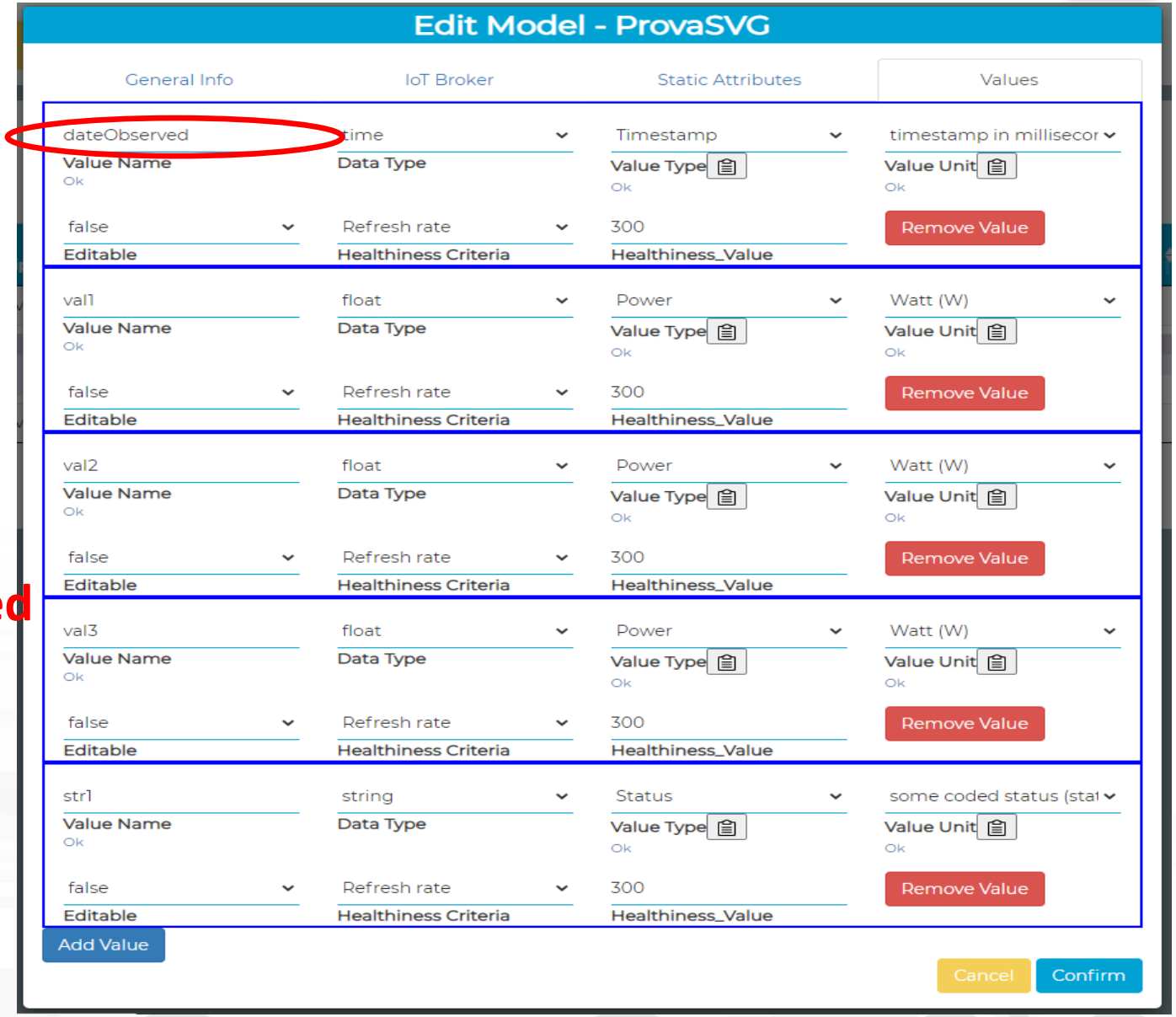


IOT Device Model

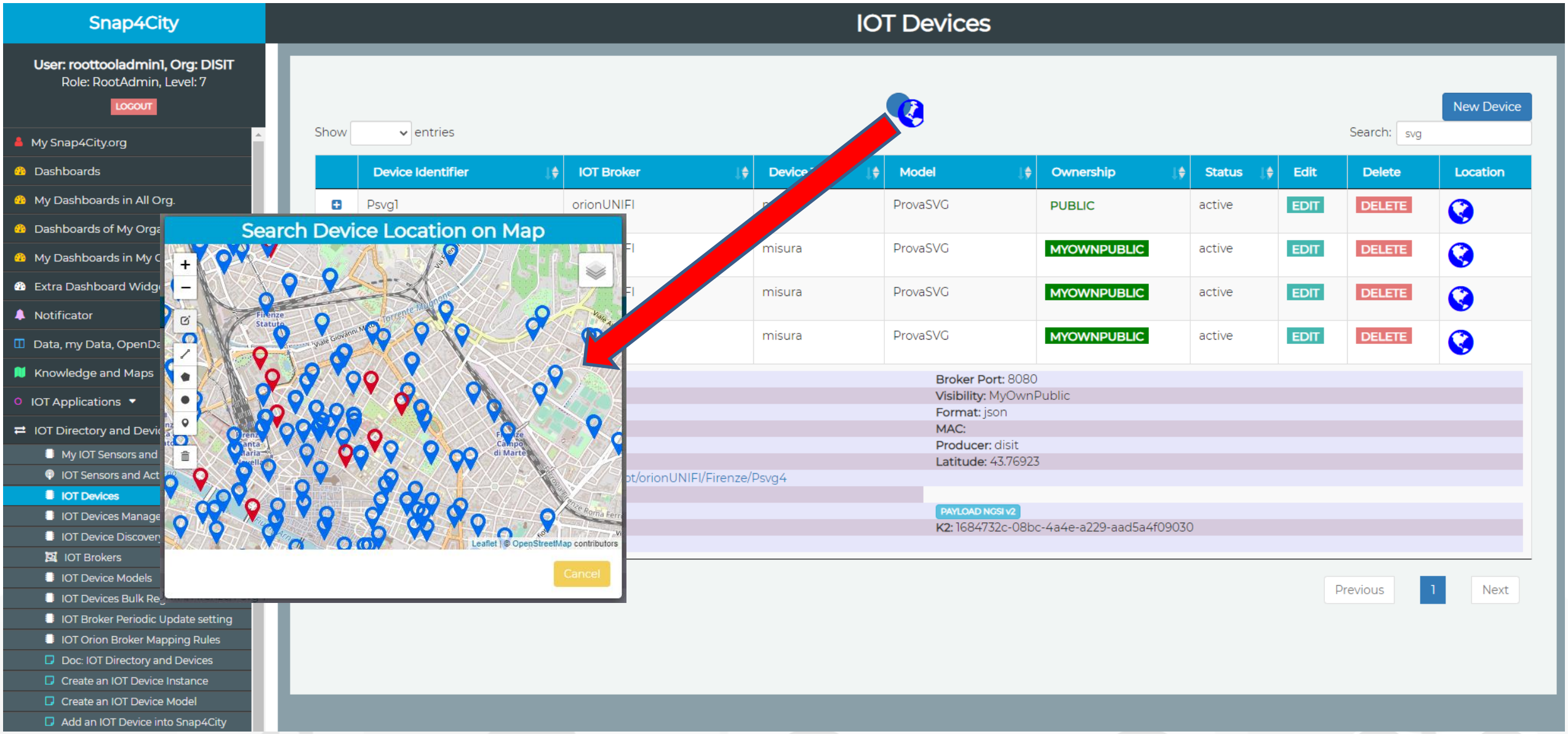


To exploit the simplified IOT Dev Creation, you have to create the IOT Device Model as here Do once and create as many IOT Device you like in shorter steps

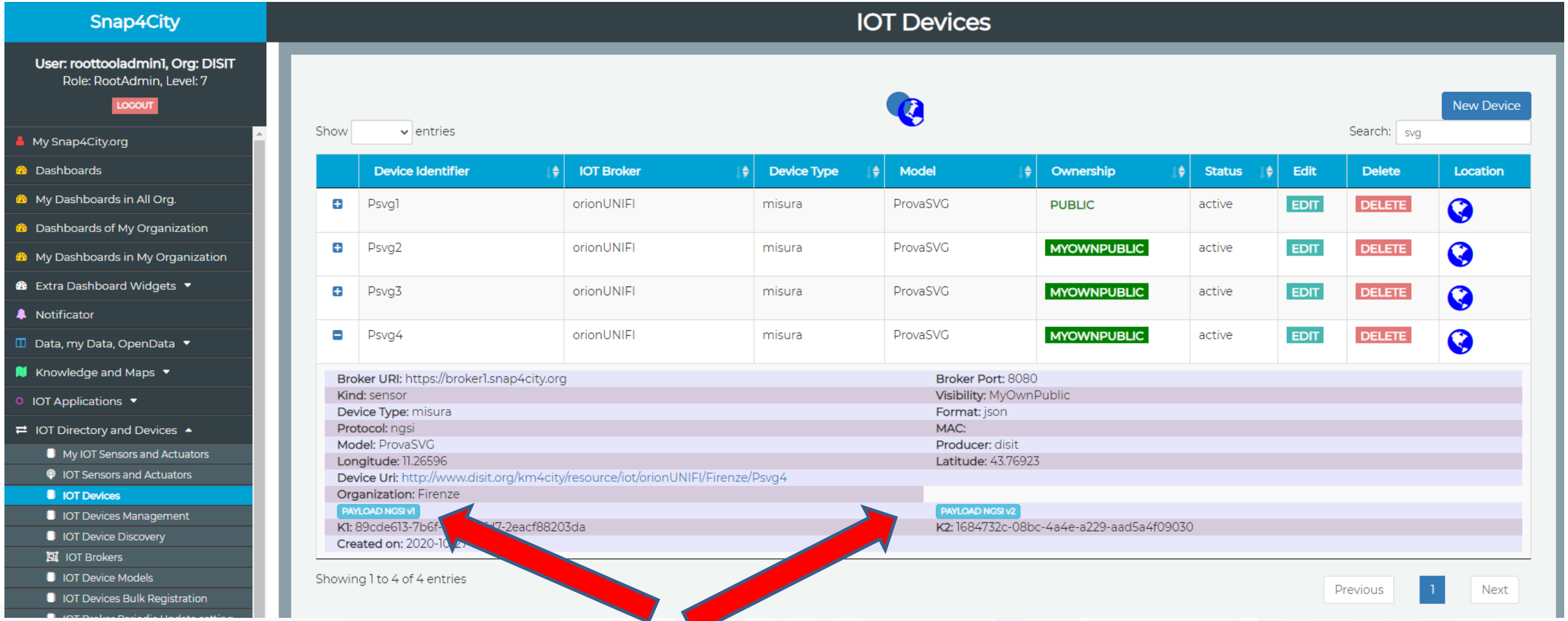
B)



IOT Devices


Snap4City
User: roottooladmin1, Org: DISIT
Role: RootAdmin, Level: 7
LOGOUT
My Snap4City.org
Dashboards
My Dashboards in All Org.
Dashboards of My Orga
My Dashboards in My C
Extra Dashboard Widg
Notificator
Data, my Data, OpenDa
Knowledge and Maps
IOT Applications
IOT Directory and Devic
My IOT Sensors and
IOT Sensors and Act
IOT Devices
IOT Devices Manage
IOT Device Discovery
IOT Brokers
IOT Device Models
IOT Devices Bulk Re
IOT Broker Periodic Update setting
IOT Orion Broker Mapping Rules
Doc: IOT Directory and Devices
Create an IOT Device Instance
Create an IOT Device Model
Add an IOT Device into Snap4City
IOT Devices
New Device
Show entries
Search: svg
Device Identifier
IOT Broker
Device
Model
Ownership
Status
Edit
Delete
Location
Psvg1
orionUNIFI
ProvaSVG
PUBLIC
active
EDIT
DELETE
misura
MYOWNPUBLIC
Broker Port: 8080
Visibility: MyOwnPublic
Format: json
MAC:
Producer: disit
Latitude: 43.76923
ot/orionUNIFI/Firenze/Psvg4
PAYLOAD NGSI V2
K2: 1684732c-08bc-4a4e-a229-aad5a4f09030
Previous
1
Next
Search Device Location on Map
Leaflet | © OpenStreetMap contributors
Cancel

See IOT Devices



Example of msg.payload to be used on IOT App compliant with the NGSI V1 and V2

Examples of NGSI V1 and V2, JSON Payload

NGSI V1



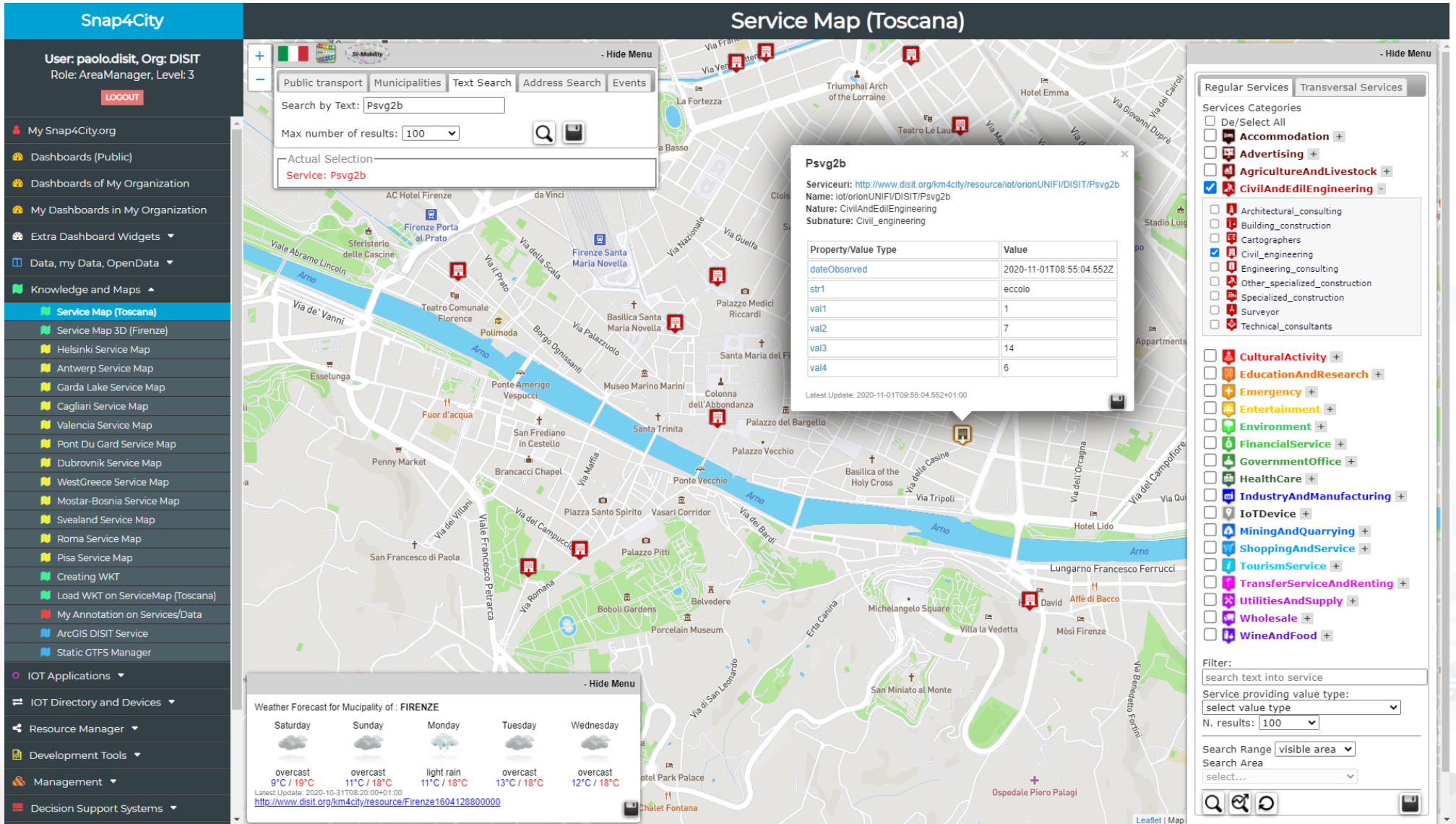

```
{
"type":"misura","id":"Psvg4",
"attributes":[
{"name":"dateObserved","type":"time",
   "value":"2020-11-06T13:25:12.191Z"},
{"name":"latitude","type":"float","value":"43.76923"},
{"name":"longitude","type":"float","value":"11.2659s"},
{"name":"str1","type":"string","value":"la mia"},
{"name":"val1","type":"float","value":"3.6"},
{"name":"val2","type":"float","value":5.77},
{"name":"val3","type":"float","value":“6.78"}
          ]
}
```


NGSI V2

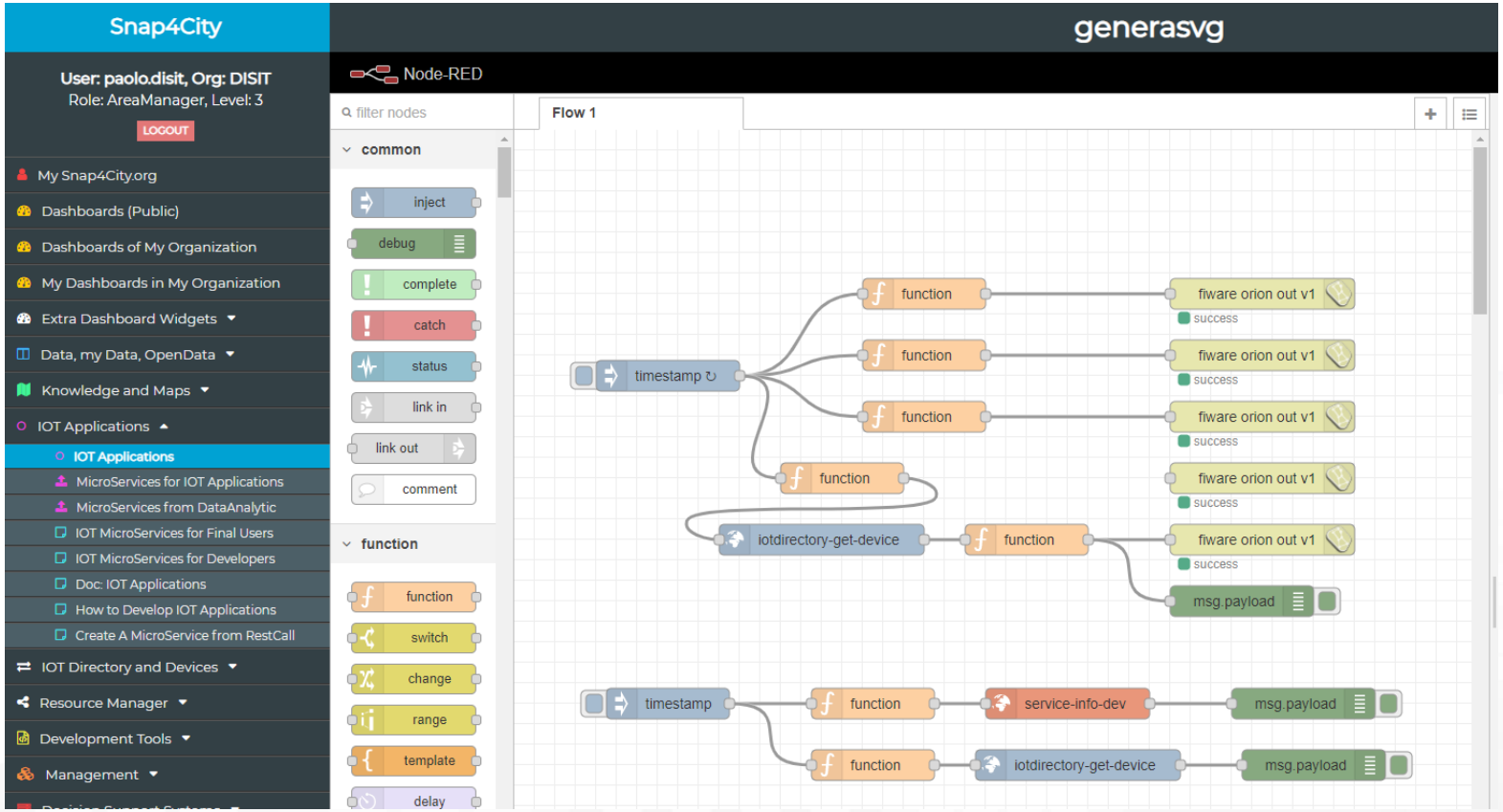



```
{
"id":"Psvg4","type":"misura",
"dateObserved":{"type":"time",
   "value":"2020-11-06T13:25:12.191Z"},
"latitude":{"type":"float","value":"43.76923"},
"longitude":{"type":"float","value":"11.26596"},
“str1":{"type":"string","value":"eccolo"},
"val1":{"type":"float","value":4.3},
"val2":{"type":"float","value":“4.6"},
"val3":{"type":"float","value":""}
}
```

Opla! See them Created on ServiceMap

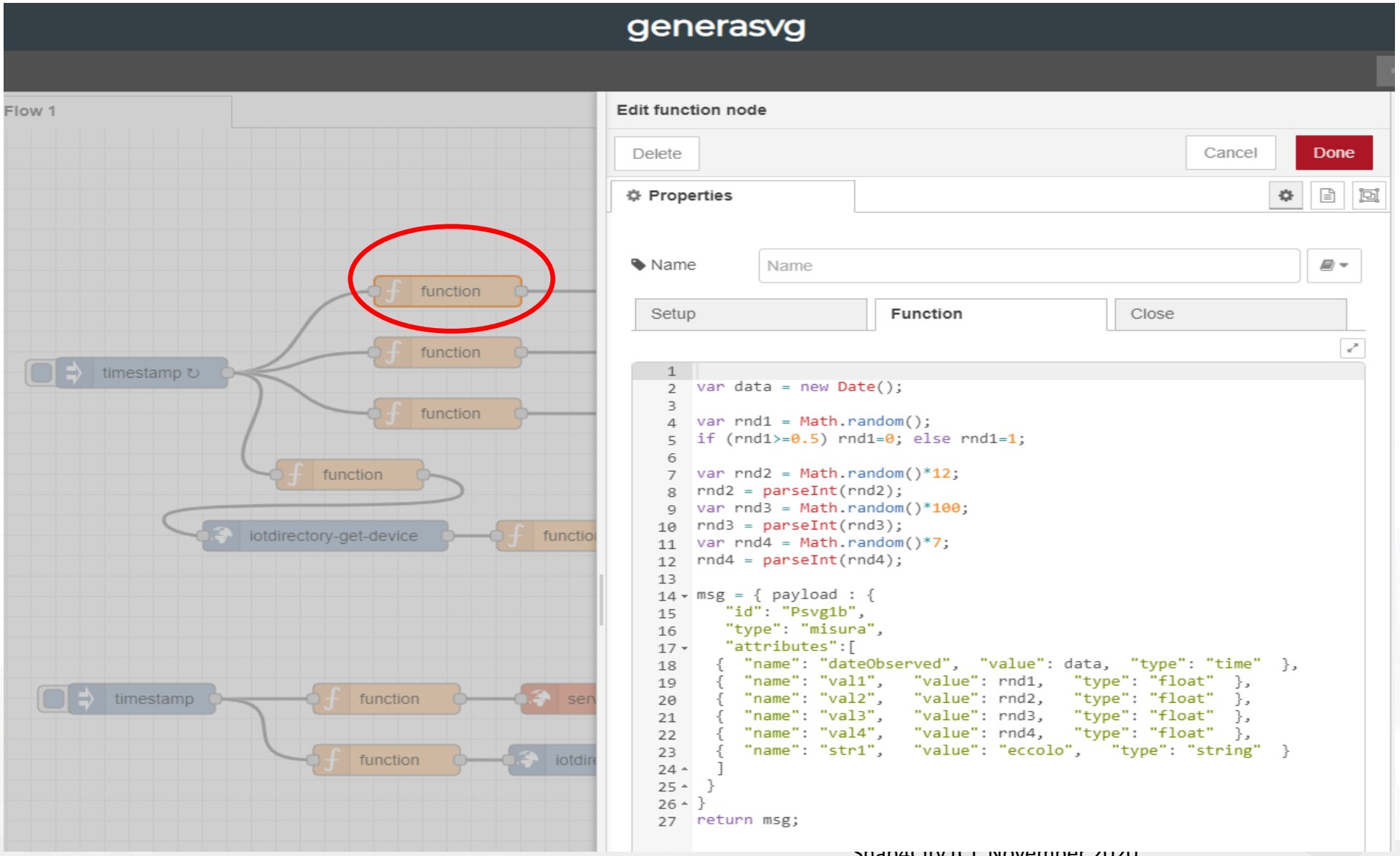


Saving Data in the IOT Device: Data Shadow



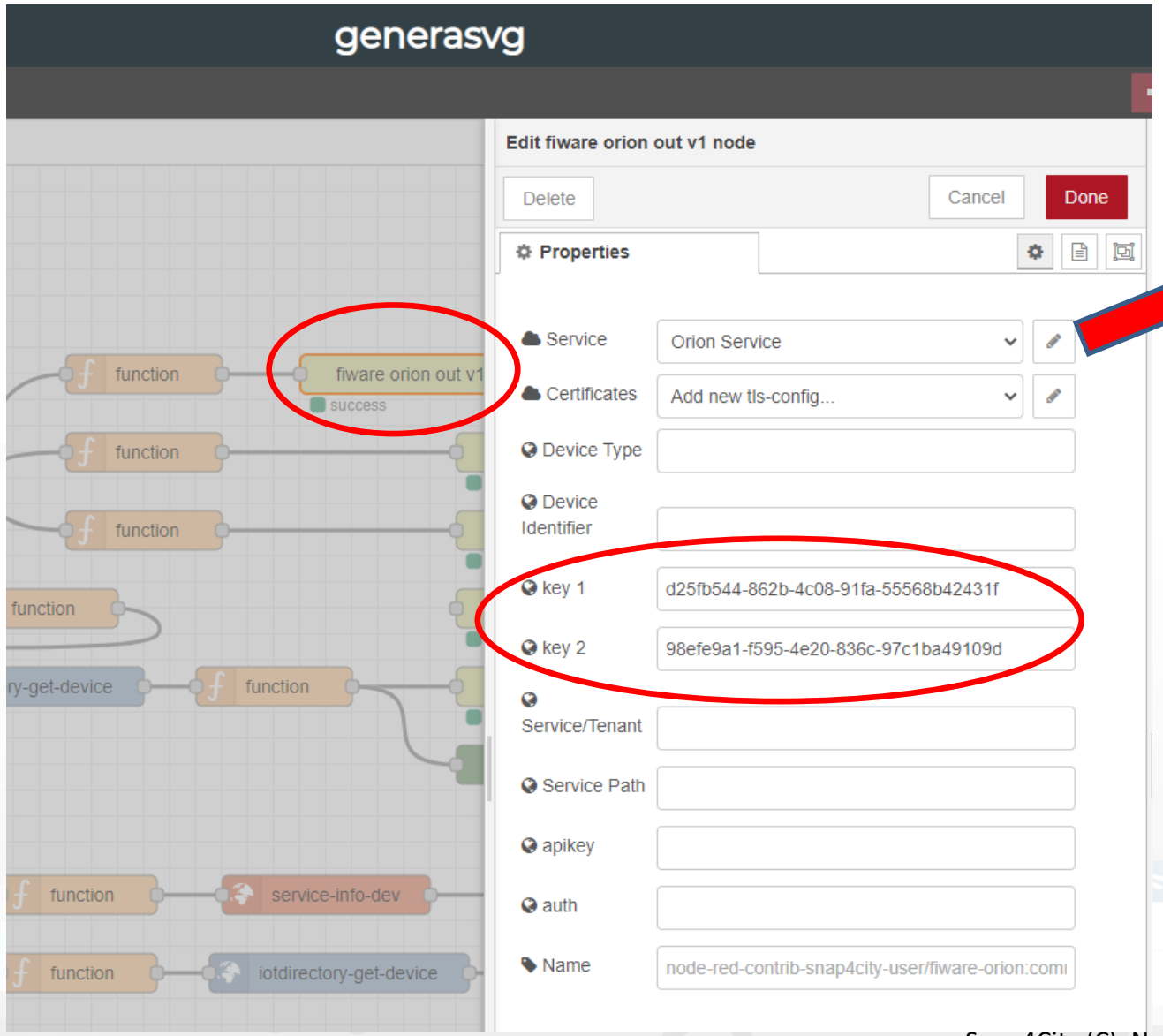
- Generation of random values of the 4 IOT devices
- Send the values to IOT Broker,
  - which in turn send automatically the value to the Storage
- Inspection of the IOT Device Information

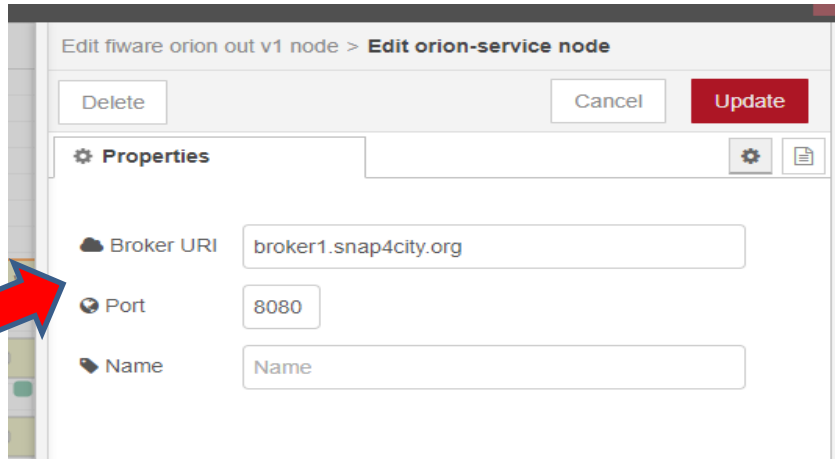
Function Block



- Data values Random generation
- dateObserved with the actual datetime
- Example in NGSI V1

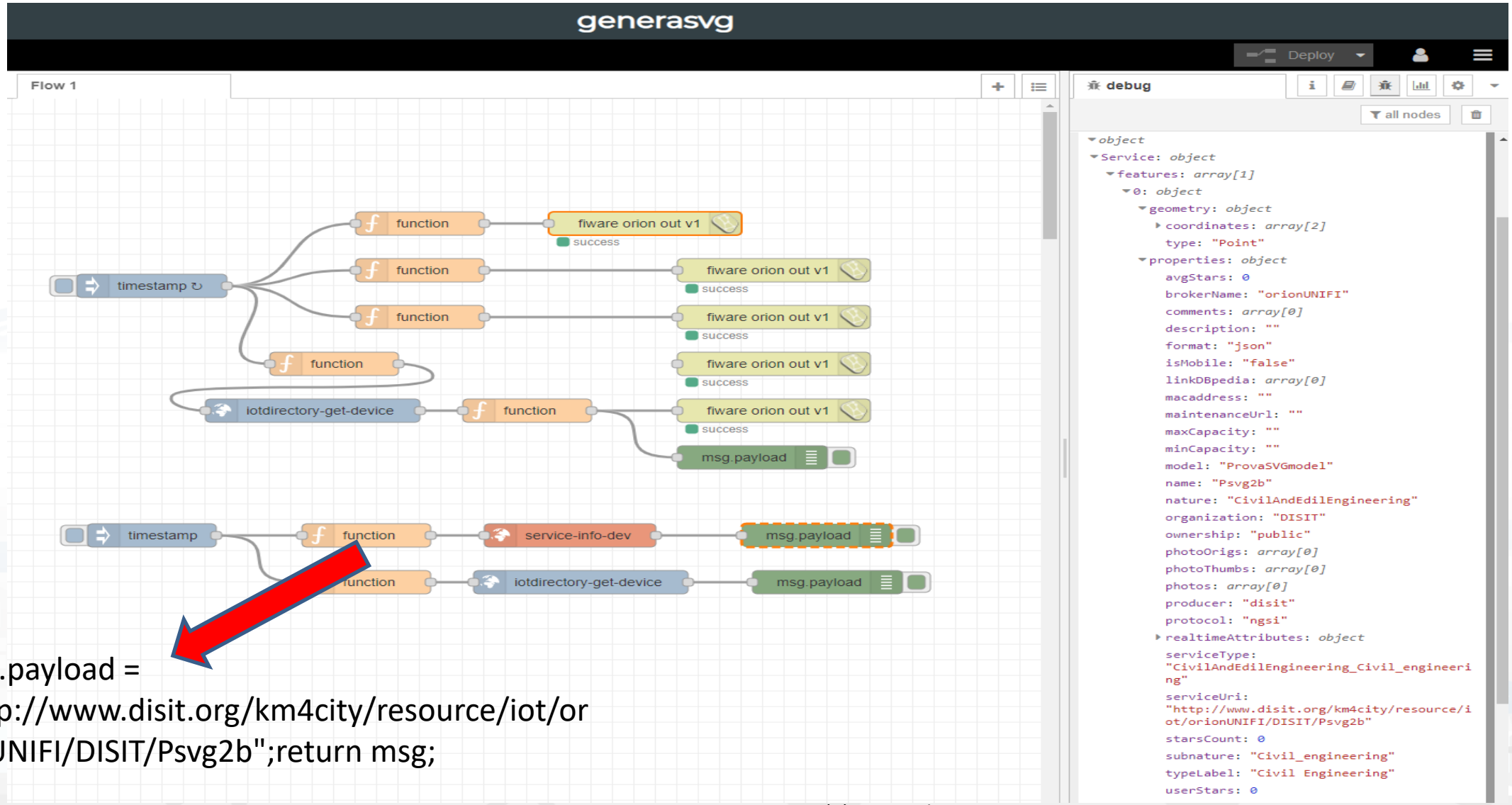
Send data to Broker





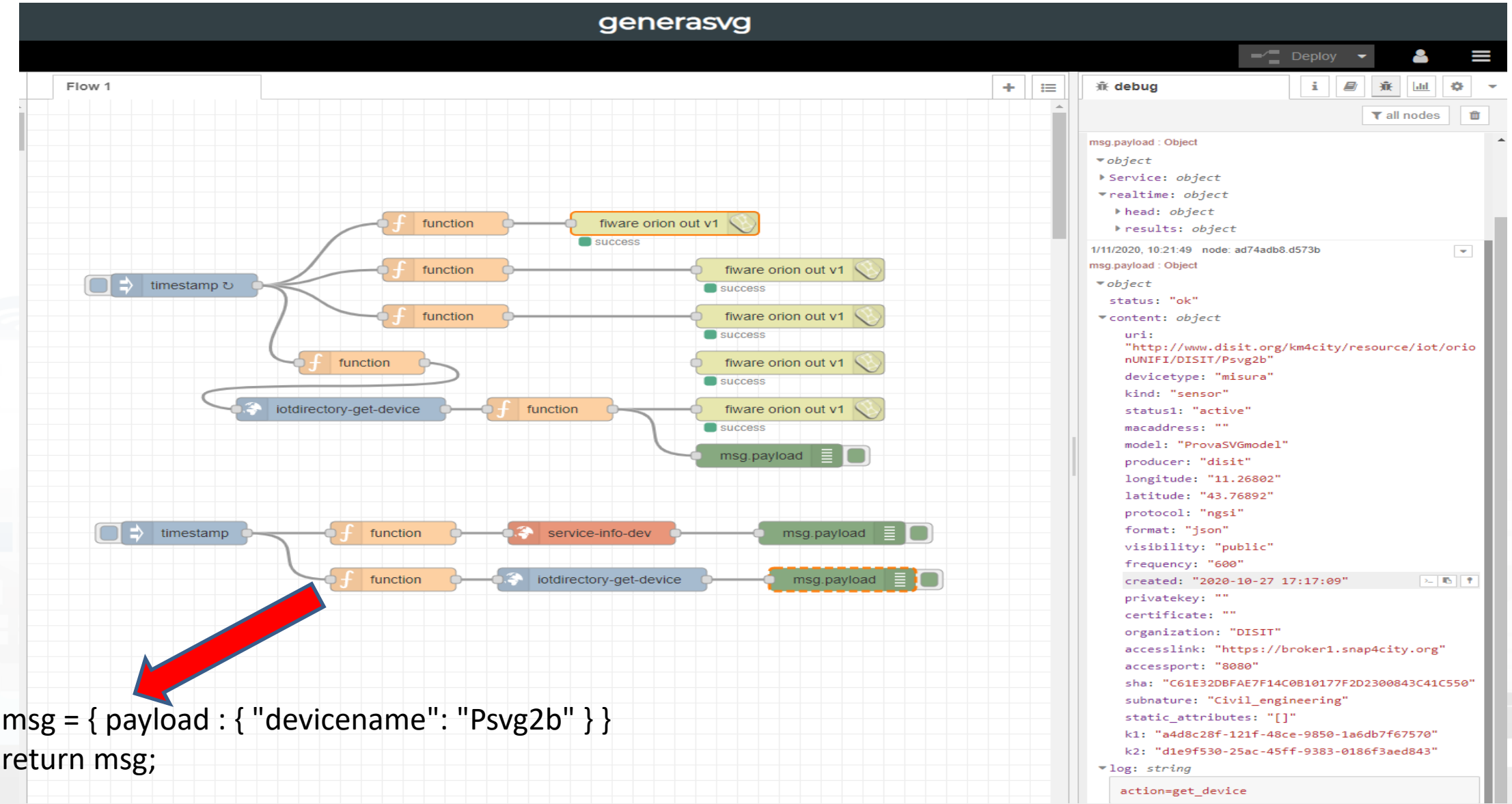
- This case is with K1 and K2 directly written into the Orion OUT V1 Node

Service Info Dev



msg.payload = "http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/Psvg2b";return msg;

IOTdirectory-get-device information

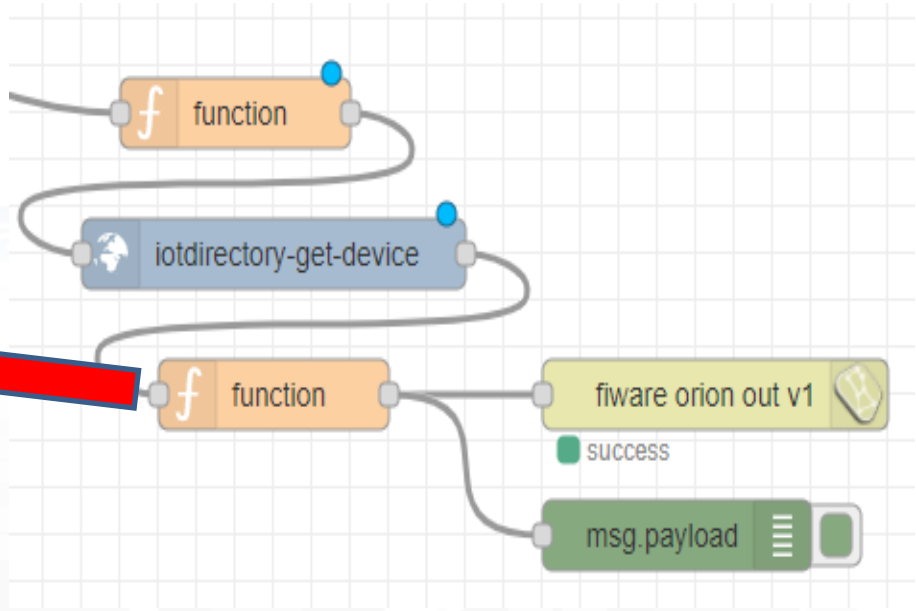



```
msg = { payload : { "devicename": "Psvg2b" } }
return msg;
```

Exploitiong IOT Directory for getting K1,K2

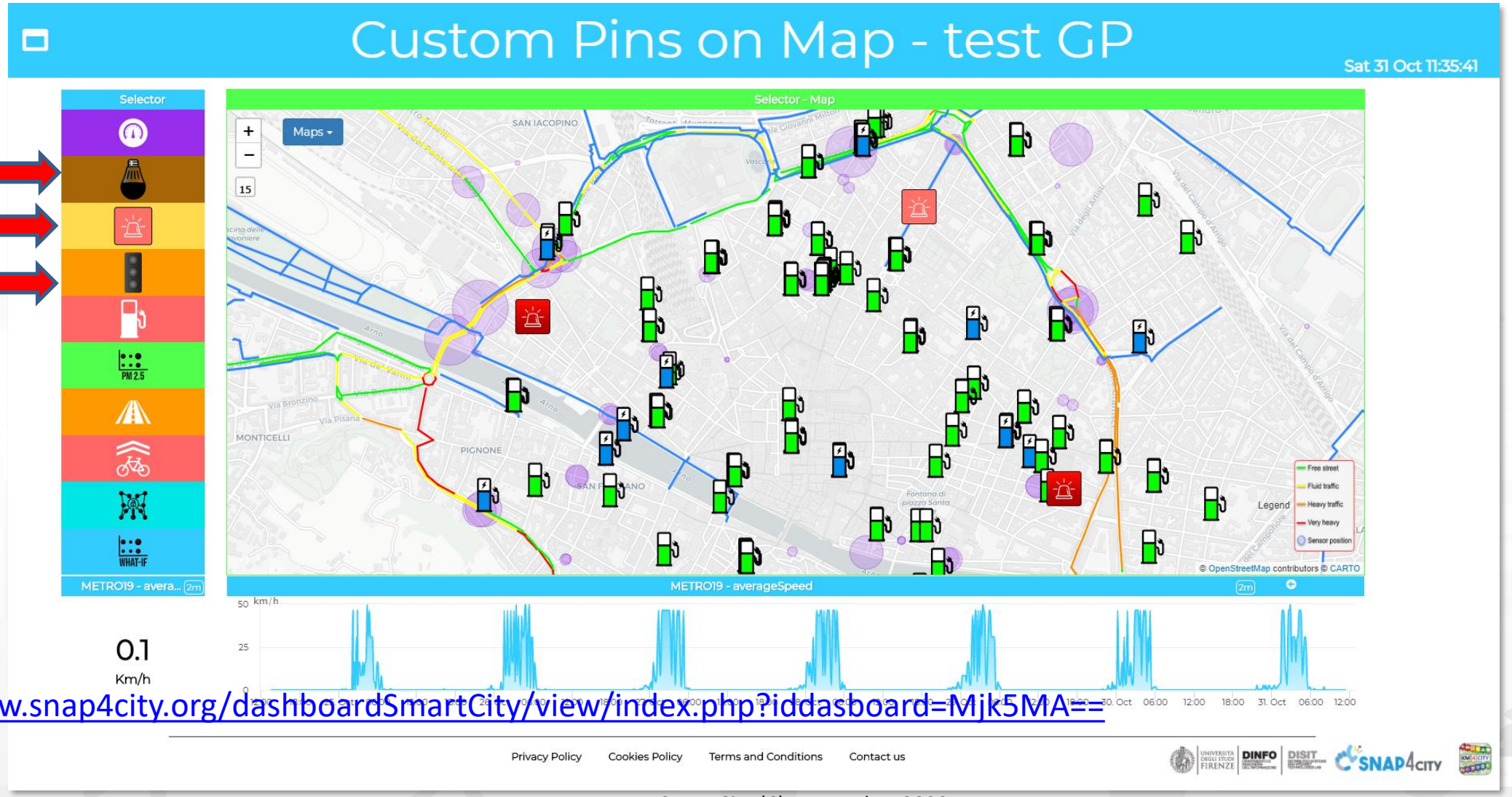
```
msg = {  payload : {
  "id": flow.get("devid"),
  "type": "misura",
  "attributes":[  { "name": "dateObserved",  "value": data,  "type":
"time"  },
          { "name": "val1",   "value": rnd1,   "type": "float"  },
          { "name": "val2",   "value": rnd2,   "type": "float"  },
          { "name": "val3",   "value": rnd3,   "type": "float"  },
          { "name": "val4",   "value": rnd4,   "type": "float"  },
          { "name": "str1",   "value": "eccolo",   "type": "string"  } ]
           },
 auth : {
    "k1": msg.payload.content.k1,
    "k2": msg.payload.content.k2
 }
}
```

// payload in NGSI V1 format



// the values of K1, K2 arrive from
// the iotdirectory-get-device

The four IOT Devices on Dashboard



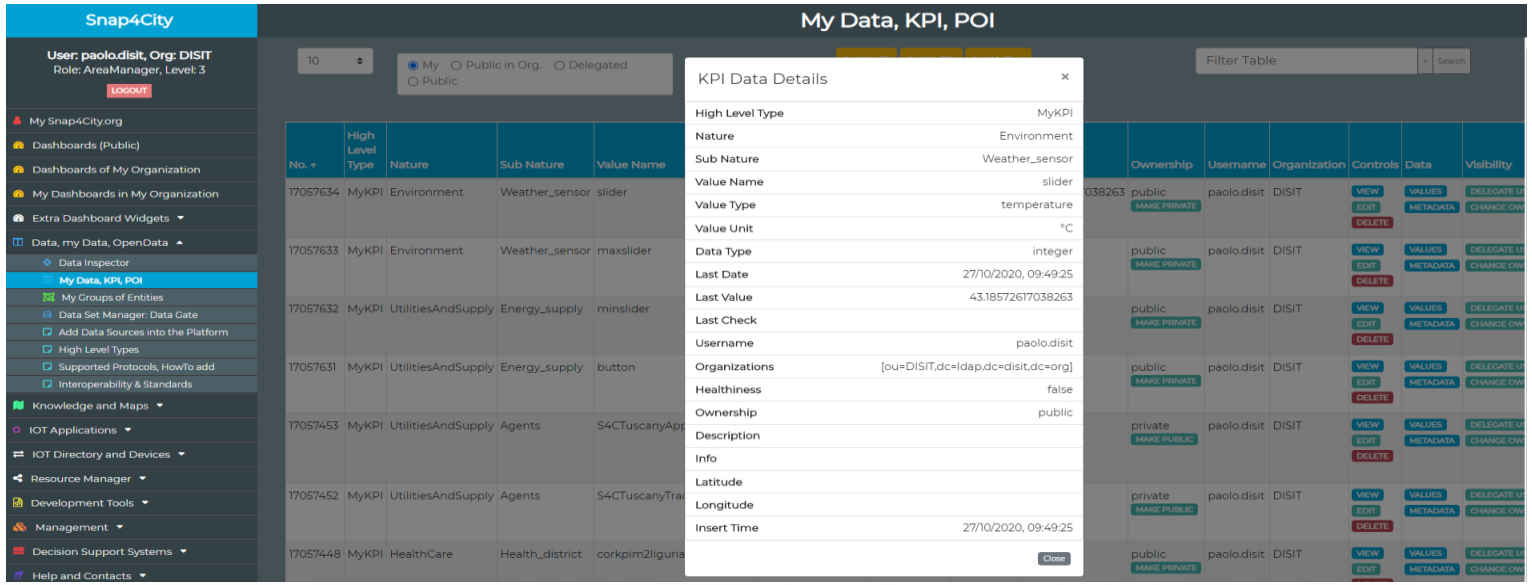
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=Mjk5MA==

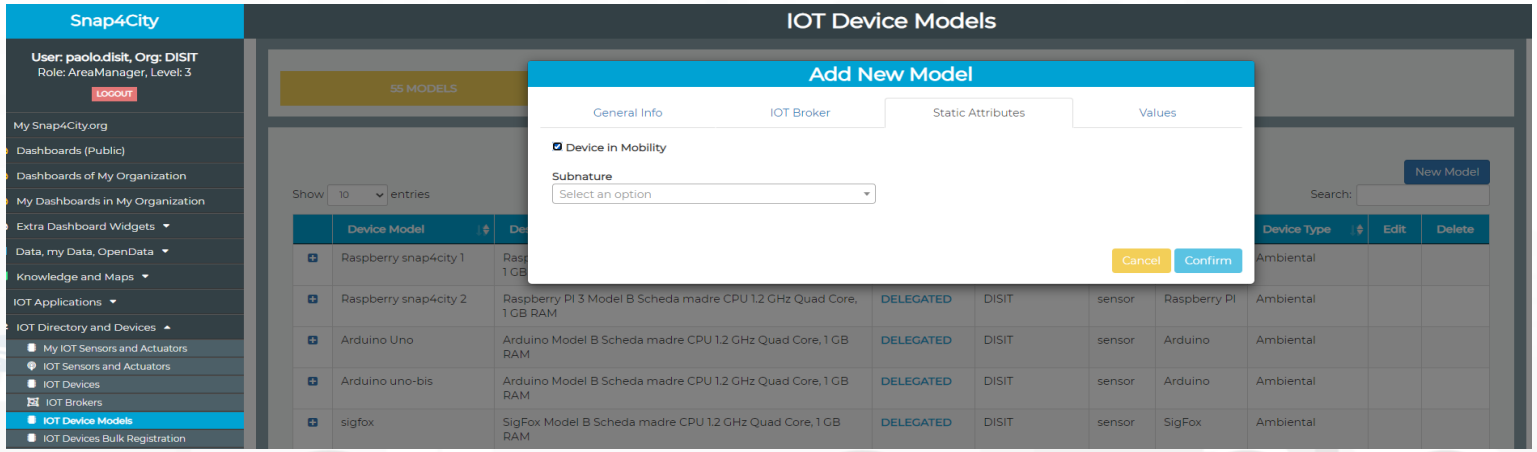
A Complete Example: IOT Device Mobile



Working with Sensor Data from Moving Devices

- Moving Data can be collected by using:
  - **MyKPI**: in which each MyKPI has a ValueName, Unit, Type, etc.. And also GPS location
  - **IOT Device in Mobility**: which generates a new HLT SensorMobile, creating a TimeSeries with changing value and GPS coordinates over time





MyKPI: Tracking of Devices and Mobiles

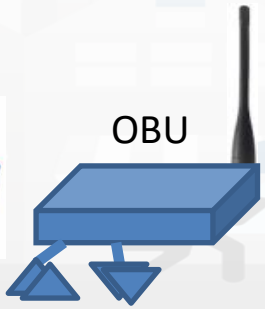
- Real Time Trajectories for
  - Mobile Phone
  - Moving IOT Devices
  - OBU, Vehicular Kits
  - Multiple tracks
  - Day by day
- Micro Application


Mobile
PAX Counter

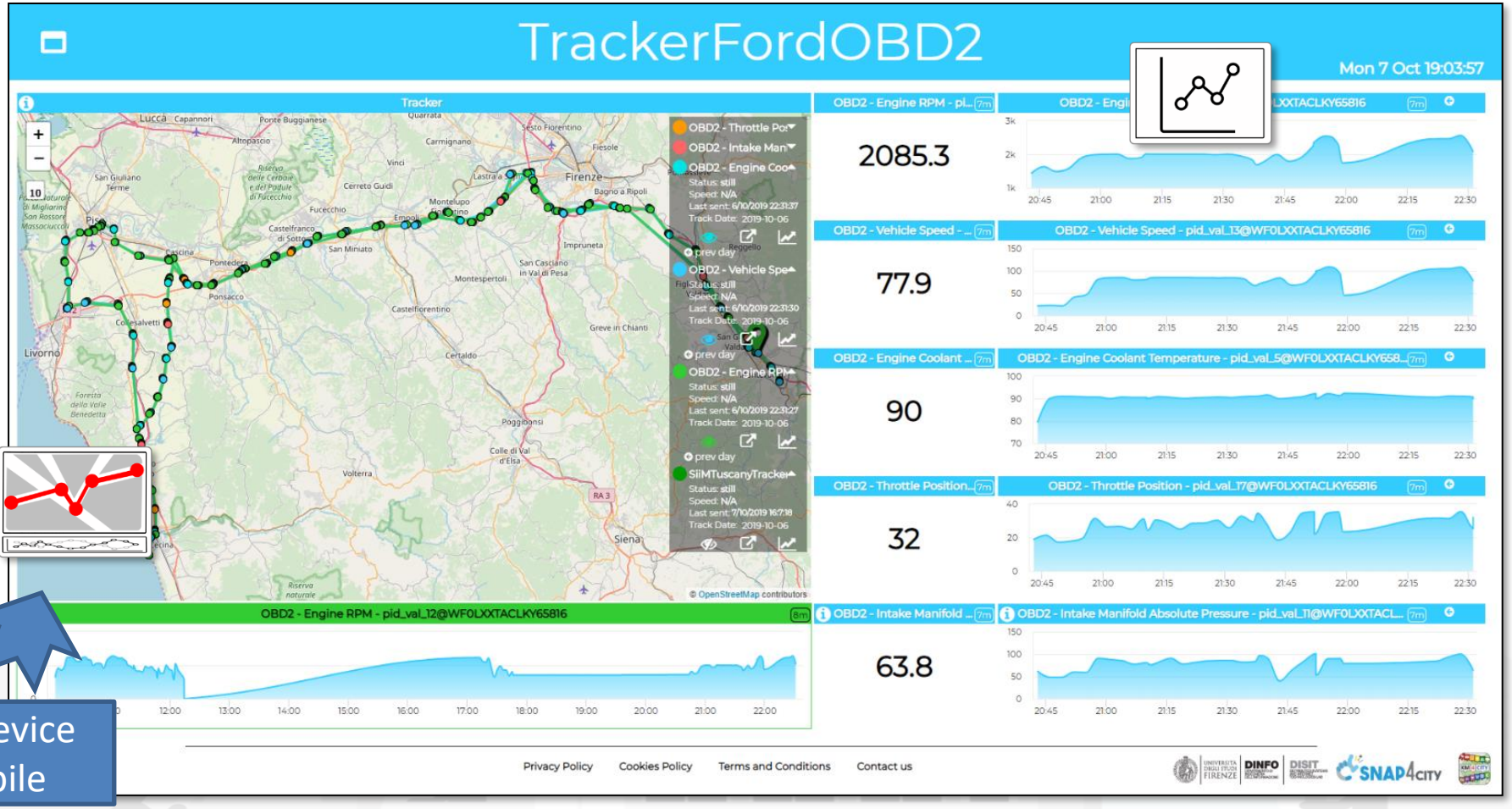
Apps



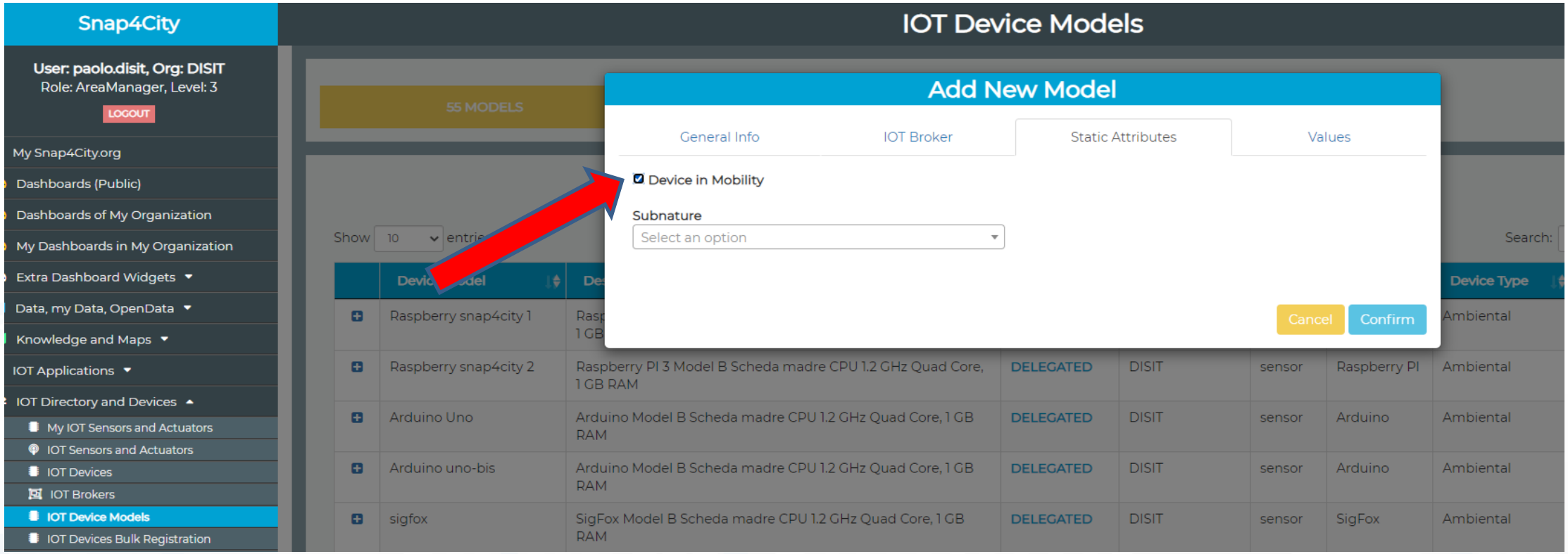

OBD2


OBU

IOT Device
Mobile



IOT Device in Mobility



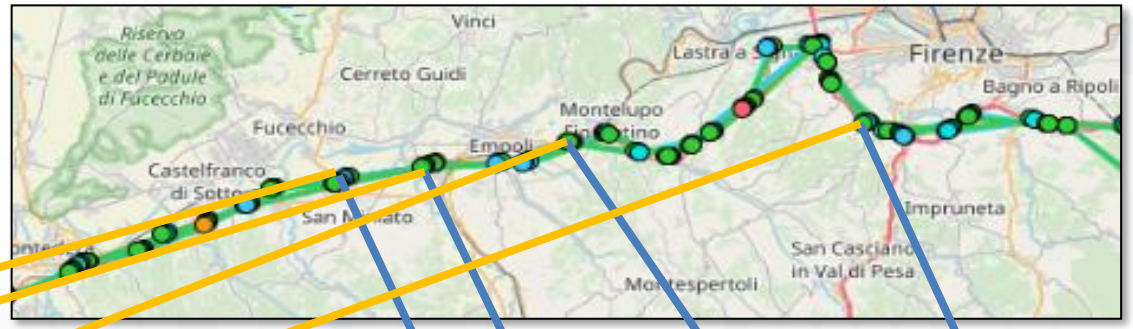
- **IOT Device in Mobility**: which generates a new HLT SensorMobile, creating a TimeSeries with changing value and GPS coordinates over time

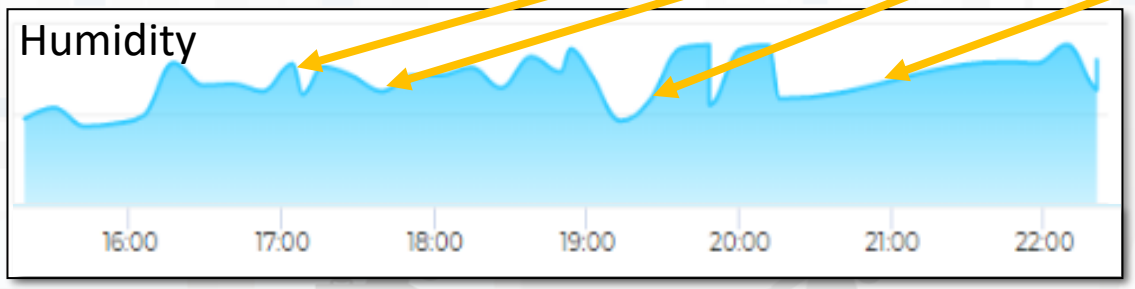
IOT Devices in Mobility as time series

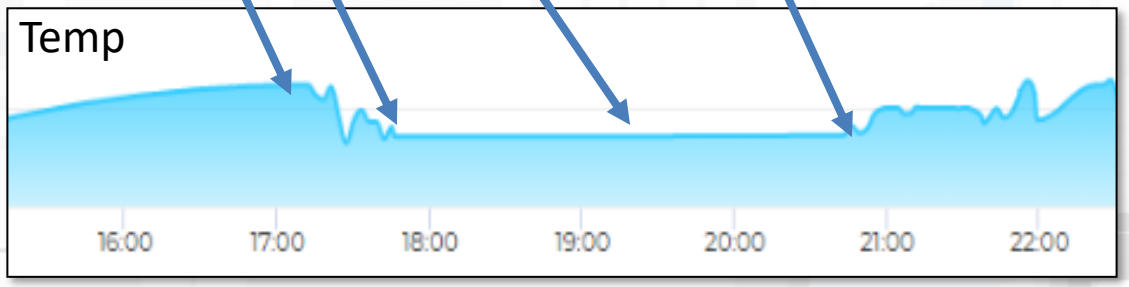
Message (
timestamp: «02-04-2020 at 10:30»,
Temperature: 29.34,
Humidity: 35,
Latitude: 11.456,
Longitude: 43.24314
)

- A set of data coming from an IOT Device with multiple sensor become a time series of values for devices.
- For example taking a new measure every 10 minutes (Red Lines)
- Each new measure in Snap4City is conventionally time located in «dateObserved»
- LAT,LONG is changing over time

dateObserved	Temp	Humid.	LAT	LON
02-04-2020 10:30	34.5	23	11.11	43.21
02-04-2020 10:40	36.5	24	11.13	43.12
02-04-2020 10:50	36.0	22.5	11.45	43.18







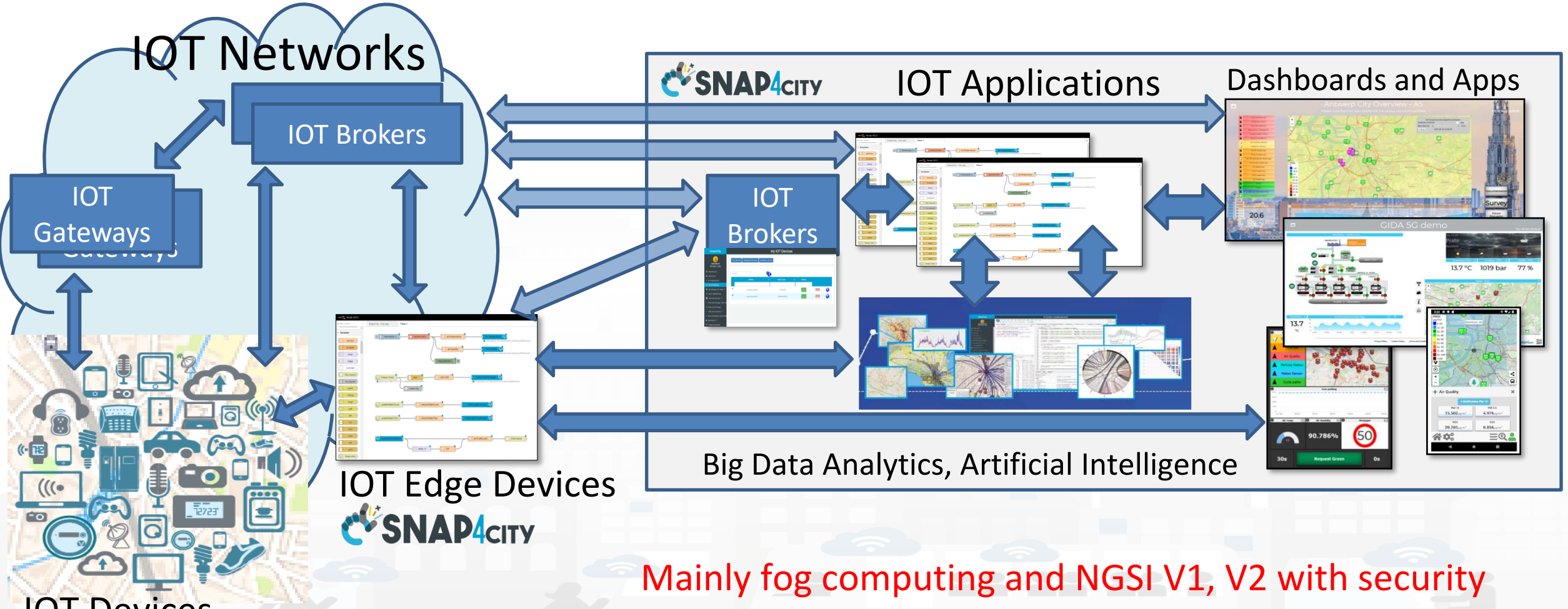
TOP

IOT Devices, hardware-software integration





Snap4City Services also on IOT Edge!!!



Mainly fog computing and NGSI V1, V2 with security

Proprietary IOT Devices as well as Open Hardware / Open Software



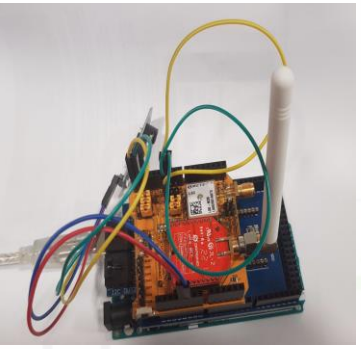
UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB | FIWARE | LoRa | Wi Fi | SNAP4INDUSTRY | KM 4 CITY | Bluetooth

IOT Devices

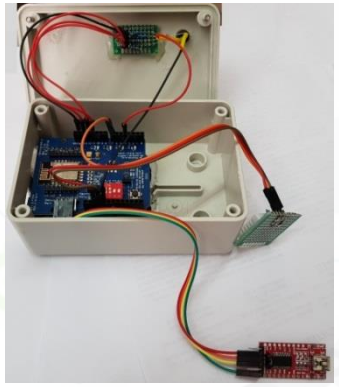
Sensors/ Actuators

IOT Edge Devices

LoraWAN + Arduino + I2C, NGSI



Arduino, Wi-Fi, NGSI



Snap4All IOT Button ESP, NGSI, Wi-FI, BT



Snap4All PAX Counter LoraWAN WIFI, NGSI, GPS



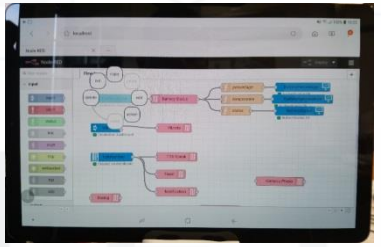
Any Sensor / Actuator
Open to other protocols

IOT Edge NodeRED: Raspberry Pi, NGSI, WiFi, RJ45,..



IOT Edge NodeRED: Android, LINUX, Windows, …





LoraWan Gateway: IOT Edge, NGSI, WIFI, RJ45, GPS

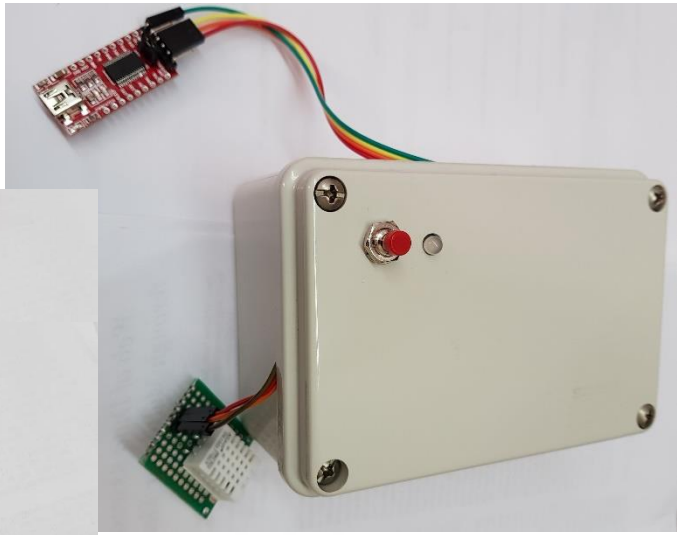


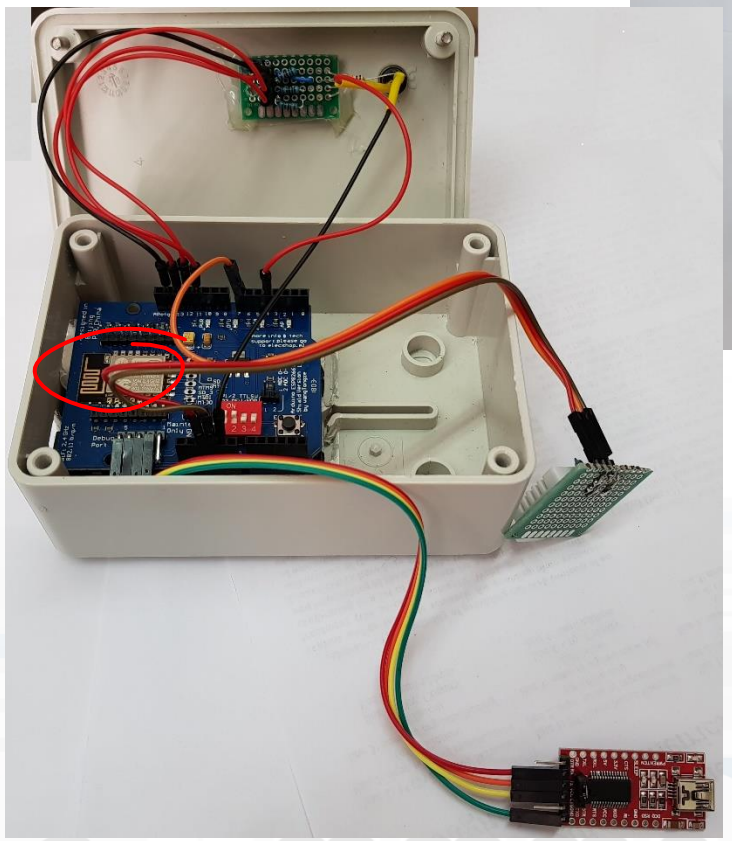
IOT Dev Management: activities

- **IOT Devices connected to Snap4 can be open or proprietary**
- **IOT Devices**: a large range of protocols, formats and kind
  - IOT Devices (single or in bulk) are **registered** on IOT Directory and thus according to Knowledge base are registered to be used in IOT Applications, Dashboards, etc. with Shadow values, etc.
  - IOT Models are saved on IOT Directory for shortening the registration process
  - IOT Device healthiness is automatically monitored
- **IOT Devices can be public or private**
  - **Full support of Proprietary protocols and devices**
  - **Providing Open Hardware and Open Software** IOT Devices/IOT Edge: NGSI fully secure
- **IOT Edge** are devices with some computing capability, realized by using: Raspberry, Android, Linux, Windows, etc.
  - **Release as**: OS images on SD, APK for Android, Virtual Machine, Docker Container, etc.
- IOT Devices are connected via **Secure Encrypted Mutual Authenticated** channel of communication, TLS and certificates

IOT Device with Arduino

- Arduino Uno
- Wi-Fi shield, standard
- Mutual Authentication with certificates, or K1,K2,sha
- Secure encrypted connection, NGSI
- Open Source
- Fully Customizable
  - Any sensor
  - NGSI or any other protocol





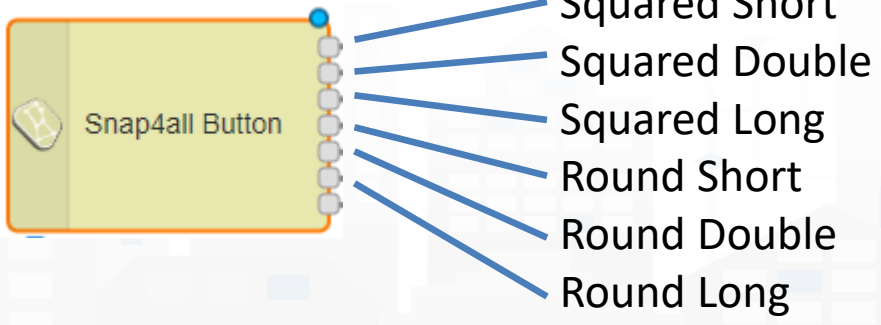
Snap4All IOT Button

- Multi Wi-Fi
- Ready to use BLE
- ESP 32 based, cheap & easy
  - low/no energy consumption/ standby
- Mutual Authentication with certificates, or K1,K2,sha
- secure encrypted connection, NGSI
- Open Source, Fully Customizable
- HW extensible to sensors

version: 3





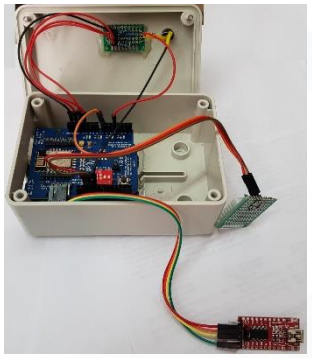


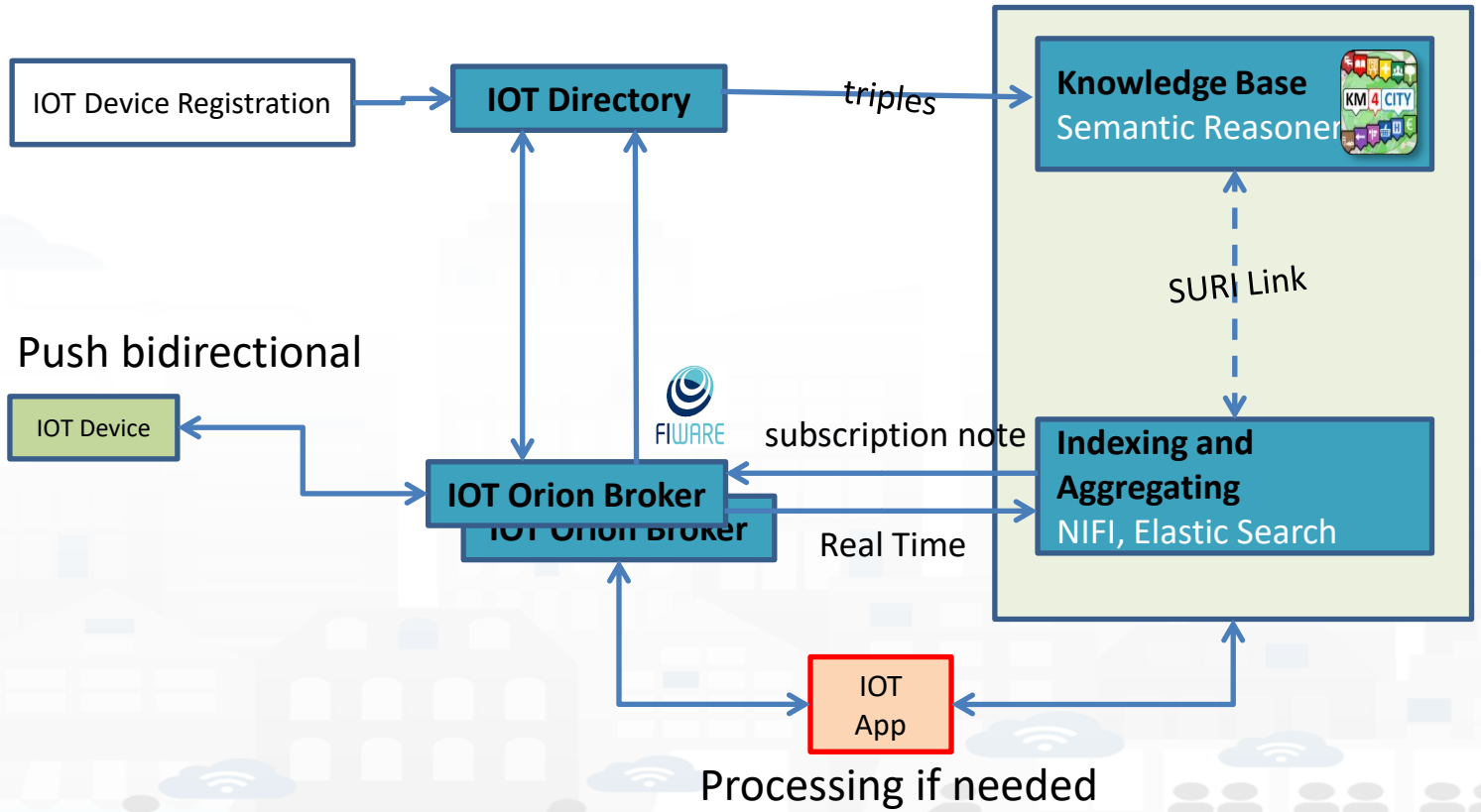
https://www.snap4city.org/276
https://www.snap4city.org/297 help config

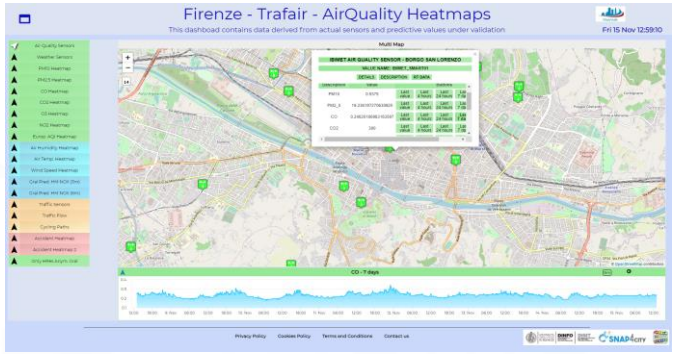
AirQINO vs SNAP4CITY

- Can be connected (i) directly to Snap4City (data driven) or (ii) Indirectly via CNR IBE (only in push)









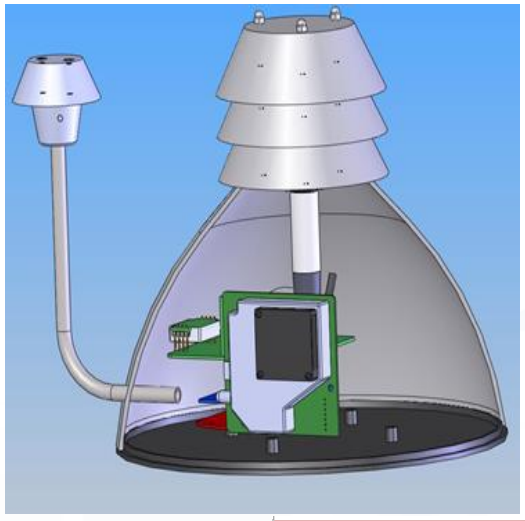
CNR IBE AirQino



AirQino

- CNR developed a circuit board "AirQino", Arduino Shield compatible, integrated with low cost and high resolution sensors, dedicated to the monitoring of environmental parameters and air quality pollutants
  - Noise, Humidity, Temperature,
  - CO, CO_2, O_3, NO_2, CH_4,
  - road pavement quality (accelerometer) and the indices of well-being (globethermometer to calculate the index of thermal comfort) in an urban environment.
- The board integrates a microprocessor unit that acquires all the sensors installed and analyses fast data from accelerometer and noise sensor.

Consiglio Nazionale delle Ricerche





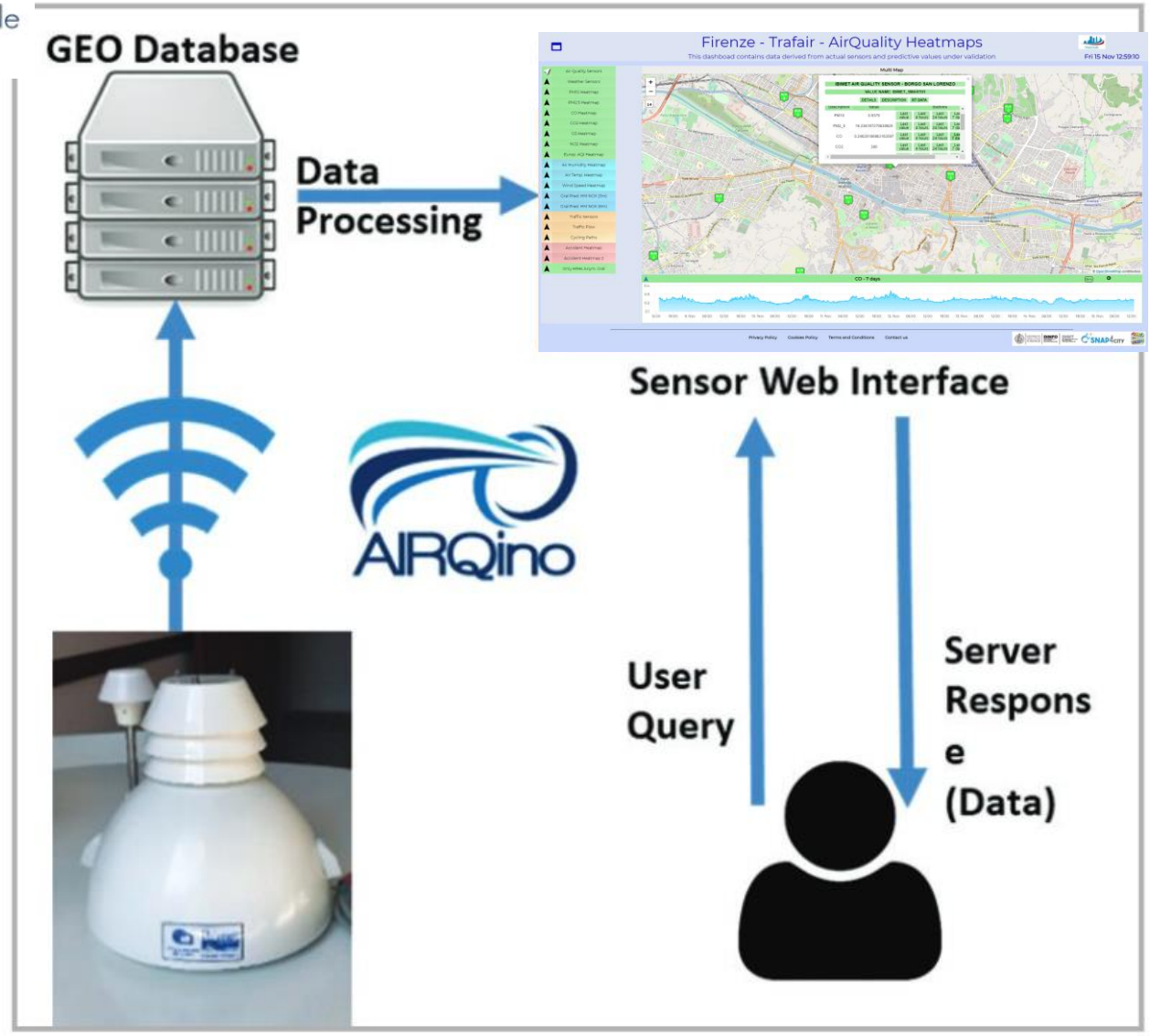
Parameter	Unit	Range
Temperature	°C	-40 – 80
Relative Umidity	%	0 – 100
CO2	ppm	0 – 2000
O3	ppb	0 – 400
NO2	ppm	0.05 – 5
CO	ppm	1 – 30
PM	µg/m3	0 – 999
VOC	ppm	1 – 100

https://www.snap4city.org/658 how to set up on Snap4City

https://www.snap4city.org/508

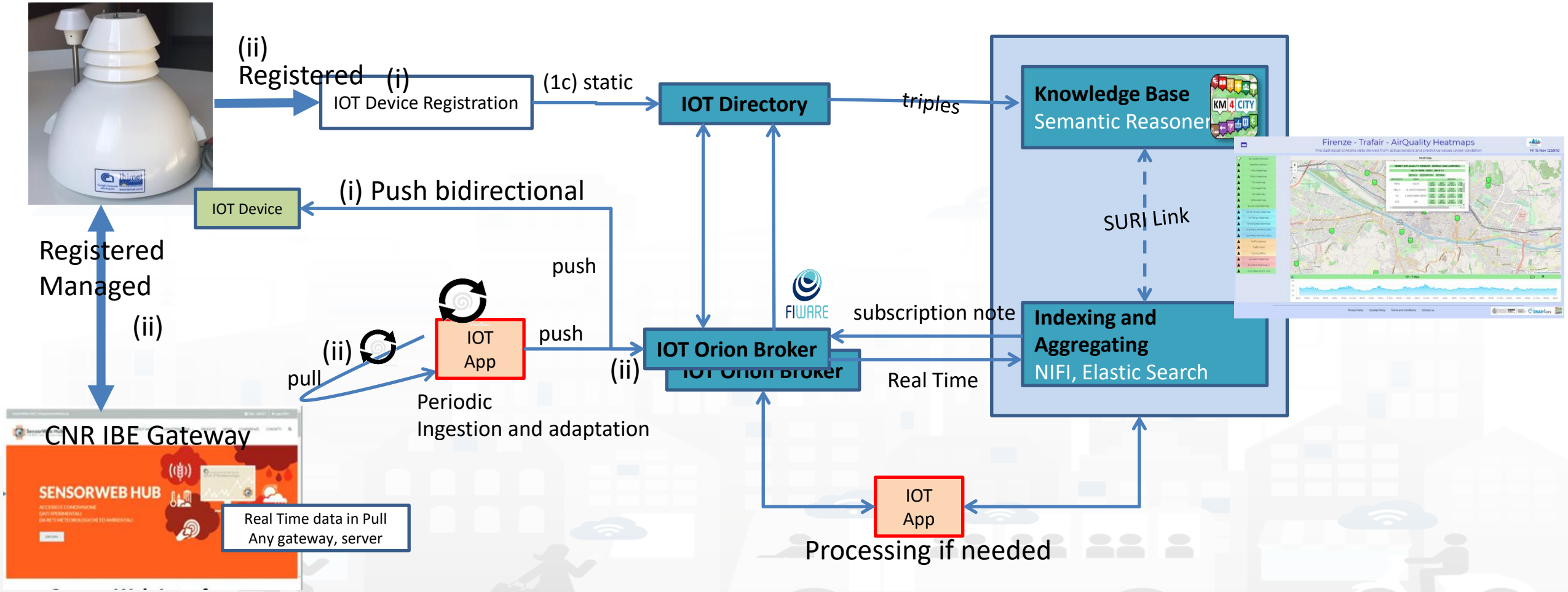
https://www.snap4city.org/download/video/tn/ARQuino-CNR.pdf


T2 Peretola Aeroporto
1012
GEST


GEO Database
Data Processing
Firenze - Trafair - AirQuality Heatmaps
Sensor Web Interface
AIRQino
User Query
Server Response (Data)

AirQINO vs SNAP4CITY

- Can be connected (i) directly to Snap4City (data driven) or (ii) Indirectly via CNR IBE (only in push)



Libelium Waspmote Plug&Sense Smart Environment PRO



Libelium





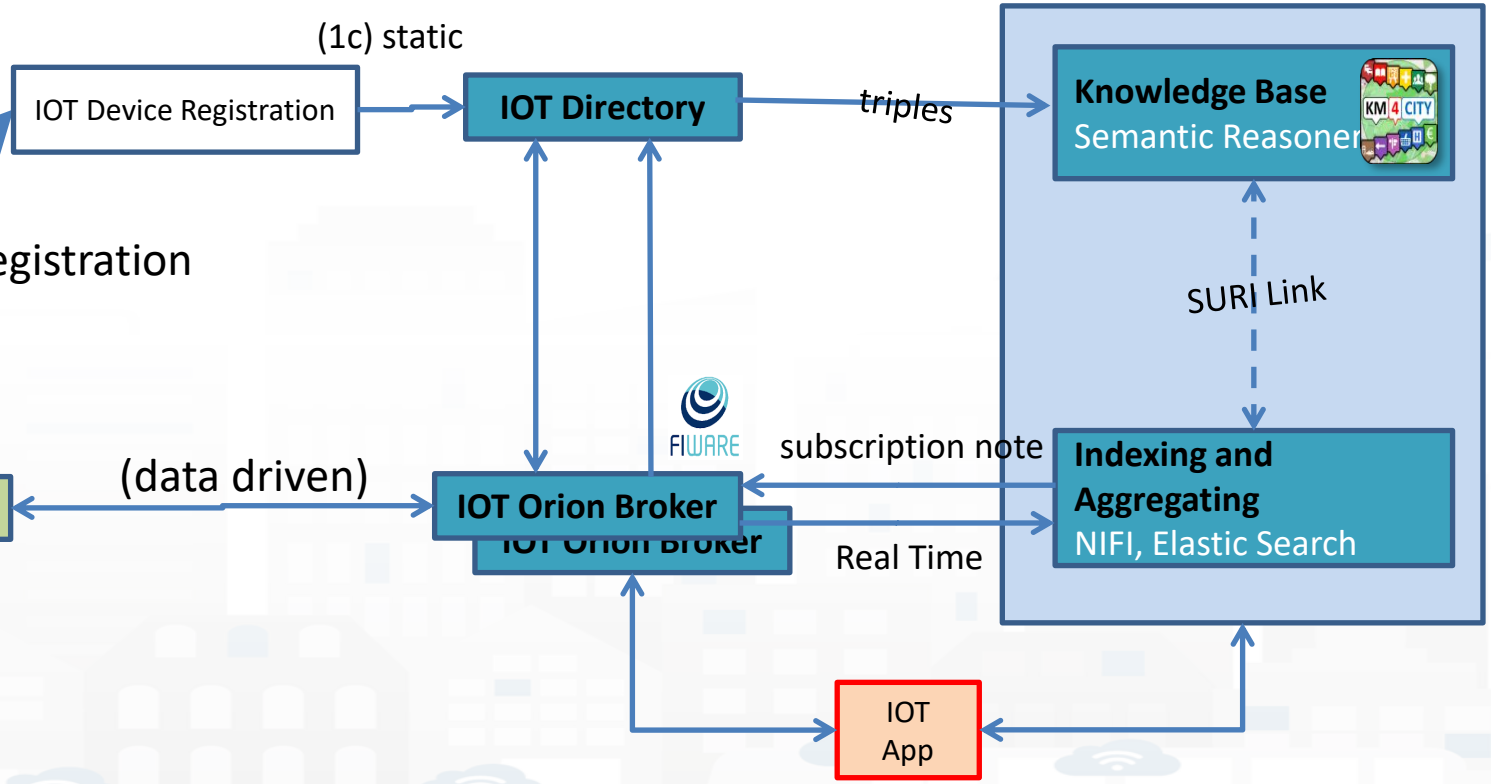
- PM10
- Temp
- Humidity
- Pm2.5
- NO
- NO2
- CO2
- Etc.

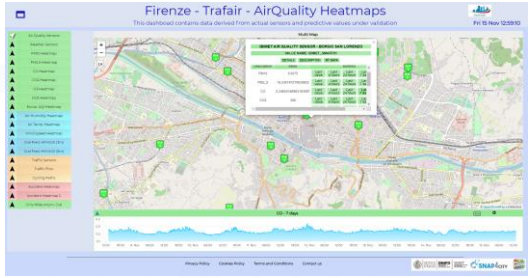
https://www.snap4city.org/659 how to set up on Snap4City

Libelium vs SNAP4CITY

- Can be directly connected to Snap4City (data driven)



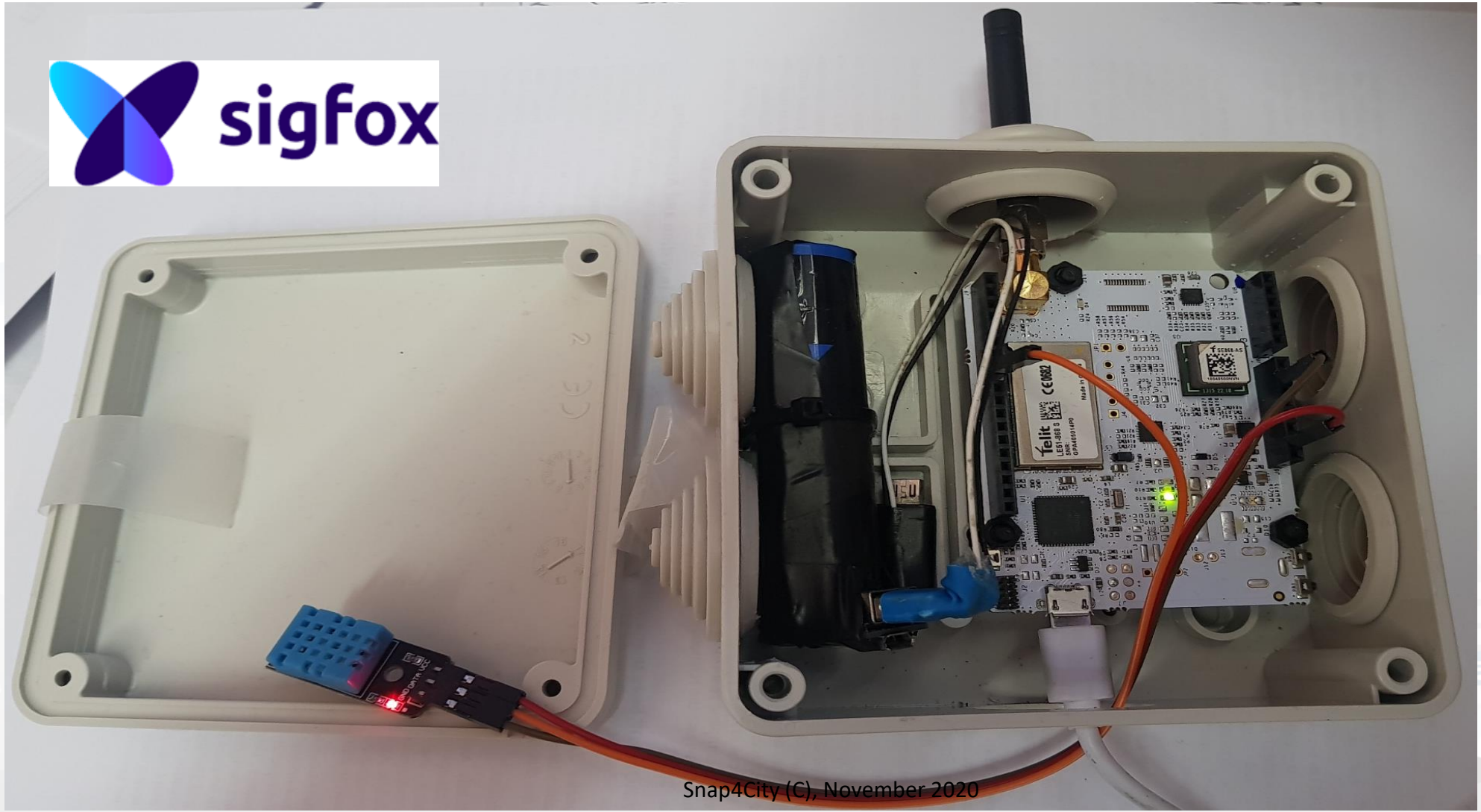




SigFox Integration



SigFOX: example of a develoment platform



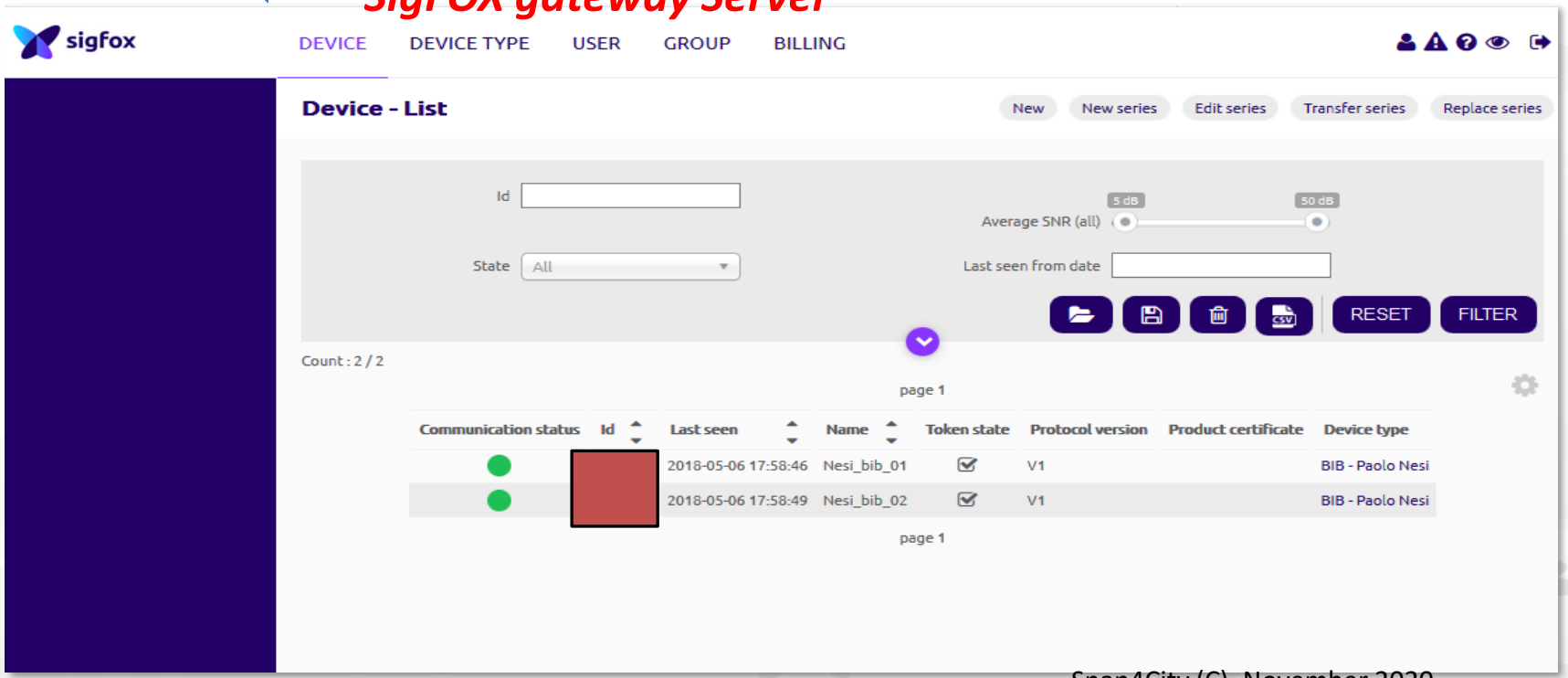
SigFOX Server Side



Registered
Managed

- Proprietary Protocol, freq similar to Lora
- Final users, consumers may buy SigFox devices and subscribe to network to register their devices
- 1 msg per every 10 minute, max num msg per day, per year…

SigFOX gateway Server



page 1

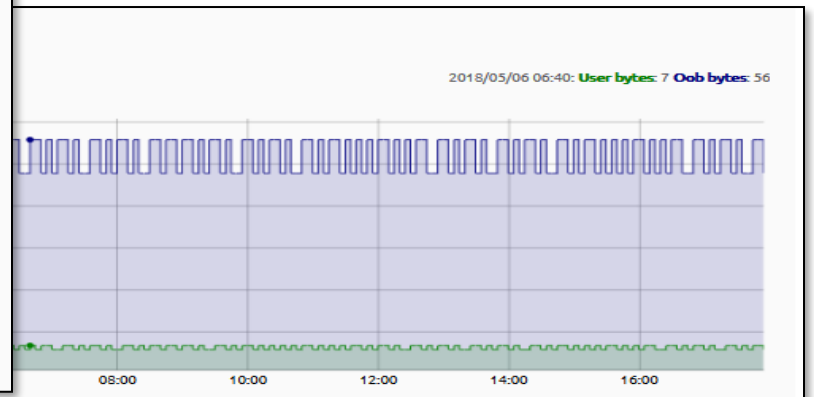
Frame reception time

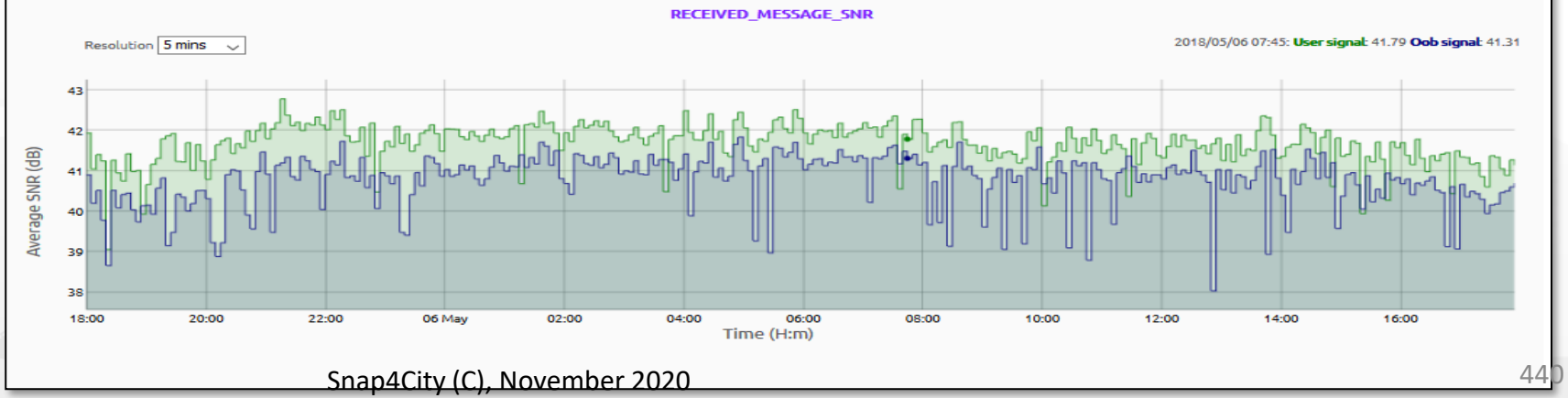
Time	Delay (s)	Header	Data / Decoding	Location	Base station reception attributes: Base station	RSSI (dBm)	SNR (dB)	Freq (MHz)	Frames	Callbacks
2018-05-06 18:03:27	< 1	0000 ack required	24		28A8	-122.00	29.40	868.1491	3/3	
					2896	-136.00	15.60	868.1420	3/3	
					25F2	-119.00	32.03	868.1373	3/3	
2018-05-06 18:03:25	< 1	0010	09dd0b4f0b040103 Temp: 26.0 °C VDD idle: 3.037 V VDD tx: 2.895 V RSSI: -97.0		25F2	-120.00	31.57	868.1187	1/3	
					28A8	-122.00	29.05	868.1185	1/3	
					2627	-141.00	10.48	868.1173	1/3	
2018-05-06 18:02:51	< 1	0000 ack required	24		28A8	-122.00	29.39	868.1357	3/3	
					2896	-136.00	14.81	868.1347	3/3	
					2884	-134.00	17.36	868.1229	3/3	
2018-05-06 18:02:23	< 1	0010	09d30b4a0b0e0102 Temp: 27.0 °C VDD idle: 3.027 V VDD tx: 2.890 V RSSI: -98.0		23DB	-110.00	41.00	868.1449	1/3	
					2896	-137.00	14.40	868.1442	1/3	
					2889	-137.00	13.67	868.1447	1/3	
2018-05-06 18:01:48	< 1	0000 ack required	24		23DB	-109.00	41.66	868.1553	1/3	
					2889	-136.00	15.06	868.1550	3/3	
					28C8	-139.00	11.81	868.1546	1/3	

SNAP4CITY

KM4CITY

SigFOX





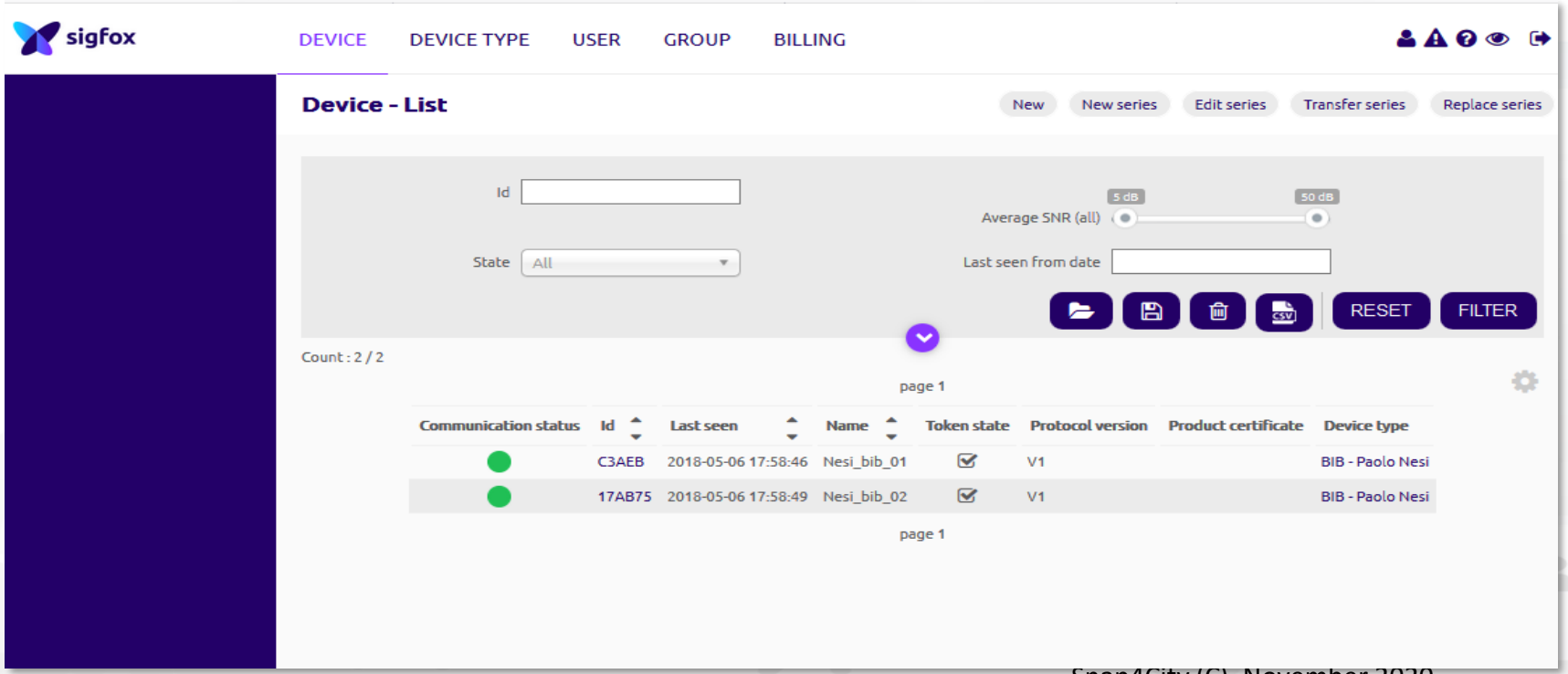
sigfox

SigFOX vs SNAP4CITY



Registered
Managed

- Possible connection in PUSH and PULL
- Ingestion via IOT Application on Cloud or on IOT App on Edge
- Suggested connection in PULL





SigFOX vs SNAP4CITY

- Can be connected Indirectly via SigFox gateway (in push or pull), here represented in PULL



(i) Registered (ii)

IOT Device Registration

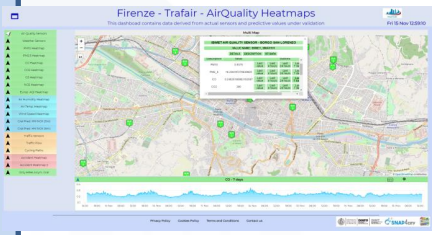
(1c) static

IOT Directory

triples

Knowledge Base
Semantic Reasoner

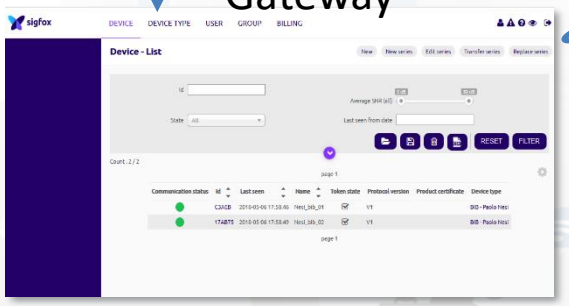
SURI Link



Registered
Managed

(ii)

Gateway



pull

(ii)

IOT App

periodic Adapter

push

IOT Orion Broker

FIWARE

subscription note

Indexing and Aggregating
NIFI, Elastic Search

Real Time

IOT App

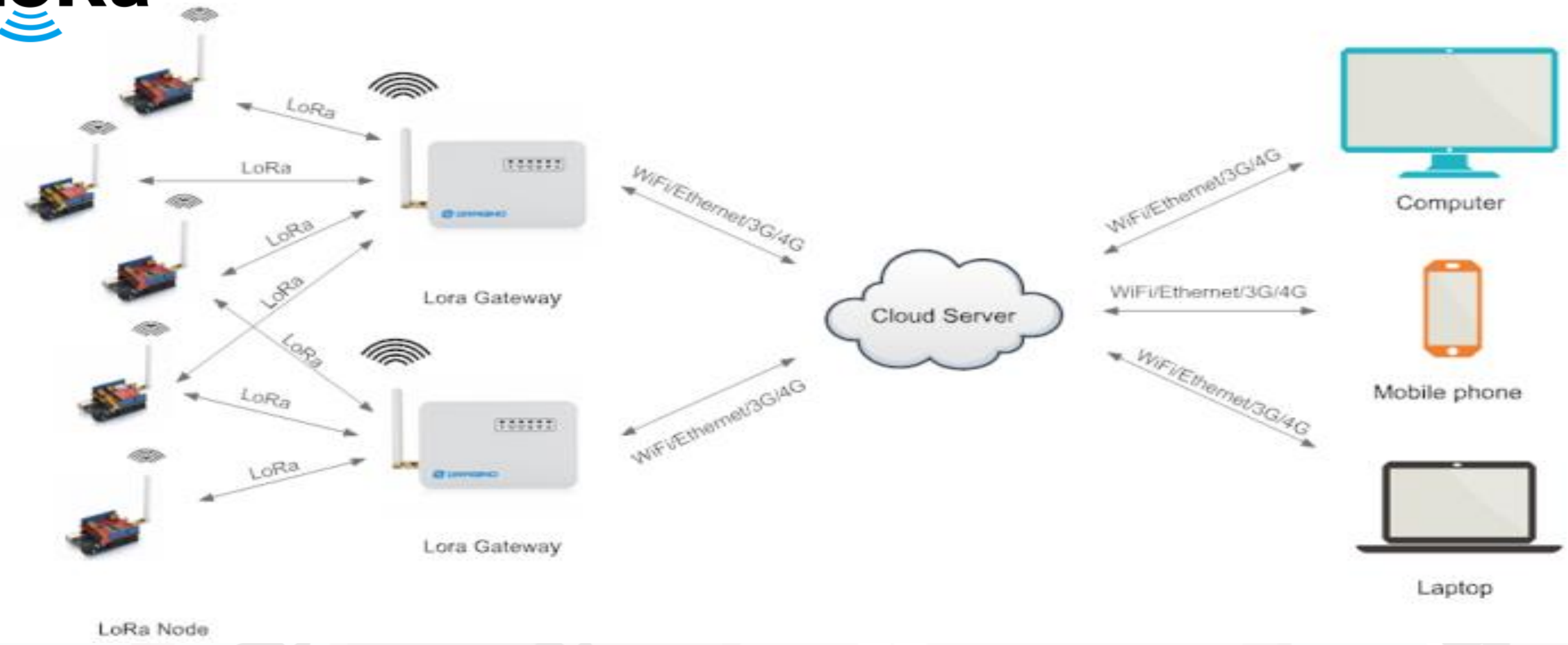
Processing if needed

Real Time data in Pull
Any gateway, server

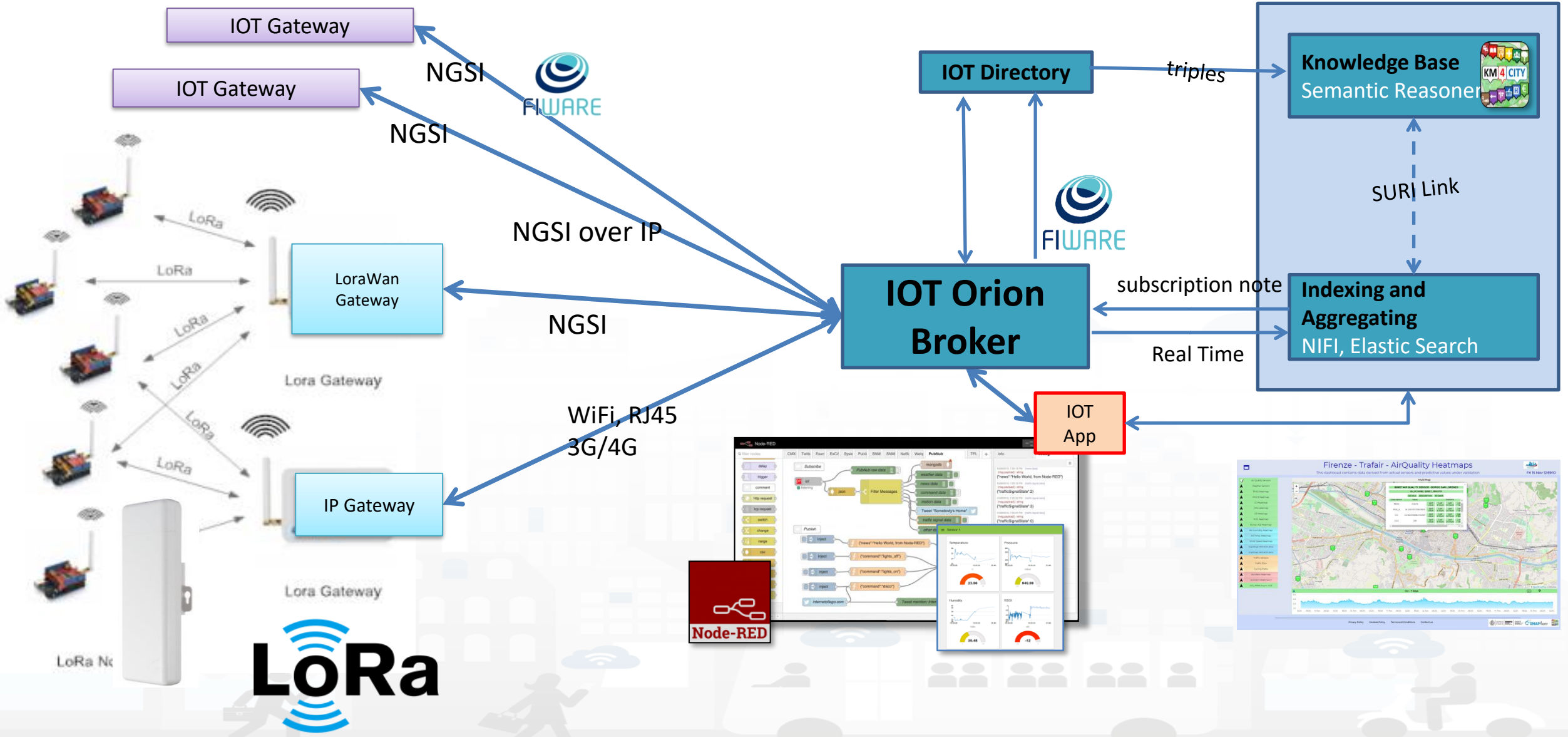
Lora IOT Gateway vs NGSI



LoraWAN Dragino (Arduino)

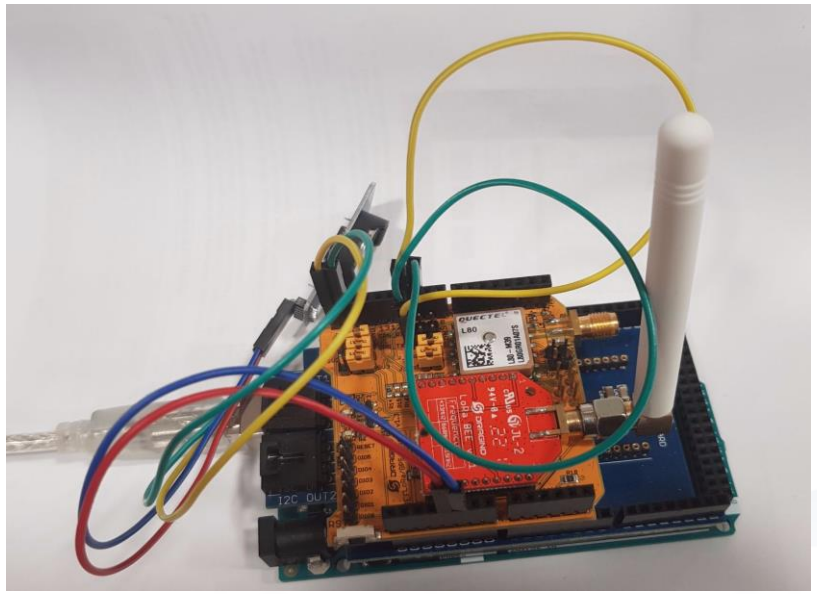


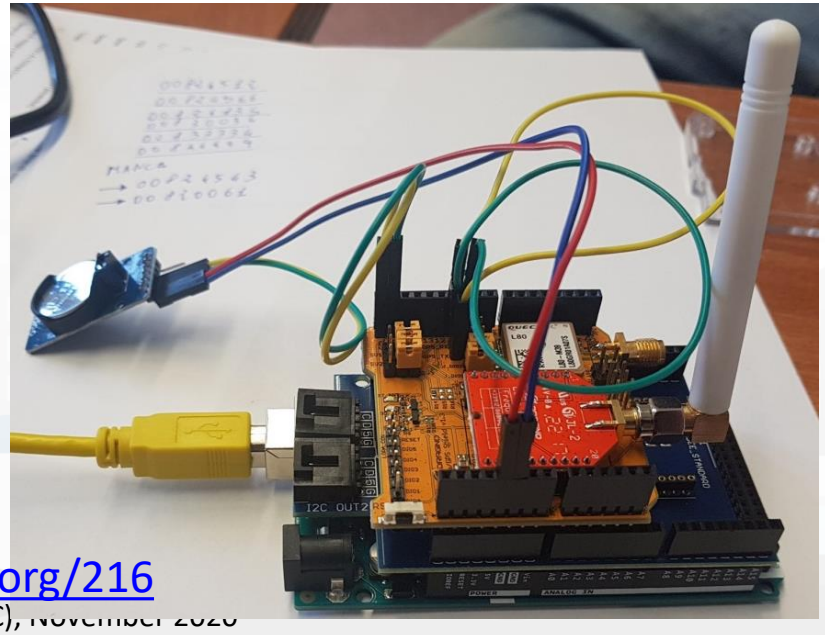
Lora IOT Management



Lora IOT Device, Arduino

- Arduino Uno, Mega
- LoraWan Connection
- Any sensor, + I2C
- Fully Customizable
- Open Source
- NGSI or any other protocols
- Gateway: Lora-NGSI Snap4





https://www.snap4city.org/216

LoraWan Gateway/Edge out of the Box

- Raspberry Pi Based LoraWan Gateway
  - Powered 5V, Wi-Fi, RJ45, …
  - GeoLocated GPS Antenna
- **IOT Edge Snap4City**
  - **Including** Node-RED, IOT APP
- Logical UpLink: LoraWAN TheThingsNetwork, NGSI V1, V2 (mutual authenticated Snap4City) toward IOT Broker



Physical UpLink as: Wi-Fi, RJ45

IOT Gateway / IOT Edge



Measuring any kind of sensors values

Controlling Energy Power

Measuring
Energy Consumption

Any kind of notification channel

IOT Edge:
Node-RED
+
Snap4City

Local Control

Modbus

SCADA

OPC UA

DCS

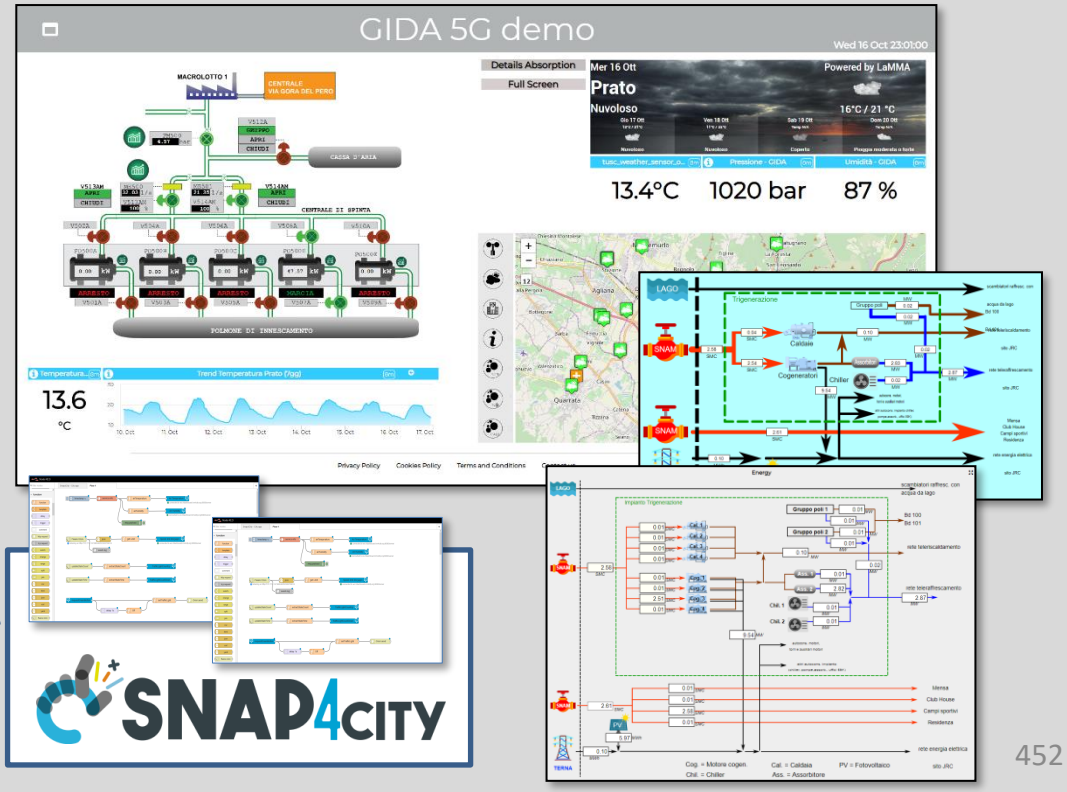
PLC

ODBC

Administrative Servers

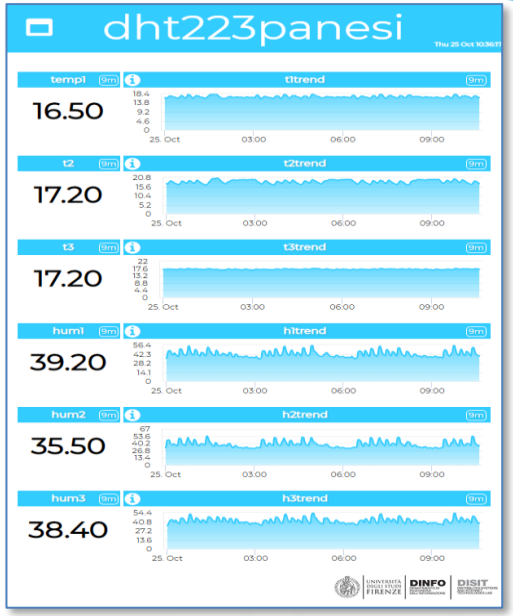
Alexa: Voice Commands

Contextual (smart city/home) data, Data Analytics
Historical Data, Remote Control, Mobile App

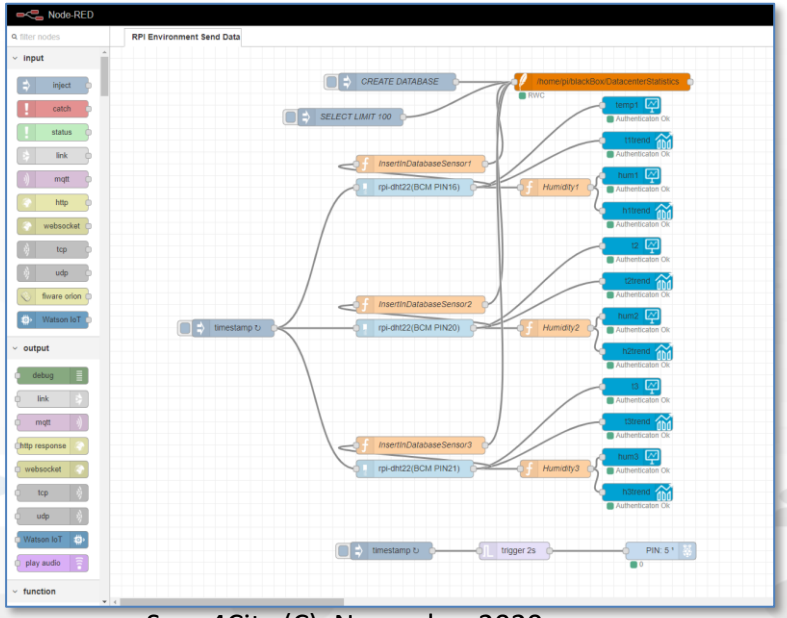


IOT Edge on Raspberry Pi

- Raspberry Pi
- Mutual Authentication with certificates
- Secure encrypted connection
- IOT Application inside
- Any sensor
- Any protocol from IOT devices
- NGSI or any other protocol
- Fully Customizable
- Local and Cloud Dashboard
- **Special MicroServices**









MicroServices:
- DHT
- ModBus
- any shield
- etc....

Raspberry for Edge

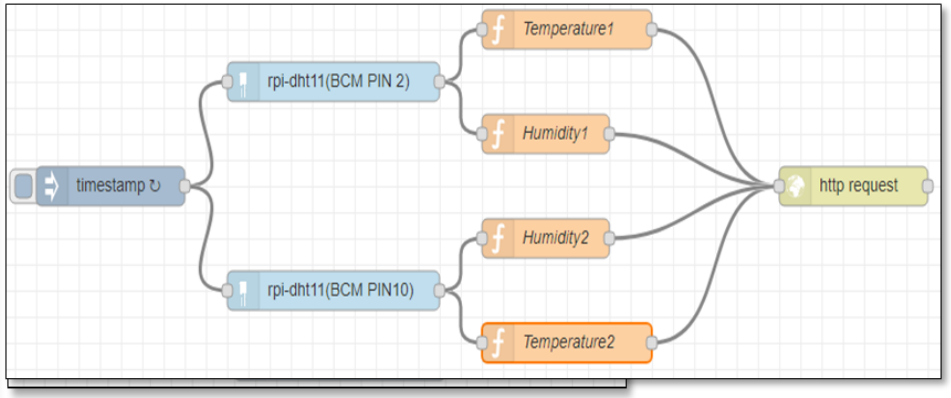


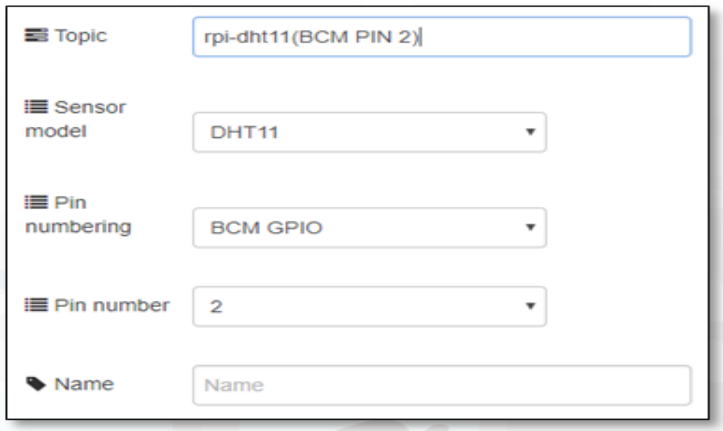


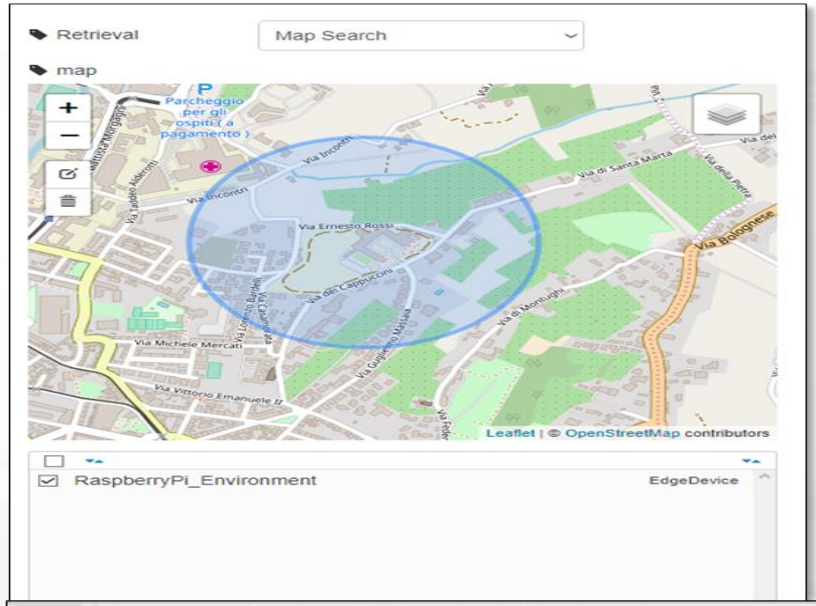




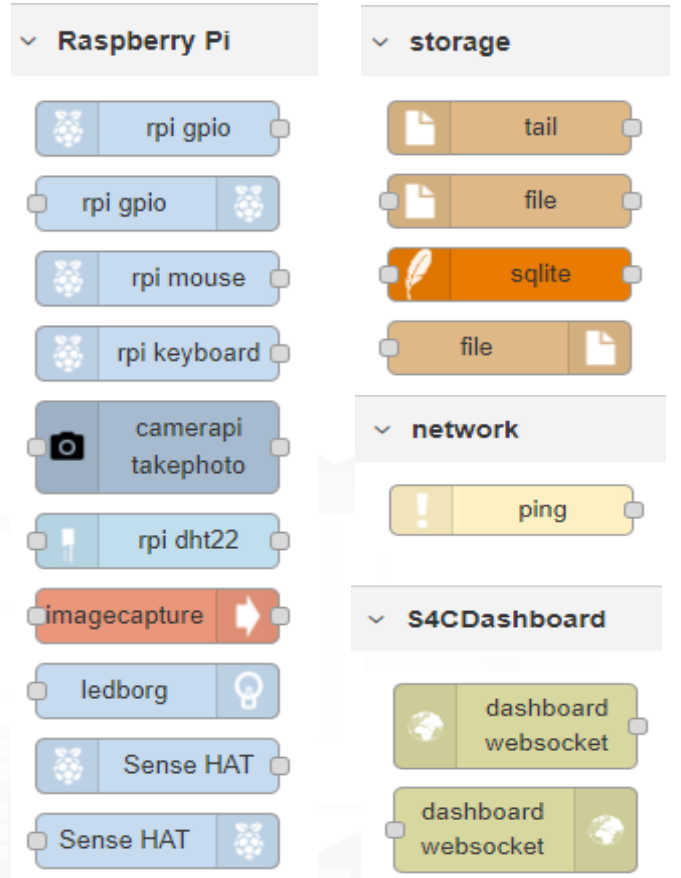
3.3V Power - 1	2 - 5V Power
SDA1 - GPIO02 - 3	4 - 5V Power
SCL1 - GPIO03 - 5	6 - Ground
GPIO04 - 7	8 - GPIO14 - TxD
Ground - 9	10 - GPIO15 - RxD
GPIO17 - 11	12 - GPIO18
GPIO27 - 13	14 - Ground
GPIO22 - 15	16 - GPIO23
3.3V Power - 17	18 - GPIO24
MOSI - GPIO10 - 19	20 - Ground
MISO - GPIO09 - 21	22 - GPIO25
SCLK - GPIO11 - 23	24 - GPIO8 - CE0
Ground - 25	26 - GPIO7 - CE1
SD - 27	28 - SC
GPIO05 - 29	30 - Ground
GPIO06 - 31	32 - GPIO12
GPIO13 - 33	34 - Ground
GPIO19 - 35	36 - GPIO16
GPIO26 - 37	38 - GPIO20
Ground - 39	40 - GPIO21








```
1  msg.payload={"temperature1":{"value":msg.payload , "type":"Float"}};
2  return msg;
```



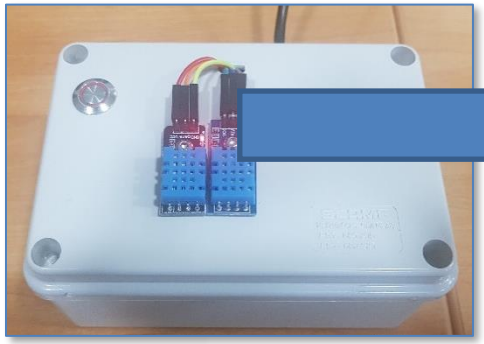
Snap4City on Raspberry Pi, IOT edge

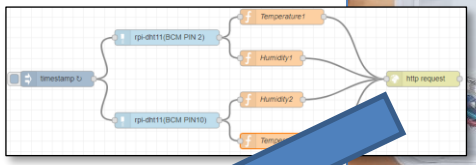
IOT Edge Computing

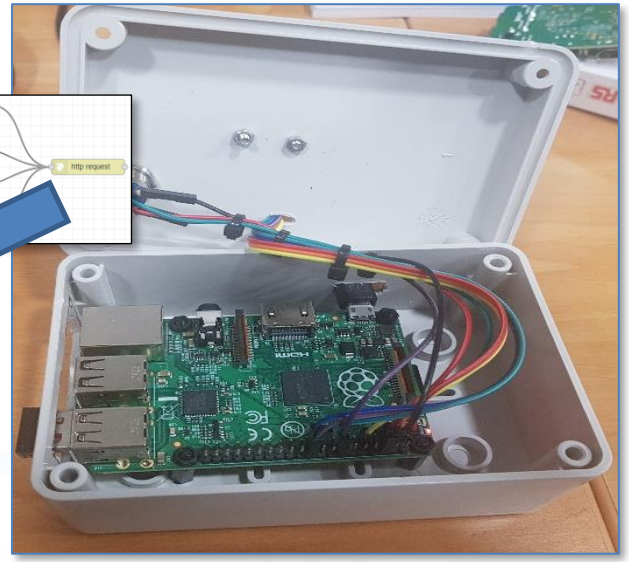
City user

Would like to:

- **Monitor and exploit temperature and humidity**
- **Manage sensors**
- **Perform edge computing**
- **Using these data for multiple applications**







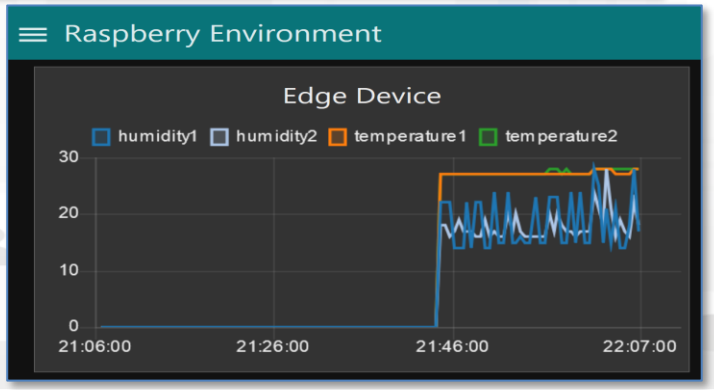
IOT Broker

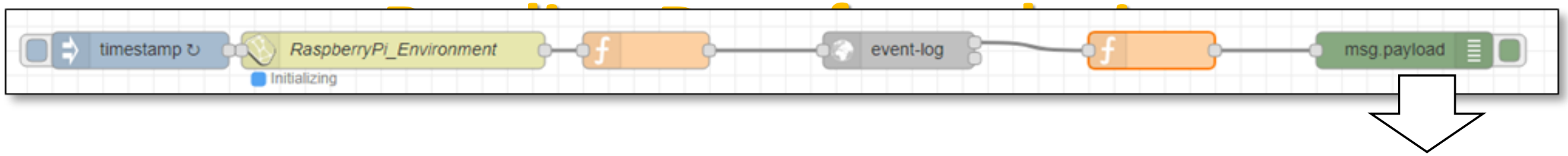
Click here



Steps:

1. **Registering the device and sensors**
2. **Create flow on edge device using NodeRed with Snap4City, sending data to Broker**
3. **Use data from Broker on Snap4City IOT App**





19/3/2018, 22:20:48 node: 1fc37579.28dbfb

msg.payload : string[459]


```
"[{"type":"EdgeDevice","id":"RaspberryPi_Environment","attributes":
[{"name":"geolocalization_lat","type":"Float","value":"43.798778"},
{"name":"geolocalization_lon","type":"Float","value":"11.253522"},
{"name":"humidity1","type":"Float","value":"30.00"},
{"name":"humidity2","type":"Float","value":"33.00"},
{"name":"model","type":"String","value":""},
{"name":"temperature1","type":"Float","value":"26.00"},
{"name":"temperature2","type":"Float","value":"26.00"}]}]"
```


19/3/2018, 22:21:02 node: 1fc37579.28dbfb

msg.payload : string[459]


```
"[{"type":"EdgeDevice","id":"RaspberryPi_Environment","attributes":
[{"name":"geolocalization_lat","type":"Float","value":"43.798778"},
{"name":"geolocalization_lon","type":"Float","value":"11.253522"},
{"name":"humidity1","type":"Float","value":"30.00"},
{"name":"humidity2","type":"Float","value":"35.00"},
{"name":"model","type":"String","value":""},
{"name":"temperature1","type":"Float","value":"26.00"},
{"name":"temperature2","type":"Float","value":"26.00"}]}]"
```


19/3/2018, 22:21:08 node: 1fc37579.28dbfb

msg.payload : string[459]


```
"[{"type":"EdgeDevice","id":"RaspberryPi_Environment","attributes":
[{"name":"geolocalization_lat","type":"Float","value":"43.798778"},
{"name":"geolocalization_lon","type":"Float","value":"11.253522"},
{"name":"humidity1","type":"Float","value":"30.00"},
{"name":"humidity2","type":"Float","value":"35.00"},
{"name":"model","type":"String","value":""},
{"name":"temperature1","type":"Float","value":"26.00"},
{"name":"temperature2","type":"Float","value":"26.00"}]}]"
```

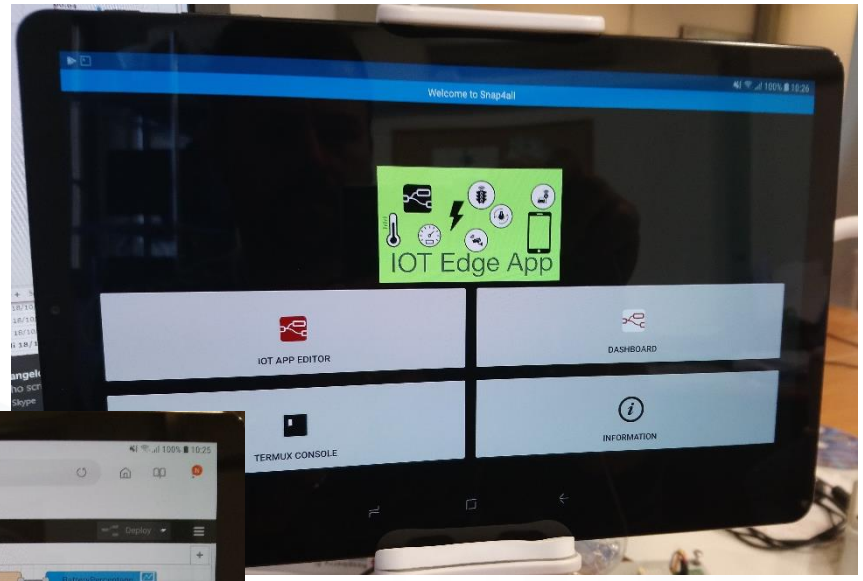

19/3/2018, 22:21:20 node: 1fc37579.28dbfb

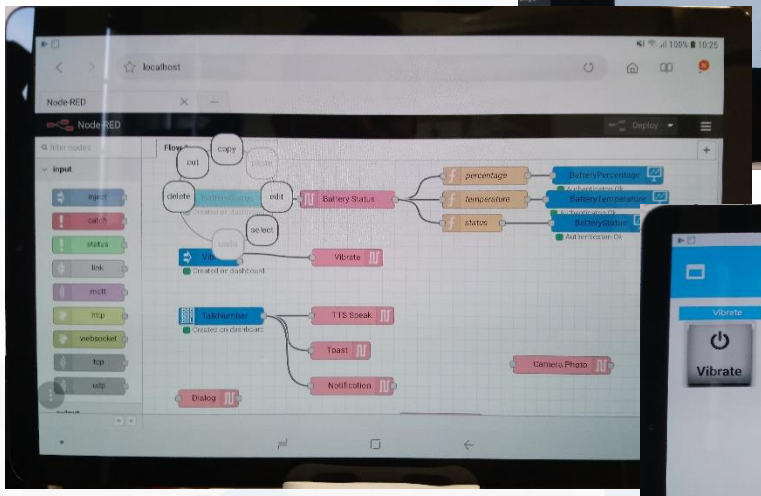
msg.payload : string[459]

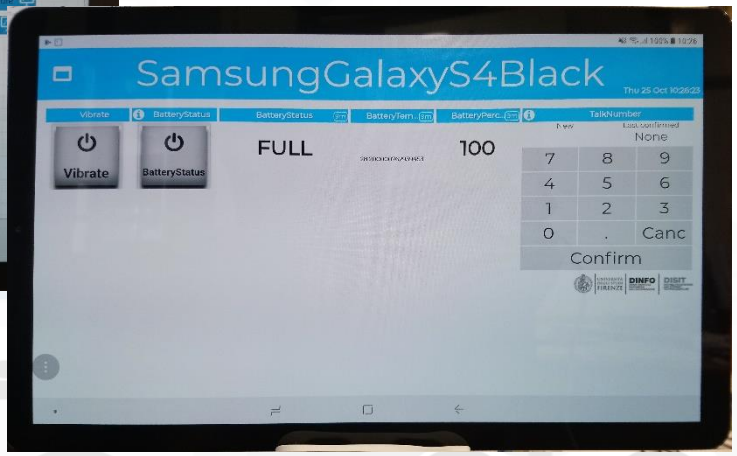

```
"[{"type":"EdgeDevice","id":"RaspberryPi_Environment","attributes":
[{"name":"geolocalization_lat","type":"Float","value":"43.798778"},
{"name":"geolocalization_lon","type":"Float","value":"11.253522"},
{"name":"humidity1","type":"Float","value":"30.00"},
{"name":"humidity2","type":"Float","value":"35.00"},
{"name":"model","type":"String","value":""},
{"name":"temperature1","type":"Float","value":"26.00"},
{"name":"temperature2","type":"Float","value":"26.00"}]}]"
```

IOT Edge Snap4All App for Android

- **Android**, any version, App from: https://www.snap4city.org/download/video/Snap4All.apk
- **Mutual Authentication** with certificates
- *Secure encrypted connection*, NGSI
- **IOT Application inside**
- **Any sensor** + Local device sensors
- **Any protocol** from IOT devices
- **NGSI** or any other protocol
- **Fully Customizable**
- Local and Cloud Dashboard
- **Special MicroServices**



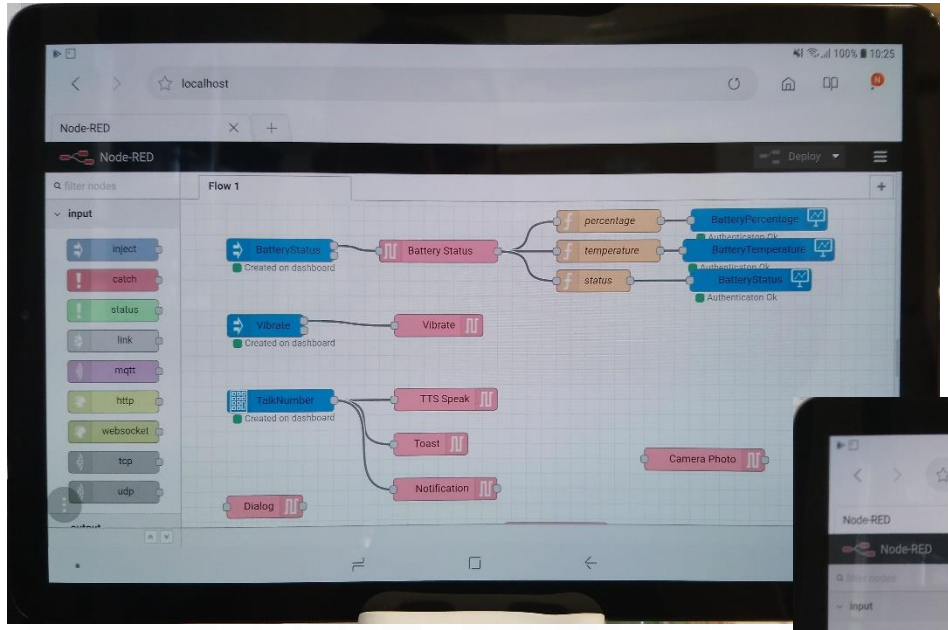


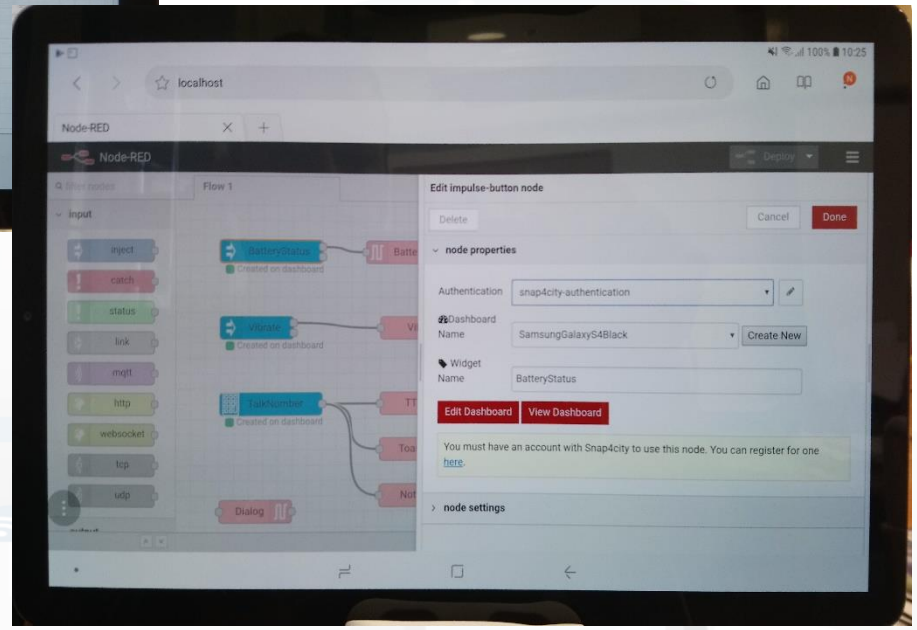


https://www.snap4city.org/drupal/node/278

IOT Edge Snap4All App for Android

- termux-battery-status
- termux-camera-info
- termux-clipboard-get
- termux-contact-list
- termux-telephony-cellinfo
- termux-telephony-deviceinfo
- termux-tts-engines
- termux-camera-photo
- termux-clipboard-set
- termux-dialog
- termux-download
- termux-location
- termux-tts-speak
- termux-vibrate
- termux-sms-inbox
- termux-toast
- termux-sms-send
- termux-share
- termux-notification
- termux-wifi-connectioninfo
- termux-wifi-scaninfo

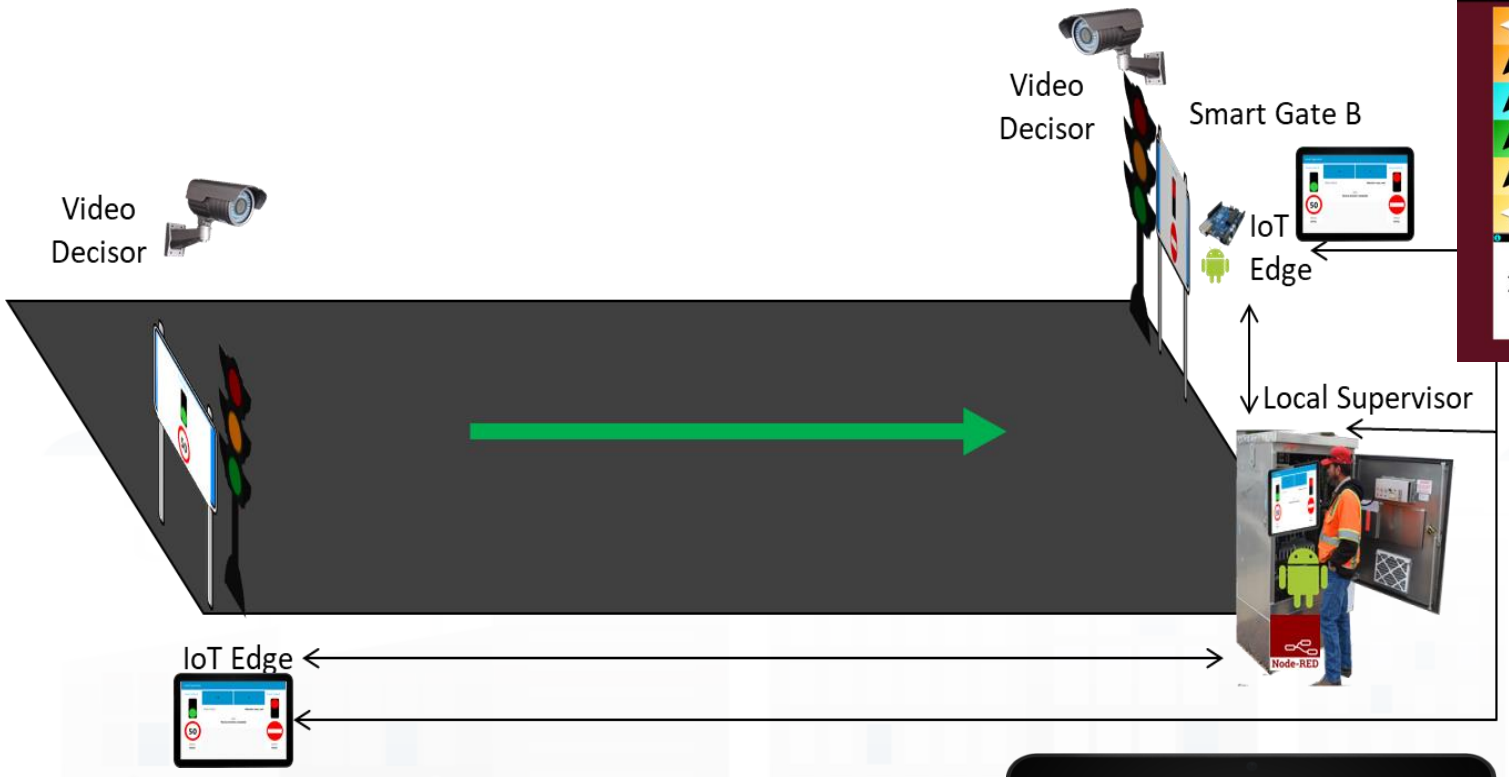


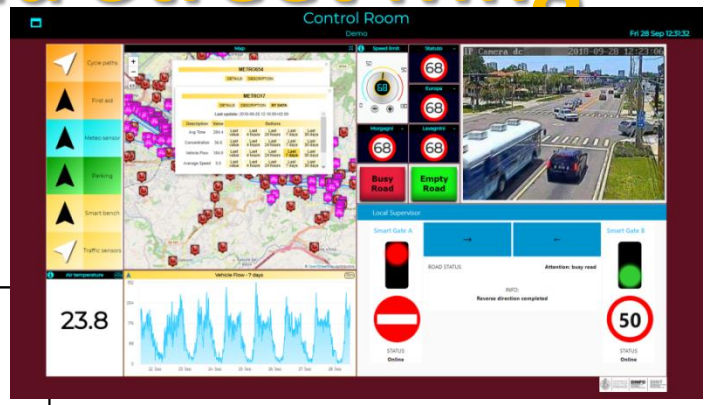


MicroServices:

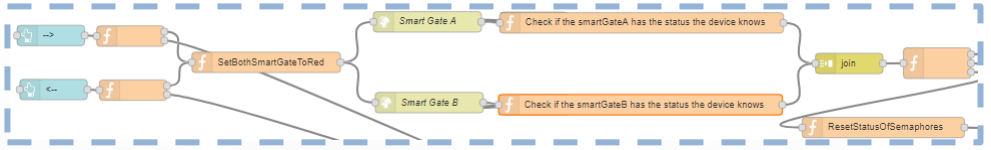
- Snap4City
- Termux Snap4City specific
- etc.

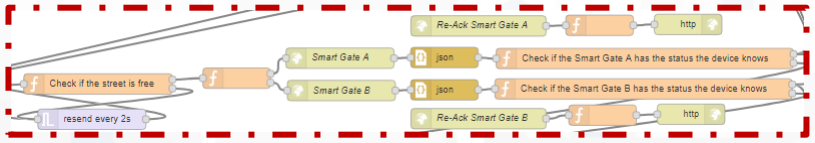
Sii-Mobility: Dynamic Signage and Street Mng

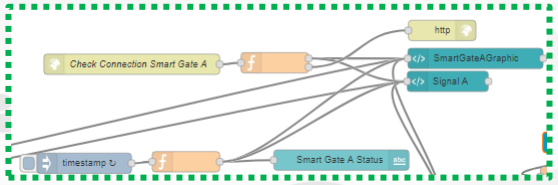




Control Room







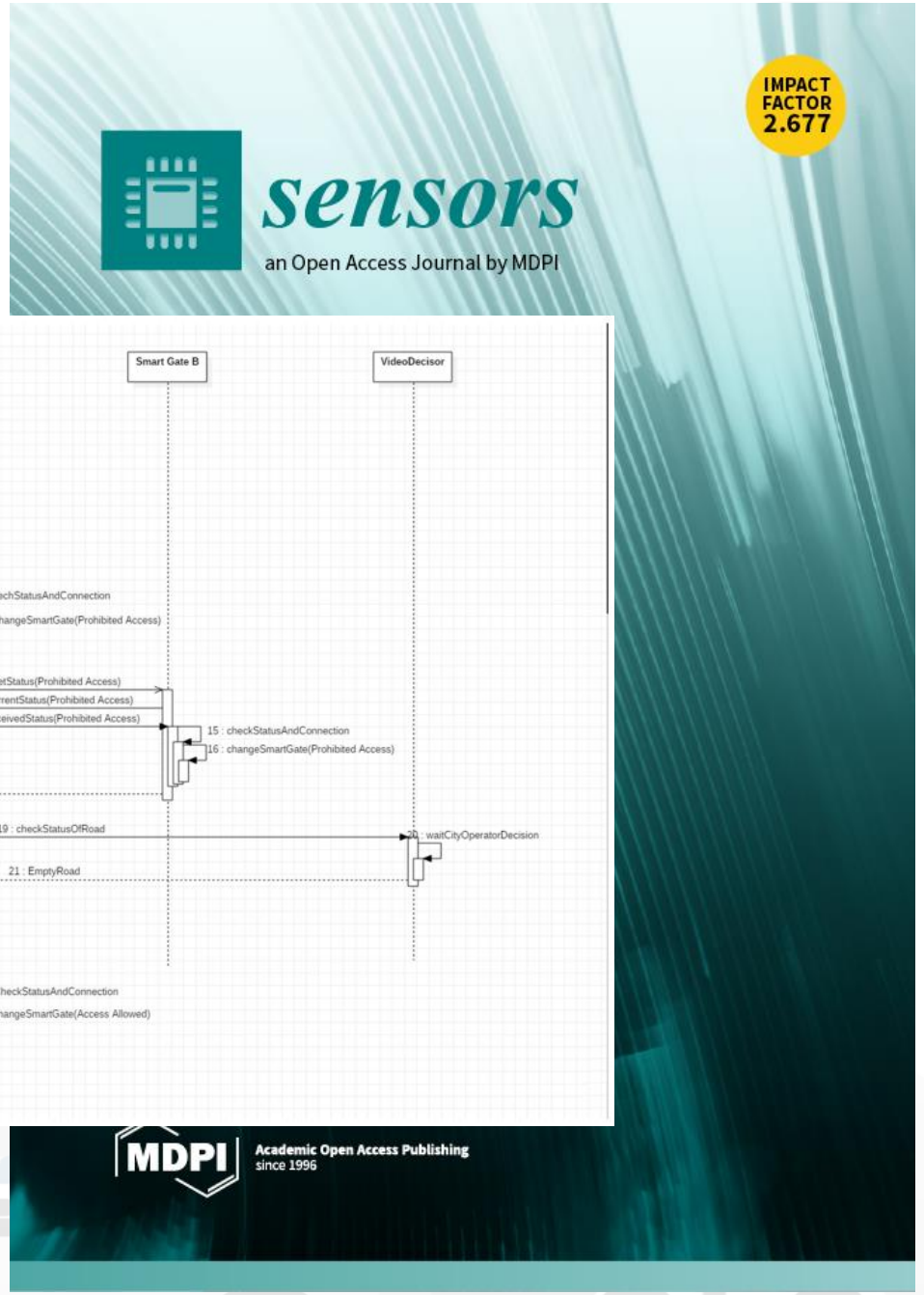
Local Control Dashboard



Safe and resilient solution manging Degradated conditions

IOT for Mobility Infrastructure

- C. Badii, P. Bellini, A. Difino, P. Nesi, "*Sii-Mobility: an IOT/IOE architecture to enhance smart city services of mobility and transportation*", Sensors, MDPI, 2019
- https://www.mdpi.com/1424-8220/19/1/1/pdf



IOT Tracking Devices



PaxCounter devices







- **Fix PaxCounter LoraWan**
  - sniffing on: Wi-Fi, Bluetooth
  - Sending data via LoraWan
- **Mobile PaxCounter LoraWan**
  - sniffing on: Wi-Fi, Bluetooth
  - Sending data via LoraWan
- **Fix PaxCounter, multiple out**
  - Sending data via LoraWan and Wi-Fi
  - sniffing on: Wi-Fi, Bluetooth

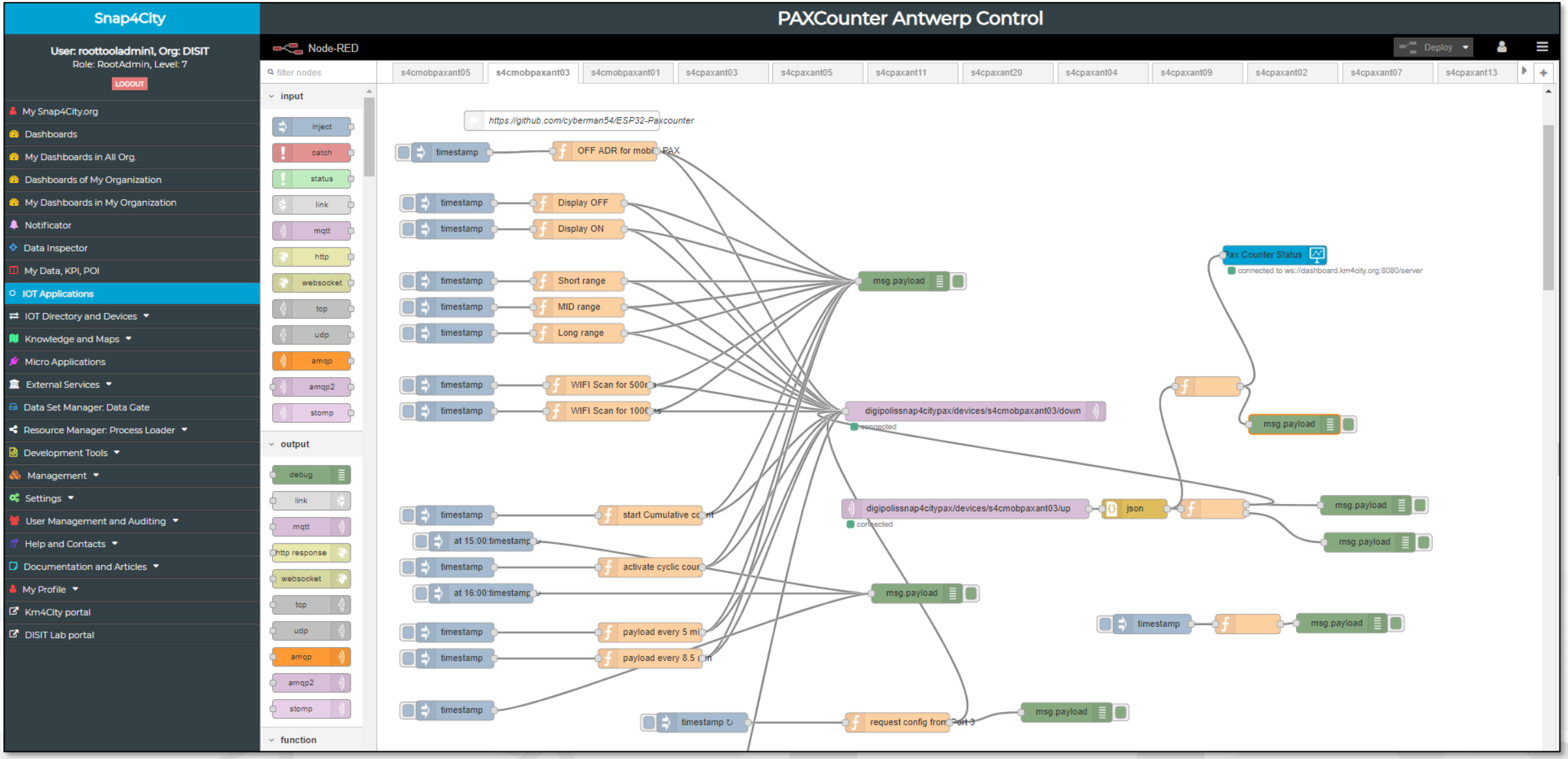
LoRa® Wi Fi Bluetooth™

https://www.snap4city.org/456

Programmable PAX counting



IOT App behind



PAXCounter real time and trend

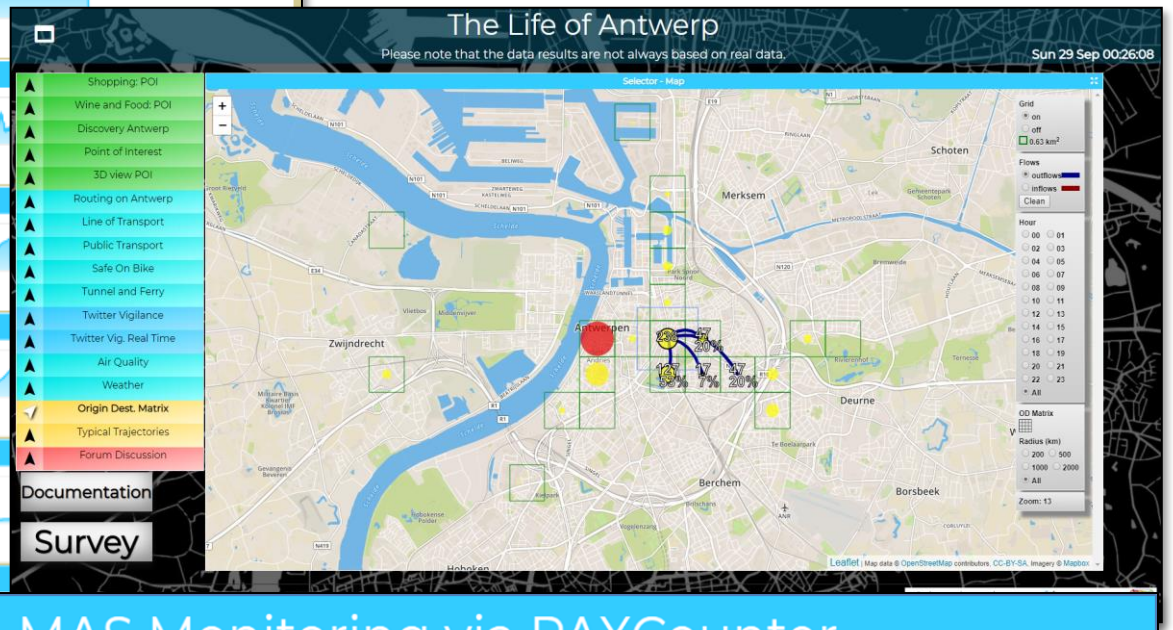
Mon 30 Sep 17:18:48

s4cpaxant01	3	s4cpaxant11	72
s4cpaxant02	1157	s4cpaxant12	15
s4cpaxant03	17	s4cpaxant13	295
s4cpaxant04	7	s4cpaxant14	13
s4cpaxant05	4	s4cpaxant15	59
s4cpaxant06	17	s4cpaxant16	30
s4cpaxant07	952		



SNAP4CITY

KM 4 CITY



MAS Monitoring via PAXCounter

Mon 30 Sep 17:31:32

pax03 in	17	s4cpaxant03 - ingang gelijkvloers balie	s4cpaxant03 - wifi / s4cpaxant03delta - wifi
pax05 8ste	9	s4cpaxant05 - 8ste verdieping, ingang	s4cpaxant05 - 8ste ingang H24 / s4cpaxant05 change on passages
pax20 2de	13	s4cpaxant20 - 2de verdieping, uitgang	s4cpaxant20 - 2de uitgang H24 / s4cpaxant20delta - change on passages
pax11 4de	74	s4cpaxant11 - 4de verdieping, uitgang	s4cpaxant11 - 4de uitgang H24 / s4cpaxant11delta - change on passages

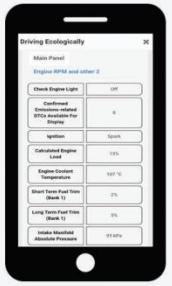
IOE – Vehicle Monitoring

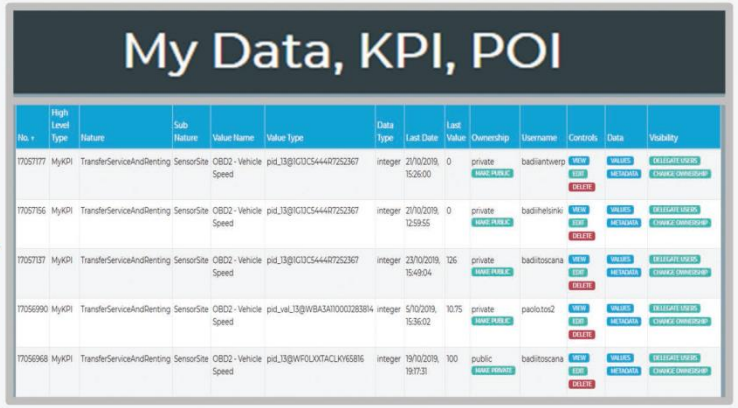


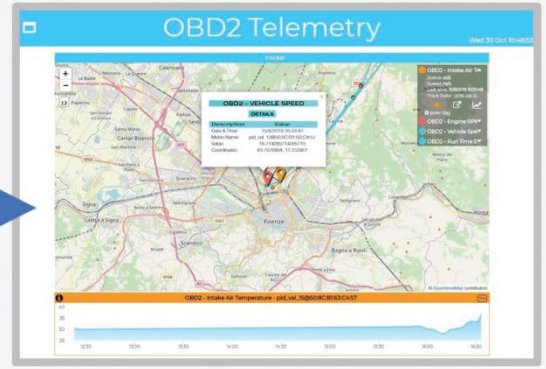
CANBUS sniffer



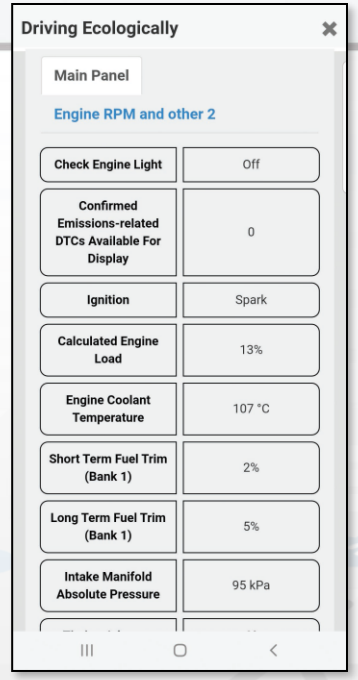
Bluetooth

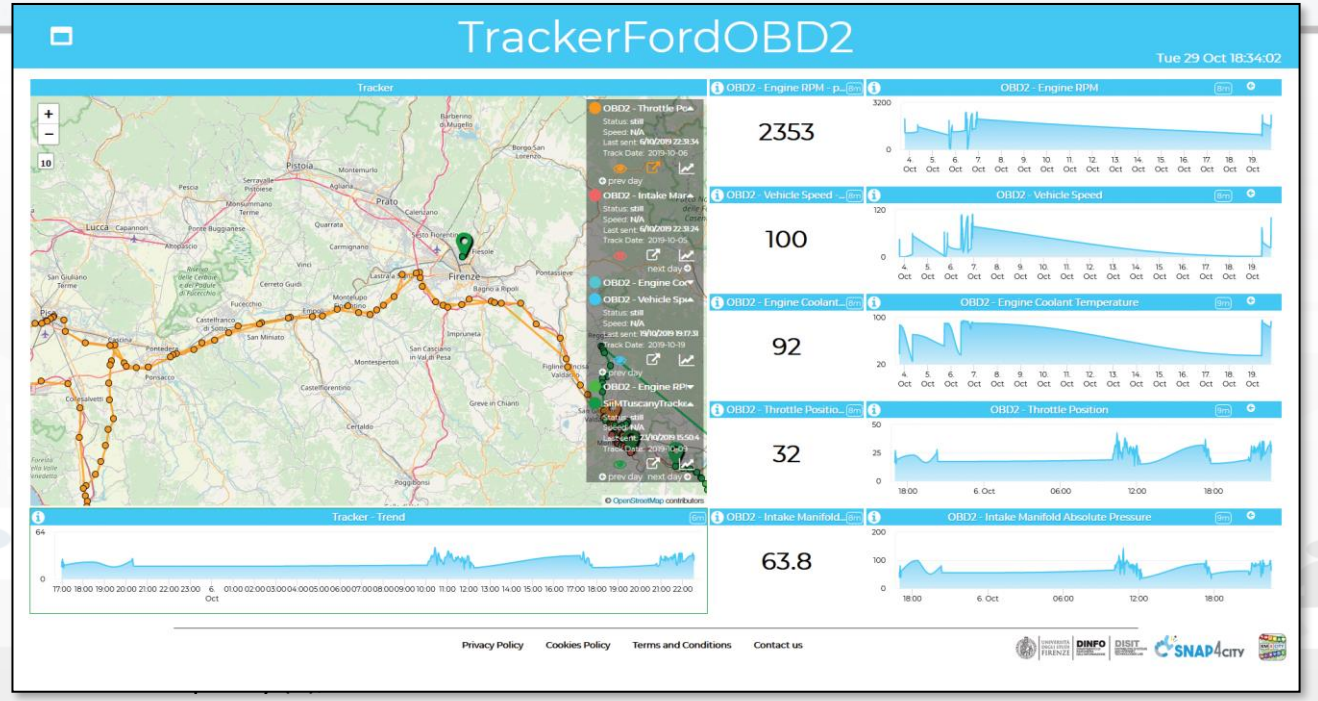






Tuscany in a Snap Mobile App on Android

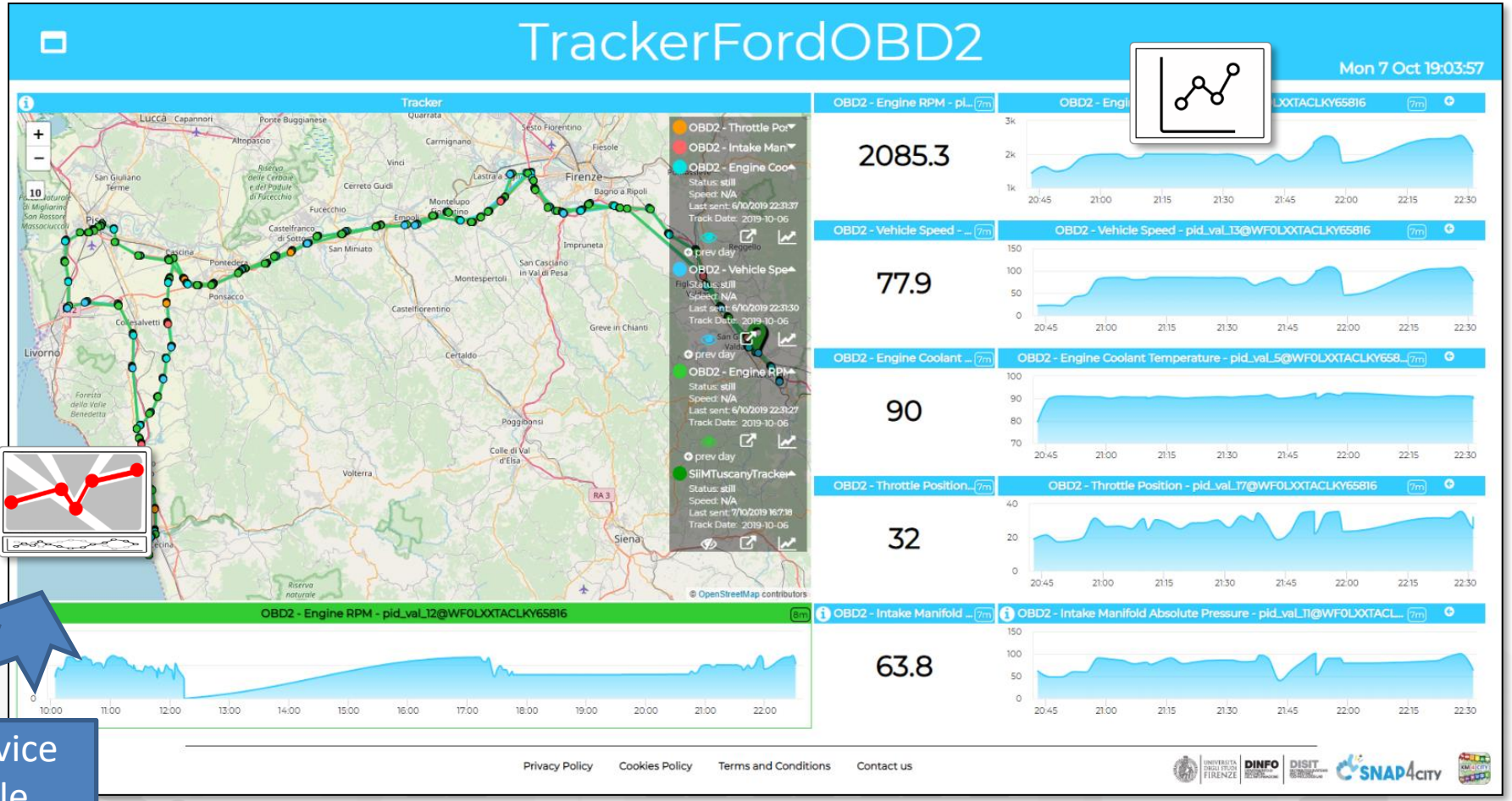




MyKPI: Tracking of Devices and Mobiles

- Real Time Trajectories for
  - Mobile Phone
  - Moving IOT Devices
  - OBU, Vehicular Kits
  - Multiple tracks
  - Day by day
- Micro Application





IOT Devices and IOT Edge (Self Training)

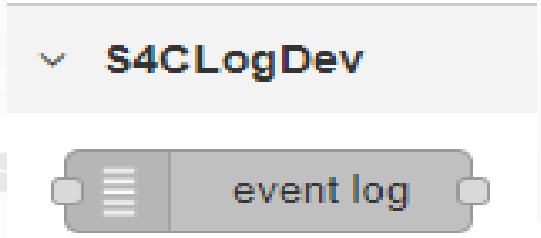
- **A large range of Devices** can be used on Snap4City:
  - Proprietary or Open HW/SW.
  - Devices of/for makers on which we provide Open source code
- **Documentation and instructions:**
  - TC9.4 - IOT application exploiting Edge computing with Raspberry
  - TC9.7 - Connection from LoraWan Dragino/arduino to Orion broker
  - Snap4City: Arduino & ESP8266 IOT Device NGSI
  - Snap4City IOT Devices Registration
  - Snap4All IOT Button: based on ESP32, NGSI compliant secure connection
  - IDE Setup for Snap4All IOT Button, and source code
  - Registering IOT Edge: example of Raspberry Pi, total security
  - Creating: IOT Device, Raspberry Pi based, totally compliant with Snap4City

Logging Internal Events

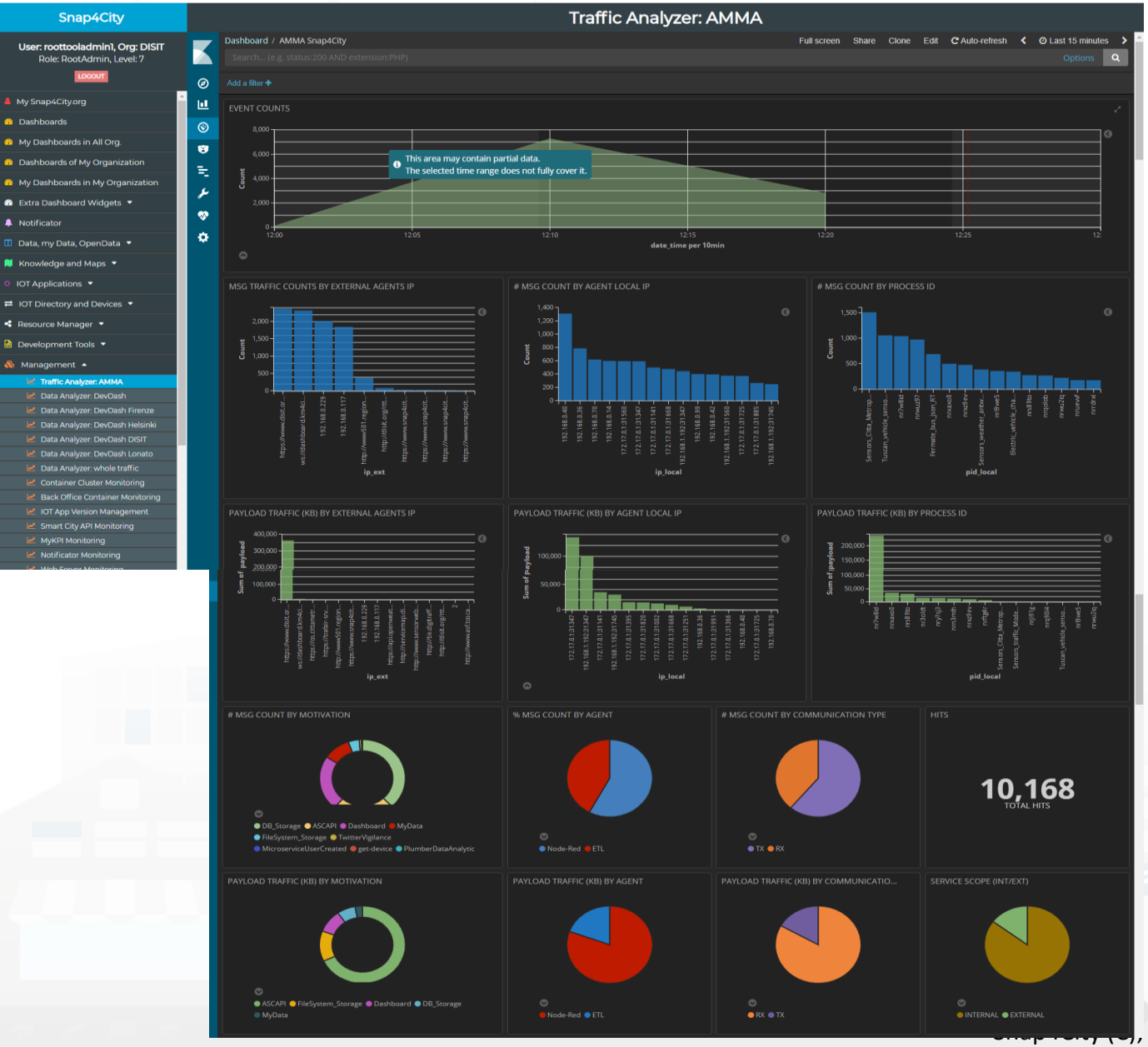


Event Logger

- There are API for Event Logger, REST API
  - They are automatically used by most of the Snap4City MicroServices
  - They log in standard Rsyslog API
    - https://www.snap4city.org/56
- The Logs regarding messages passed and usage are logged and accessed with the AMMA tools that is based on Elastic Search and Kibana.
  - Former version was made in Hbase and SOLR, and Banana
- Additional Logs events can be logged by using a dedicated MicroService in Node-RED, IOT Apps

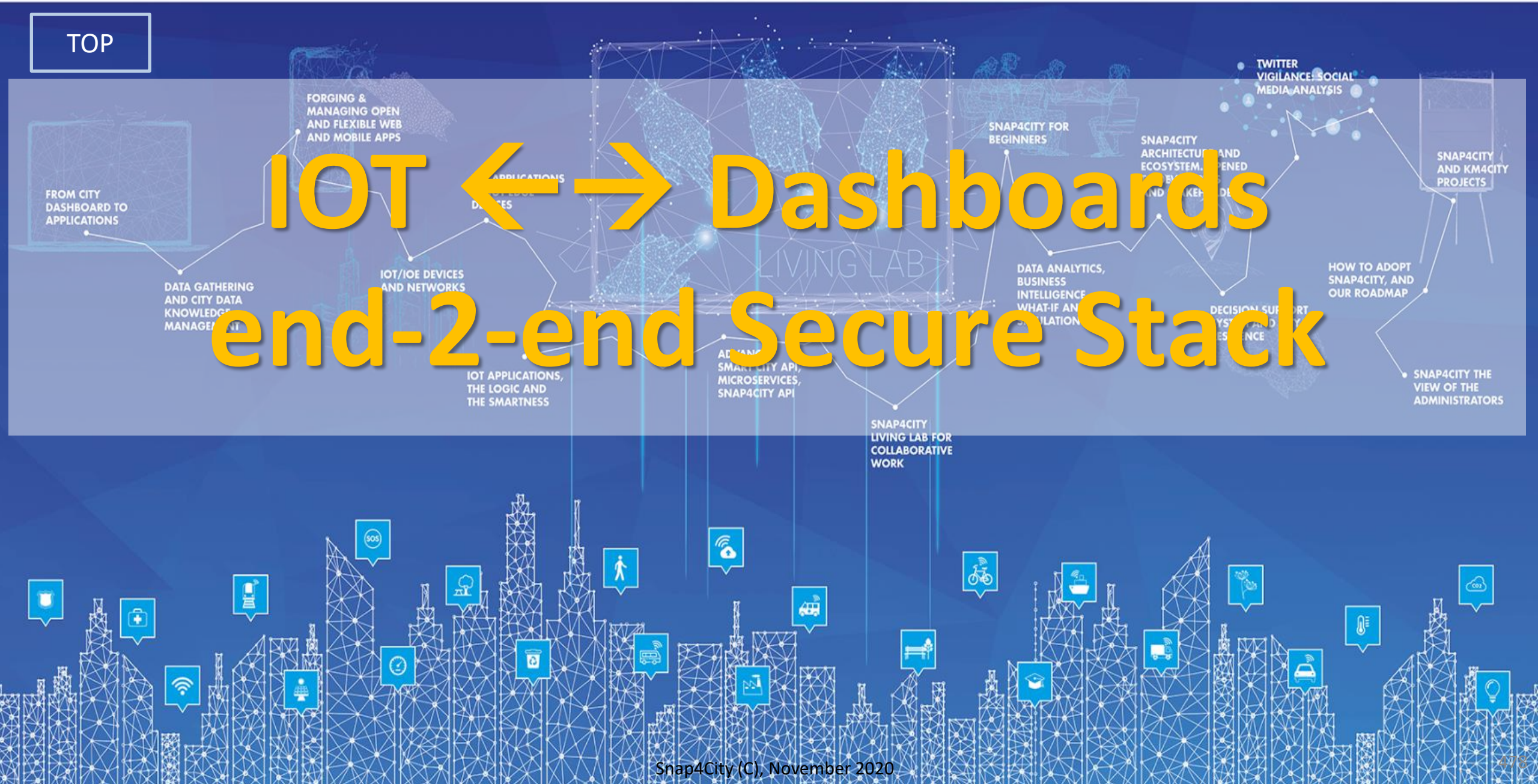


AMMA



- ***Managing and Monitoring Data-Traffic in the BackOffice***
- Data Traffic Analyzer
  - Business intelligence
  - Faceted searches
  - Drill down on time
- Several different views and details on data traffic among the main entities in the platform:
  - IOT APP
  - Storage
  - Data sources,
  - ….

TOP



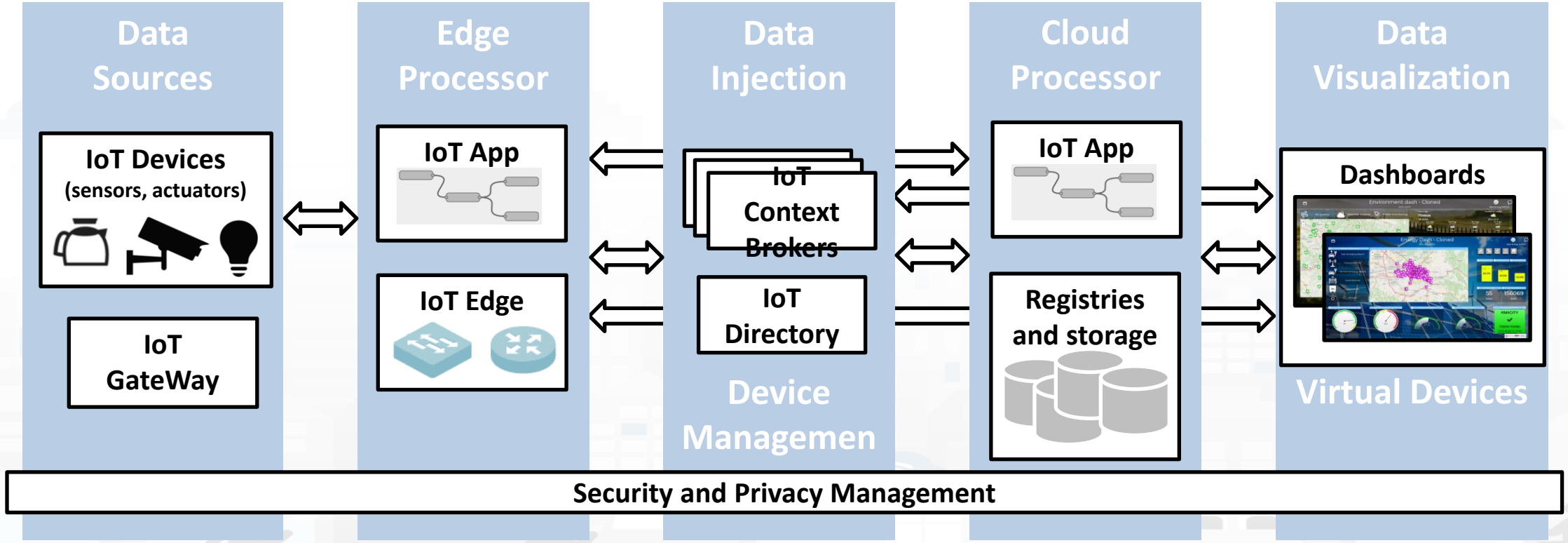
IOT ←→ Dashboards end-2-end Secure Stack

Complexity in Smart City IOT Platforms

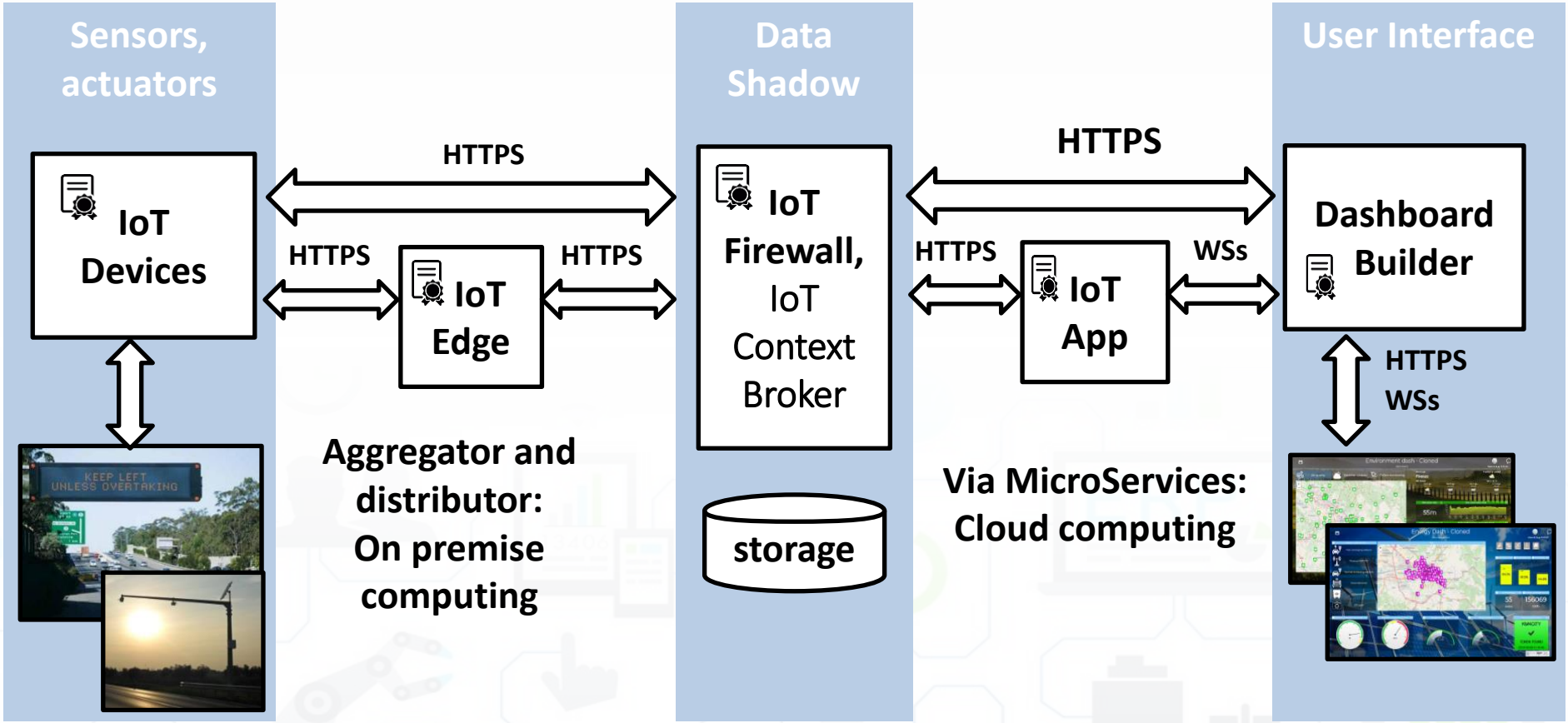
End to End security

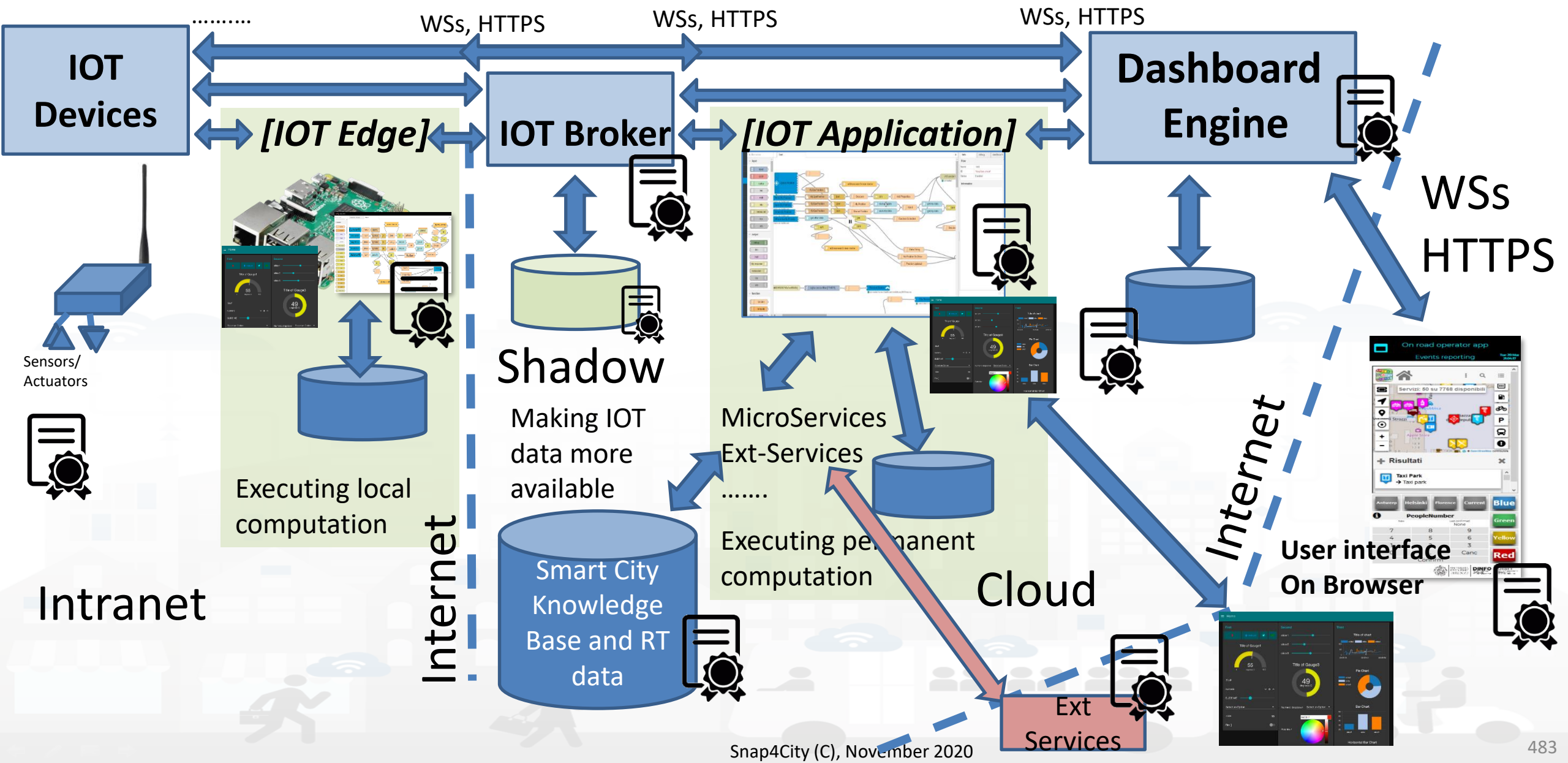
– From IOT Devices to Dashboard (user interface)

- H2M
- M2M



The secure stack

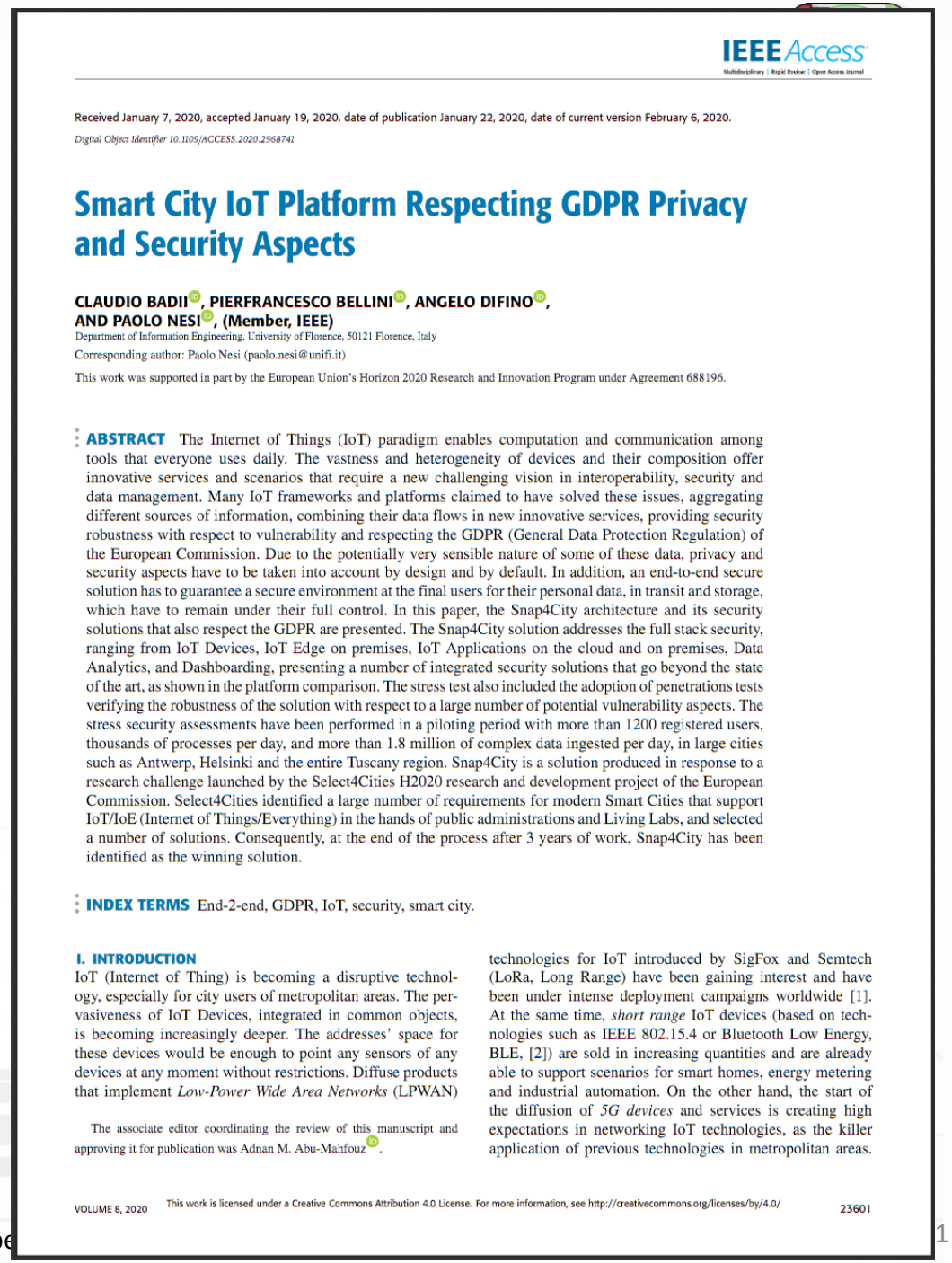



.....
WSs, HTTPS
WSs, HTTPS
WSs, HTTPS
IOT Devices
[IOT Edge]
IOT Broker
[IOT Application]
Dashboard Engine
WSs
HTTPS
Sensors/
Actuators
Executing local computation
Shadow
Making IOT data more available
MicroServices
Ext-Services
.....
Executing permanent computation
Internet
Smart City Knowledge Base and RT data
Intranet
Cloud
Internet
Ext Services
User interface On Browser

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

IEEE Access

- C. Badii, P. Bellini, A. Difino, P. Nesi, "Smart City IoT Platform Respecting GDPR Privacy and Security Aspects", accepted for publication on IEEE Access, 2020. 10.1109/ACCESS.2020.2968741 https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8966344



IEEE Access

Received January 7, 2020, accepted January 19, 2020, date of publication January 22, 2020, date of current version February 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2968741

Smart City IoT Platform Respecting GDPR Privacy and Security Aspects

CLAUDIO BADII, PIERFRANCESCO BELLINI, ANGELO DIFINO, AND PAOLO NESI, (Member, IEEE)

Department of Information Engineering, University of Florence, 50121 Florence, Italy

Corresponding author: Paolo Nesi (paolo.nesi@unifi.it)

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Program under Agreement 688196.

ABSTRACT The Internet of Things (IoT) paradigm enables computation and communication among tools that everyone uses daily. The vastness and heterogeneity of devices and their composition offer innovative services and scenarios that require a new challenging vision in interoperability, security and data management. Many IoT frameworks and platforms claimed to have solved these issues, aggregating different sources of information, combining their data flows in new innovative services, providing security robustness with respect to vulnerability and respecting the GDPR (General Data Protection Regulation) of the European Commission. Due to the potentially very sensible nature of some of these data, privacy and security aspects have to be taken into account by design and by default. In addition, an end-to-end secure solution has to guarantee a secure environment at the final users for their personal data, in transit and storage, which have to remain under their full control. In this paper, the Snap4City architecture and its security solutions that also respect the GDPR are presented. The Snap4City solution addresses the full stack security, ranging from IoT Devices, IoT Edge on premises, IoT Applications on the cloud and on premises, Data Analytics, and Dashboarding, presenting a number of integrated security solutions that go beyond the state of the art, as shown in the platform comparison. The stress test also included the adoption of penetrations tests verifying the robustness of the solution with respect to a large number of potential vulnerability aspects. The stress security assessments have been performed in a piloting period with more than 1200 registered users, thousands of processes per day, and more than 1.8 million of complex data ingested per day, in large cities such as Antwerp, Helsinki and the entire Tuscany region. Snap4City is a solution produced in response to a research challenge launched by the Select4Cities H2020 research and development project of the European Commission. Select4Cities identified a large number of requirements for modern Smart Cities that support IoT/IoE (Internet of Things/Everything) in the hands of public administrations and Living Labs, and selected a number of solutions. Consequently, at the end of the process after 3 years of work, Snap4City has been identified as the winning solution.

INDEX TERMS End-2-end, GDPR, IoT, security, smart city.

I. INTRODUCTION

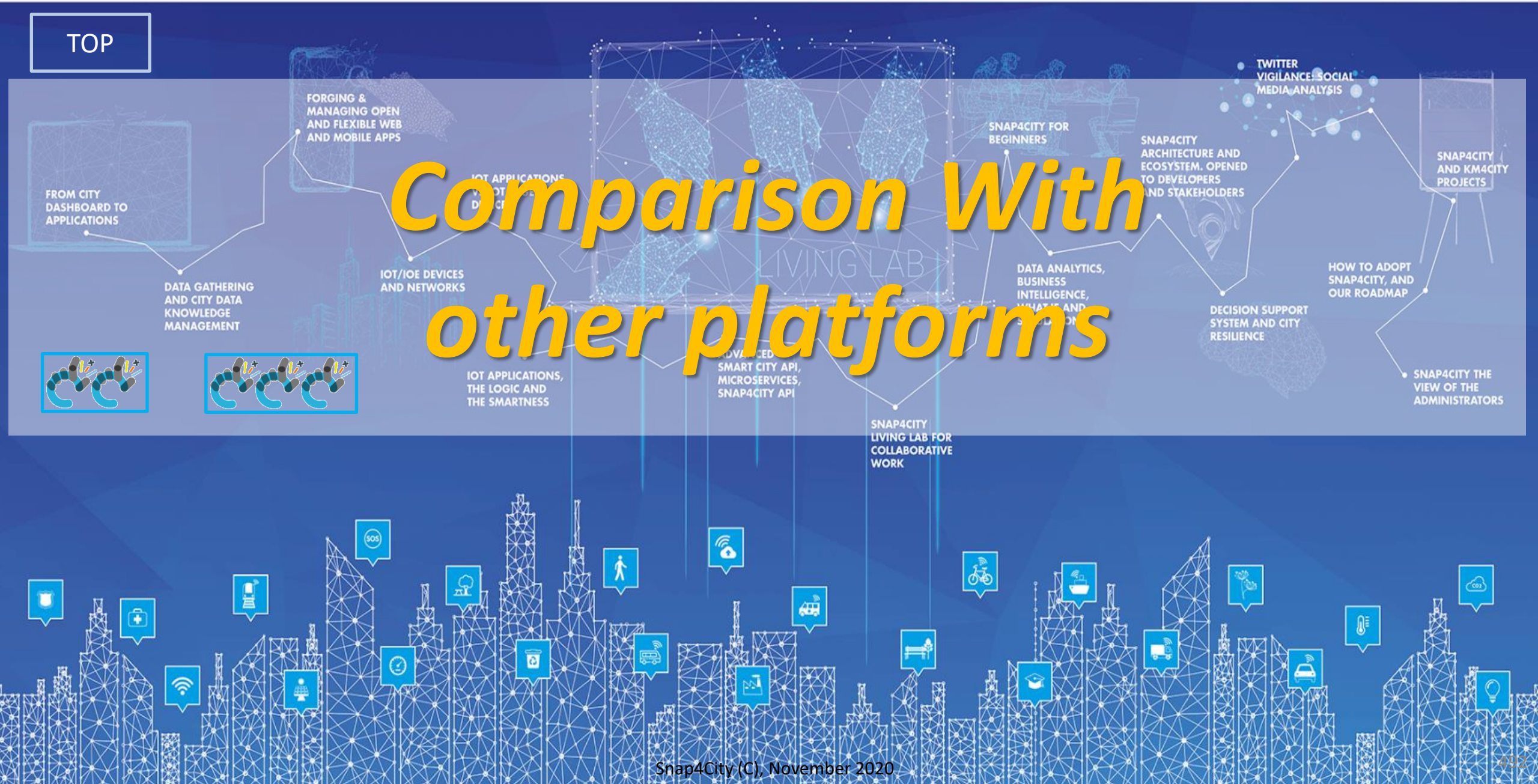
IoT (Internet of Thing) is becoming a disruptive technology, especially for city users of metropolitan areas. The pervasiveness of IoT Devices, integrated in common objects, is becoming increasingly deeper. The addresses' space for these devices would be enough to point any sensors of any devices at any moment without restrictions. Diffuse products that implement *Low-Power Wide Area Networks* (LPWAN) technologies for IoT introduced by SigFox and Semtech (LoRa, Long Range) have been gaining interest and have been under intense deployment campaigns worldwide [1]. At the same time, *short range* IoT devices (based on technologies such as IEEE 802.15.4 or Bluetooth Low Energy, BLE, [2]) are sold in increasing quantities and are already able to support scenarios for smart homes, energy metering and industrial automation. On the other hand, the start of the diffusion of *5G devices* and services is creating high expectations in networking IoT technologies, as the killer application of previous technologies in metropolitan areas.

The associate editor coordinating the review of this manuscript and approving it for publication was Adnan M. Abu-Mahfouz.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

TOP

Comparison With other platforms

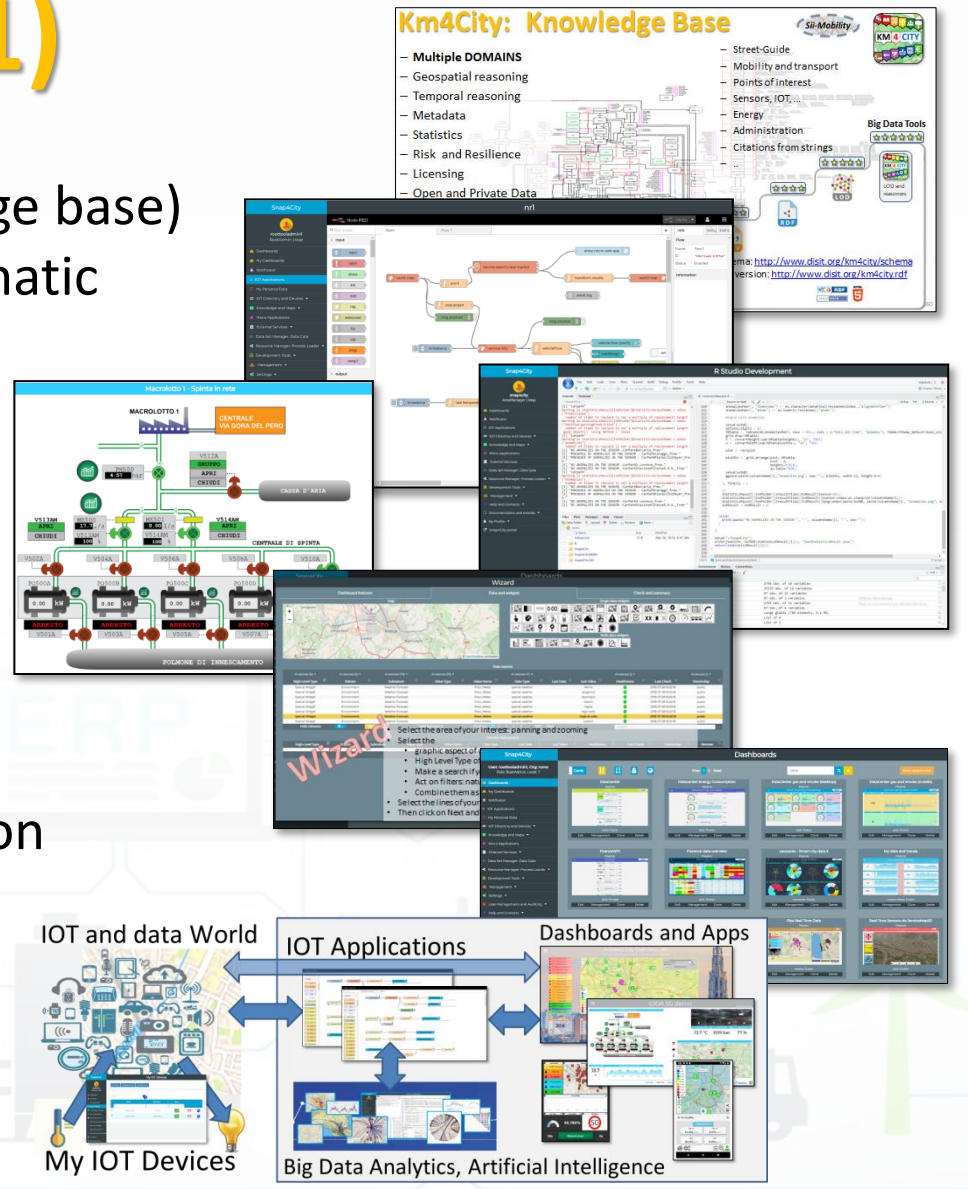


State of the Art Solutions vs Snap4City

	IOT Discovery Abstraction	Authentication, Authorization	Security end-2-end, secure on IOT and Dashboards	Open HW and Open SW	Integrated Community management	Data Types: IOT Devices, IOT App, Dashboard, Data	Data Type: Publish/share, Delegation, Consent and change	Data Type: Download and Delete	Auditing on Data Type Access	Open Source end-to-end	Scalability IOT	Visual Programming end-to-end applications	Advanced Smart City API, MicroServices	Multi Domain Semantic Platform	Standard based Modules and IOT, Open Devices	Resource Sharing	Data Analytics integrated	Dashboard H24/7, protected connection	Multi-protocol on IOT
		G				G	G	G	G										
Snap4City	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
KAA [53]	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	N	Y	N	(Y)	N	N	Y	Y
Thingsboard [55]	Y	Y	Y	Y	N	Y	N	Y	Y	Y	Y	N	N	N	N	N	N	Y	MQTT,coap, http
IOT eclipse.org [56]	N	N	N	(Y)	N	Y	N	N	N	Y	Y	N	N	N	Y	N	N	N	Y
IOT IGNITE [57]	N	Y	N	Y	N	Y	N	Y	Y	Y	Y	Y	N	N	N	N	N	Y	MQTT
FIWARE [47]	N	Y	N	Y	N	N	N	Y	N	Y	(Y)	(N)	Y	N	Y	N	N	Y	Y
ARM mbed IoT [48]	Y	Y	Y	Y	Y	N	(N)	N	Y	Y	Y	N	N	N	Y	N	N	Y	Limited
Airvantage [51]	Y	Y	Y	Y	N	Y	N	Y	Y	Y	Y	N	N	N	N	N	N	Y	MQTT, HTTP
AWS [43]	Y	Y	Y	Y	N	Y	(N)	Y	Y	N	Y	N	N	N	Y	Y	(Y)	Y	Limited
Azure IOT [44]	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	N	N	N	Y	Y	(Y)	Y	Limited
PTC ThingWorkx [59]	N	Y	Y	Y	Y	Y	N	N	Y	N	Y	Y	N	N	Y	N	N	Y	Y
Bosch IoT Suite [58]	Y	Y	Y	Y	Y	(Y)	(N)	Y	Y	N	Y	Y	Y	N	Y	N	Y	Y	Y
CISCO Jasper [55]	Y	Y	Y	Y	N	(Y)	(N)	N	Y	N	Y	N	N	N	N	--	(Y)	Y	N
Siemens MindSphere [60]	Y	Y	Y	(Y)	N	Y	(N)	Y	Y	N	Y	Y	N	N	Y	N	Y	Y	Y
Carriots [54]	Y	Y	Y	(Y)	N	Y	N	N	Y	N	Y	N	N	N	--	N	N	Y	MQTT
Google IOT [45]	Y	Y	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	N	N	N	(Y)	(Y)	MQTT, HTTP
Homekit Apple [50]	Y	Y	Y	Y	N	Y	N	N	Y	N	(Y)	N	N	N	N	Y	N	Y	Limited
Smarthing Samsung [52]	Y	Y	Y	Y	Y	Y	(Y)	Y	Y	N	(Y)	N	N	N	N	N	N	Y	Limited

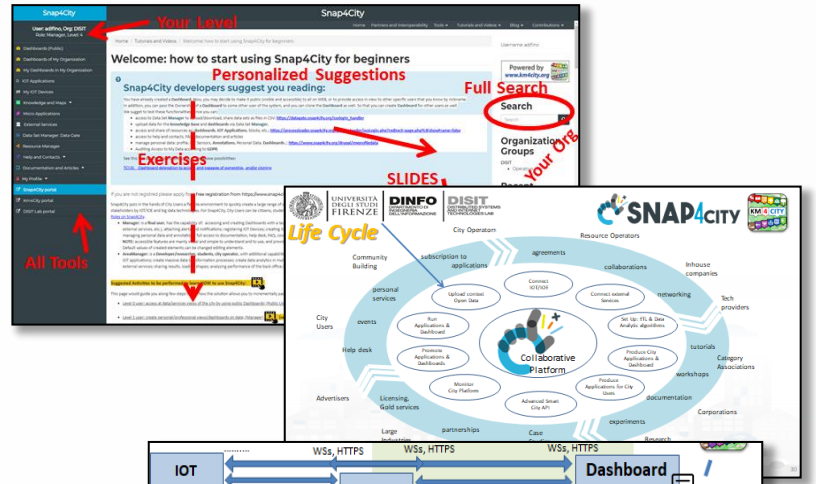
Unique of Snap4City Platform (1)

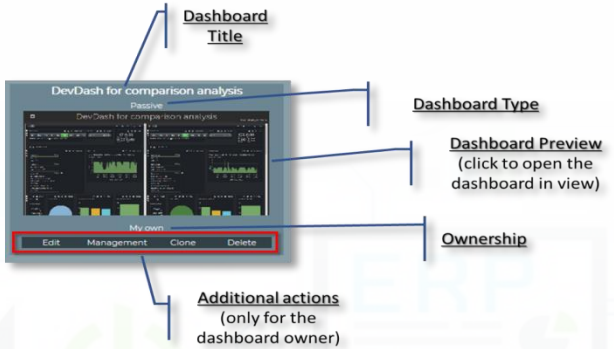
- Data ingestion and model
  - **Unified data model** (exploited in the Wizard and Knowledge base)
  - **Semantic Reasoner** modelling city entities, supporting sematic search, expert system, digital Twin, etc.
  - **IOT Directory abstracting complexity of IOT Devices, Edge, Brokers**, protocols and data formats
- Data Analytics and Data Processes
  - **Flexible and extensible IOT Applications**
  - **Data Analytic:** multiple programming languages,
- Visual Analytics, dashboarding, Apps
  - **Wizard: expert system** for immediate dashboard production matching data vs graphics representation
  - **Dashboards specialized** multidomain for Smart Cities
  - **Custom Widgets and Synoptics**
  - **Ready to use Mobile App, instant App, MicroApplication**
  - **Strategies** formalization supports

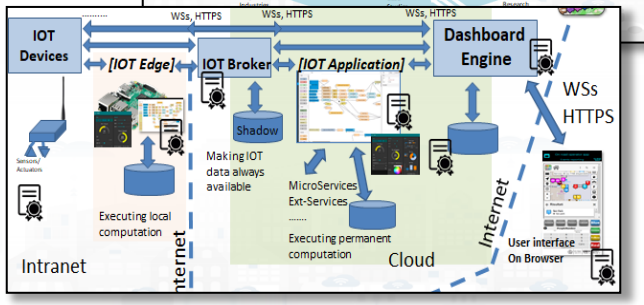


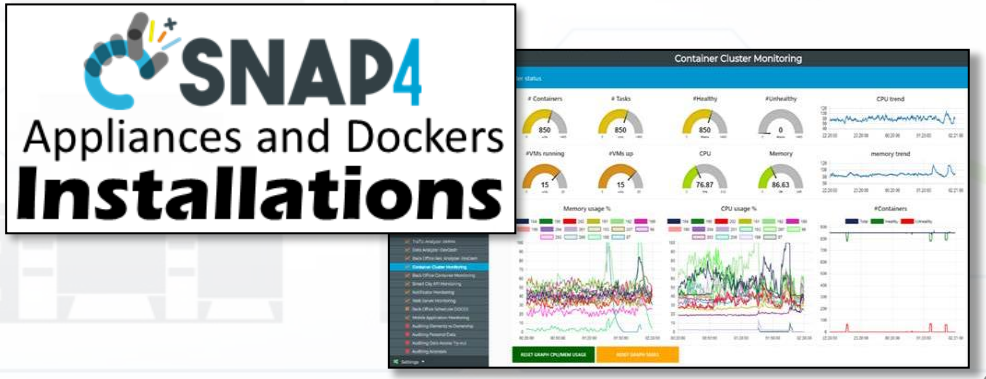
Unique of Snap4City Platform (2)

- Openness to any developers
  - **Living Lab support for coworking, sharing, and delegating**
  - **Multitenacy, multiorganization and geoareas**
  - **Advanced Smart City APIs and MicroServices**
  - **100% Open Source, Open hardware**
- Security and Privacy
  - **End-2-end encrypted communication**, on devices, platform, ... dashboards
  - **GDPR compliant** privacy/security
- Non functional
  - **on cloud and on premise,** your private installation
  - **Ready to use Appliance** Virtual Machines and/or Containers for a modules and tools.
  - **Flexible, Modular, Elastic, scalable and robust**









Fi-Ware vs Snap4City



FI-WARE

SMART CITIES REFERENCE ARCHITECTURE

VS

SNAP4INDUSTRY

KM4CITY

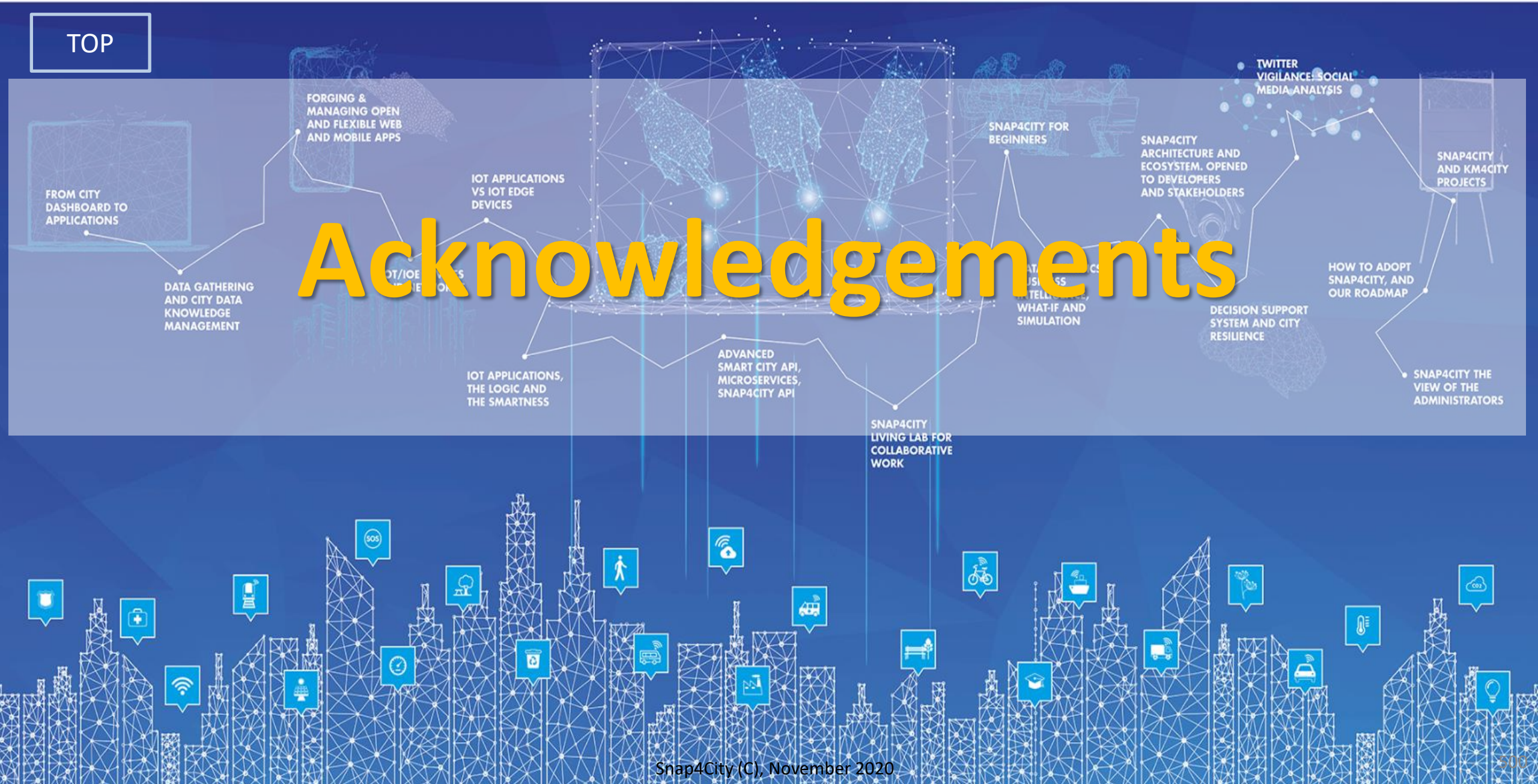
- Is open to the Development of Applications leaving large space to developers
- Is centered on the Orion Broker that result central in the architecture: any Broker or data source is sending data to Orion
- No data shadow at the beginning, only recently they are adding data shadow on IOT Broker
- Security level is not clear, partially demanded to developers
- Visual Flexible IOT processing is not clearly provided
- **Limited API** for IOT data access
- **Knowage BI** presents several limitations in showing Smart City Data
- Market place on Open Data
- Support of Developers via Fi-Ware
- Deployed as VM and Dockers
- open source, not the application parts

- Is open to the Development of Applications leaving large space and providing a **large set of ready to use applicative tools** and solutions to build their solutions on top or aside.
- Is fully distributed, **any kind of data source** can be ingested, automatically to form the Data Shadow.
- **Orion Broker is only one of the Brokers** that can be used. It can be also protected by Snap4City tech, with Mutual Authentication
- **Visual Flexible IOT processing is provided** as IOT App that is Node-RED plus Snap4City MicroServices suites
- **Advanced Smart City API** are provided on top of Knowledge Base
- **Dashboard Builder** has been designed for Smart City Data and automated dashboards' production
- **Market Place** for promoting, publishing and sharing Open Data, tools, processes, experiences
- Full Support for Living Lab of the city, coworking, tutorials
- Deployed as VM and Dockers, on cloud and on premise
- Fully support for Multi-tenacy
- 100% open Source

- **Snap4City is an official Fi-Ware Solution via**
  - NGSI V1, V2 The IOT Orion Broker
  - IOT Orion Broker can connect JSON, MQTT, Lightweight M2M, LoraWAN, OPC, SigFOX, etc. see Fi-Ware Https://www.fiware.org
- **Snap4City is compatible** with all the above protocols
  - via IOT Orion Broker,
  - via IOT Applications.
  - via direct connection on ETL processes on their corresponding IOT brokers, and/or
- **Snap4City is also compatible** with many other protocols, see the table reported in page: https://www.snap4city.org/65

- In Snap4City you can chose to connect your devices at Snap4City Platform in different manners:
  - (a) directly to Snap4City with some Broker, or on IOT App, Brokers, MyKPI
  - (b) via an IOT Orion Broker (external IOT Broker or those provided by Snap4City), or
  - (c) via any third party IOT Brokers in any protocol you have.
- **Snap4City has**
  - **Improved IOT Orion Broker** with the so called Orion Broker Filter (Orion Broker Filter, NGSI Security Wrapper) which is a secure wrapper for NGSI V1 and V2 protocol for enforcing Mutual Authentication, Security, roles, etc.
  - **Produced open hardware and open software NGSI Compliant:** as
    - **IOT Devices** with mutual authentication and security based for NGSI on: Android, Arduino and ESP32, IOT Button, etc.
    - **IOT Edge** devices with mutual authentication and security based for NGSI on: Raspberry PI, Windows, Linux.

TOP



Acknowledgements

DISIT lab roadmap vs model and tools' usage

2013
Km4City Ontology 1.1

- Tuscany, Road Graph
- Mobility
- culture, tourism
- Events
- Parking
- Services
- Linked open graph

2014
- Weather Forecast
- Real Time Wi-Fi
- Entertainment
- LOD

- Twitter Vigilance
- Social Media Analytics, Sentiment Analysis

Km4City 1.4

2015
Km4City 1.5

- Resilience Decision Support
- Smart First Aid
- User Behaviour Analysis, predictions
- Risk Analysis

ckan, resolute

H2020 (2015-18)

2016
(2016-21) H2020 REPLICATE

- Smart Energy
- Sustainable Mobility
- Control Room
- Dashboard

Km4City 1.6.2

Sii-Mobility (2016-21) MIUR

SII-MOBILITY SCN

- Infomobility
- Mobile App
- Routing
- Multimodality

FIWARE

2017
Km4City 1.6.4

Sii-Mobility

- Origin-Destination and trajectories
- Traffic Reconstruction
- Offer Analysis
- OBU, smart devices

E0l5 digital ecosystem (Powered by EXPO MILANO 2015)

(2016-19) MIUR

GHOST SIR

- Sardinia Region Smart City Strategies and plan

GREEN IMPACT C.A.P.A.C.I.T.Y.

GREEN IMPACT
POR FESR 2014-2020

- Industry 4.0
- Critical Plant
- Monitoring

- IOT/IOE, IOT App
- Living Lab
- Maker Support
- IOT Edge
- Smart City IOT
- GDPR,
- Privacy & Security

life weee (2017-20)

- Smart Waste

2018
MOSAiC MObility 4.0 for SmArt (i) City (2018-20)

- Mobility Demand / Offer Analytics and Strategy

Sii-Mobility

- User engagement
- Bike Sharing
- Data Analytics ++
- Social Predictions
- OBD2

IOT/IOE

Km4City 1.6.6

Node-RED

SNAP4CITY

SELECT for Cities H2020 (2017-19)

2019
5G (2018-21)

- 5G tech
- Energy
- Industry 4.0
- Synoptics

TRAFAIR

CEF (2018-21)

- Traffic and Mobility Impact on Pollution
- NOX predictions

EUROPEAN OPEN SCIENCE CLOUD

SIRIUS

LAID Smart Bed

- Smart Health

GREEN FIELD PEAS Soda4.0 (NaOH) Optimization and Automated Decision System 4.0

- Industry 4.0

bee smart city

- Smart Lonato

Reverberi Enetec Gruppo MPES

- PCP Award



2020
…2022

2021
- CAPELON
- Sweden

smartGARDAlake

Interreg MARITTIMO-IT FR-MARITIME MOBI MART

- Smart Mobility
- PISA, PUMS
- Living lab

Interreg Mediterranean HERIT-DATA

- Smart Tourism
- 6 Pilots
- Data Analytics
- Extended platform

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

Main running instances

- Sii-Mobility → mobility and transport, sustainability
- REPLICATE → ICT, smart City Control room, Energy, IOT
- RESOLUTE → Resilience, ICT, Big Data
- GHOST → Strategies, smart city
- TRAFAIR → Environment & transport
- MOSAIC → mobility and transport
- WEEE Life → Smart waste, environment
- Smart Garda Lake → Castelnuovo del Garda
- 5G → Industry 4.0 vs SmartCity
- Green Impact → Industry 4.0, Chemical Plant
- SmartBed (laid → smart health
- Green Field Peas (soda) → Industry 4.0, Chemical plant
- MobiMart and PISA Agreement → data aggregation, mobility and transport, Living Lab
- Lonato del Garda → smart parking, environment
- Herit Data → tourism, culture and management
- ISPRA JRC → site management and services
- Capelon (Sweden) → smart light solutions

Acknowledgements

- Thanks to the European Commission for founding. All slides reporting logo of **Snap4City** https://www.snap4city.org **of Select4Cities H2020** are representing tools and research founded by European Commission for the **Select4Cities** project. **Select4Cities** has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation Programme (grant agreement n° 688196)
- **TRAFAIR is a CEF** project. All slides reporting logo of TRAFAIR project are representing tools and research founded by the EC on CEF programme http://trafair.eu/
- Thanks to the European Commission for founding. All slides reporting logo of **REPLICATE H2020** are representing tools and research founded by European Commission for the REPLICATE project. **REPLICATE** has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation Programme (grant agreement n° 691735).
- Thanks to the European Commission for founding. All slides reporting logo of **RESOLUTE H2020** are representing tools and research founded by European Commission for the RESOLUTE project. **RESOLUTE** has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation Programme (grant agreement n° 653460).
- Thanks to the MIUR for co-founding and to the University of Florence and companies involved. All slides reporting logo of **Sii-Mobility** are representing tools and research founded by MIUR for the Sii-Mobility SCN MIUR project.
- **Km4City** *is an open technology and research line of DISIT Lab exploited by a number of projects. Some of the innovative solutions and research issues developed into projects are also compliant and contributing to the Km4City approach and thus are released as open sources and are interoperable, scalable, modular, standard compliant, etc.*

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

DISIT thanks to

Herit Data: Tourism and Mng. https://herit-data.interreg-med.eu/
Snap4City: IOT/IOE smart city www.snap4city.org
Trafair: CEF project with several Cities http://trafair.eu/
Mosaic: Mobility and transport model
Km4City: http://www.km4city.org
REPLICATE H2020, SCC1, EC flagship
http://replicate-project.eu/
Sii-Mobility SCN MIUR: http://www.sii-mobility.org
Feedback: retail and GDO Big Data analytics
5G with 3G-Wind, Open Fiber, Estra
Coll@bora Social Innovation, MIUR:
http://www.disit.org/5479
RESOLUTE H2020, EC:
http://www.resolute-eu.org
TRACE-IT, RAISSS, TESYSRAIL, …
Mobile Emergency:
http://www.disit.org/5404



SNAP4CITY

REPLICATE

resolute

Sii-Mobility

TRAFAIR

GREEN IMPACT C.A.P.A.C.I.T.Y.

LAID Smart Bed

GREEN FIELD PEAS Soda4.0

5G

MOSAiC

Coll@bora

TESYS RAIL



feedback

Trace-IT

RAISSS

Smart COMMUNITIES Tech Italian Technology Cluster for Smart Communities

Regione Toscana

MIUR

Horizon 2020 European Union Funding for Research & Innovation

Further readings

https://www.snap4city.org/108

- HOW TO: create a Dashboard in Snap4City
- HOW TO: add a device to the Snap4City Platform
- HOW TO: add data sources to the Snap4City Platform
- HOW TO: define privacy rules for personal data, produced by the end-users own device
- HOW TO: Develop Smart Applications, Snap4City development Life Cycle
- HOW TO: HLT vs Ingestion, and HLT vs Widgets
- HOW TO: Develop an IOT Application for Data Ingestion
- HOW TO: Upload data into Knowledge Base, ServiceMap (triple upload)
- HOW TO: Create as set of Devices with BulkProcessing
- HOW TO: Create an IOT Device Model
- HOW TO: Create an IOT Device Instance from IOT Directory tool

TOP





Be smart in a SNAP!





CONTACT

DISIT Lab, DINFO: Department of Information Engineering
Università degli Studi di Firenze - School of Engineering

Via S. Marta, 3 - 50139 Firenze, ITALY
https://www.disit.org

www.snap4city.org

Email: snap4city@disit.org

Office: +39-055-2758-515 / 517
Cell: +39-335-566-86-74
Fax.: +39-055-2758570

