SNAP4CITY

KM4CITY

Be smart in a SNAP!

SNAP4APPLIANCE
Virtual Machines ready to use for Smart City and IOT Applications
DOWNLOAD

23 July 2019, 3° Training day of 3, Florence
https://www.snap4city.org/drupal/node/485

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

23 July 2019, 3° Training day of 3, Florence
https://www.snap4city.org/drupal/node/485

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

23 July, Agenda

- **Dashboard Management and GDPR**
  - Dashboard Listing, Managing, Sharing, Delegation
  - Dashboard ChatRoom and Notifications
- **DataType Management GDPR Compliant**
  - Personal Data Management GDPR Compliant: POI, Data, KPI
- **IOT Applications, Devices and Dashboards**
  - Managing IOT Applications
  - Authoring IOT Applications
- **From Simple to Data Processing IOT Applications**
  - Create a Simple IOT Application (Demo)
  - **Production of IOT Application (Exercitation)**
  - Data Processing with IOT Application (Demo)
  - **Processing Data with IOT Applications (Exercitation)**
- **Data Analytics: Examples from Snap4City**
  - Smart parking: Predictions
  - User Behavior Analysis
  - Traffic Flow Reconstruction
  - Modal and Multimodal Routing
  - Environmental Data: Predictions
  - Social Media Analysis: Early Warning, Predictions
- **Data Analytics: Enforcing and Exploiting**
  - Real Time Data Analytics: using R Studio Exploitation in IOT Applications (DEMO)
- **How to Add Functions that are not present in the Platform**
- **Development of Low Cost Sensors and Actuators**
  - Proprietary IOT Devices as Well as Open hardware / Open Software
- **IOT end-2-end Secure Stack**

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

Self Training main path

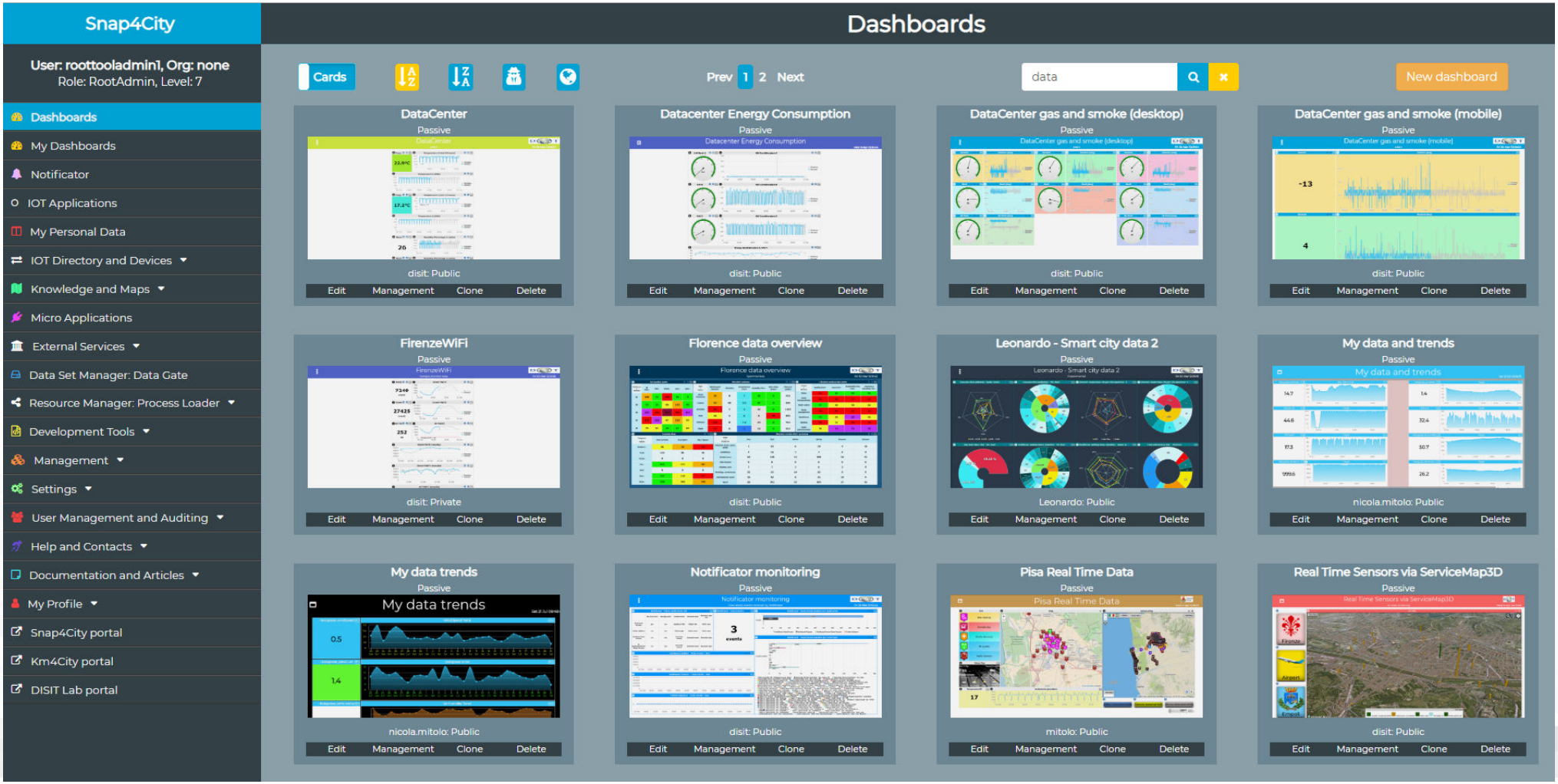
- **Please start a fully guided training cases:**
  - HOW TO: create a Dashboard in Snap4City
  - HOW TO: add a device to the Snap4City Platform
  - HOW TO: add data sources to the Snap4City Platform
  - HOW TO: define privacy rules for personal data, produced by the end-users own device

SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT CITIES

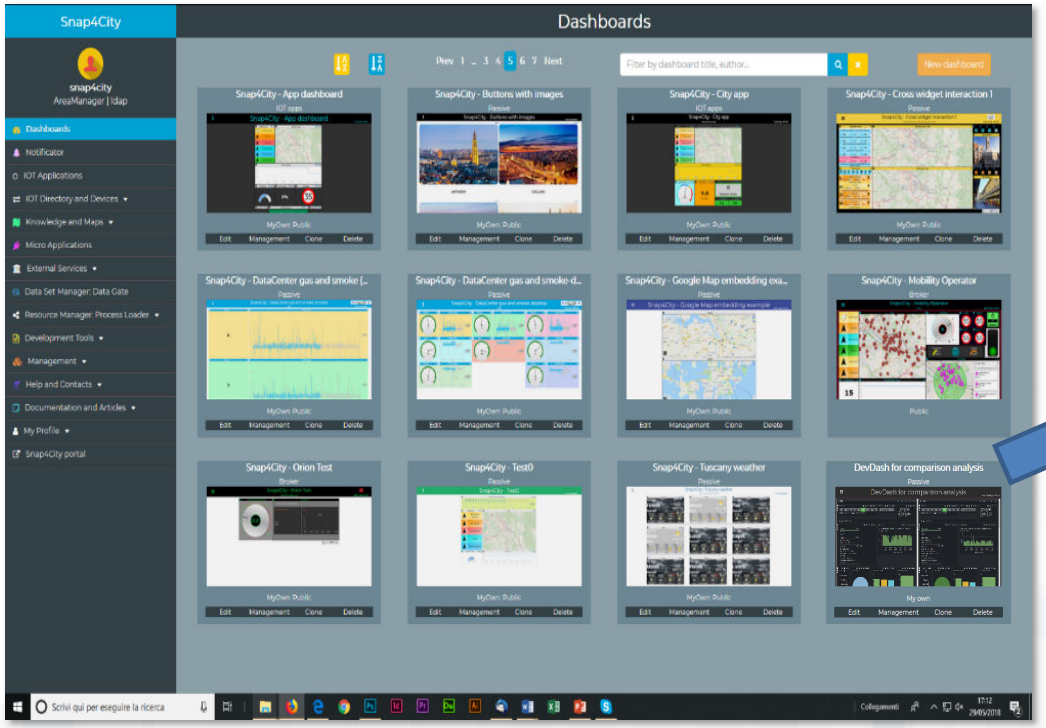
Dashboard Management and GDPR

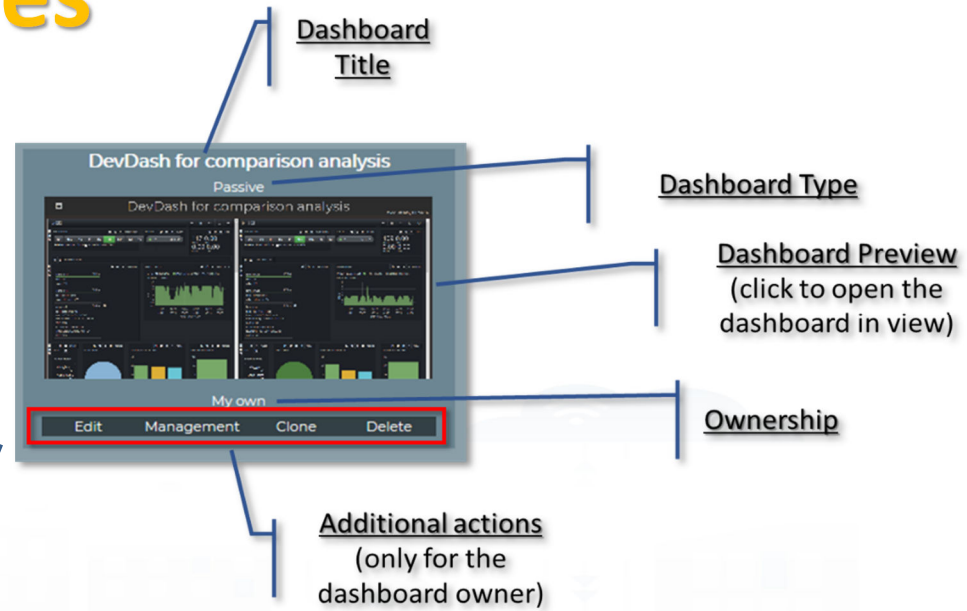
Dashboards Listing, Managing, Sharing, Delegation

Dashboard List and Editor



Dashboard Listing and Features

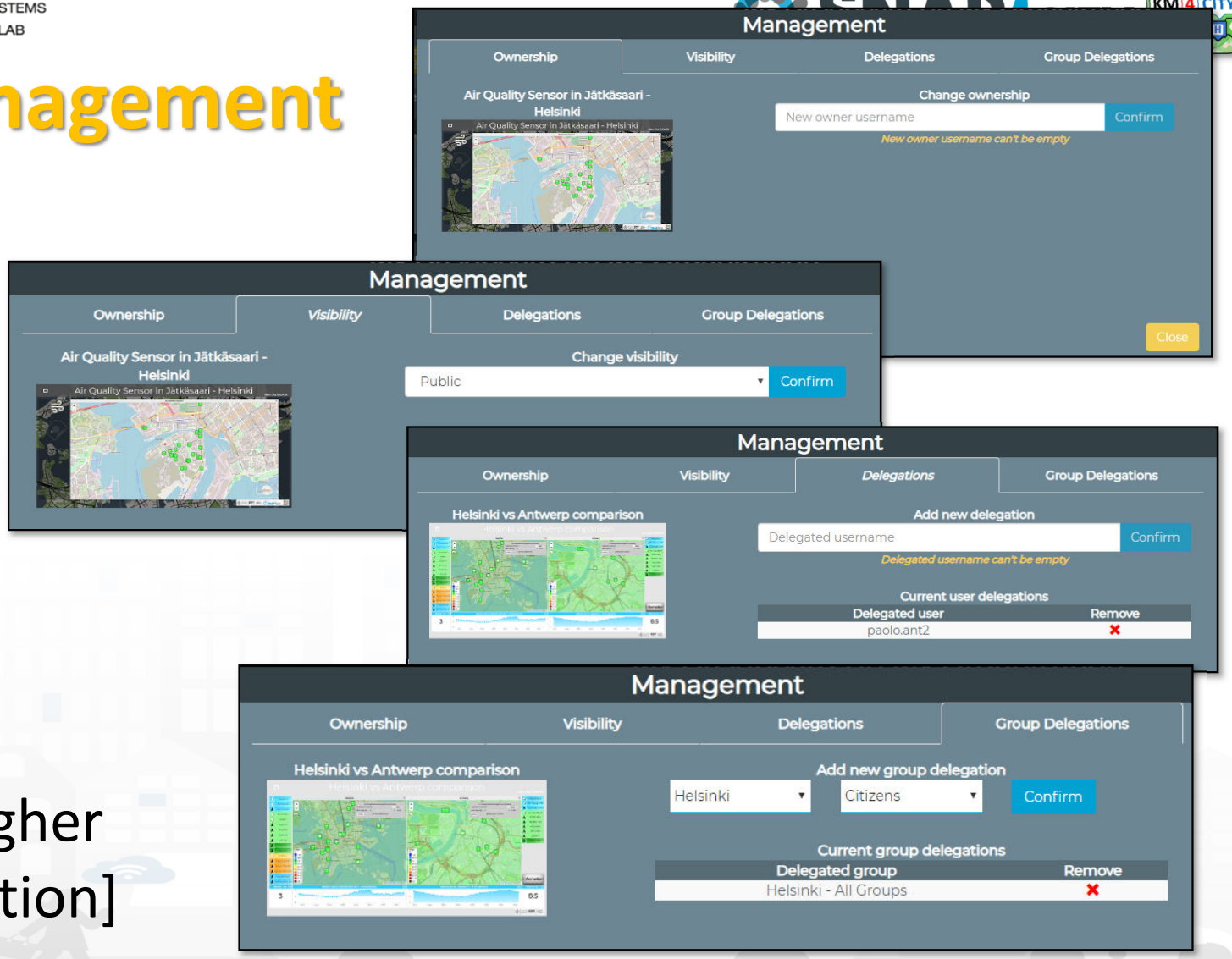




- Edit
- Management
- Clone
- Delete
- (add Thumbnail is inside Edit)

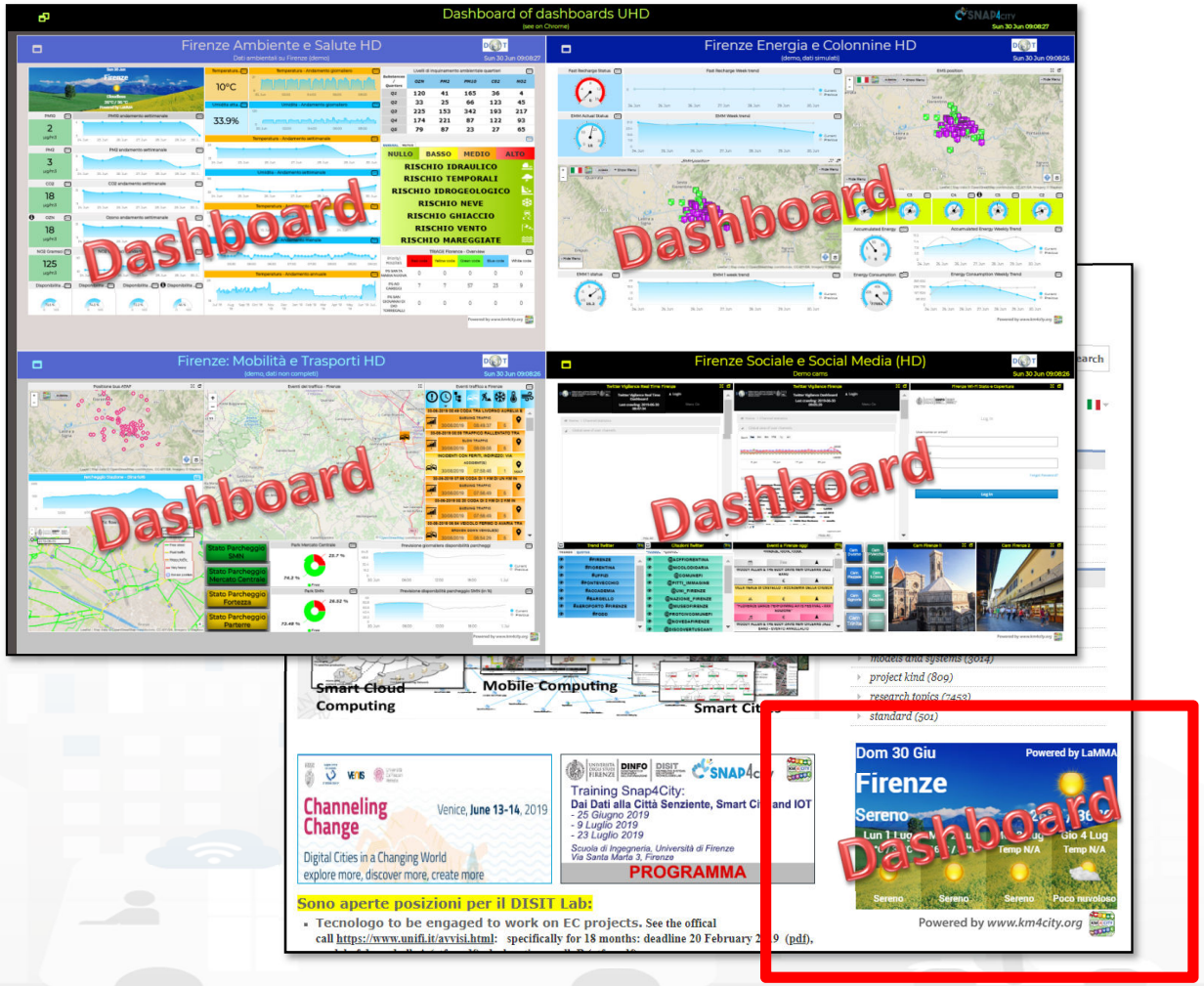
Dashboard Management

- Change Ownership
- Public or Private
- Delegation access to other users
- Delegation Access to other Groups [Higher roles cross Organization]



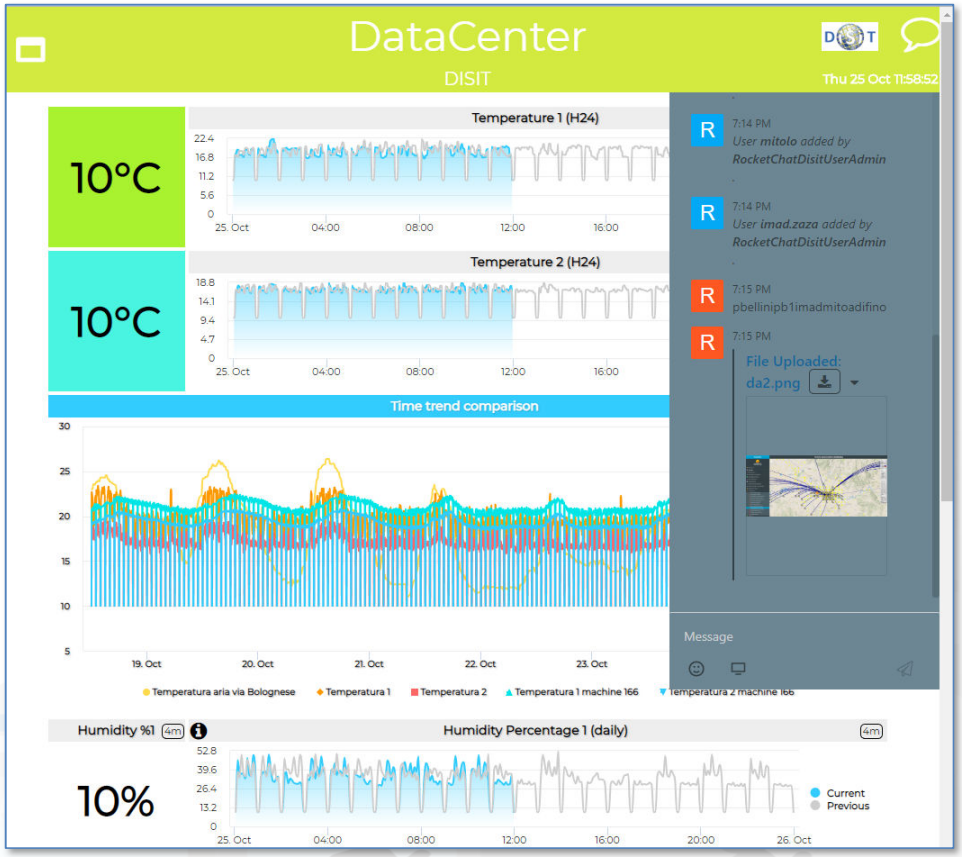
Additional Properties from Edit

- Embedding Dashboards into
  - a Dashboard
  - third Party Web Page
- Header or not
- Responsive or not
- Size
- Background Image
- Add / change Screenshot (Thumbnails)
- …etc..

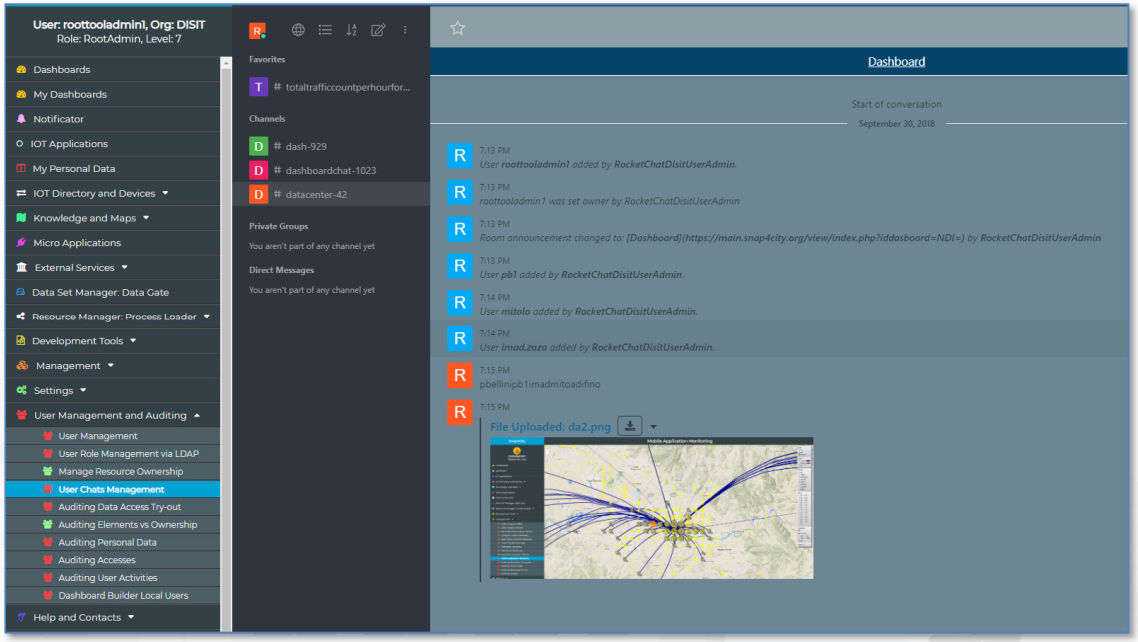


Dashboard Chatroom and Notifications

ChatRoom Per Dashboard



Chat Management



Dashboard's Chat Rooms

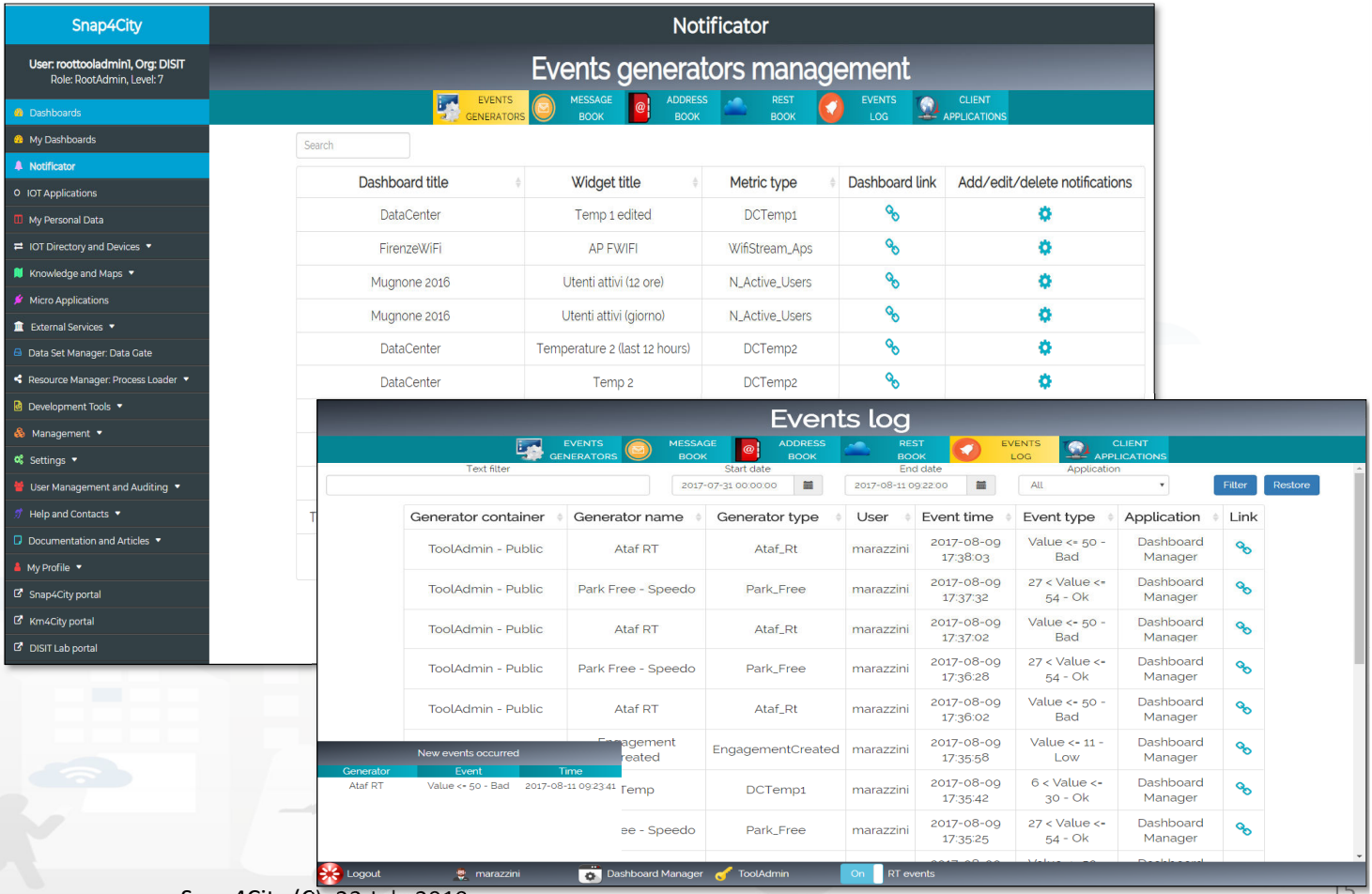
- Each Dashboard may have only one separate ChatRoom
- The Dashboard Owner can
  - Activate the Chat Room on Dashboard header in Edit
  - Add a number of users platform to chat room
- The Chat Room
  - Allows to Exchange Comments and Pictures
  - Can be Accessed on web and mobile
  - May Provoke notifications on the header of the Dashboard
  - Is accessible only under authentication
- The Administrators can access to the Log for review of the discussions

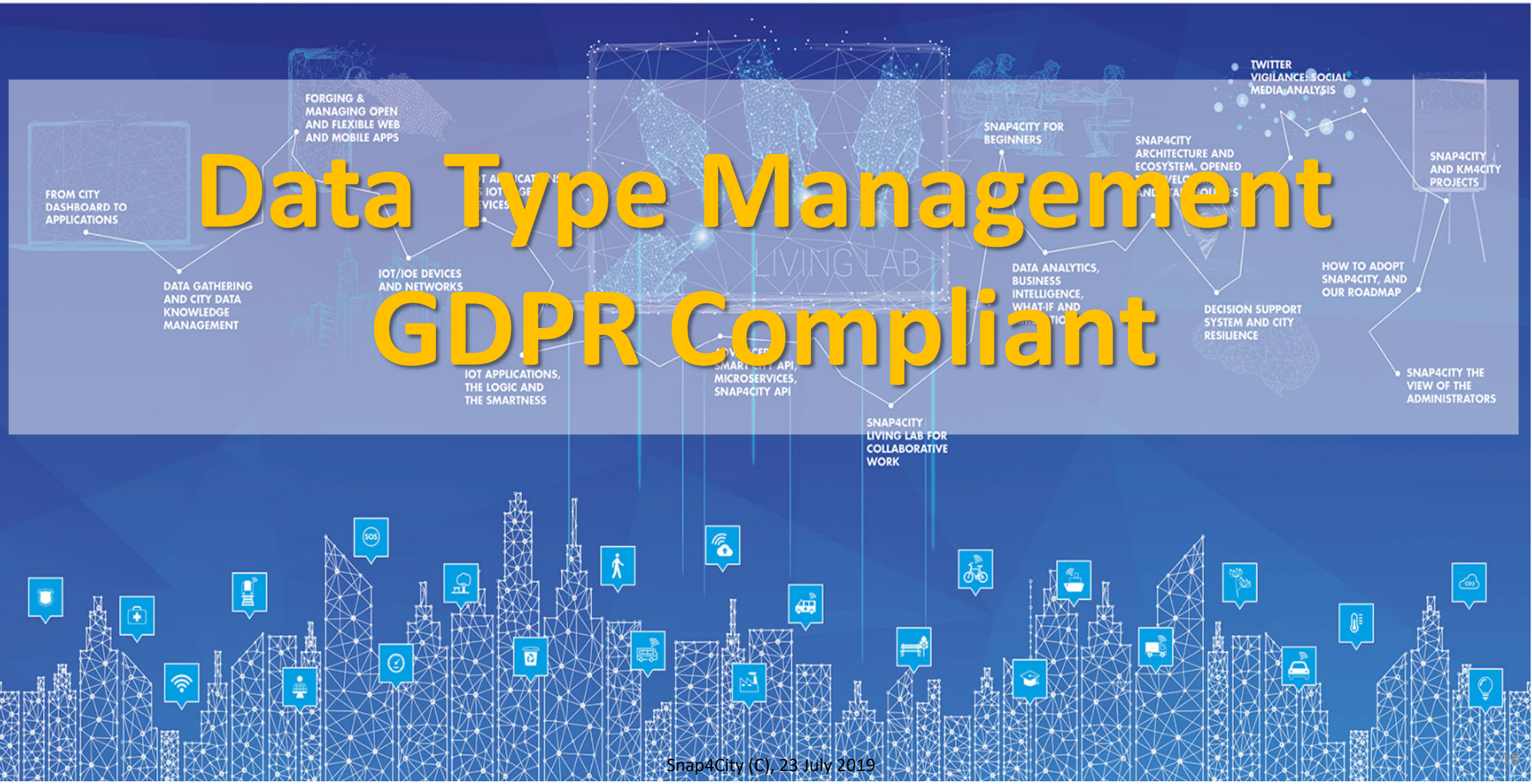
Smart City Monitoring: Notificator

Associated with metrics shown on Widgets

Alternative with IOT App

- Notifications may arrive via Facebook, Telegram, SMS, email, etc., by exploiting IOT App behind the dashboard
- Integration with workflow management system for ticketing





Data Type Management
GDPR Compliant

GDPR: General Data Protection Regulation

Users may decide to:

- provide access to who, for do what, until when consented
- accept terms of use by **signed** consent **for** data management **service**

From each service, the user is capable to:

- **See** what we collect in terms of Data Type: traces, logs, paths, profiles, accesses, IOT devices, sensors, maps, etc.
- **Download**, **delete**, **inspect** Data
- **Auditing** and **Revoke access** or **grant** access right to each **single Data**
- **Delete all Data in single shot** or singularly (**forget all about me**)

- Correctness
- Transparency
- Security
- Integrity
- Privacy
- Auditing
- …

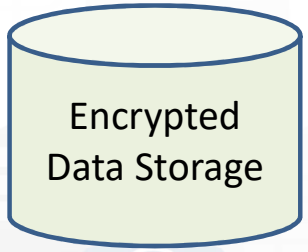
GDPR: General Data Protection Regulation

If personal data are **published by the owner:**

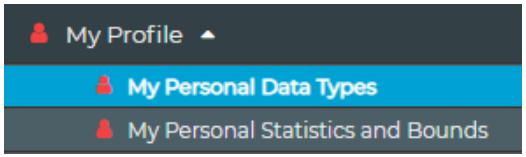
- the data are **released anonymously**,
  → also in this case they can be **revoked at any time**:

Snap4City is also compliant to GDPR **Technical Constraints** as it:

- **Performs Secure connections** in any private data exchange
- **Encrypts** data store for all private data
- **Decouples** data and personal IDs
- **Audits** private data usage



GDPR Compliant



My Personal Data Types

View | Edit | Track | Access control | Convert

This page allows you to access at your Data Types, which are your personal data that we c
most cases, a specific tool and view is provided to manage them.

- My profile data and Blogs
  - to manage your user profile data (name, email,): view, edit, delete
- My Personal Statistics and Bounds: daily or Monthly
  - to access at your statistics about the data access and volume of resources use
    that may depend on the Organization at which one belong and on the role in
- My Personal Data, My KPI and My POI
  - to manage your personal MyKPI, MyPOI and trajectories, if any: view, edit, dele
- My Personal Engagement
  - to manage your personal engagements recevied on the Mobile Apps, auditing
- My IOT Devices
  - to manage your IOT Devices in which it is possible to: edit, delete, make public
- My IOT Applications
  - to manage your IOT Applications in which it is possible to: delete, restart, char
- My Dashboards
  - to manage your Dashboards in which it is possible to: edit, delete, change owr
- My IOT sensor data service URI (for programmers)
  - to manage the Delegations to access at the ServiceURI of the knowledge base
- My IOT sensor data service GraphID (for programmers)
  - to manage the Delegations to access at the a Graph (data set) of the knowled
- My personal data by IOT App (partially deprecated)
  - to manage your MyPersonal Data, if any: view, edit, delete, delegation in acces
- My Annotation data
  - to manage the Delegation to access at the Annotations: delegation in access,
- Auditing Access to My Data
  - to audit the accesses to MyData

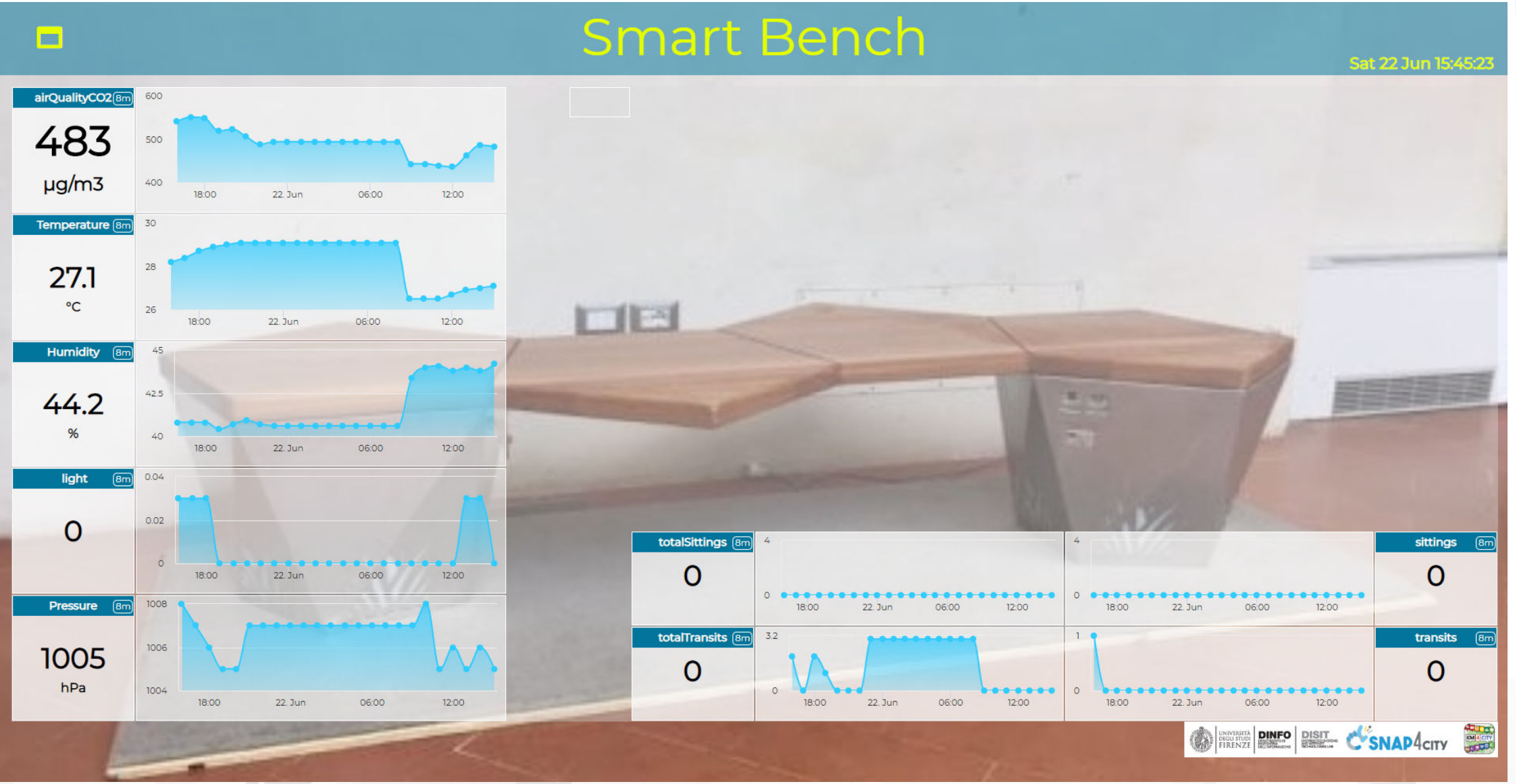
Manage Profile and MyPersonalData

For each Data Type:

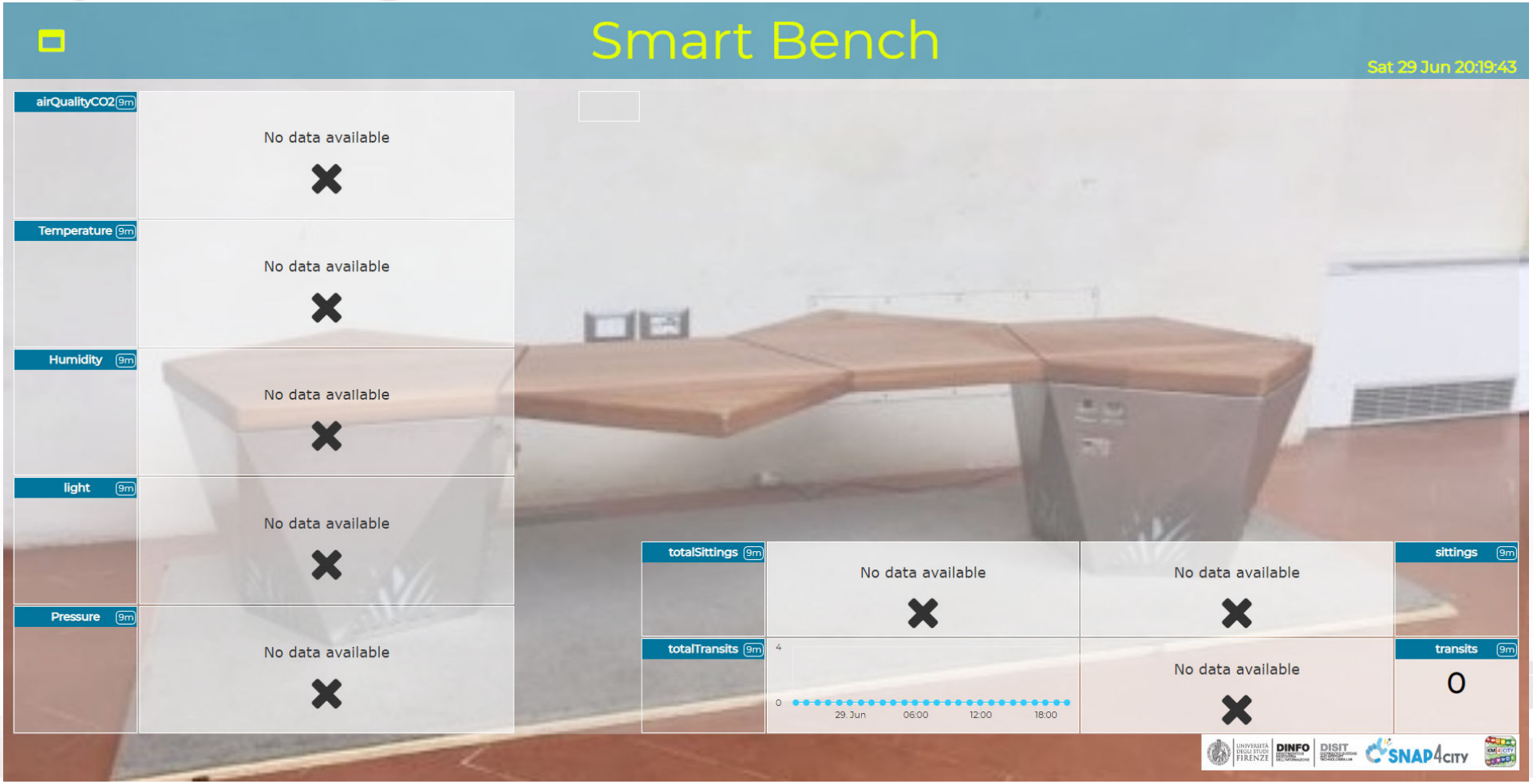
- Start as private → making them public (anonymous) and revoke
- The Owner is the only one that can: (1) modify values; (2) change the ownership
- Define/revoke Delegation to Access
- Delete/forget per Data Type and “me all”!
- Auditing

Details for Main Data Kinds

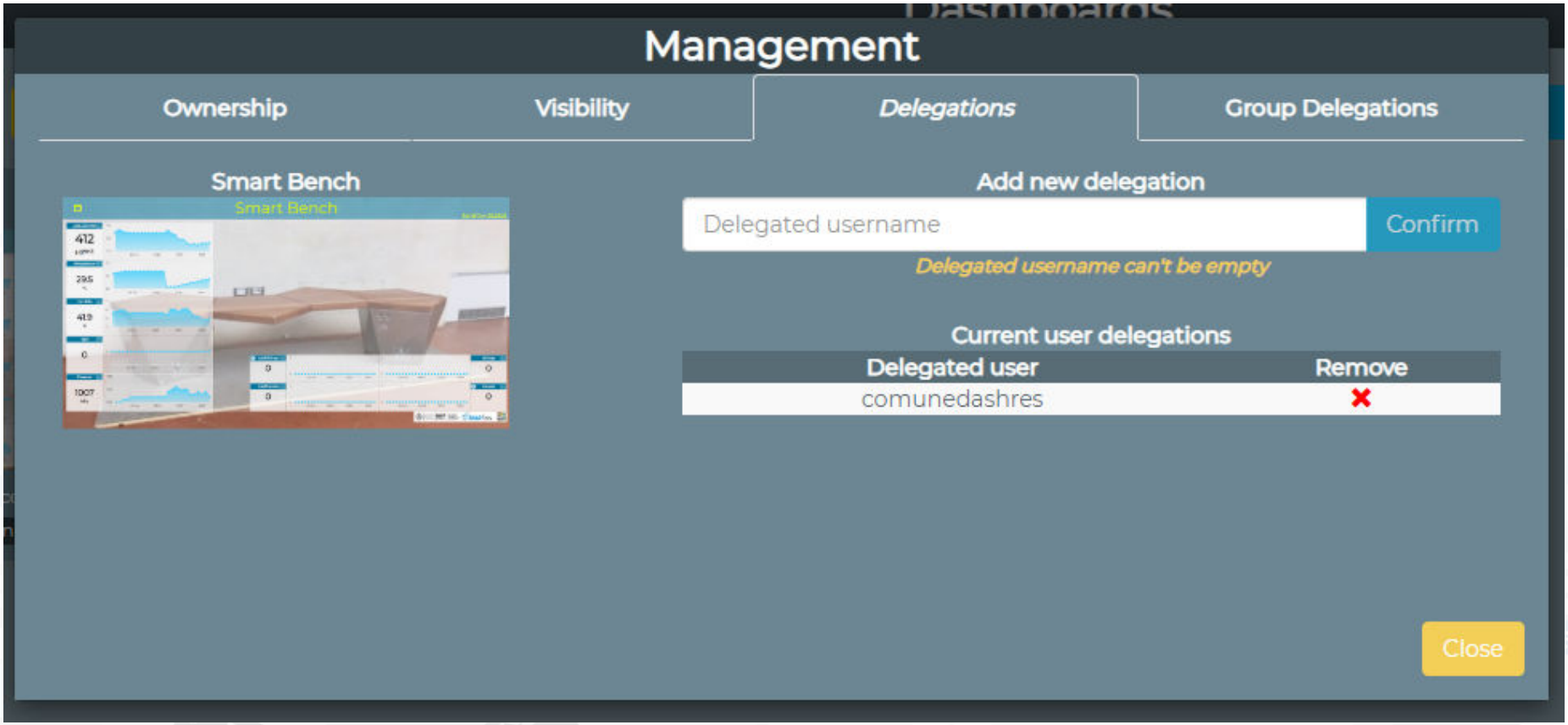
- **My Personal Data, My KPI and My POI**
  - to manage your personal MyKPI, MyPOI and trajectories, if any: view, edit, delete, delegation in access, revoke delegation, make public, change ownership
- **My Personal Engagement**
  - to manage your personal engagements received on the Mobile Apps, auditing, if any: view, delete
- **My IOT Devices**
  - to manage your IOT Devices in which it is possible to: edit, delete, make public, delegate in access, revoke delegation, change ownership
- **My IOT Applications**
  - to manage your IOT Applications in which it is possible to: delete, restart, change ownership.
- **My Dashboards**
  - to manage your Dashboards in which it is possible to: edit, delete, change ownership, delegate in access, revoke delegation, see list of delegations, make public.


Smart Bench
Sat 22 Jun 15:45:23
airQualityCO2 8m
483
µg/m3
600
500
400
18:00
22. Jun
06:00
12:00
Temperature 8m
27.1
°C
30
28
26
Humidity 8m
44.2
%
45
42.5
40
light 8m
0
0.04
0.02
0
Pressure 8m
1005
hPa
1008
1006
1004
totalSittings 8m
0
totalTransits 8m
0
sittings 8m
0
transits 8m
0
3.2
1
SNAP4CITY

Example: Delegated Dashboard but not all data



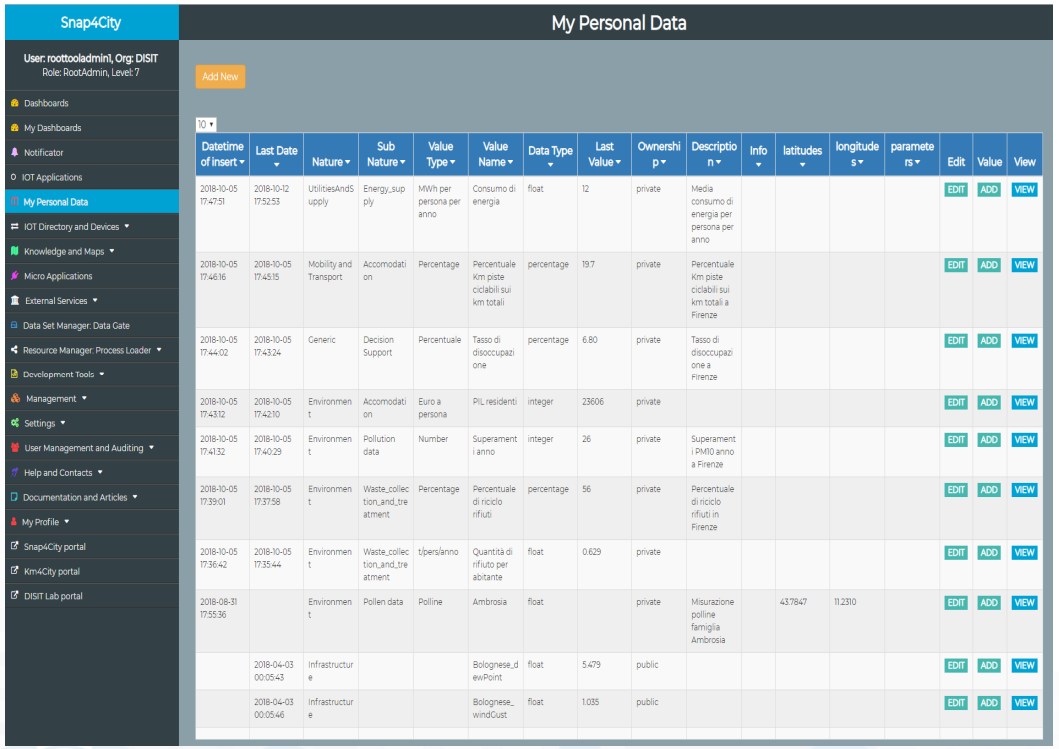
Dashboard Delegation



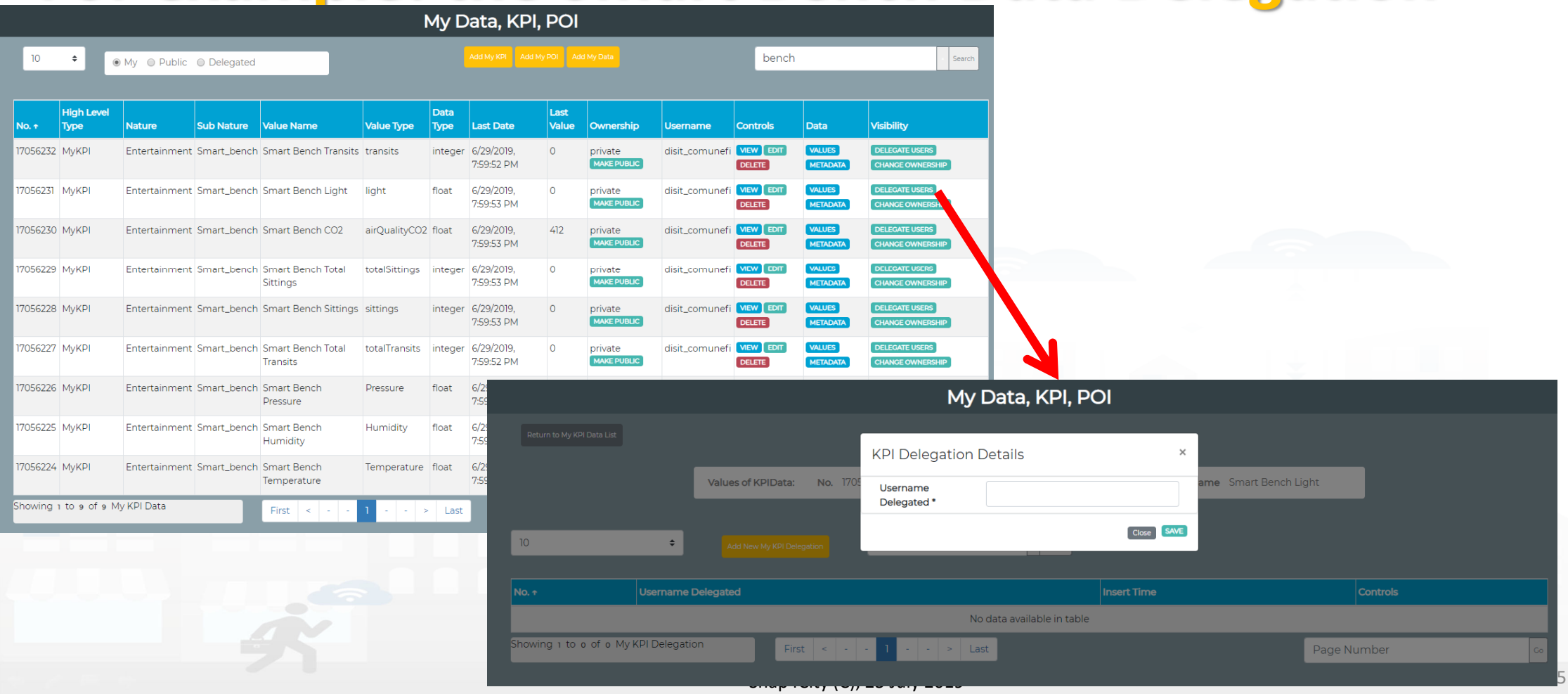
Managing Personal: KPI, Data, POIs

Which can be:

- **Acquired from any source:** SQL, SPARQL, API, etc., ODBC, JDBC, etc.
- **Computed** by means of IOT App, Data Analytics, ETL
- **Stored/retrieved** into personal safe or general storage
- **Edited** from tool, or from IOT App
- **Added** Manually or Automatically
- **Shown via Dashboard**



For example: the Smart Bench Data Delegation



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

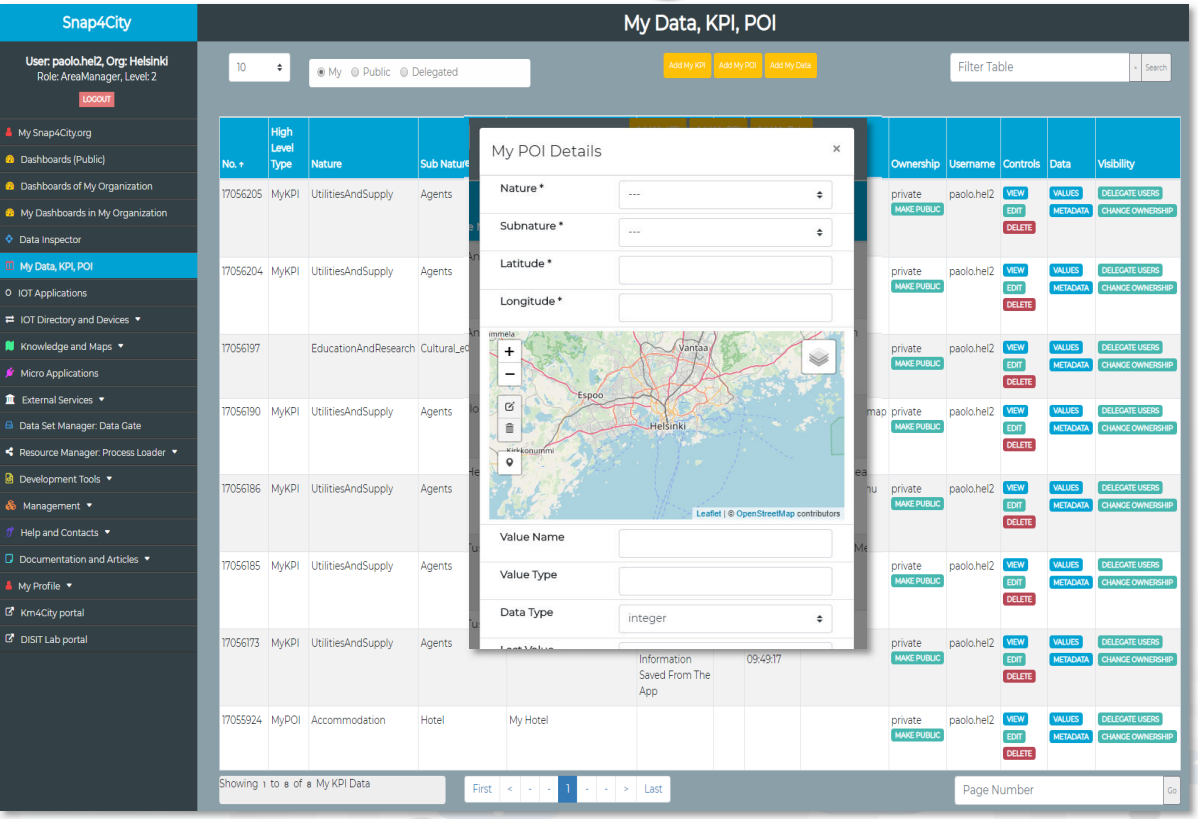
SNAP4CITY KM4CITY

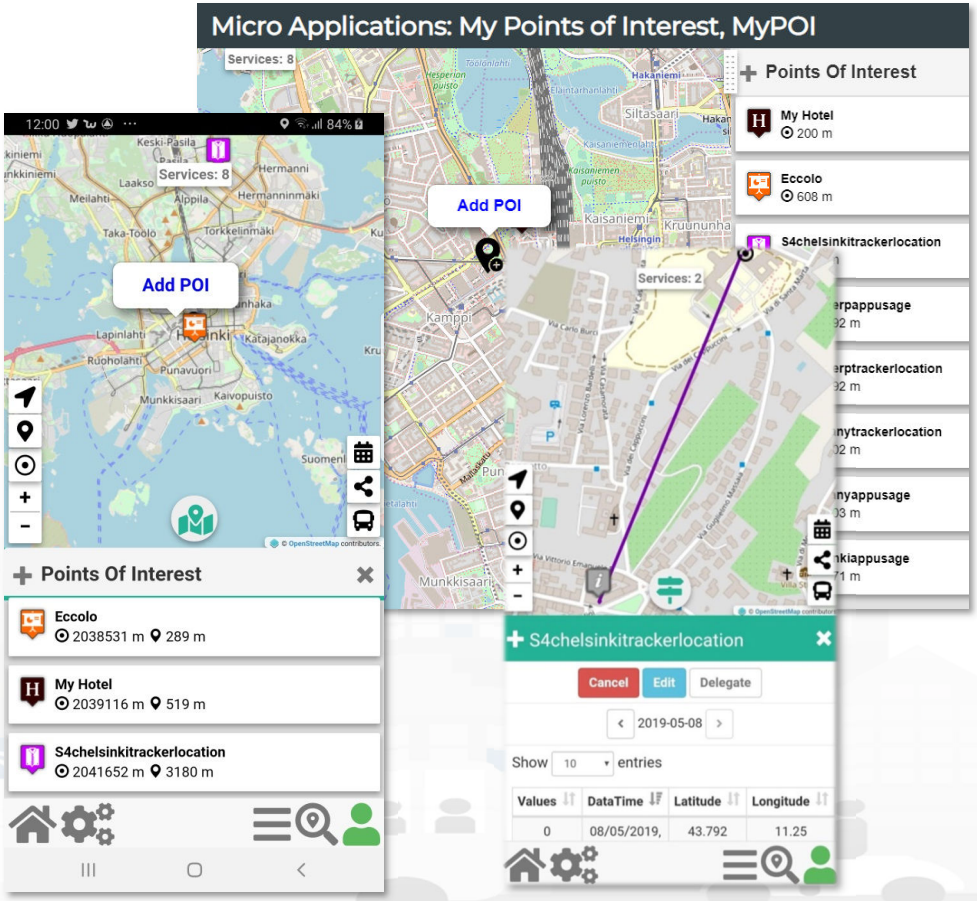
Personal Data Management, GDPR compliant: POI, Data, KPI

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

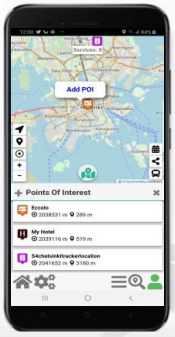
SNAP4CITY KM4CITY

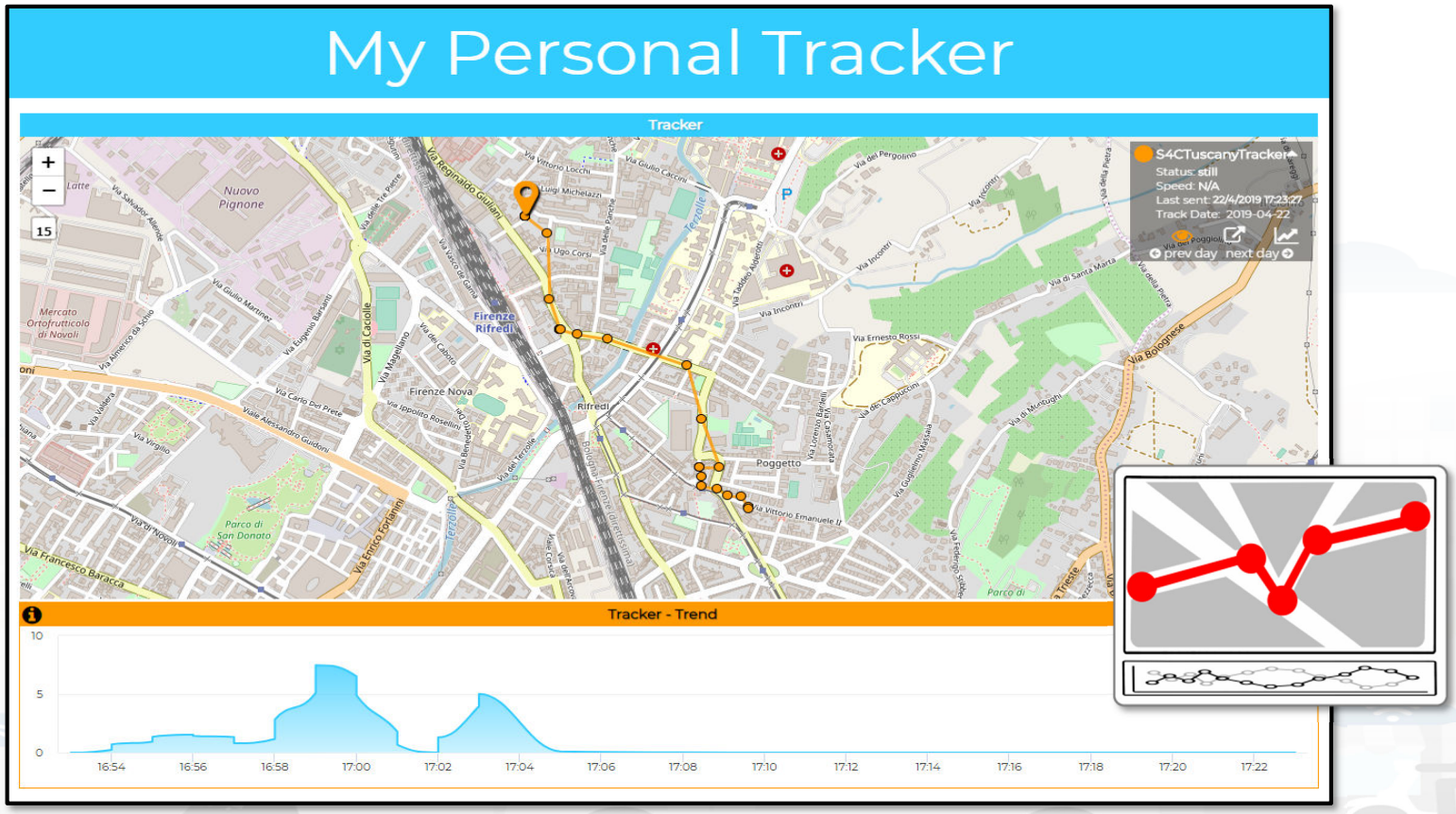
Management of MyKPI, MyPOI, …



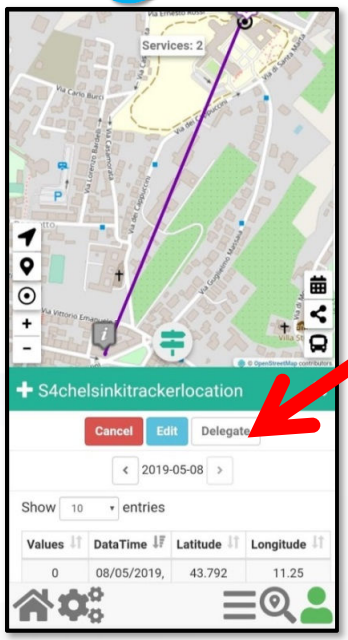


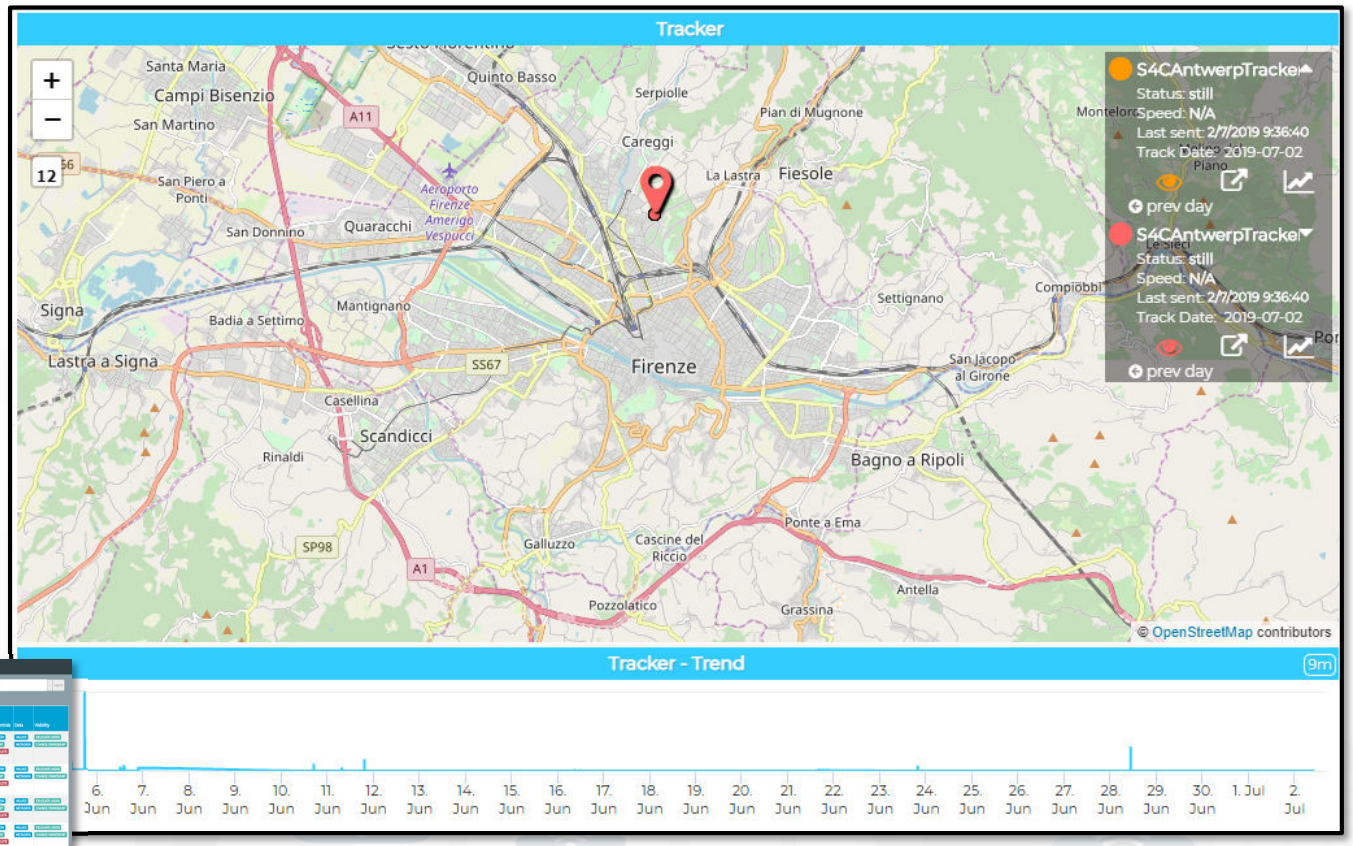
Recalling Exercize number 3

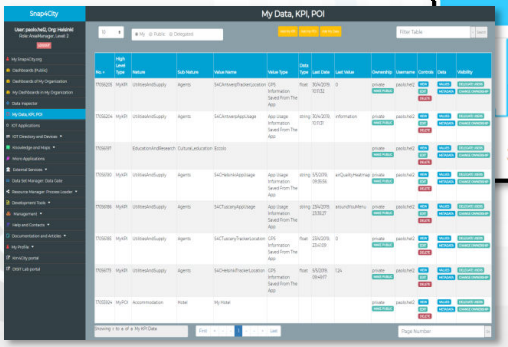




Accessing Multiple Trajectories

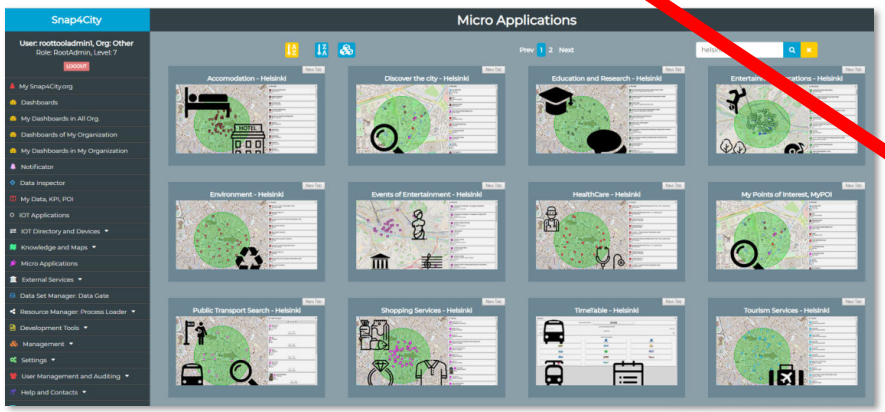


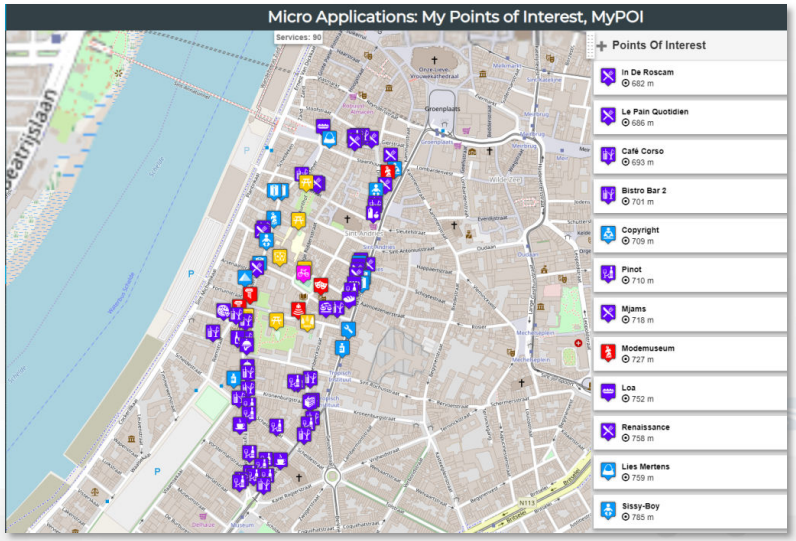


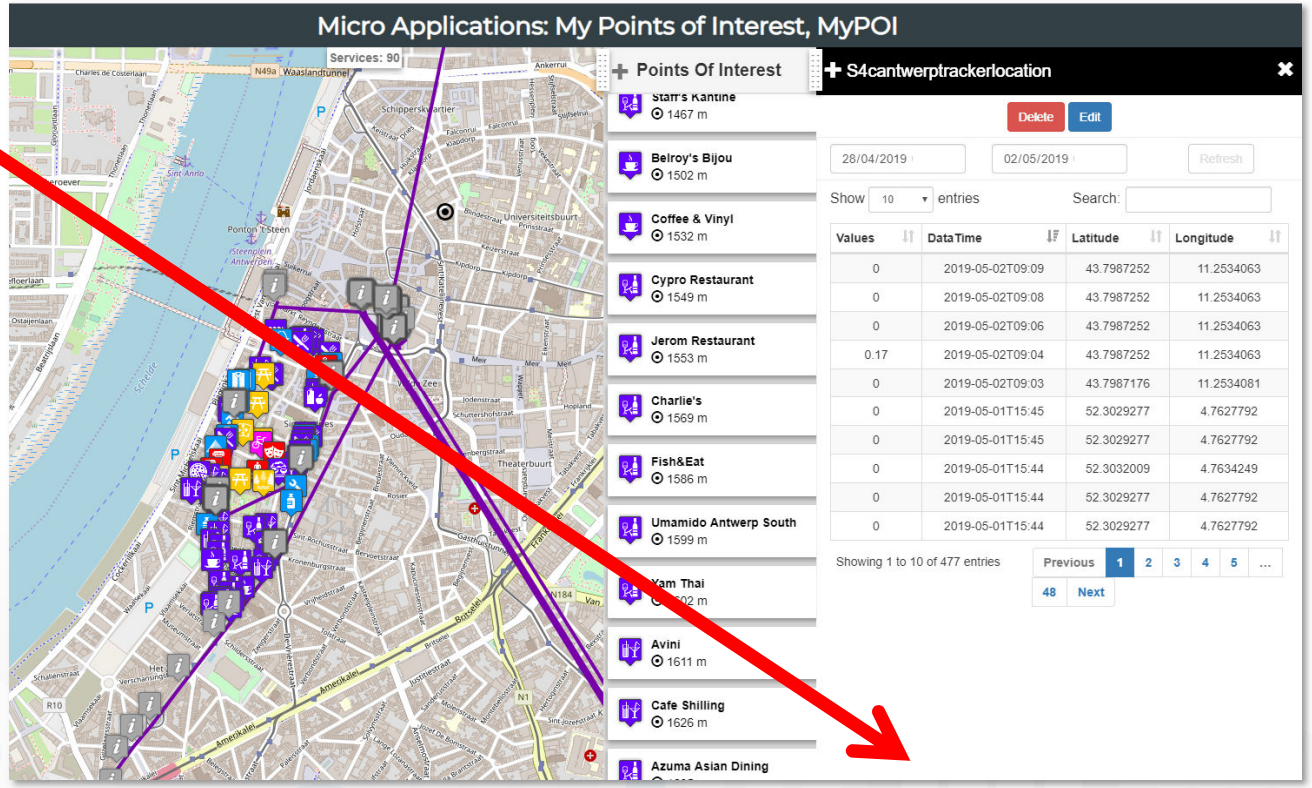


You can revoke at any instant from Web and mobile App

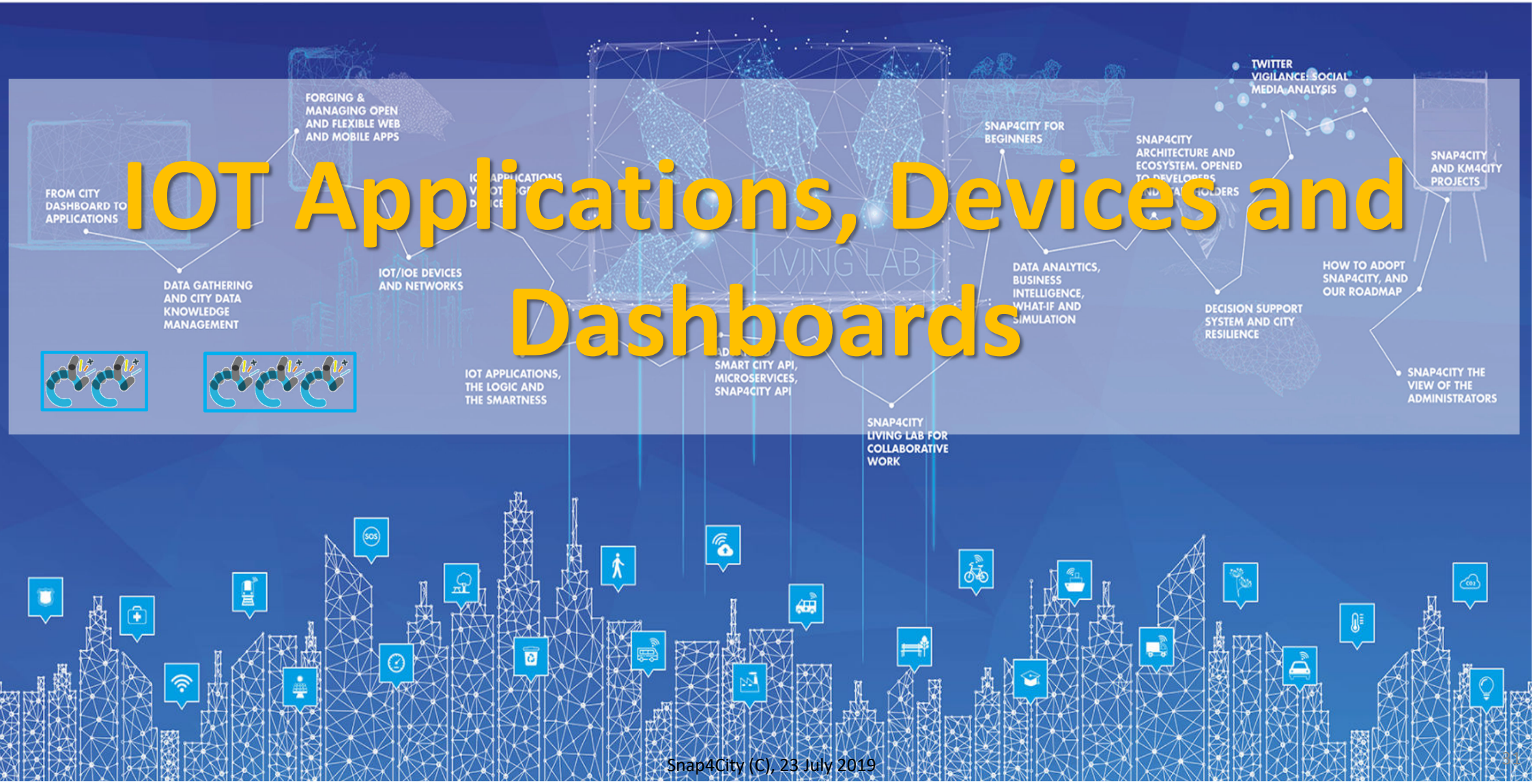
MicroApplications: several new







- Personal POI, KPI, Trajectories/trips

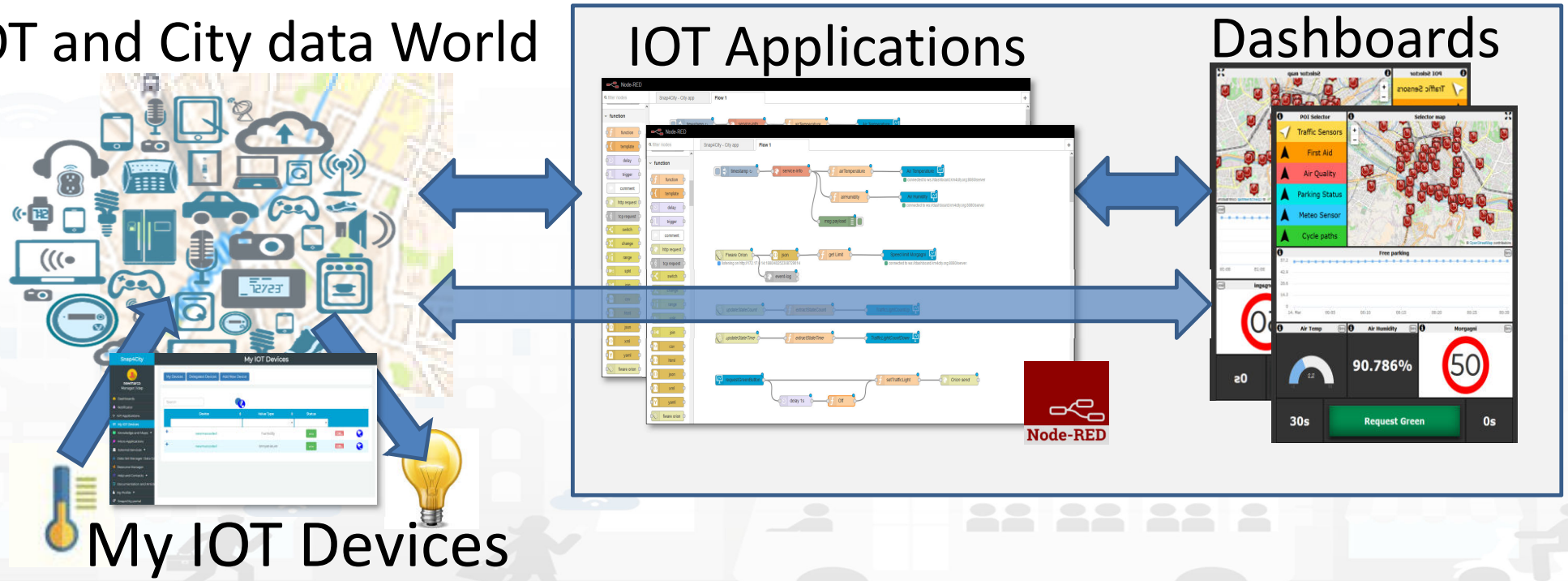


IOT Applications, Devices and Dashboards

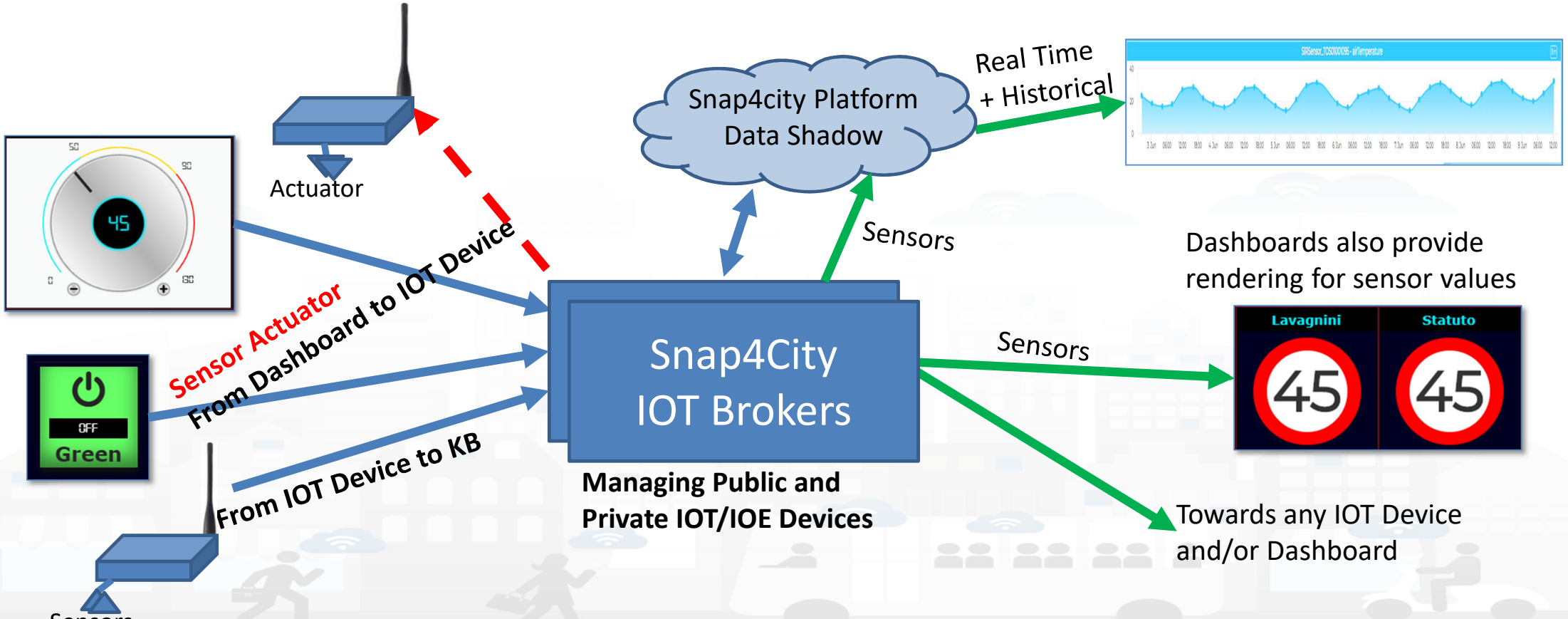
Dashboard with intelligence App

Dashboards with IOT Applications for enforcing smart and intelligence into them

Dashboard-IOT App



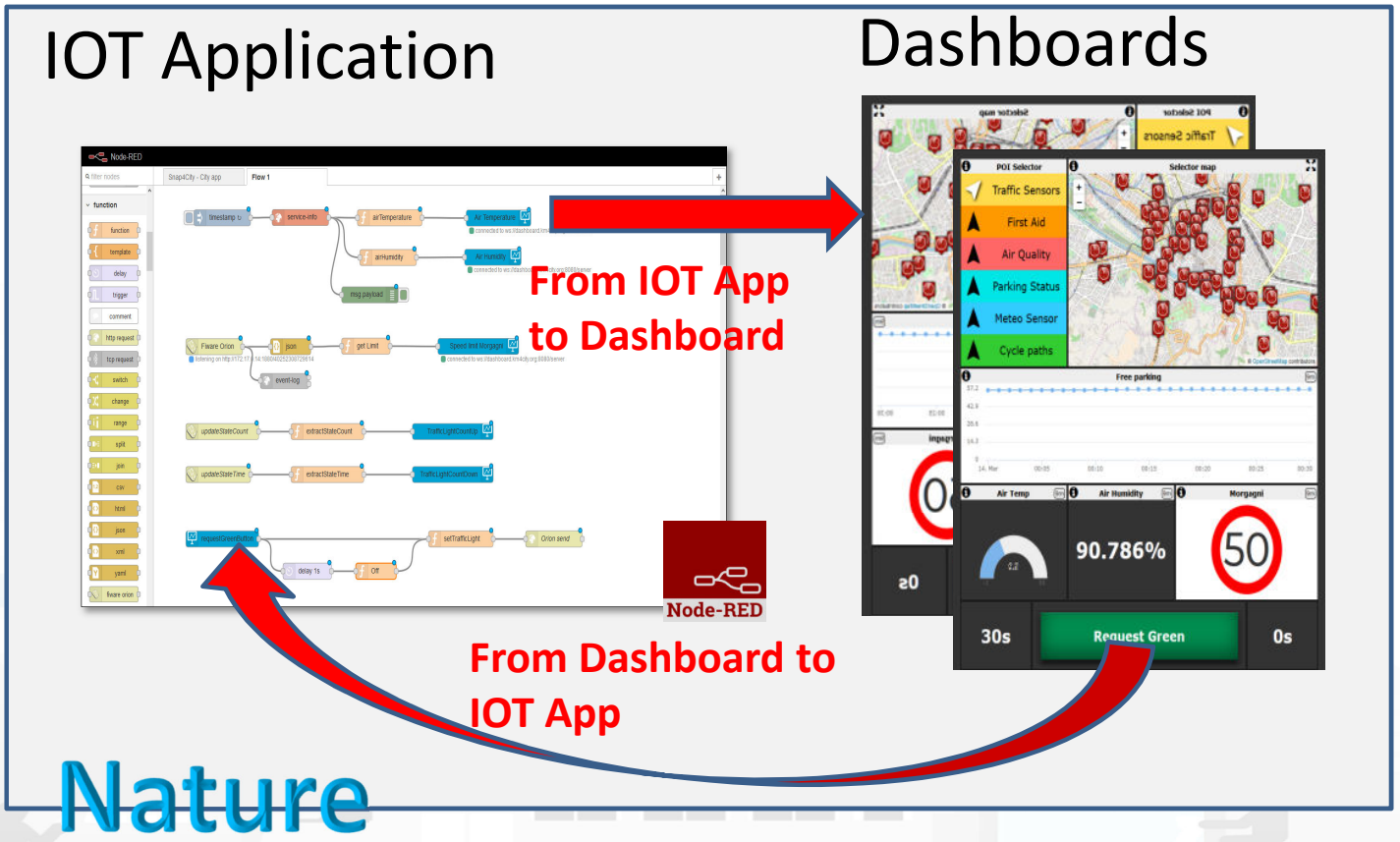
IOT Data Driven



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)



IOT/IOE on the field

On Cloud

IOT Directory

Other Connected IOT App

(1) Registration

(0) Sensors & Actuators

Raspberry pi

Sensors/ Actuators

Sensors/ Actuators

Sensors/ Actuators

Sensors/ Actuators

Sensors/ Actuators

-- PC: Win, Linux

Internet

(2) Production of IOT App on IOT edge

IOT Edge With IOT App distributed

(4) IOT App and its Dashboard are executed

(3) Production of Dashboard user interface

IOT Button

IOT Brokers

Mobiles

IOE Devices

Sensors/Actuators

IOT App can be executed on IOT Edge and/or on Cloud. MicroServices calling Dashboards, Storage and Analytics are executed on Cloud.

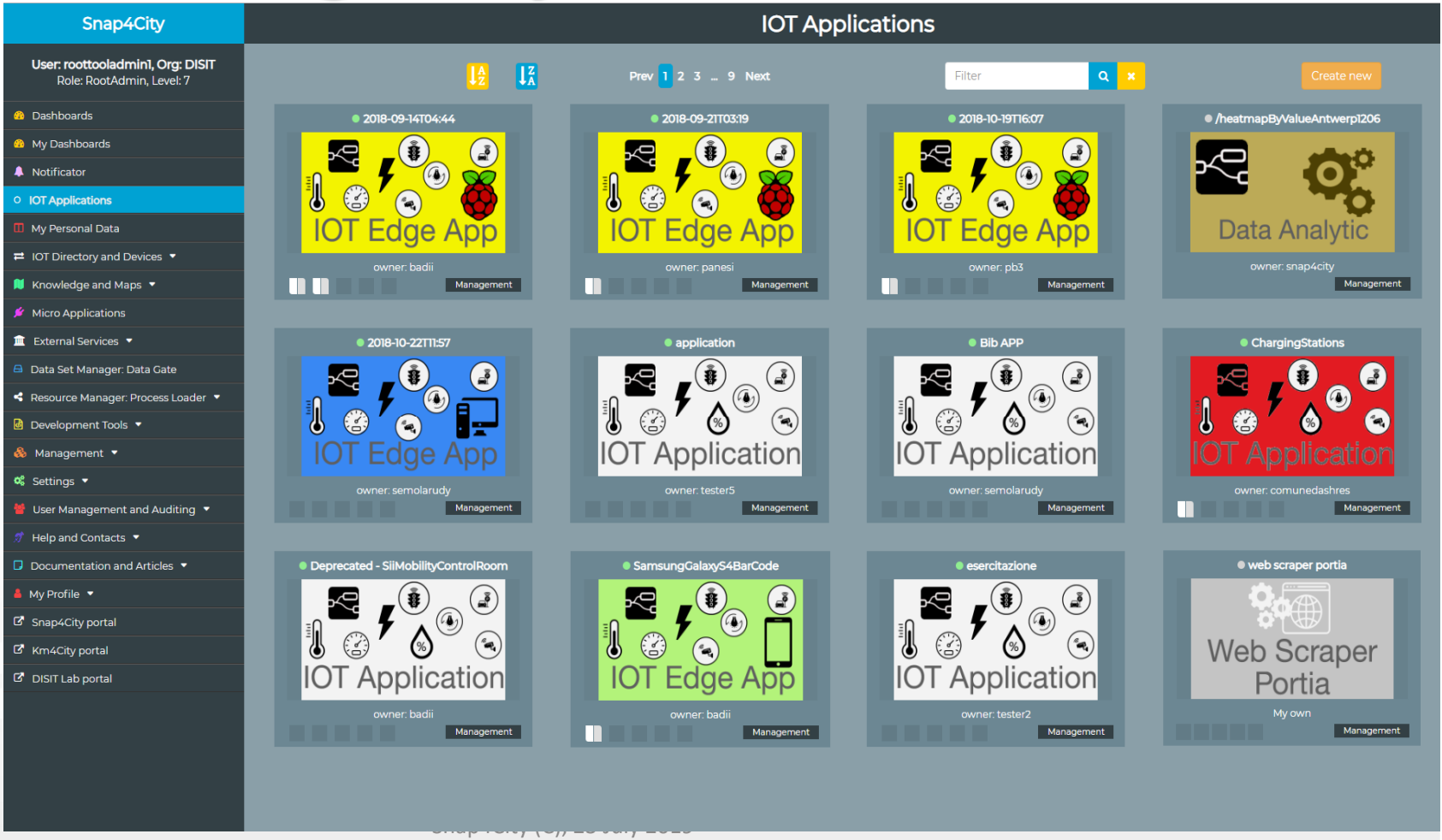
Managing IOT Applications

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

SNAP4CITY KM4CITY

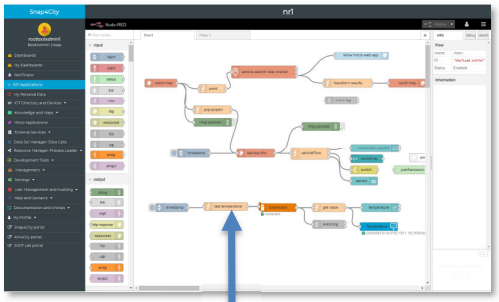
IOT Application Listing, they can be

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

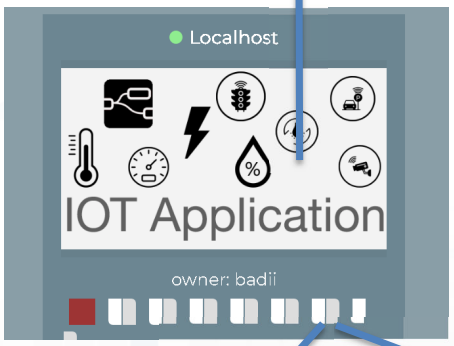


IOT Applications Listing

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

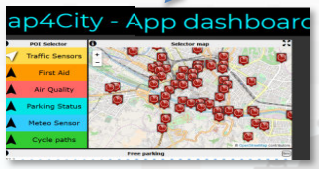


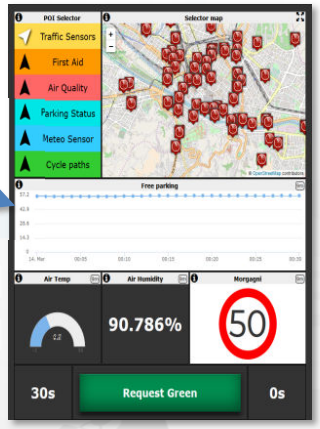
EDIT IOT APP



VIEW

EDIT





IOT App title

Ownership of the IOT App

Click the icon to edit the IOT App

Click to open the Node-RED IOT App dashboard

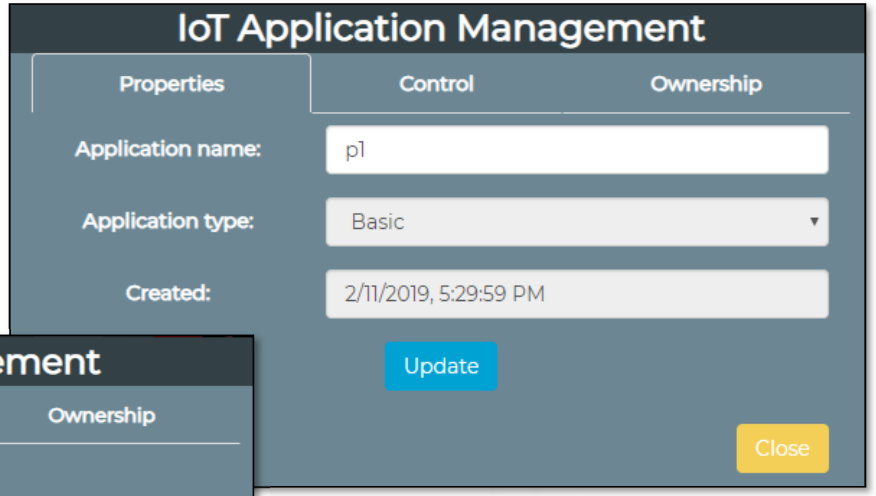
Click to edit IOT App properties

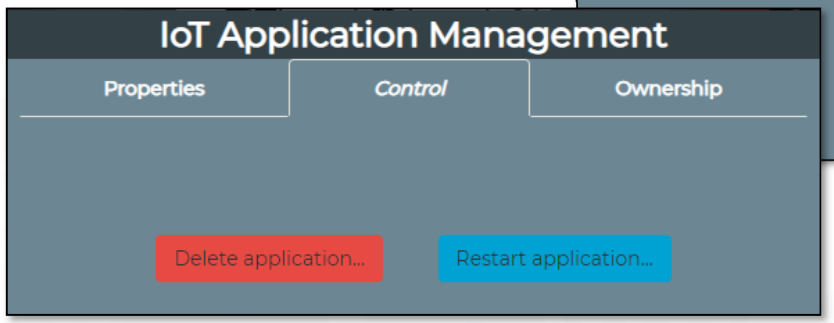
Click to view the Snap4City Dashboard

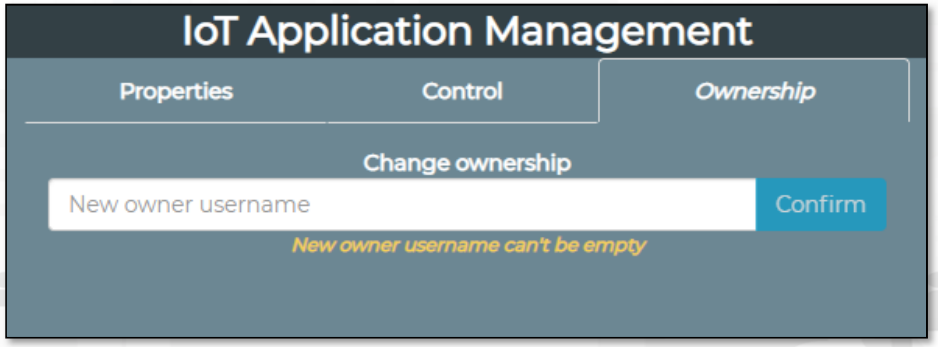
Click to edit the Snap4City Dashboard

IOT Application Management

- **Properties**
  - Name, Type, Creation date
- **Control**
  - Restart
  - Delete
- **Change of ownership**
  - Toward another Snap4City User





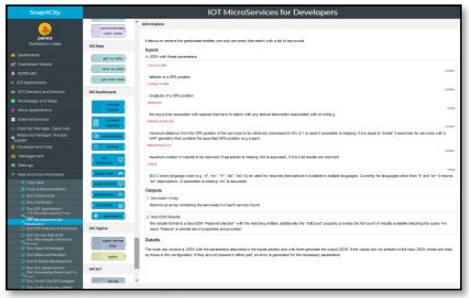


Authoring IOT Applications

IOT Applications Development

IOT Discovering

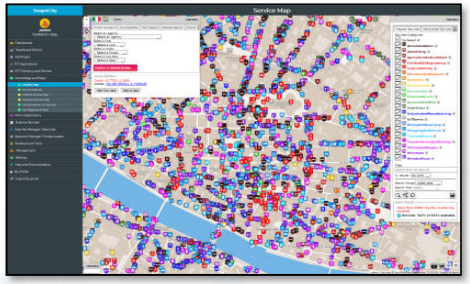
MicroServices collections



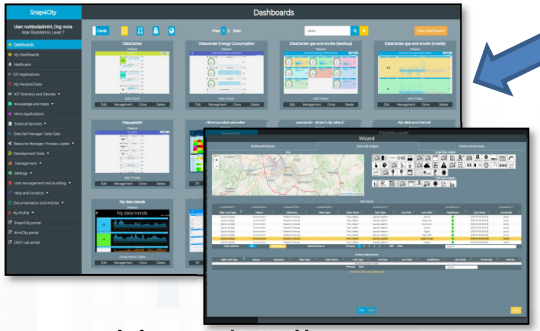
My IOT Applications

IOT App. Editor

Generating IOT App
With Dashboard



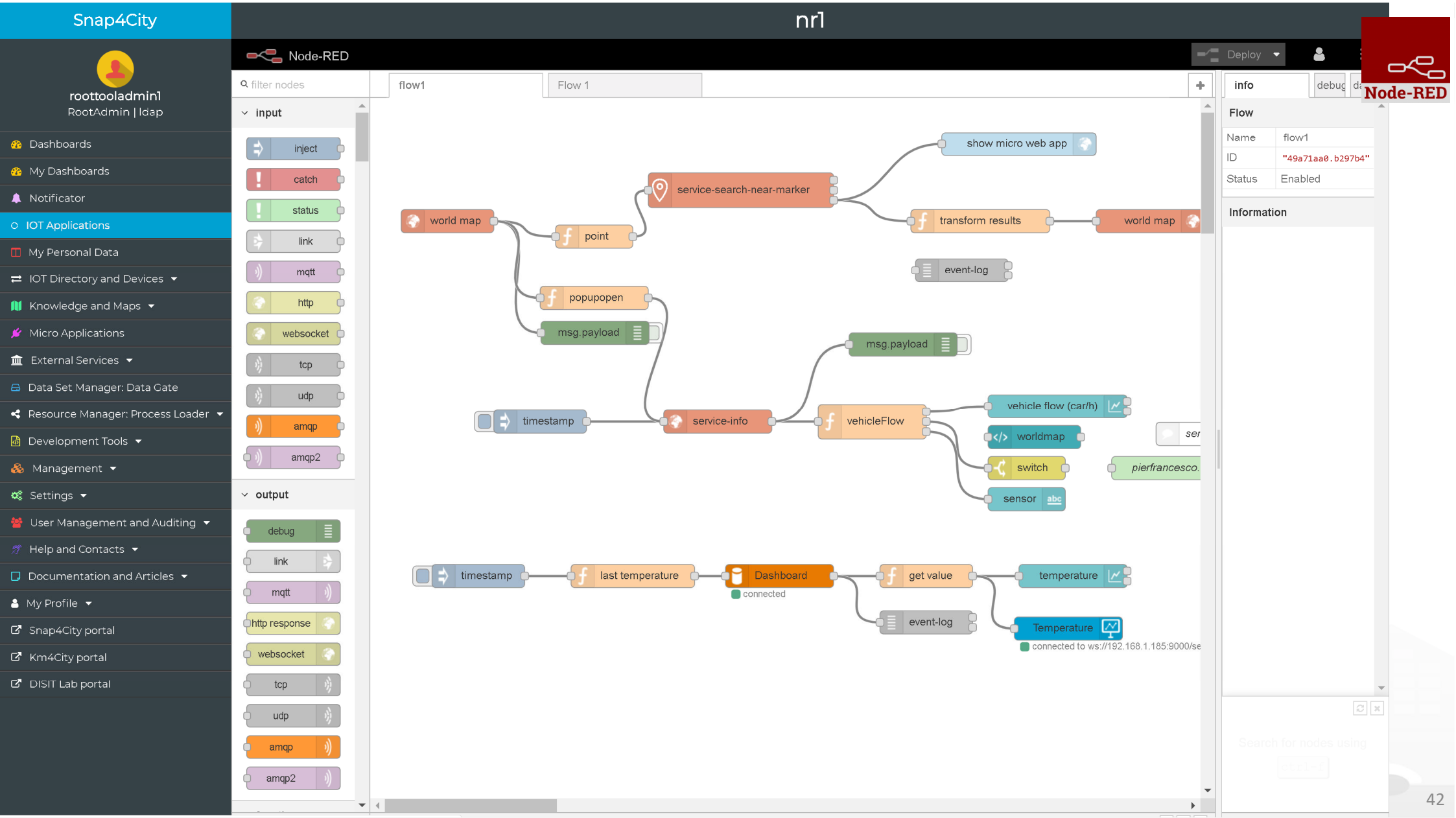
ServiceMap Discovery



Dashboard Collection,
Editor and Wizard

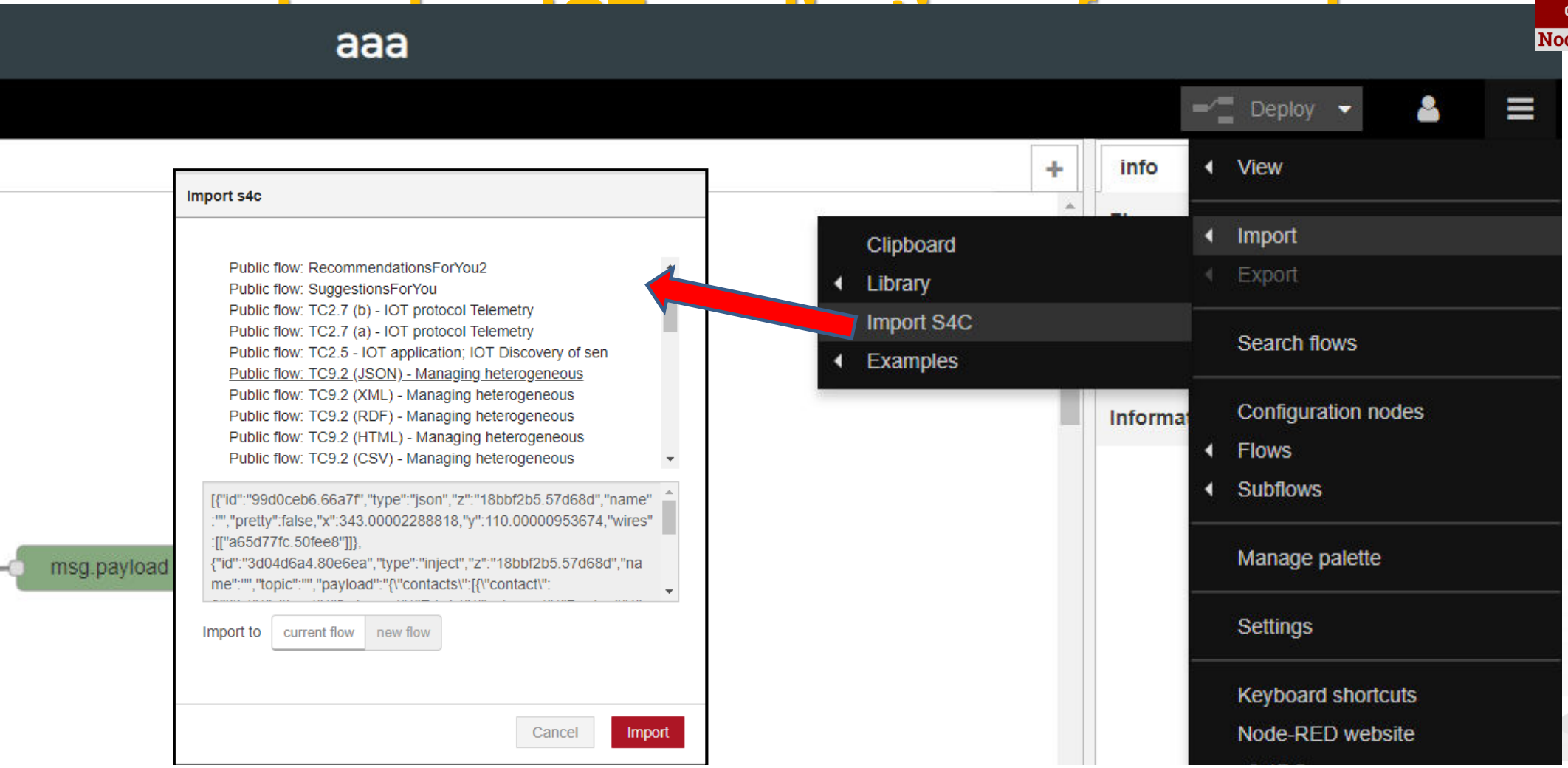
Sharing/saving
reusing IOT App

Resource Manager


Snap4City
nr1
Node-RED
Deploy
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
filter nodes
input
inject
catch
status
link
mqtt
http
websocket
tcp
udp
amqp
amqp2
output
debug
link
mqtt
http response
websocket
tcp
udp
amqp
amqp2
flow1
Flow 1
world map
point
service-search-near-marker
show micro web app
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
Node-RED

IOT Application Editor: NODE-RED

- In the IOT Application of Snap4City, it is possible to:
  - Create multiple concurrent Flows for each IOT Application
  - Execute flow that process data as: Event Driven, Batch (periodic or not)
  - Load other libraries of MicroServices/Nodes/Blocks
    - The loading is allowed only for Administrators for security reasons
  - Save/load, share, Flows, and applications with other users via the Resource Manager or with JS Foundation
  - Ask a limited number of IOT Applications.
    - The Limit may depend on the organization or on personal authorization
  - ..

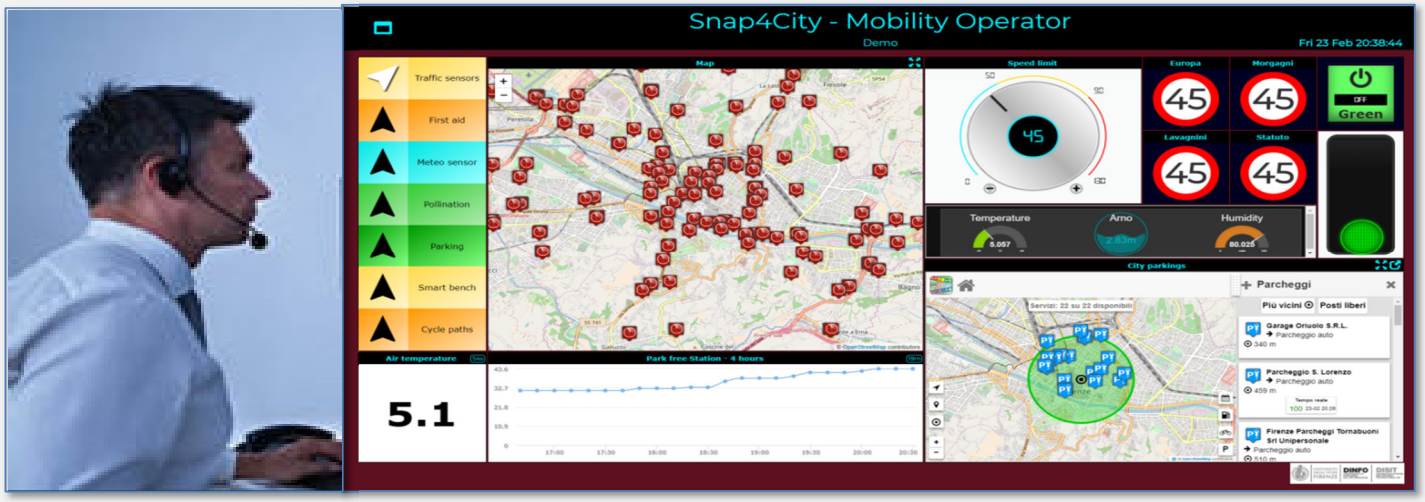

aaa
Deploy
info
View
Import s4c
Public flow: RecommendationsForYou2
Public flow: SuggestionsForYou
Public flow: TC2.7 (b) - IOT protocol Telemetry
Public flow: TC2.7 (a) - IOT protocol Telemetry
Public flow: TC2.5 - IOT application; IOT Discovery of sen
Public flow: TC9.2 (JSON) - Managing heterogeneous
Public flow: TC9.2 (XML) - Managing heterogeneous
Public flow: TC9.2 (RDF) - Managing heterogeneous
Public flow: TC9.2 (HTML) - Managing heterogeneous
Public flow: TC9.2 (CSV) - Managing heterogeneous
[{"id":"99d0ceb6.66a7f","type":"json","z":"18bbf2b5.57d68d","name":"","pretty":false,"x":343.00002288818,"y":110.00000953674,"wires":[["a65d77fc.50fee8"]]},{"id":"3d04d6a4.80e6ea","type":"inject","z":"18bbf2b5.57d68d","name":"","topic":"","payload":"{\"contacts\":[{\"contact\":
Import to
current flow
new flow
Cancel
Import
msg.payload
Clipboard
Library
Import S4C
Examples
Import
Export
Search flows
Configuration nodes
Flows
Subflows
Manage palette
Settings
Keyboard shortcuts
Node-RED website

City Dashboard + IOT App

Control Room Operator

Would like to:

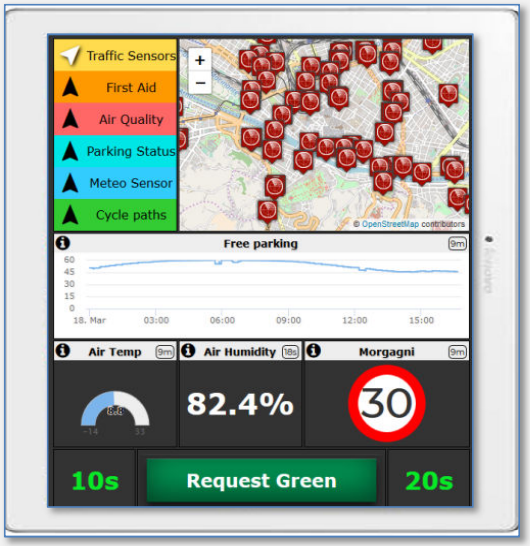
- **Monitor** traffic flow, Environment, Car parking, Cycling, First aid, temp., ..
- **Act and** monitor Dynamic Plates
- **Act and** monitor red lights



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

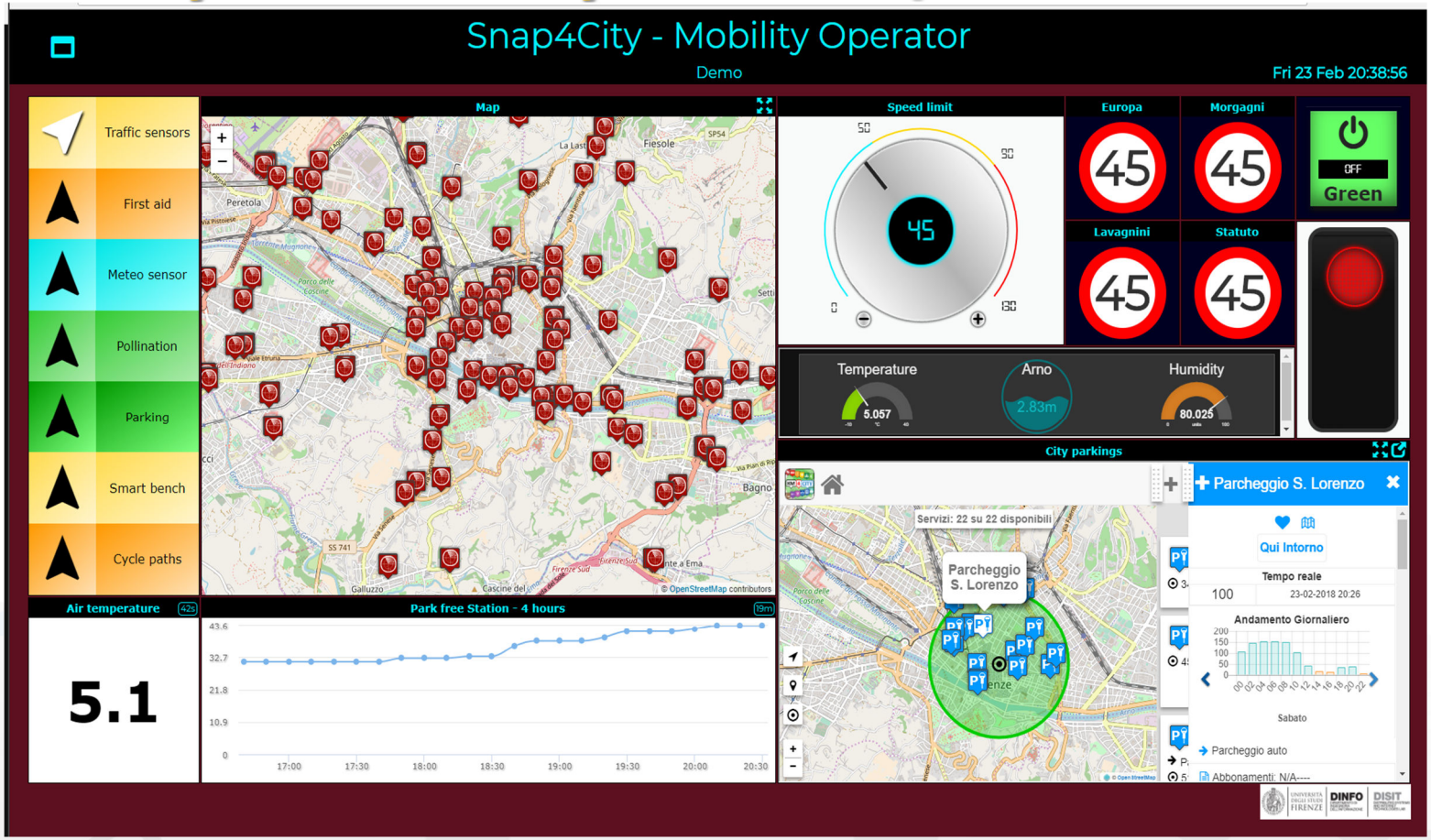
Sensors:

- Values
- Status

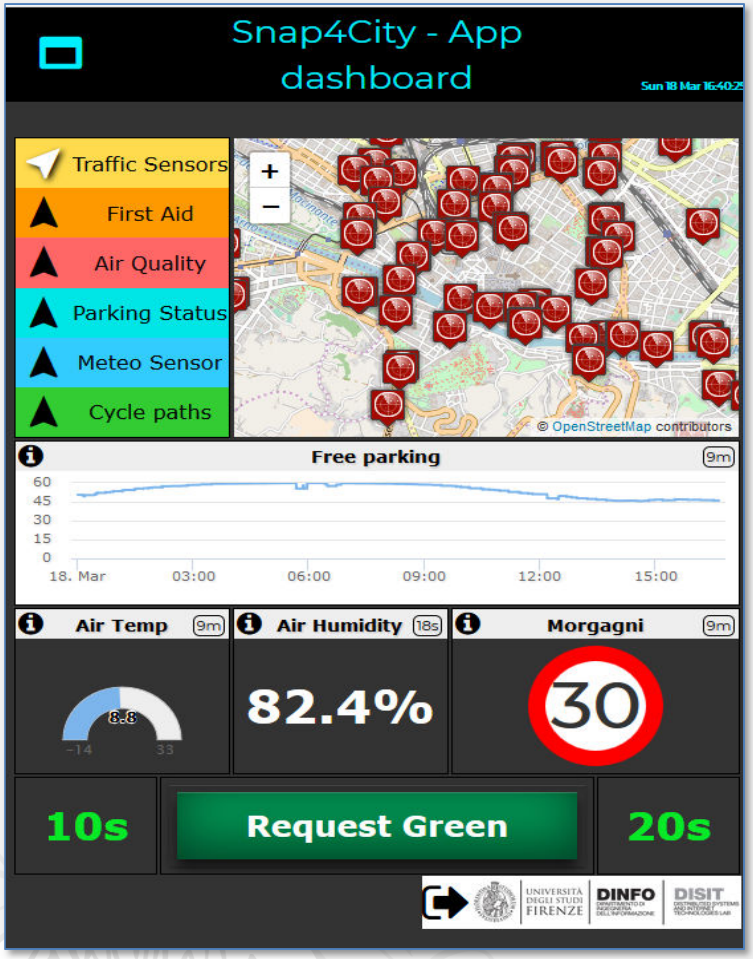
Actuators:

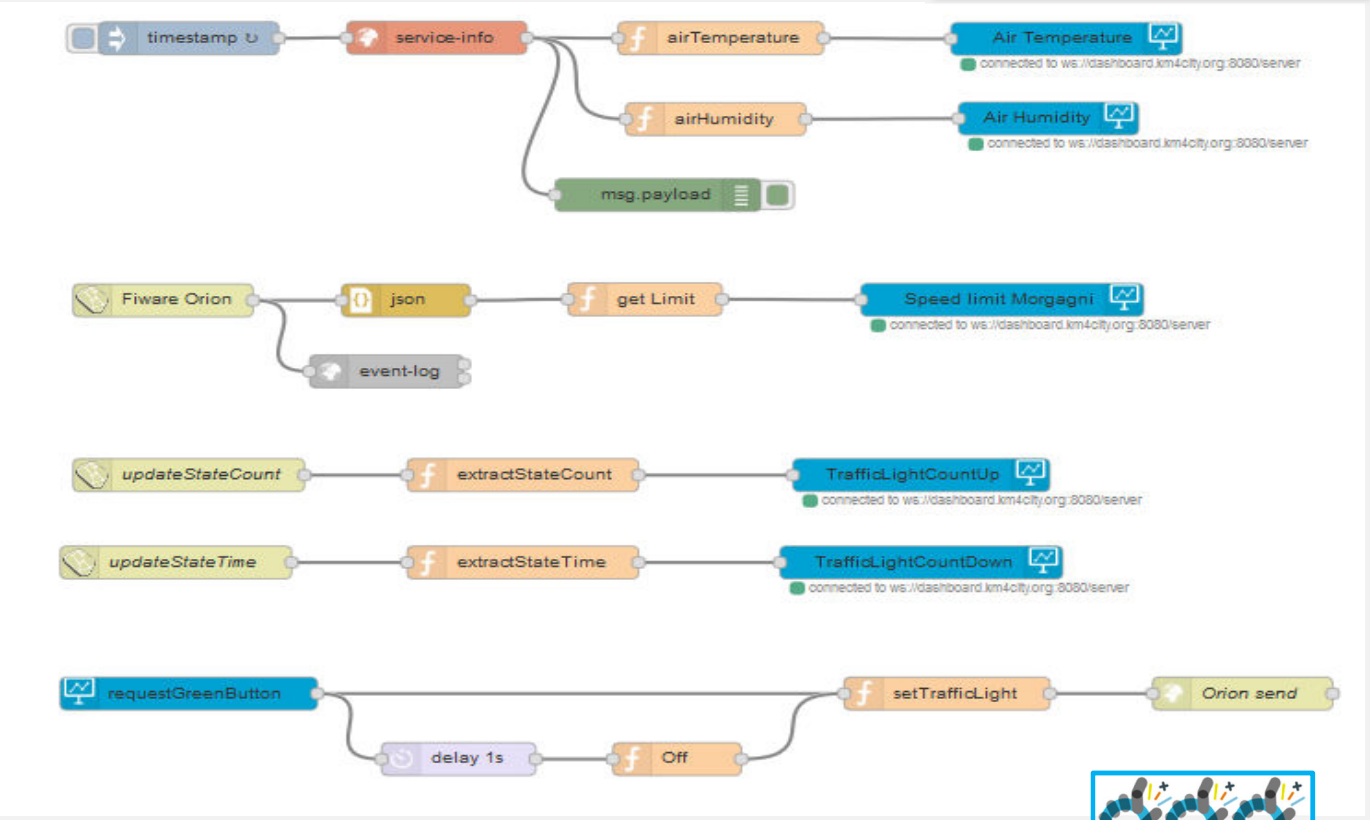
- Buttons
- Dimers
- Etc.

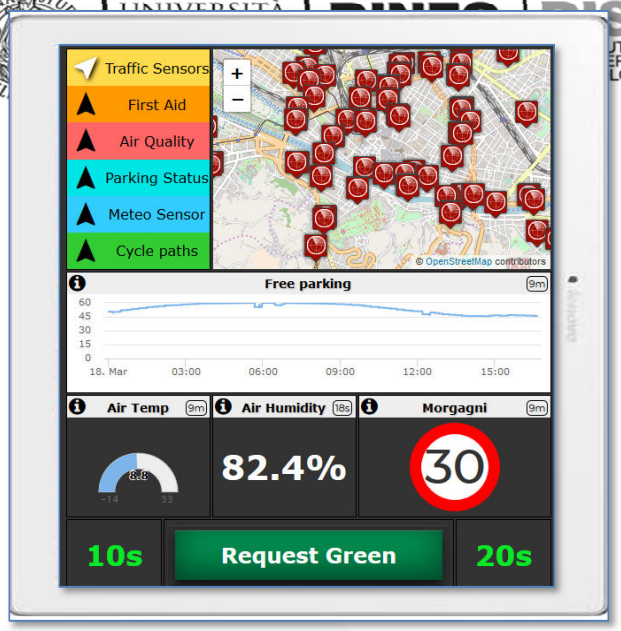
Virtual Sensors and Actuators



IOT Application with City Dashboard simple development

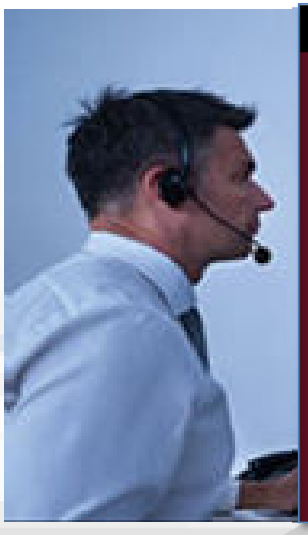


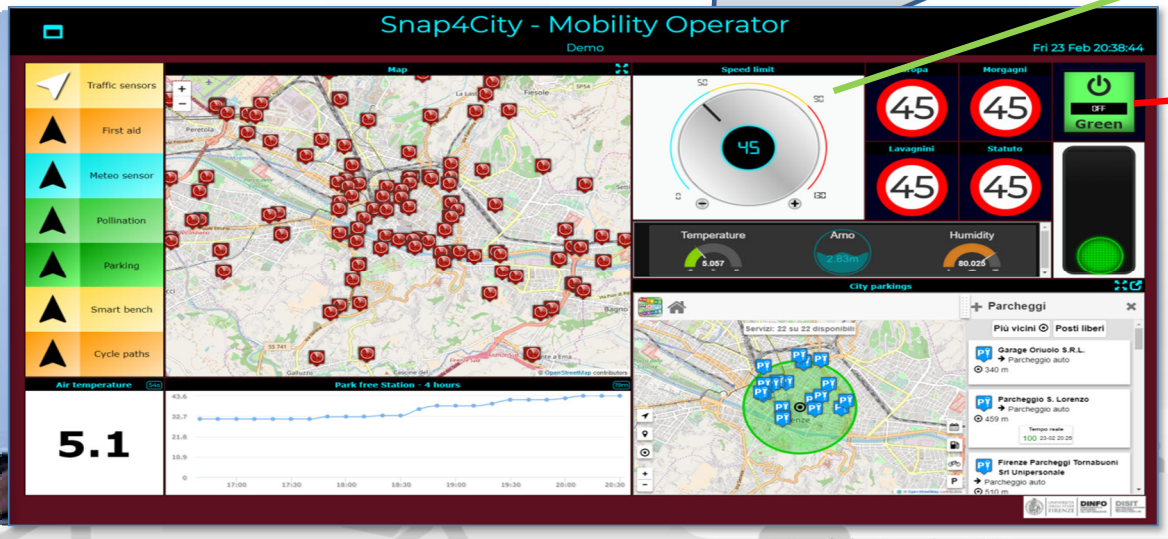


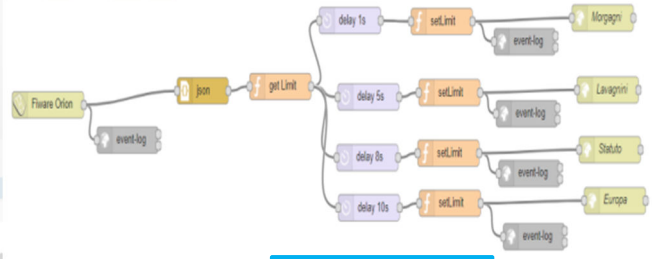


SNAP4CITY

KM 4 CITY





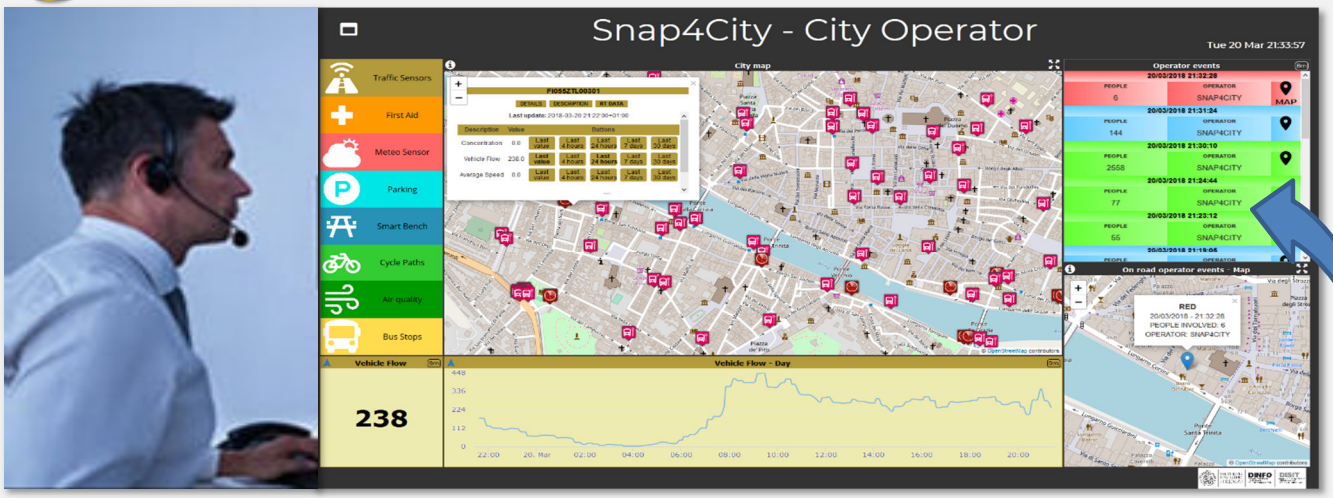


Reporting Critical Events

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

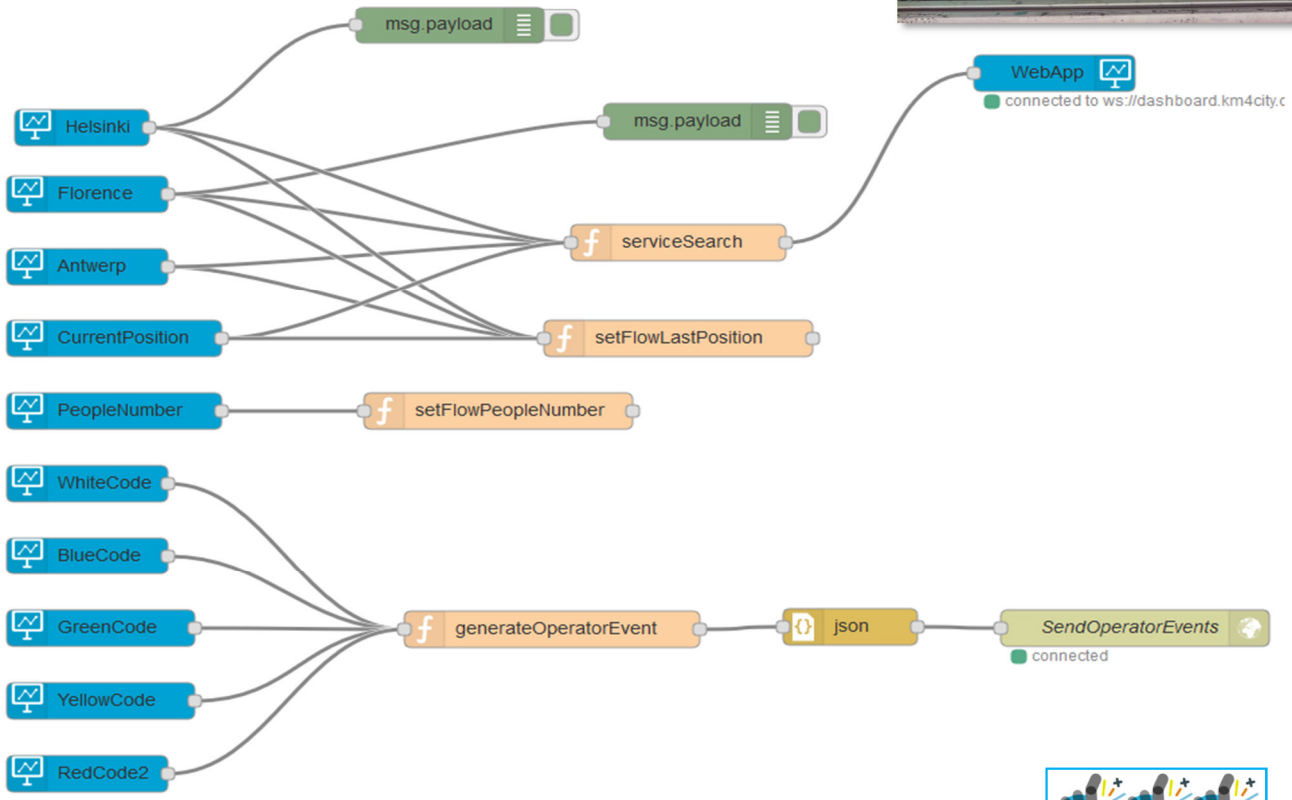


On the road operator

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

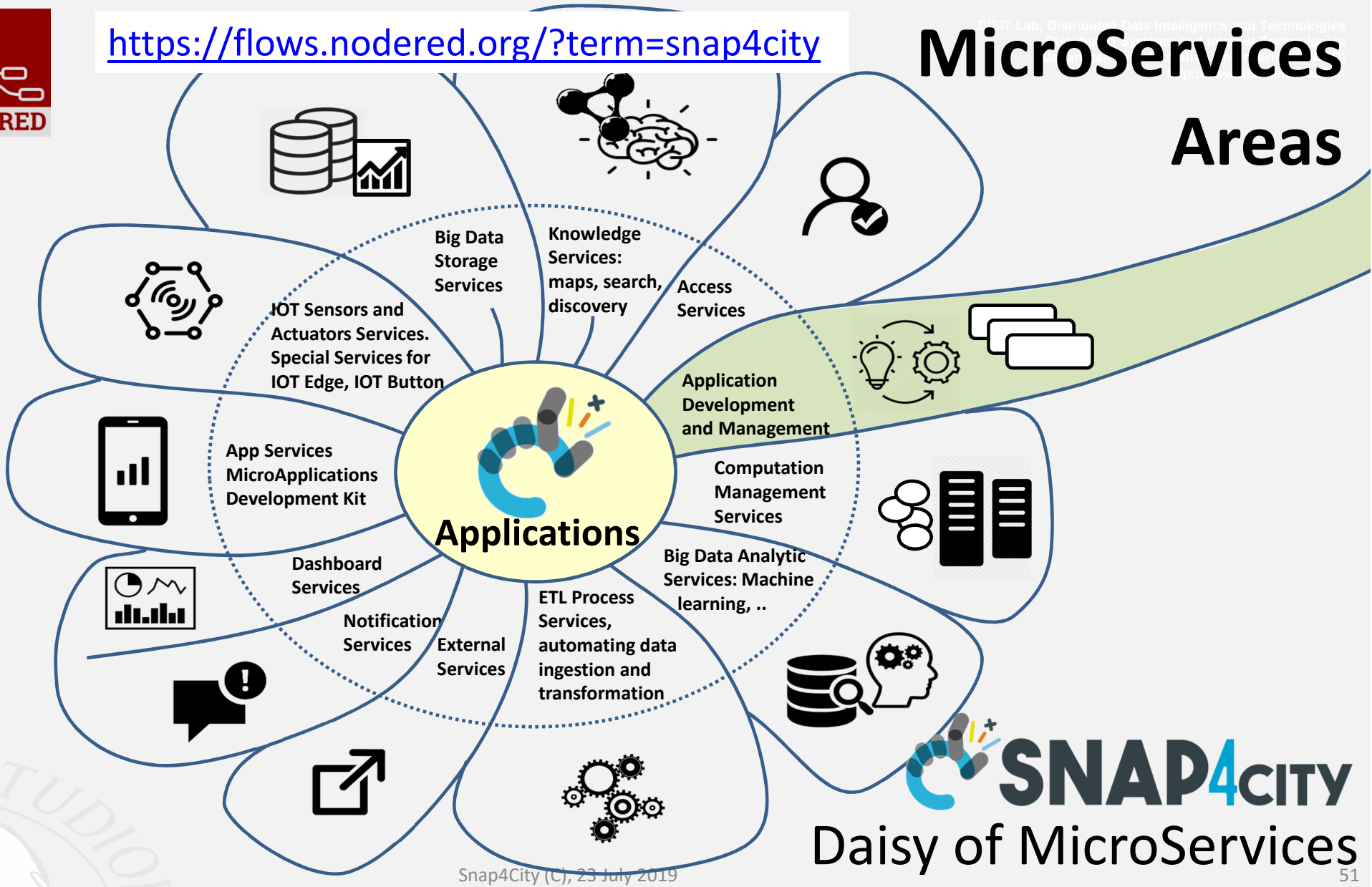


- input
- output
- function
- social
- storage
- analysis
- advanced
- lwm2m
- S4CSearchDev
- S4CMapping
- S4CManagement
- S4CDataAnalytic
- S4CBigData
- S4CSearch
- S4CData
- S4CDashboard
- S4CSigfox
- S4CIoT
- S4CLogDev
- S4CView
- S4CSocial
- location
- dashboard

Node-RED

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

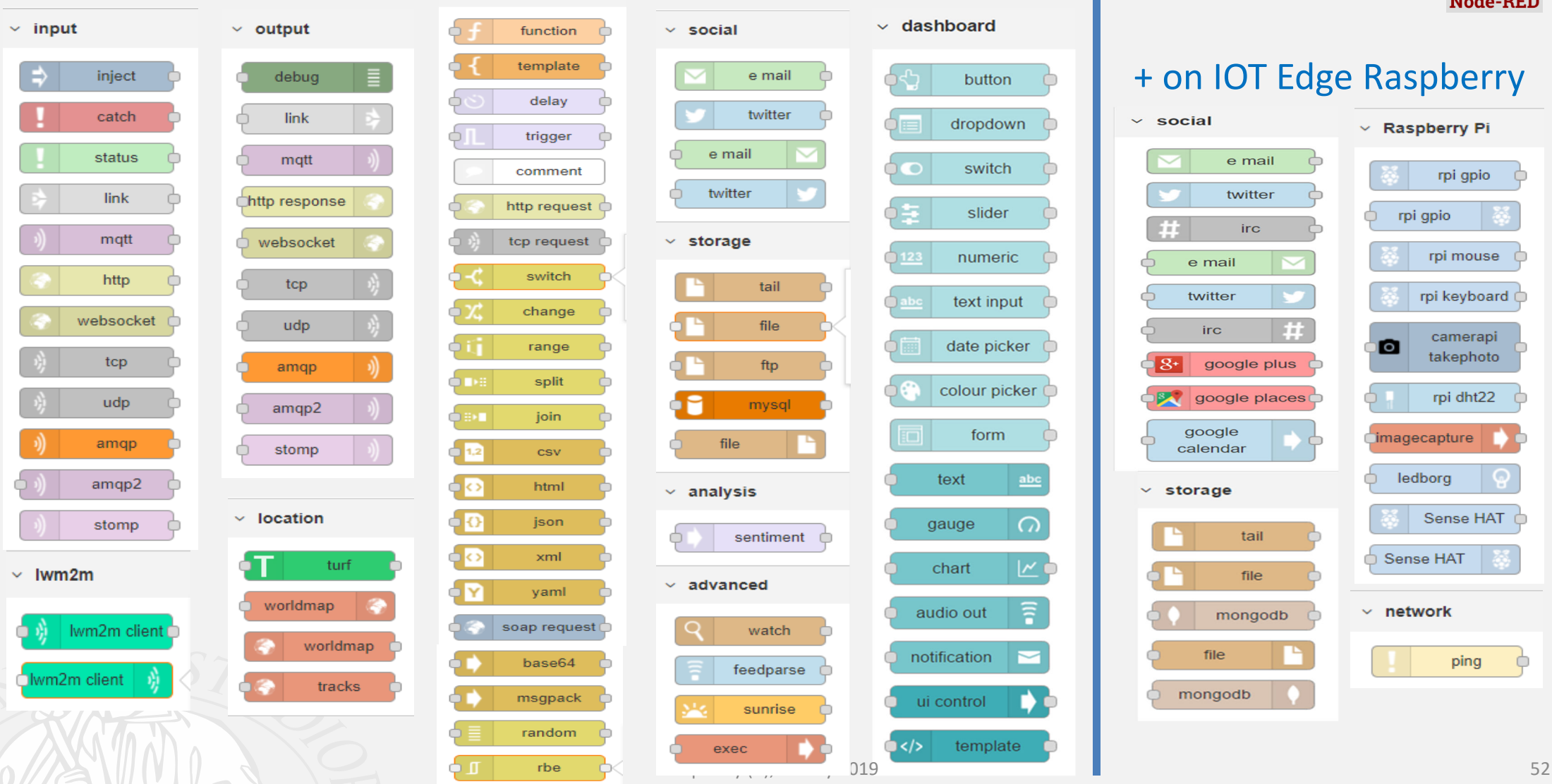
MicroServices Areas



SNAP4CITY

Daisy of MicroServices

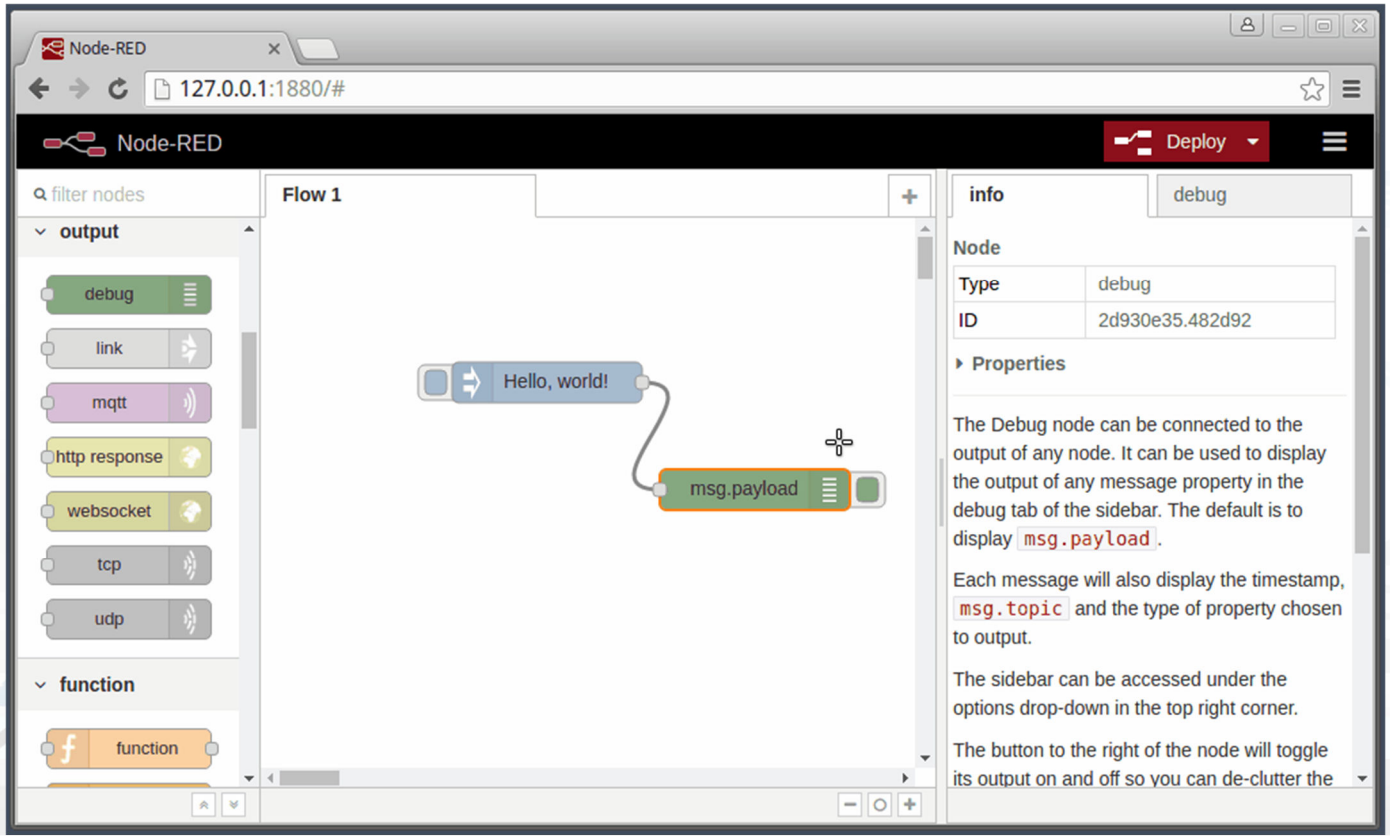
Basic Node.js Blocks on NodeRed on our Advanced IOT Apps

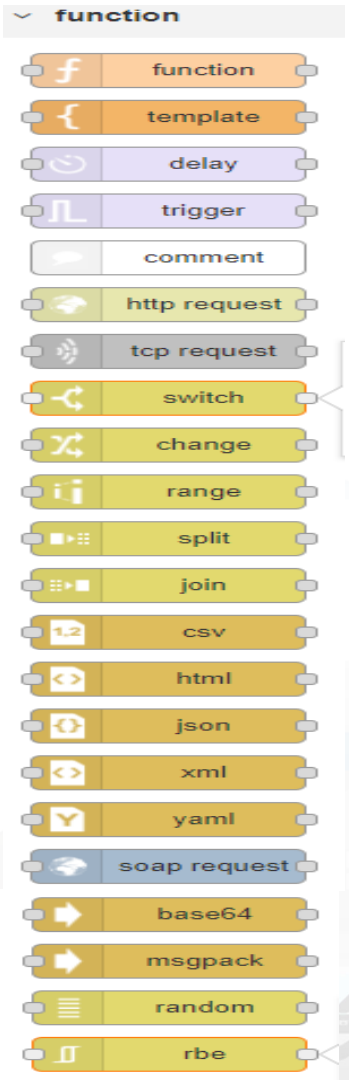


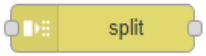
+ on IOT Edge Raspberry

Hello World of Node-RED

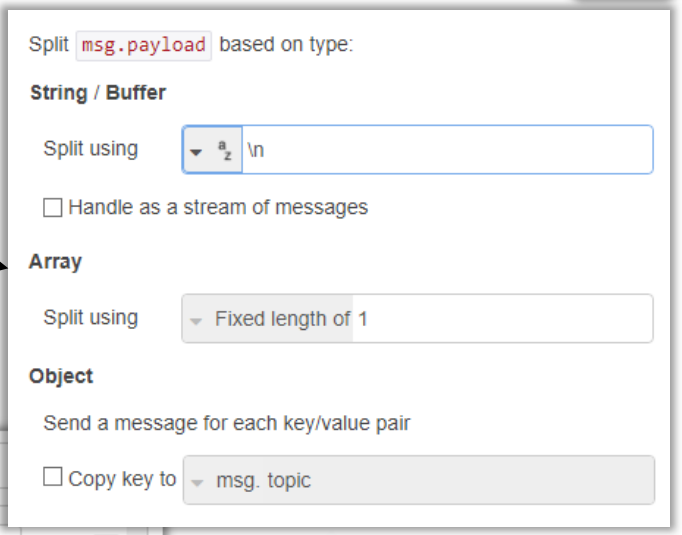
- http://developer.opto22.com/nodered/general/getting-started/node-red-hello-world/

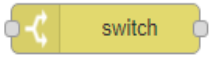




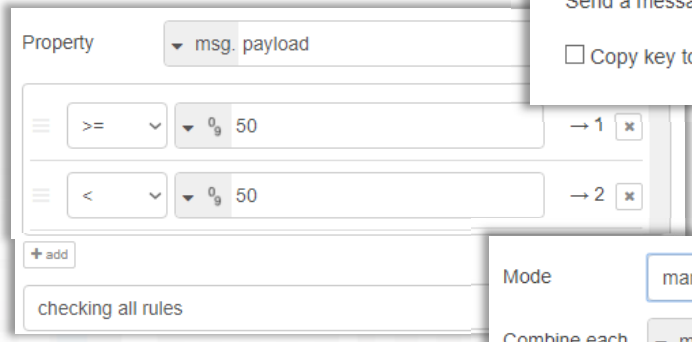


Divides the input message into multiple messages as indicated in the configuration. If you have an array at the input, you can configure it to send each element of the array individually at the output.

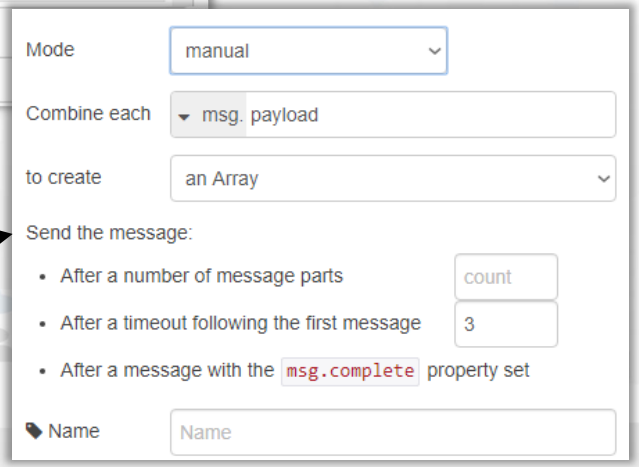


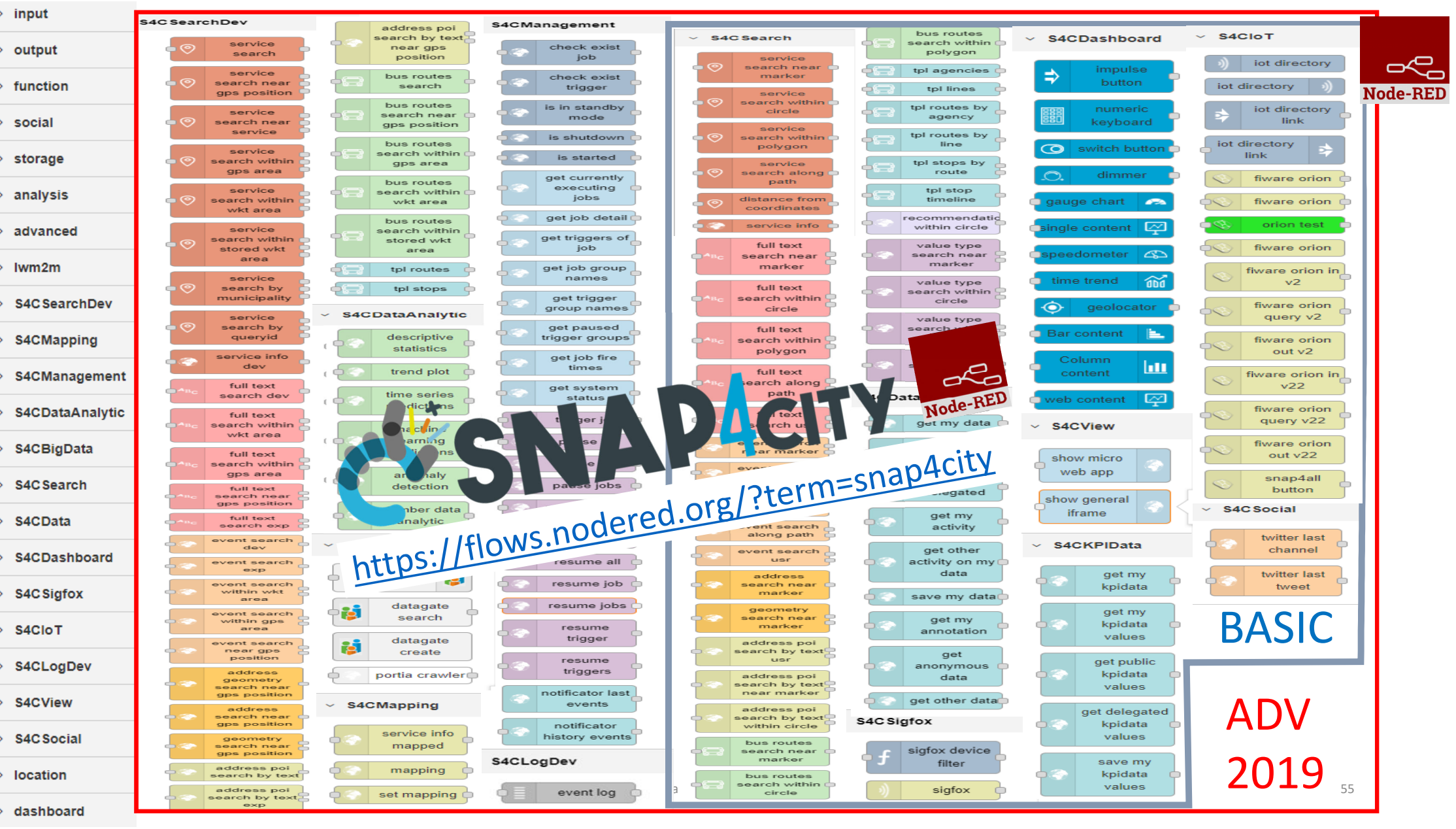


Treads the input message on possible different outputs based on a comparison made on the input message.



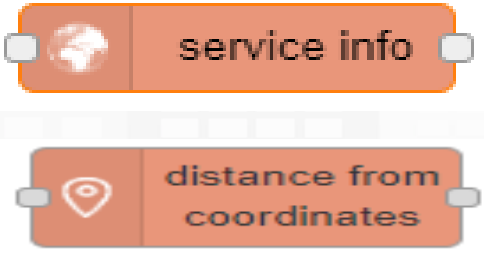
Operates in reverse order to the split. Joins the incoming messages in the mode indicated in the configuration.




input
output
function
social
storage
analysis
advanced
lwm2m
S4C SearchDev
S4CMapping
S4CManagement
S4CDataAnalytic
S4CBigData
S4C Search
S4CData
S4CDashboard
S4C Sigfox
S4CIoT
S4CLogDev
S4CView
S4C Social
location
dashboard
service search
service search near gps position
service search near service
service search within gps area
service search within wkt area
service search within stored wkt area
service search by municipality
service search by queryid
service info dev
full text search dev
full text search within wkt area
full text search within gps area
full text search near gps position
full text search exp
event search dev
event search exp
event search within wkt area
event search within gps area
event search near gps position
address geometry search near gps position
address search near gps position
geometry search near gps position
address poi search by text
address poi search by text exp
address poi search by text near gps position
bus routes search
bus routes search near gps position
bus routes search within gps area
bus routes search within wkt area
bus routes search within stored wkt area
tpl routes
tpl stops
descriptive statistics
trend plot
time series
anomaly detection
number data analytic
datagate search
datagate create
portia crawler
service info mapped
mapping
set mapping
check exist job
check exist trigger
is in standby mode
is shutdown
is started
get currently executing jobs
get job detail
get triggers of job
get job group names
get trigger group names
get paused trigger groups
get job fire times
get system status
pause jobs
resume all
resume job
resume jobs
resume trigger
resume triggers
notificator last events
notificator history events
event log
service search near marker
service search within circle
service search within polygon
service search along path
distance from coordinates
service info
full text search near marker
full text search within circle
full text search within polygon
full text search along path
event search along path
event search usr
address search near marker
geometry search near marker
address poi search by text usr
address poi search by text near marker
address poi search by text within circle
bus routes search near marker
bus routes search within circle
bus routes search within polygon
tpl agencies
tpl lines
tpl routes by agency
tpl routes by line
tpl stops by route
tpl stop timeline
recommendation within circle
value type search near marker
value type search within circle
get my data
get my activity
get other activity on my data
save my data
get my annotation
get anonymous data
get other data
sigfox device filter
sigfox
impulse button
numeric keyboard
switch button
dimmer
gauge chart
single content
speedometer
time trend
geolocator
Bar content
Column content
web content
show micro web app
show general iframe
get my kpidata
get my kpidata values
get public kpidata values
get delegated kpidata values
save my kpidata values
iot directory
iot directory
iot directory link
iot directory link
fiware orion
fiware orion
orion test
fiware orion
fiware orion in v2
fiware orion query v2
fiware orion out v2
fiware orion in v22
fiware orion query v22
fiware orion out v22
snap4all button
twitter last channel
twitter last tweet
SNAP4CITY
Node-RED
https://flows.nodered.org/?term=snap4city
BASIC
ADV
2019
55

S4CSearch

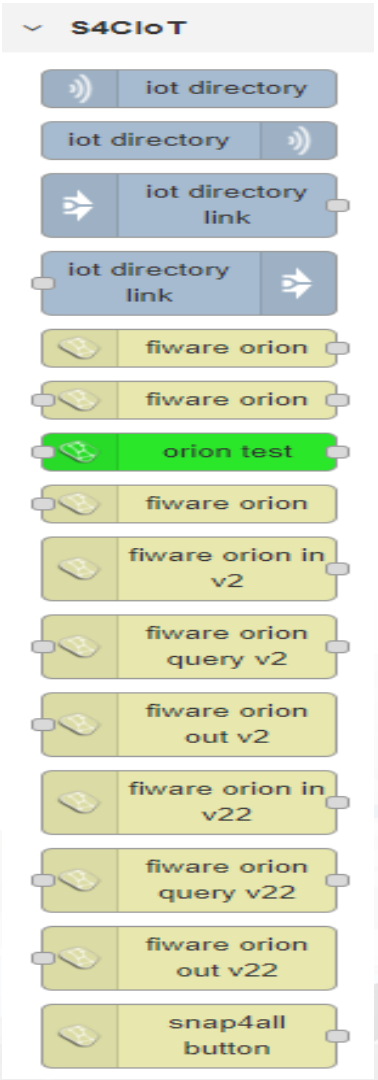
- **For example to search for:**
  - POIs:
    - near a GPS position, from text, along a path, in an area, etc..
  - Public Transport information / data
  - Suggestions
  - Public Transport Means Routes/Paths
  - Events in the area
  - Value Type (kind of data)
  - Etc.
- **To Get DATA of a Service /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



S4CSearch Adv

- 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

S4CIOT



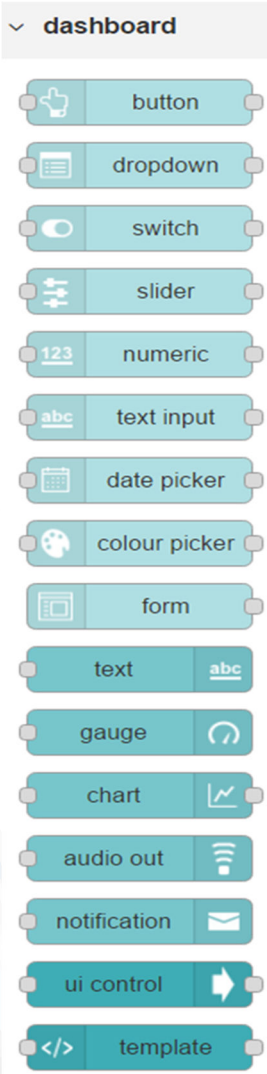
Node-RED

- 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

Dashboard

Native Local or Snap4City

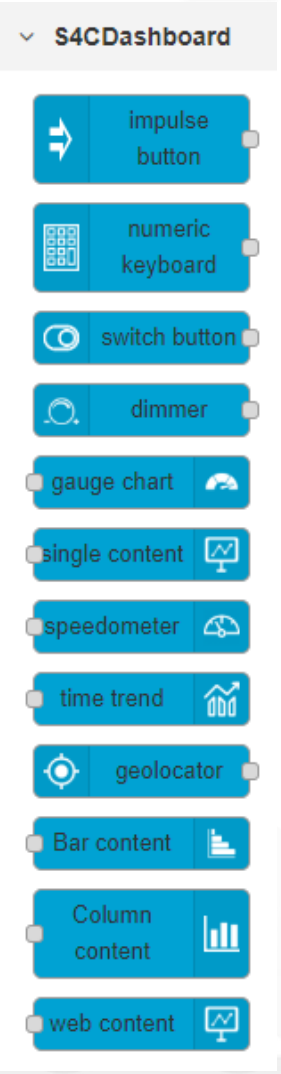


- Input/output
- non secure
- Limited in graphics
- No authentication
- No HLT
- No integration
- Etc..

- Local on IOT Edge

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

- Remote for IOT Edge via WebSocket Secure



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

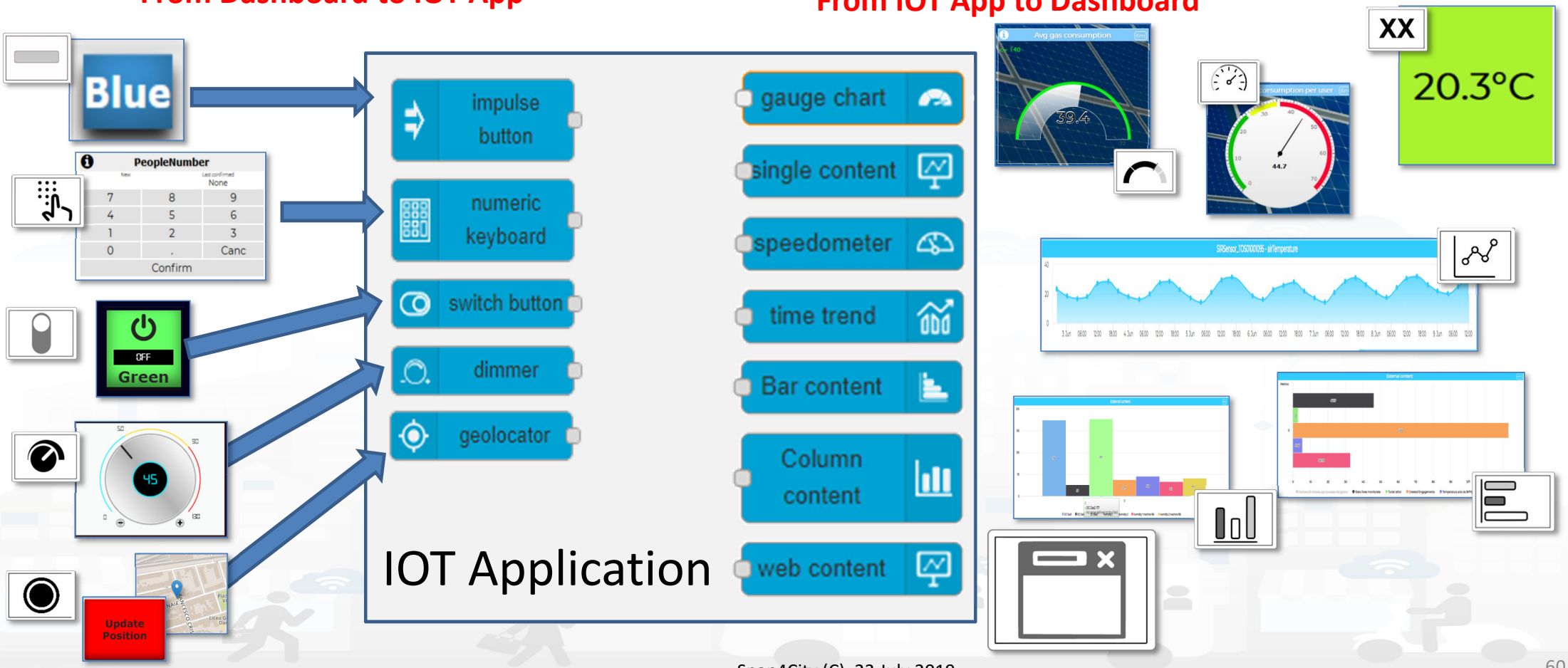
SNAP4CITY KM4CITY

Dashboard-IOT App

Nature

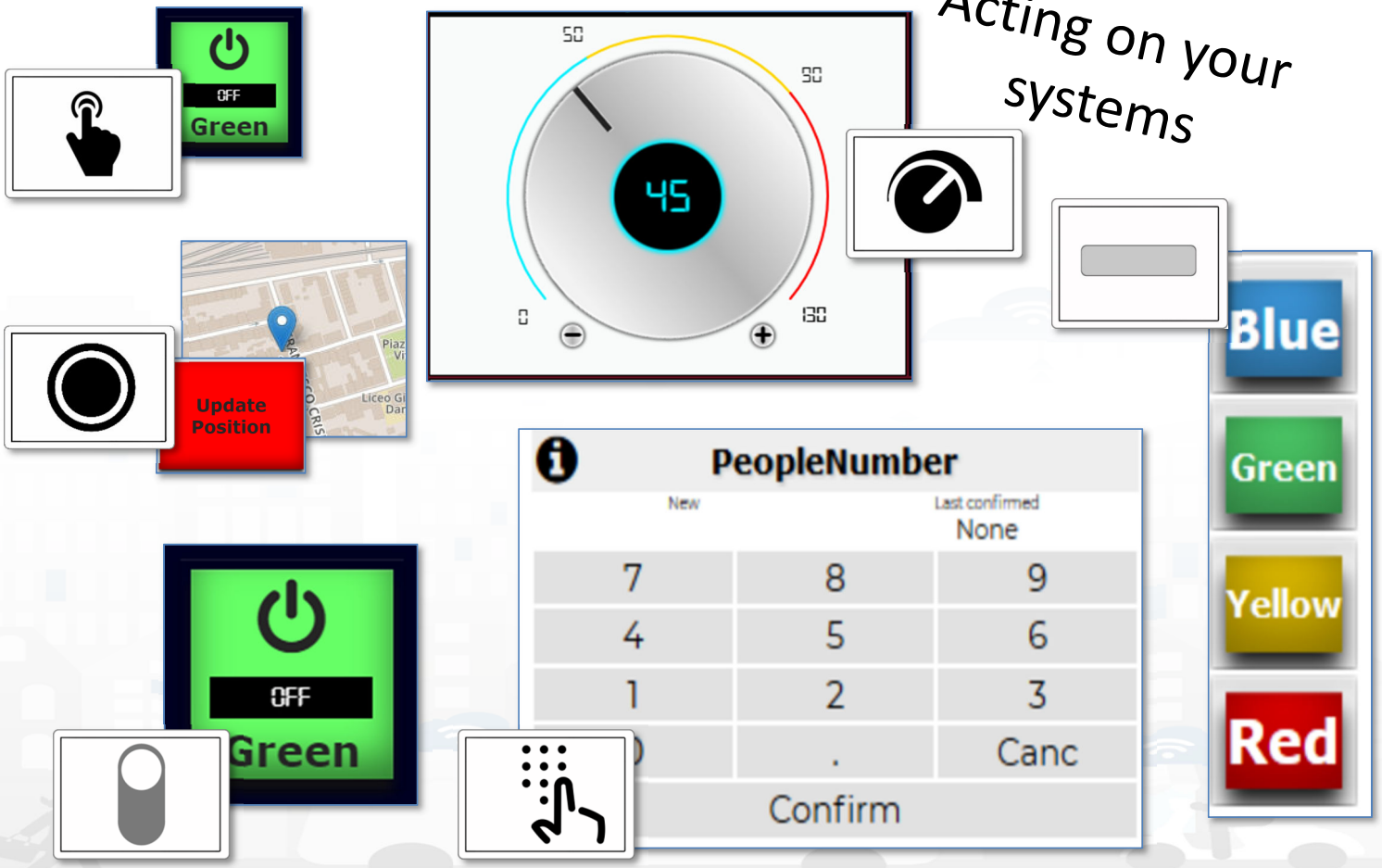
From Dashboard to IOT App

From IOT App to Dashboard



From Dashboard to IOT Devices

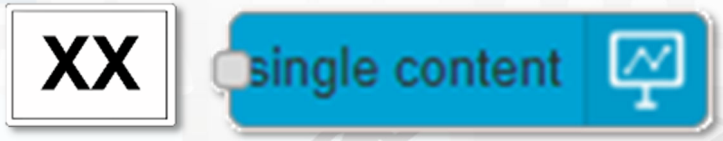
- **Widgets:**
  - Impulse Button
  - Button
  - Switch
  - Dimer/Knowb
  - KeyPad
  - geolocator
- **Registered** on some IOT brokers with NGSI mutual authentication

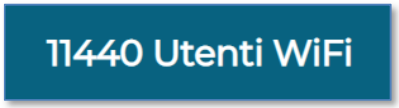


Single Content Widget (flexibility)

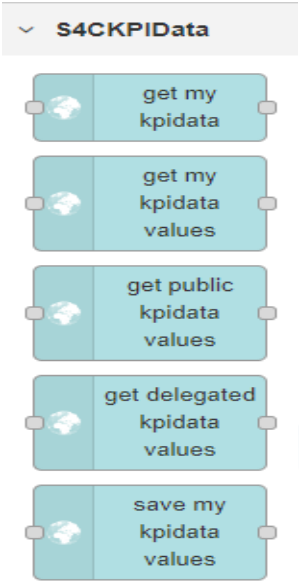
From Dashboard Editor and IOT Applications, accepts in input:

- Numbers
- String
- HTML code



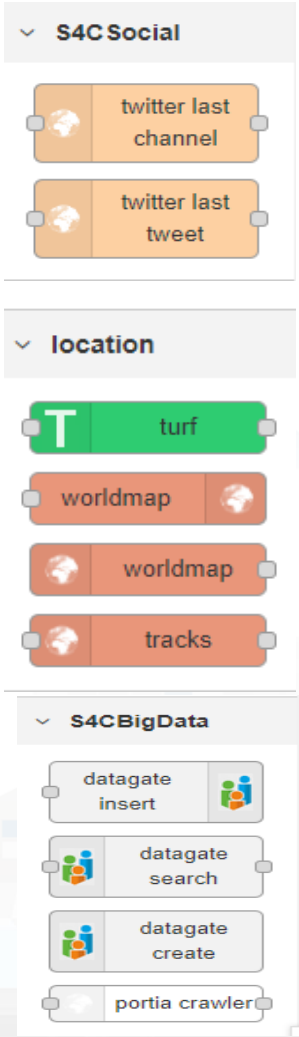


S4CKPIData



- 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



- Request metrics from Twitter Vigilance Channel service and engine of DISIT Lab

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

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

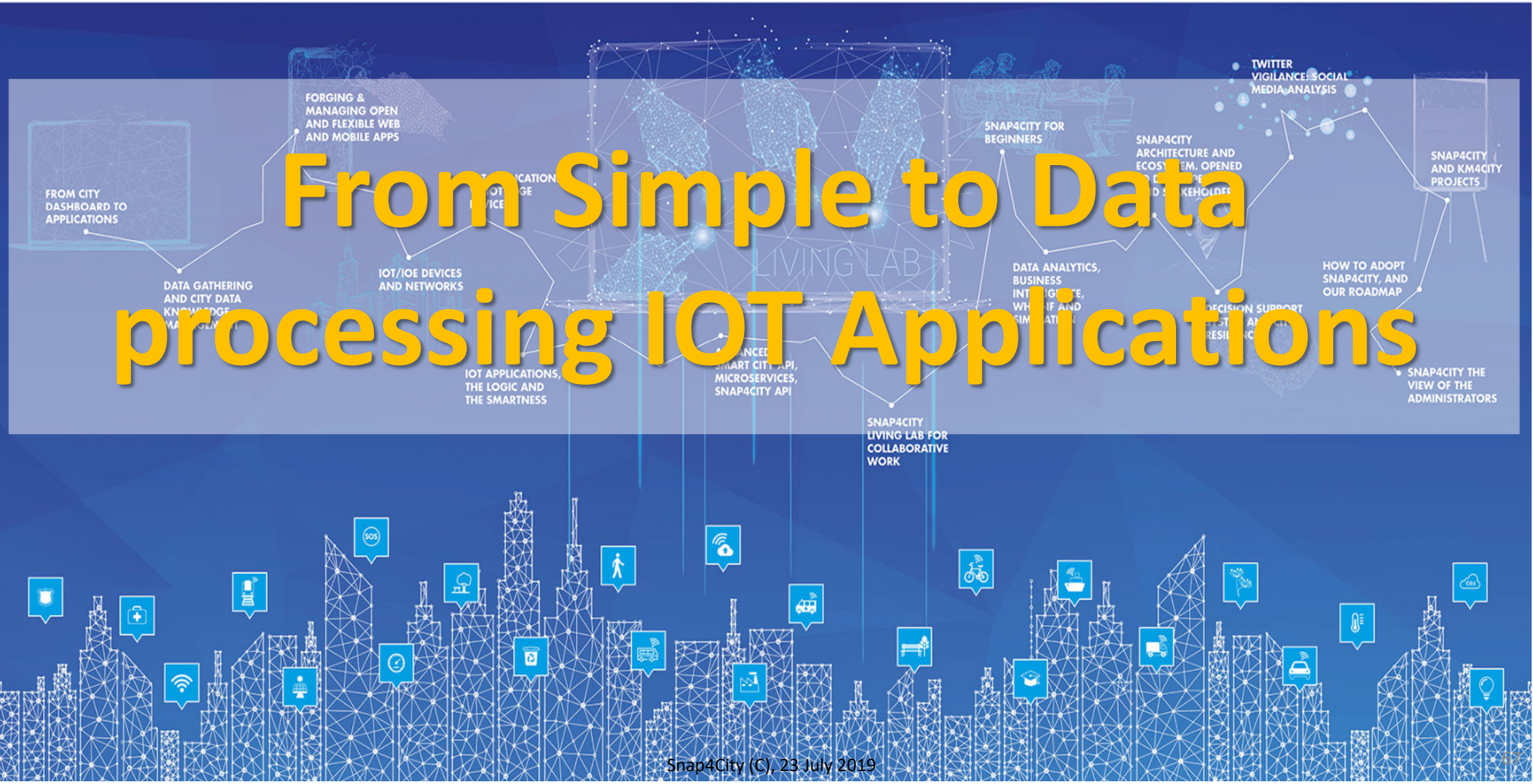
IOT Applications vs Dashboards (self training)

- 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

SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT CITIES



From Simple to Data processing IOT Applications

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)

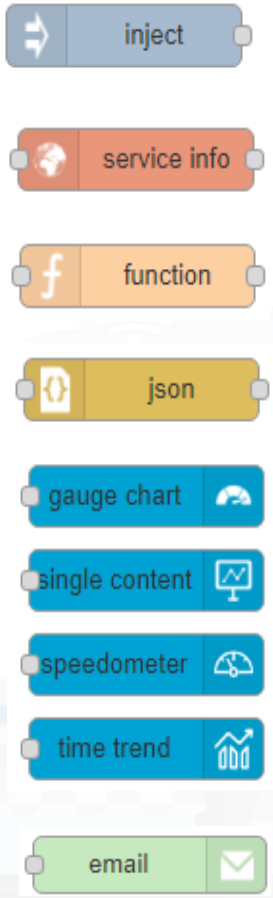
Create a Simple IOT Application (DEMO)

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)

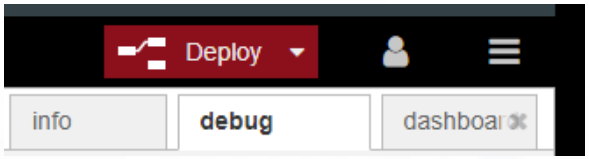
The node called single content accepts strings, numbers and html.

The others only accept numbers.

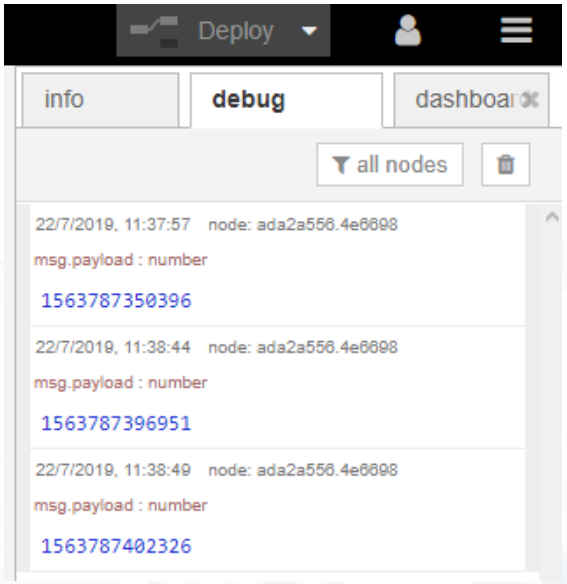
Send an email to the desired recipient. You must enter the username and password of an active email.

Step 1

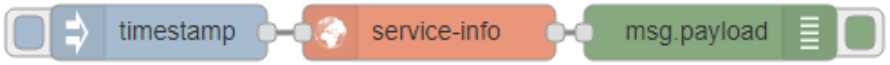
- Inject and Debug 

- Connect
- Deploy 

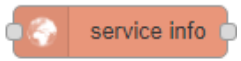
- Click and Observe
- Play with results



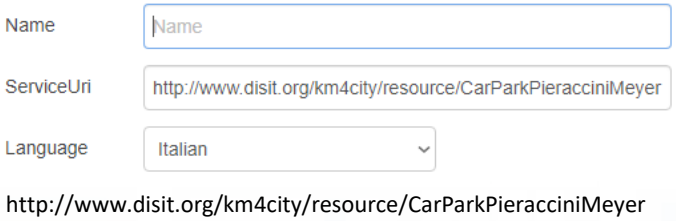
Step 2



- Service Info



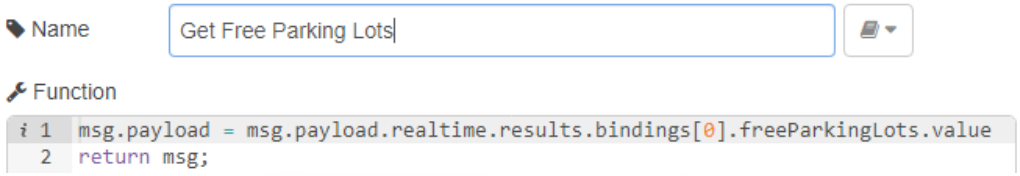
- Connect
- Configure



- Deploy
- Click and Observe
- Play with results

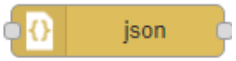
Step 3

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



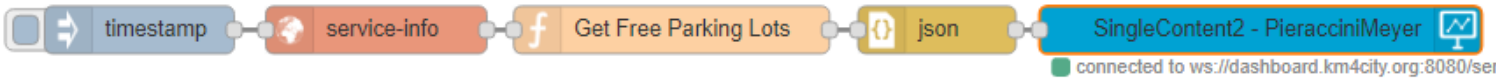
msg.payload = msg.payload.realtime.results.bindings[0].freeParkingLots.value

Step 4

- JSON 

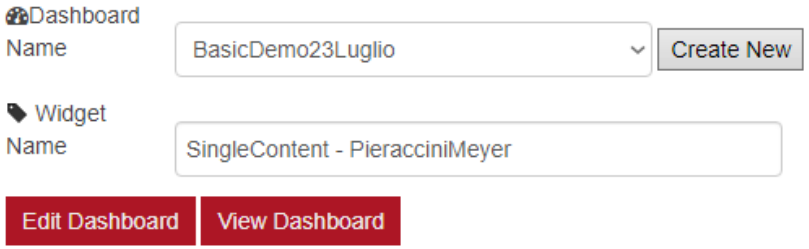
- Connect
- Deploy
- Click and Observe
- Play with results

Step 5

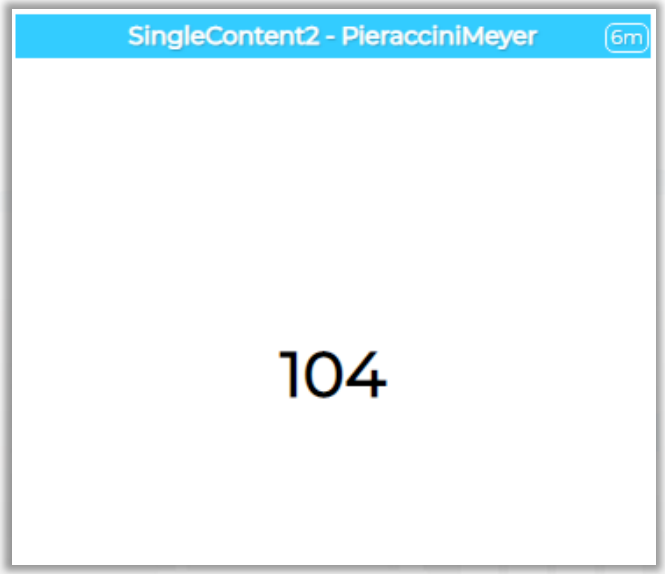


- Single content 

- Connect

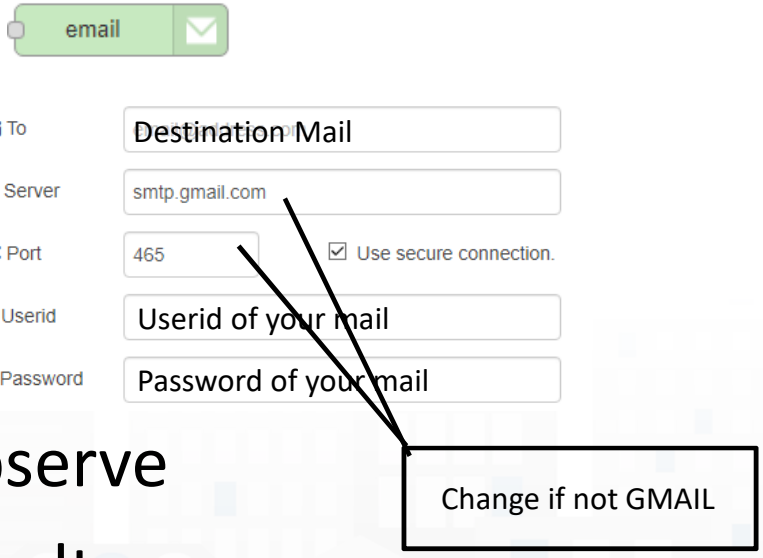


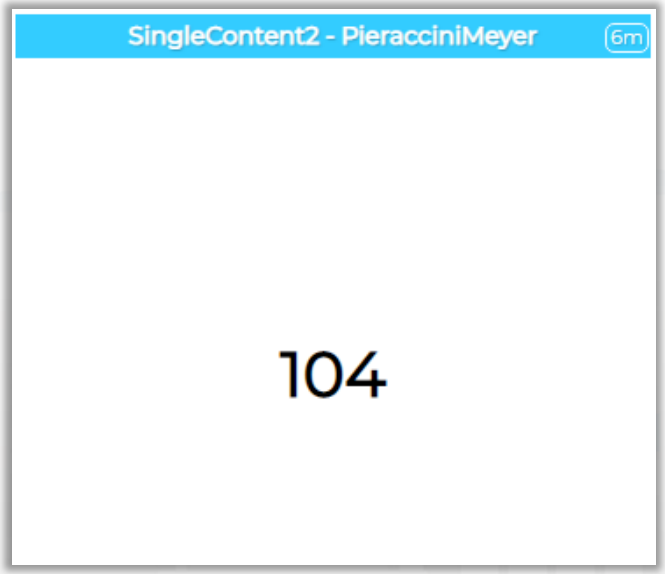
- Configure
- Deploy
- Click and Observe
- Play with results



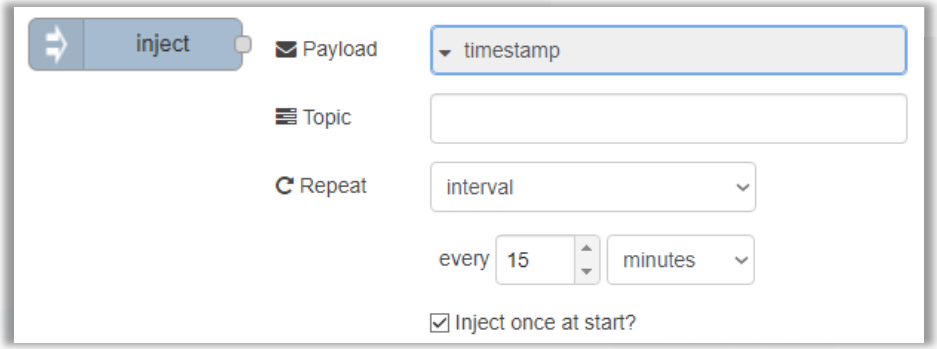
Step 6

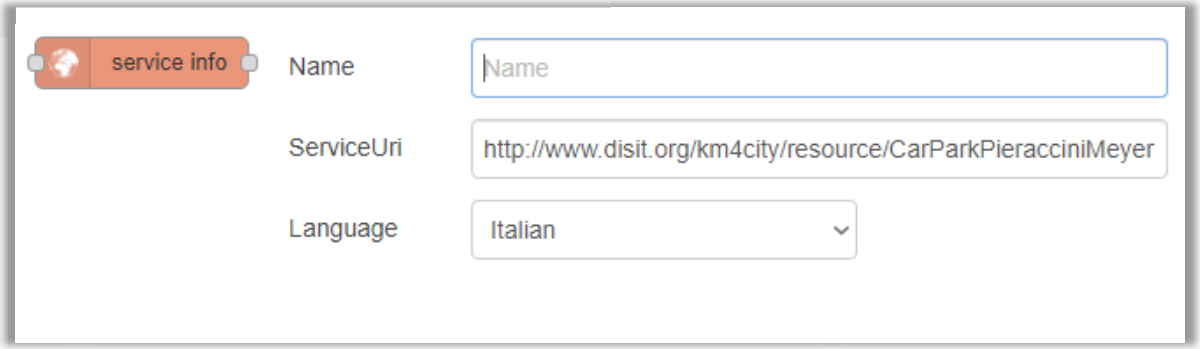
- Email
- Connect
- Configure
- Deploy
- Click and Observe
- Play with results

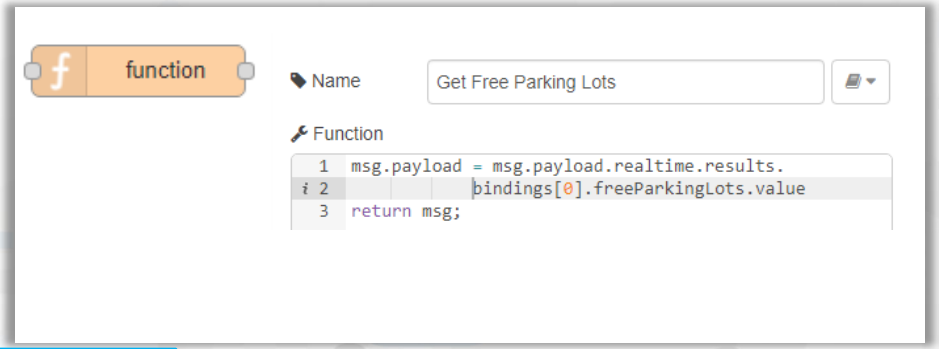


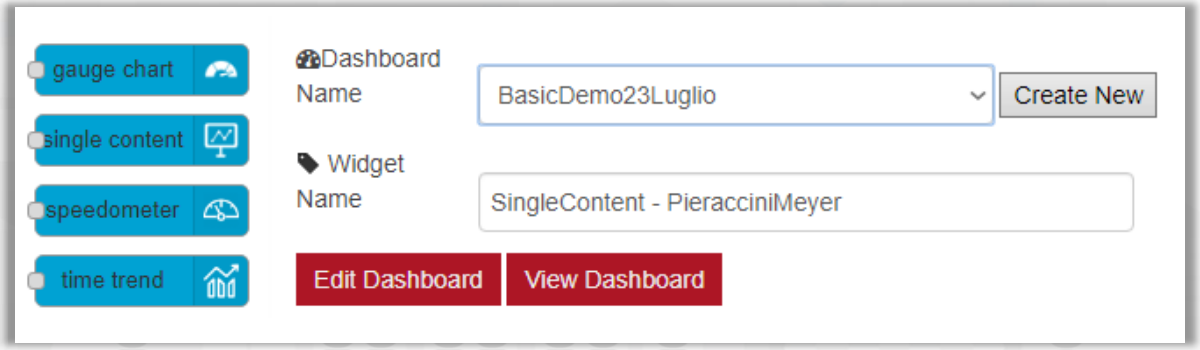


Nodes configuration







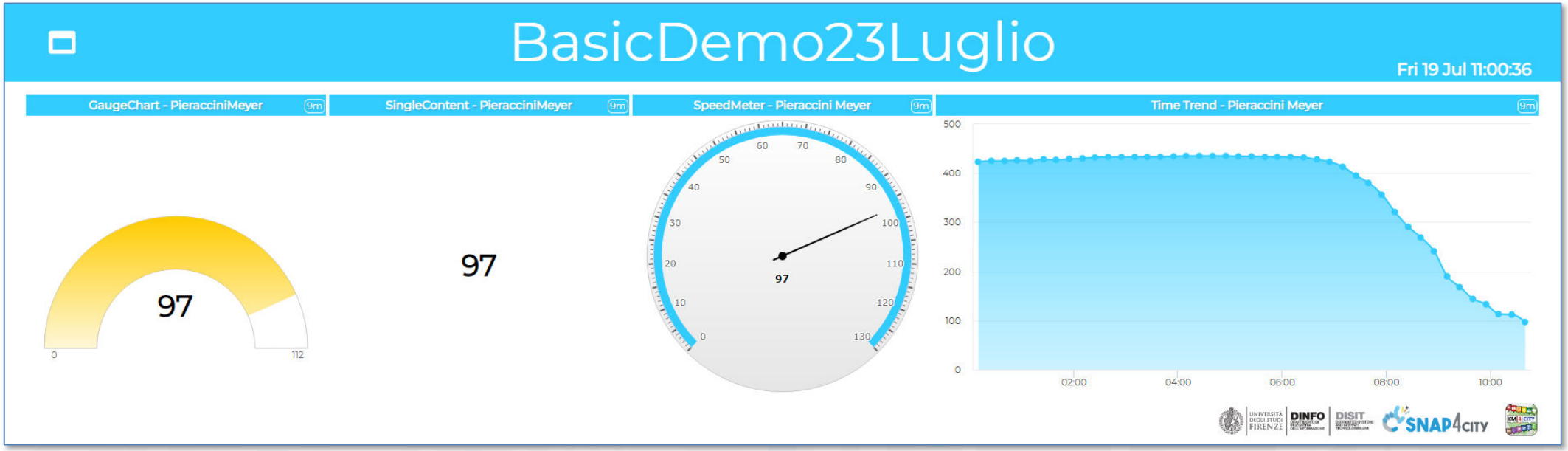


Nodes connections

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==

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)

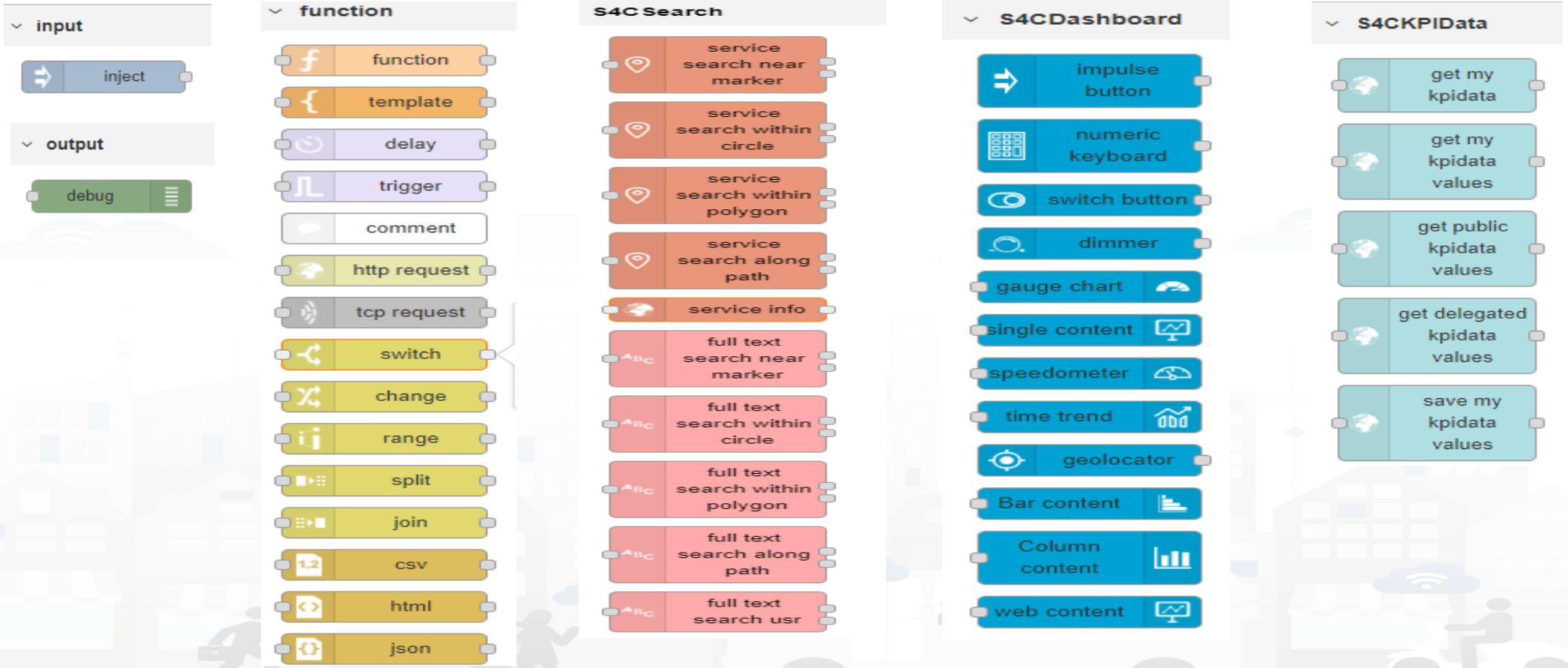
serviceUri: http://www.disit.org/km4city/resource/CarParkPieraccini Meyer

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** ☺ !

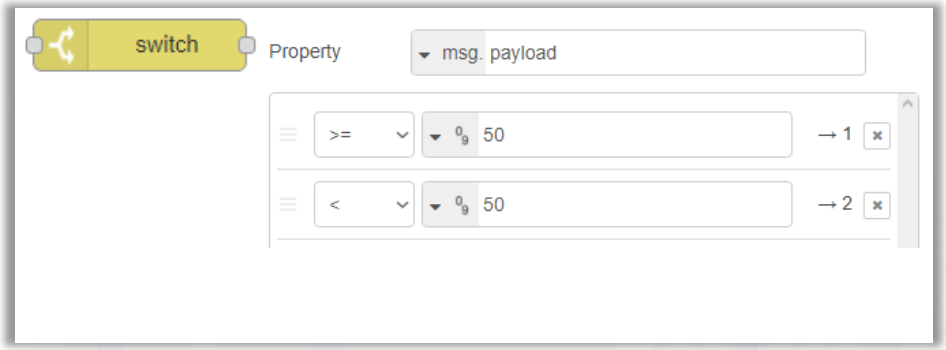
You have 15 Minutes!

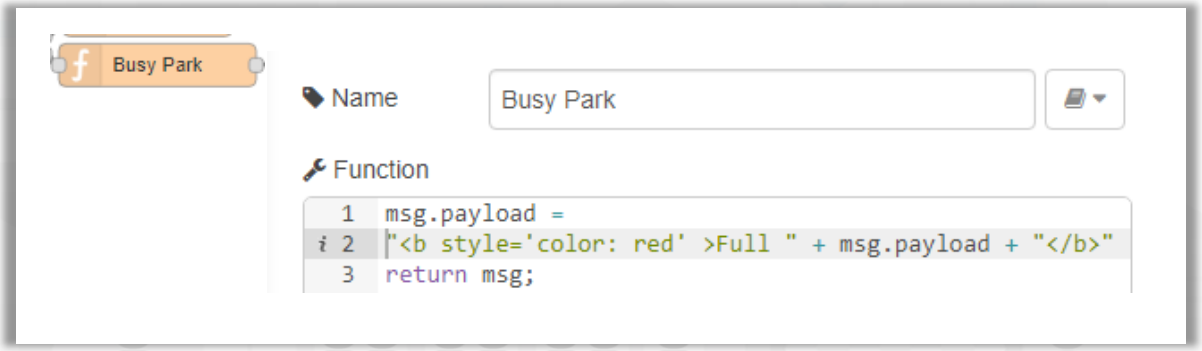
Ex1: Your NickName: ……



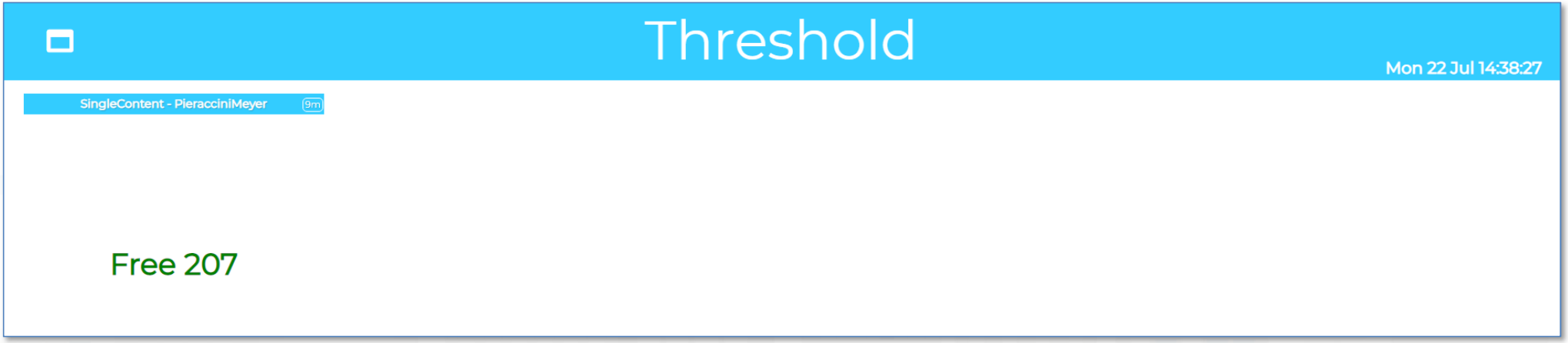
One Possible Solution

Nodes configuration





Resulting Dashboard



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

Data Processing with IOT Application (DEMO)

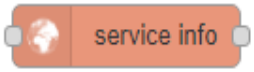
Example of more Complex IOT Application

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

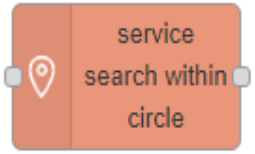
- reads a realtime values from a list of services,
- makes the sum of the value and
- publish the result on a dashboard

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)



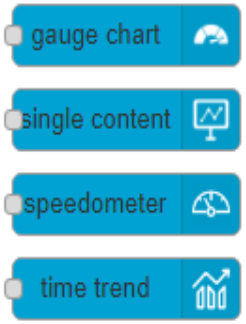
Requests detailed information for a specific service on the platform (such as a car park, hotel, etc.)



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.

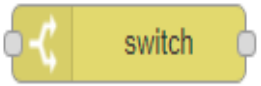
Executes Javascript code. For example, exploiting data arrived on input message and producing an output message in JSON



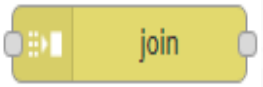
Display values in different modes on a dashboard. The node called single content accepts strings, numbers and html. The others only accept numbers.

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.

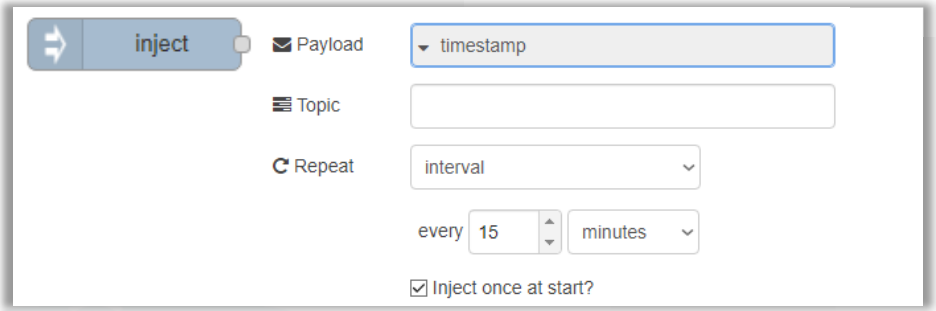


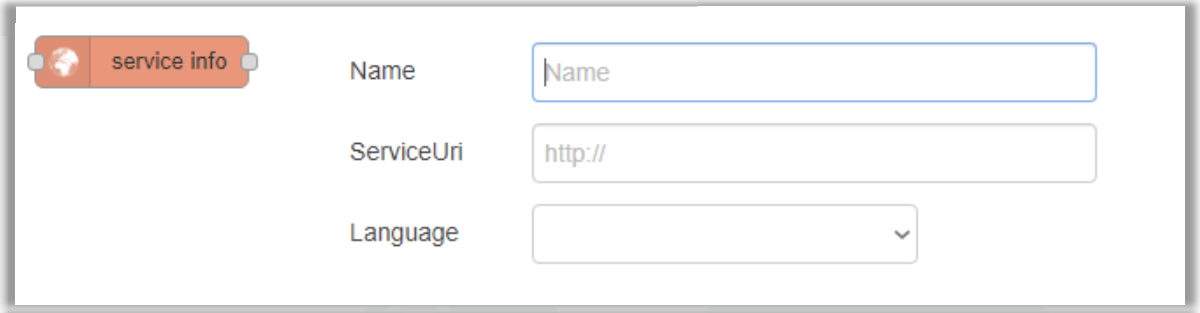
Treads the input message on possible different outputs based on a comparison made on the input message.

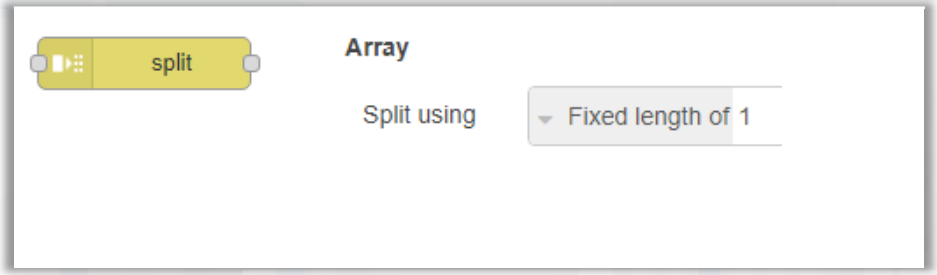


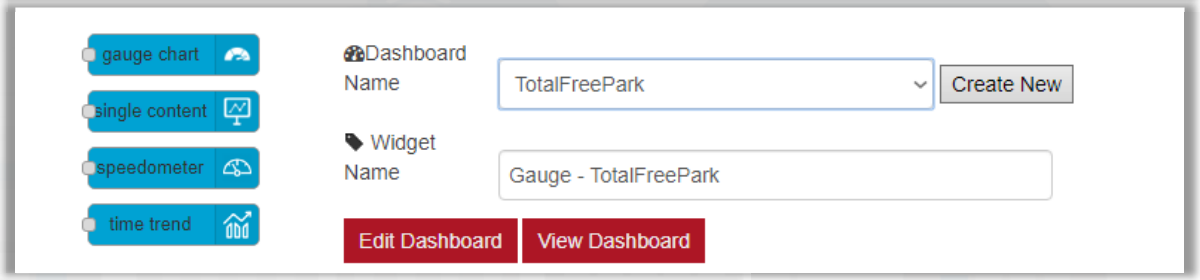
Operates in reverse order to the split. Joins the incoming messages in the mode indicated in the configuration.

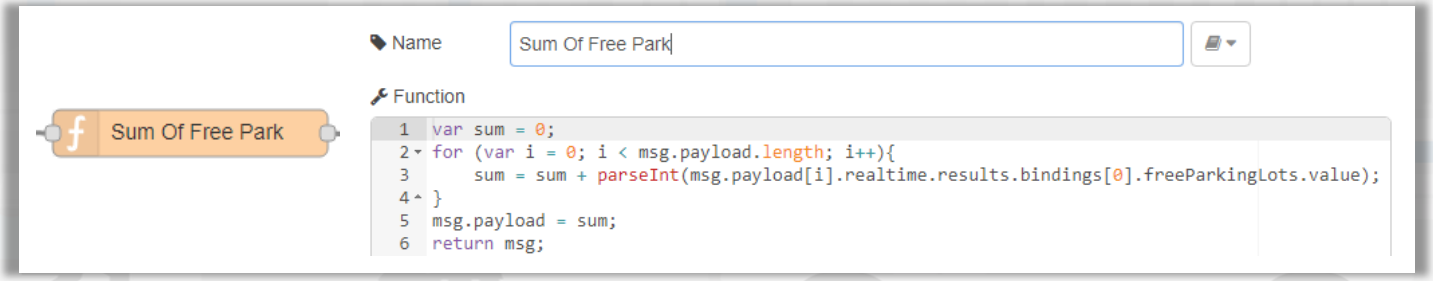
Nodes configuration 1/2



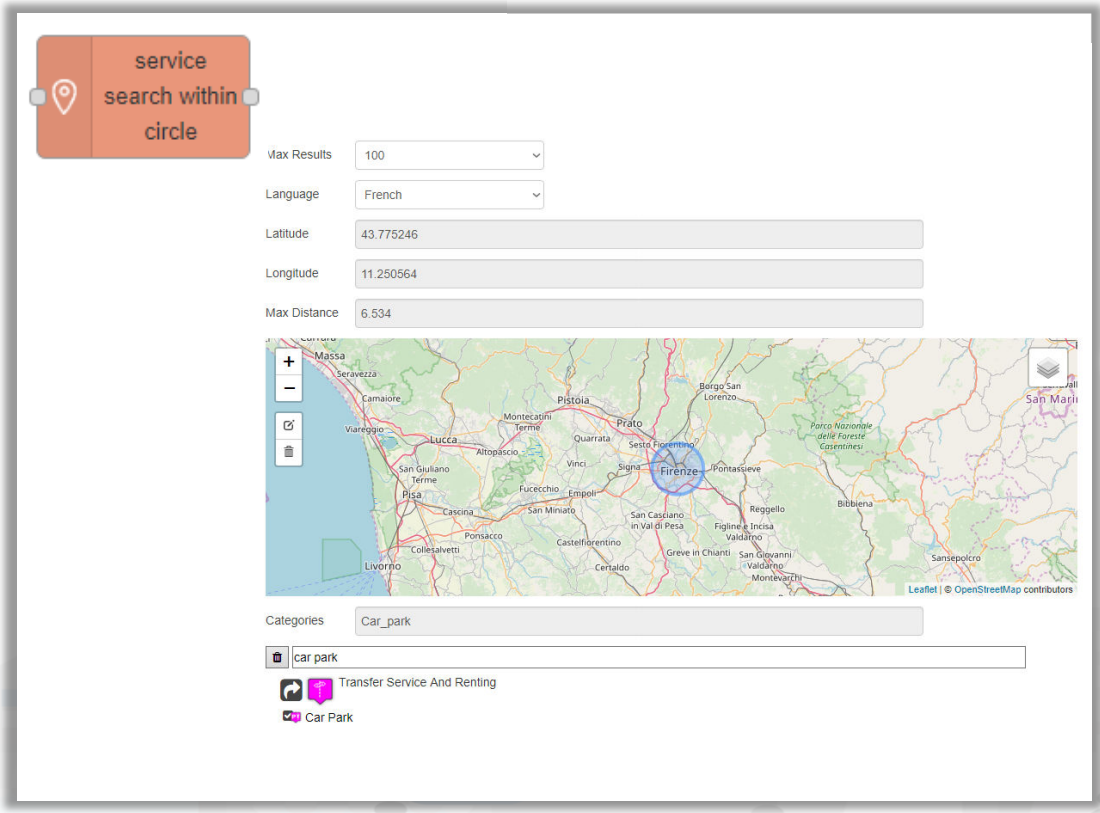


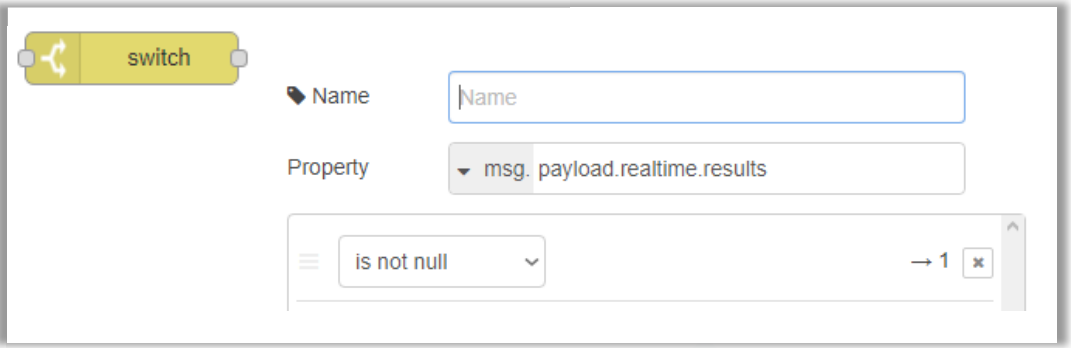


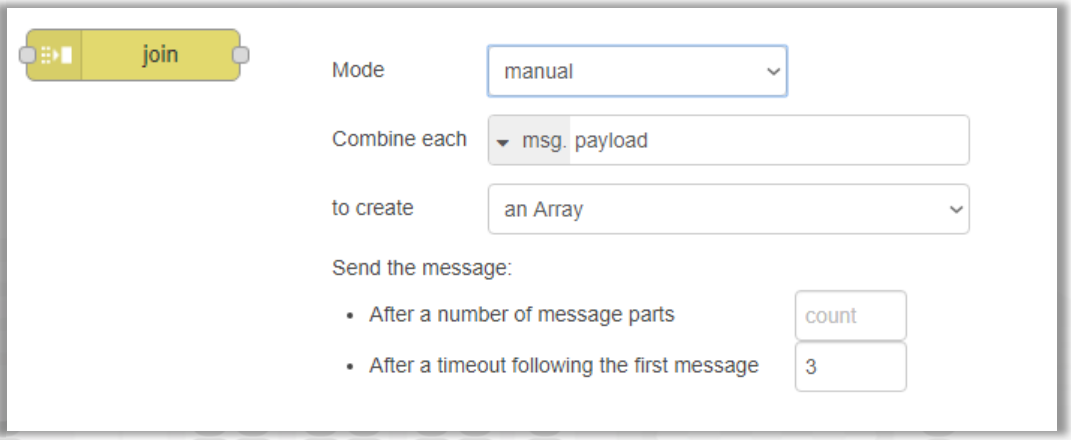




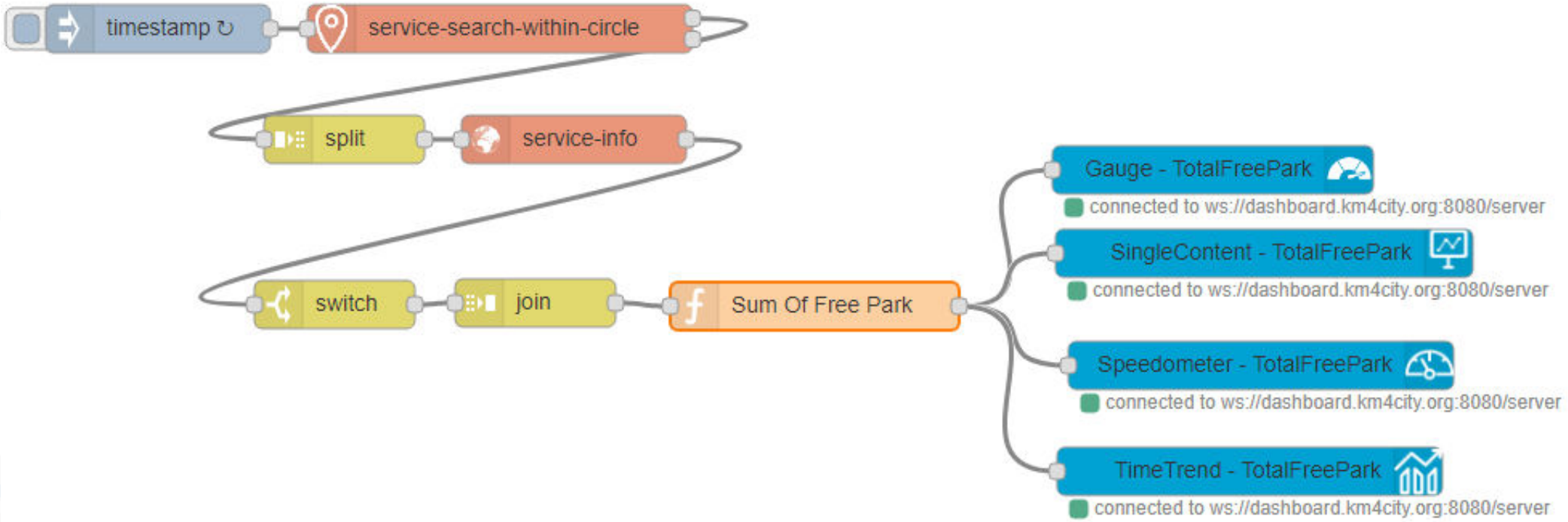
Nodes configuration 2/2



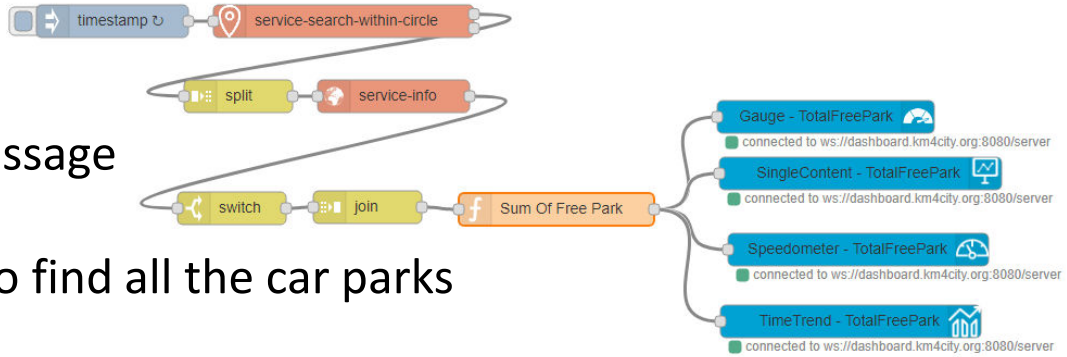




Nodes connections

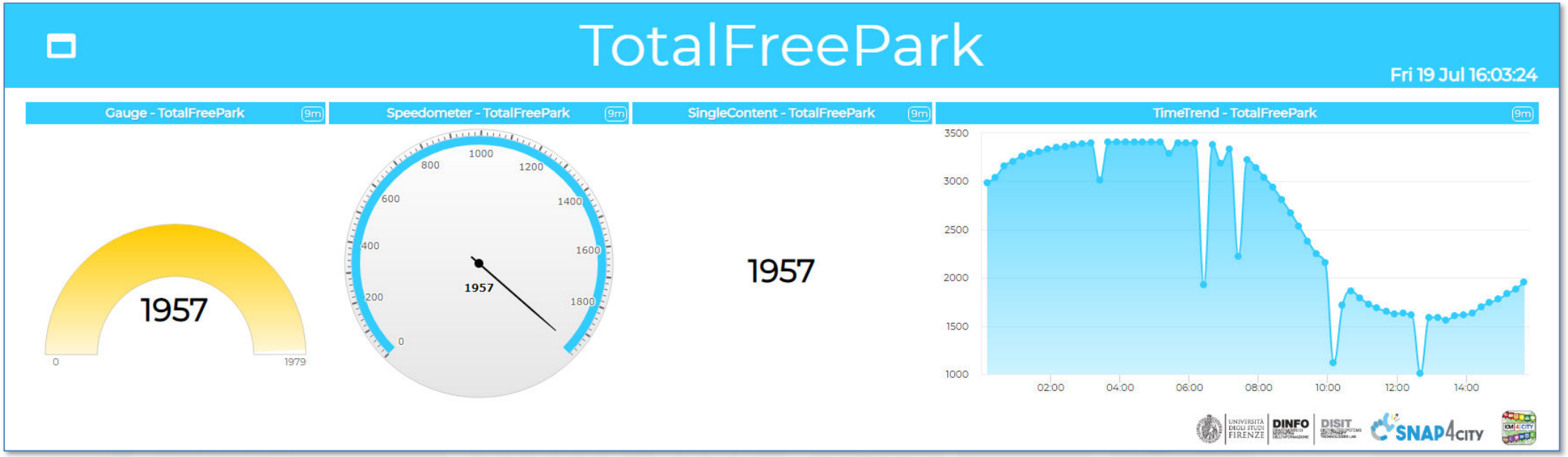


Nodes explanation 1/2



- 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==

Processing data with IOT Applications (Exercitation)

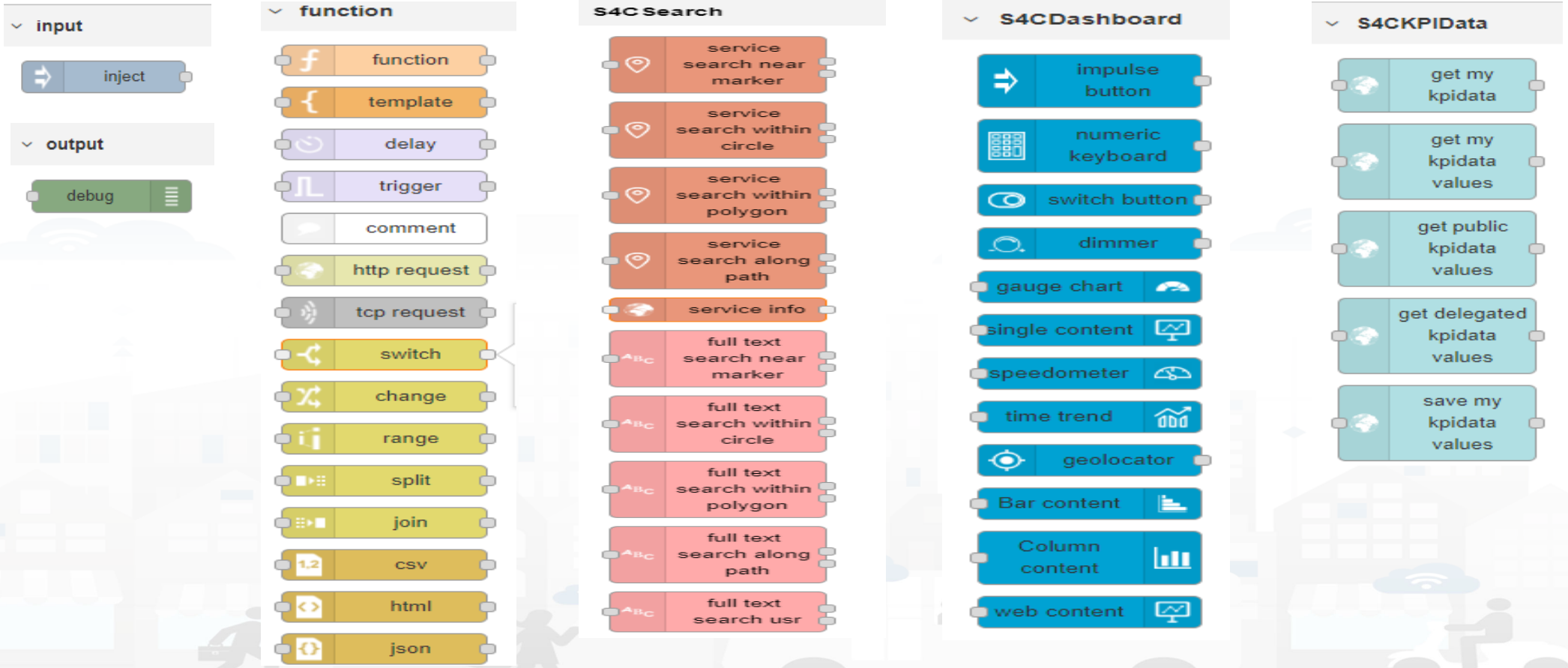
Average IoT Application

Create an IOT Application / flow that:

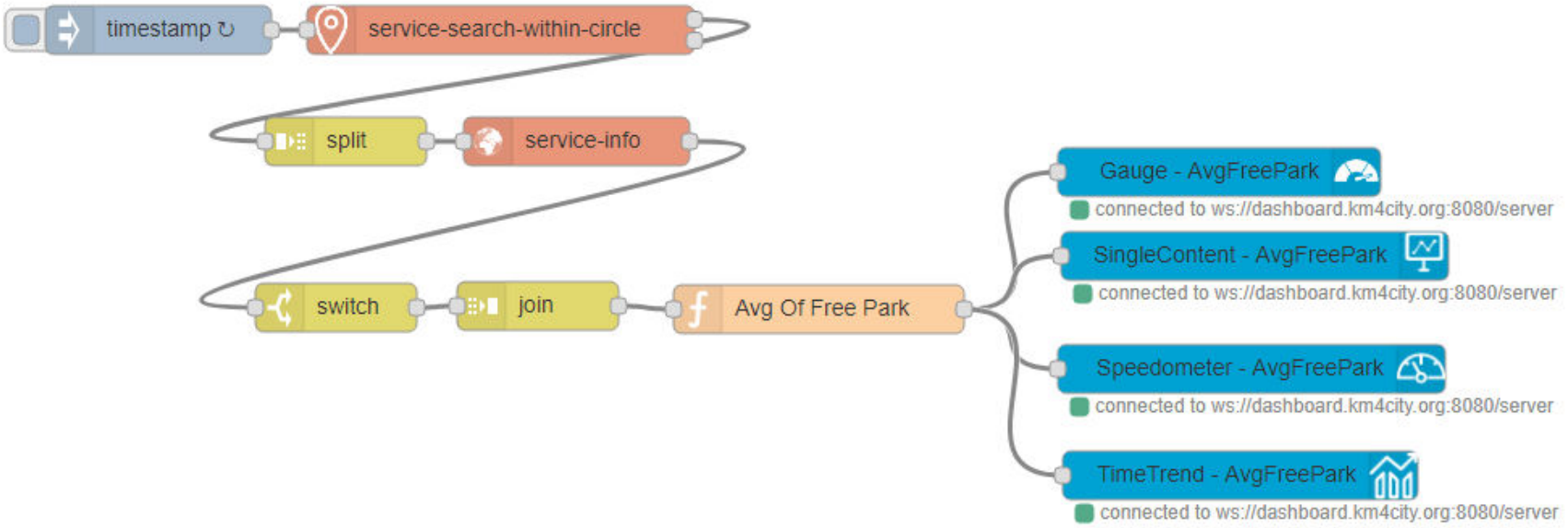
- reads a value from a list of service, for example the car parks in the Florence City Area, as seen in previous demo and
- calculates the average of Free Parking Lots and
- sends the value on a dashboard with the four possible nodes seen in the demo.

Execution Time: 20 Minutes

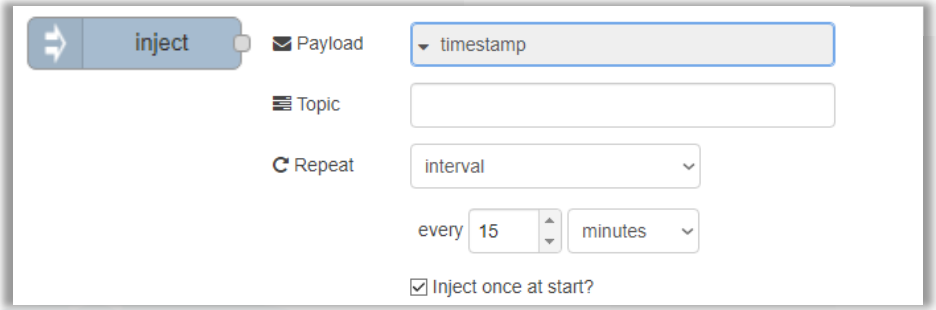
Ex2: Your NickName: ……

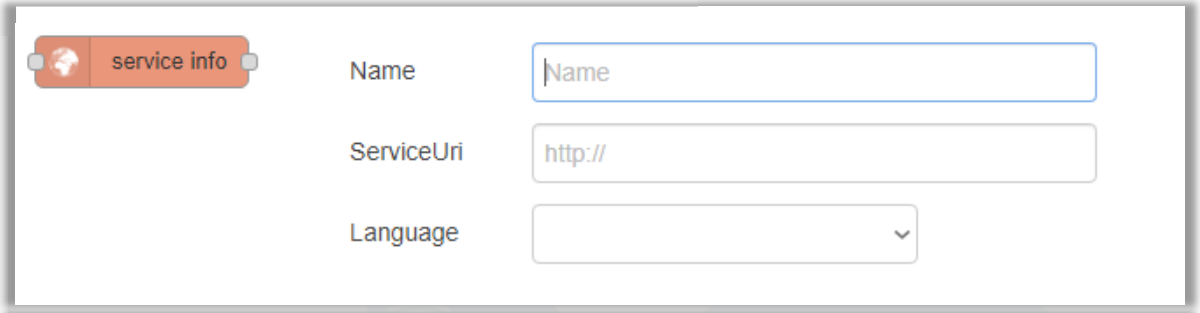


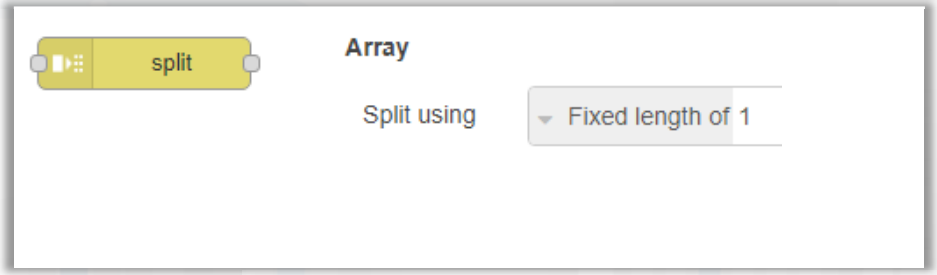
One Possible Solution

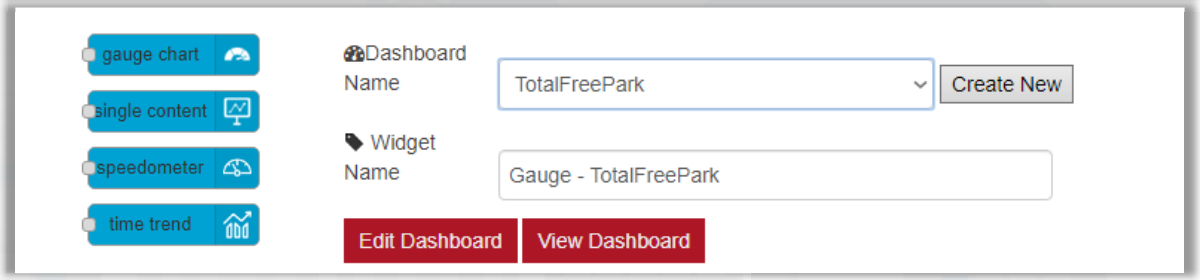


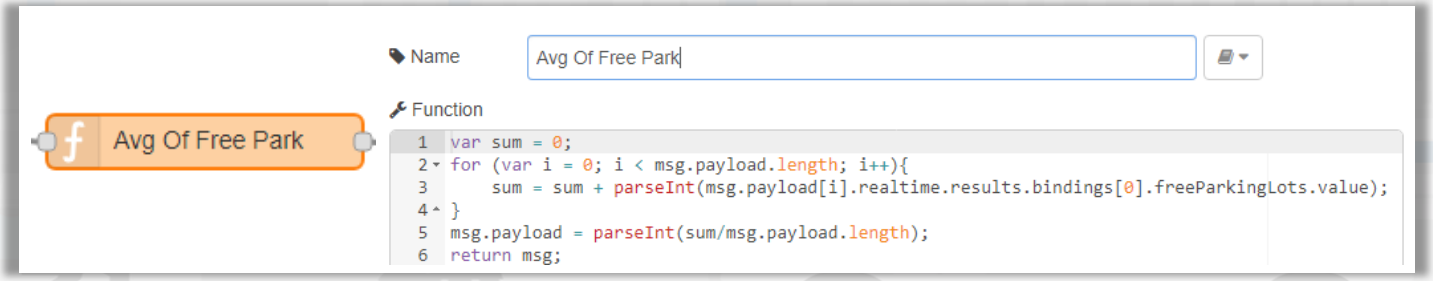
Nodes configuration 1/2



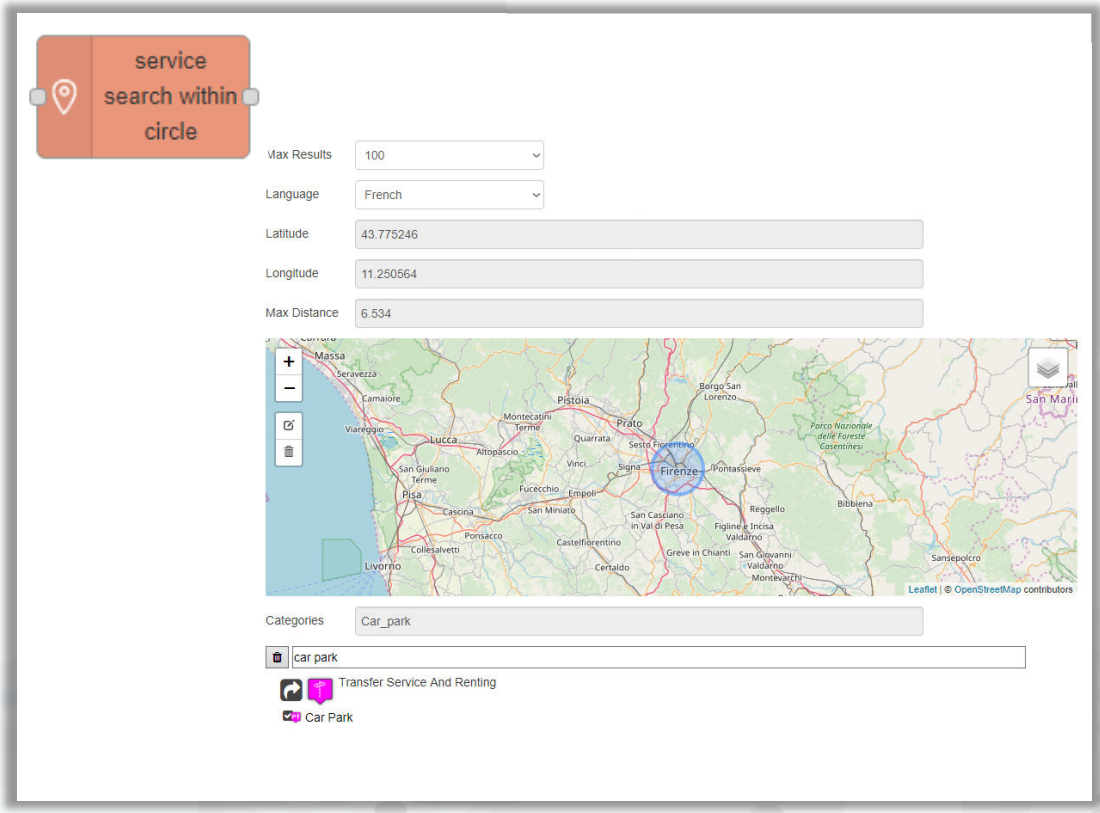


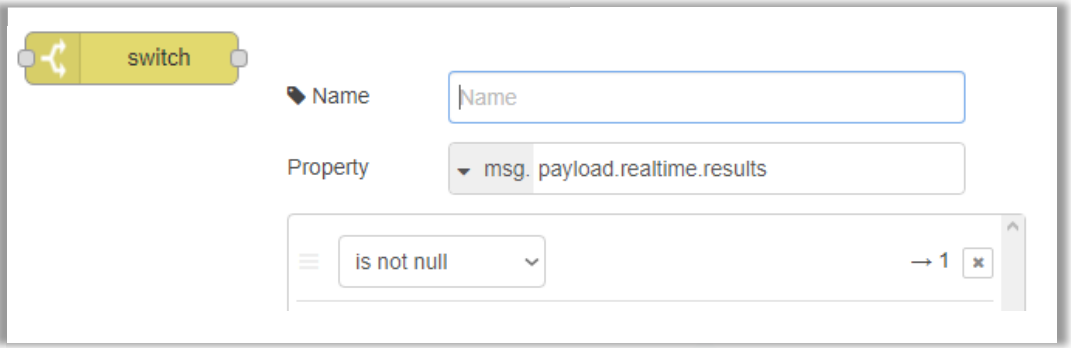


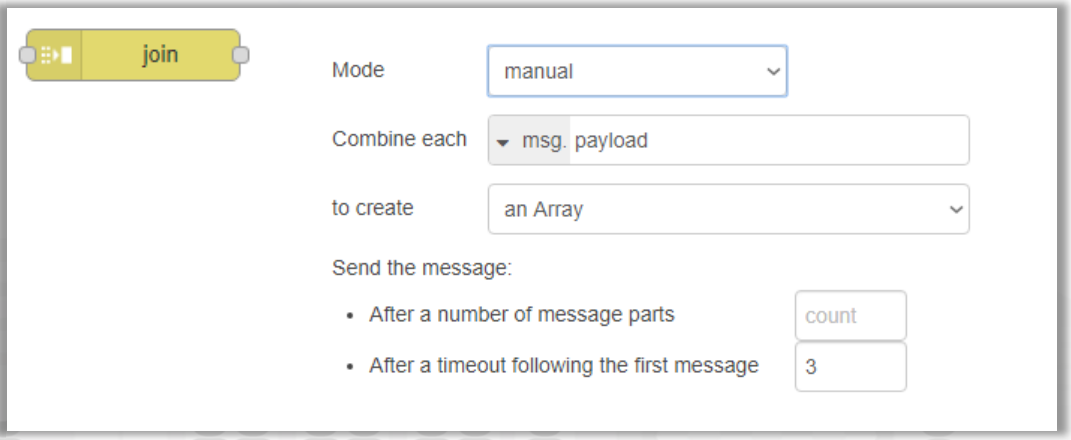




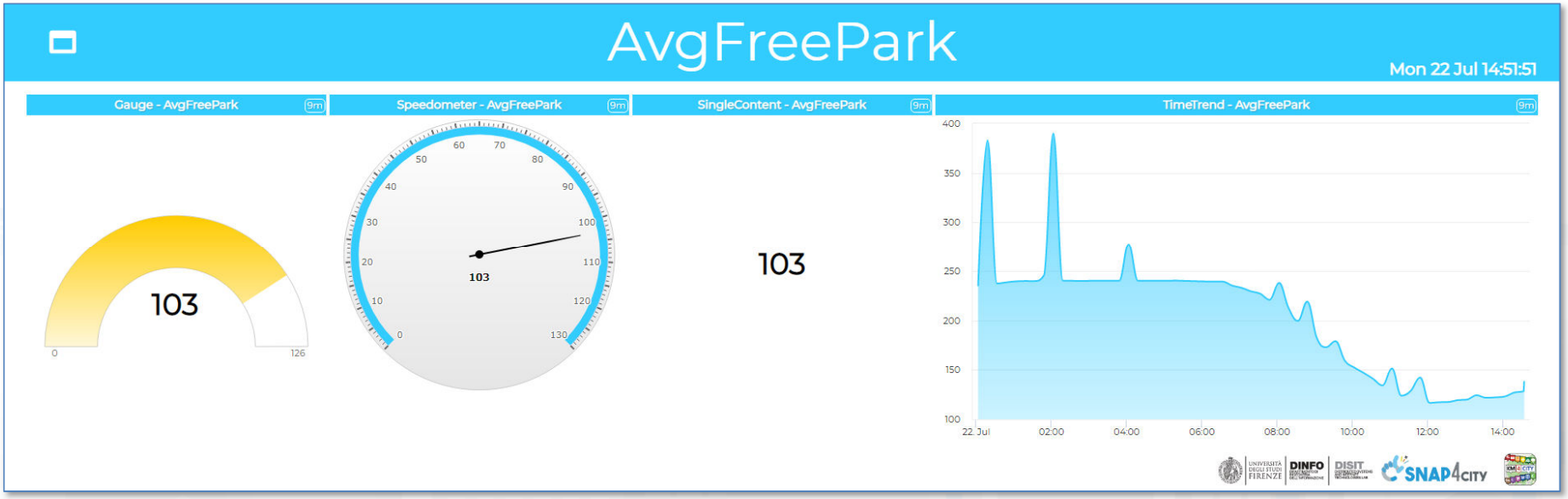
Nodes configuration 2/2



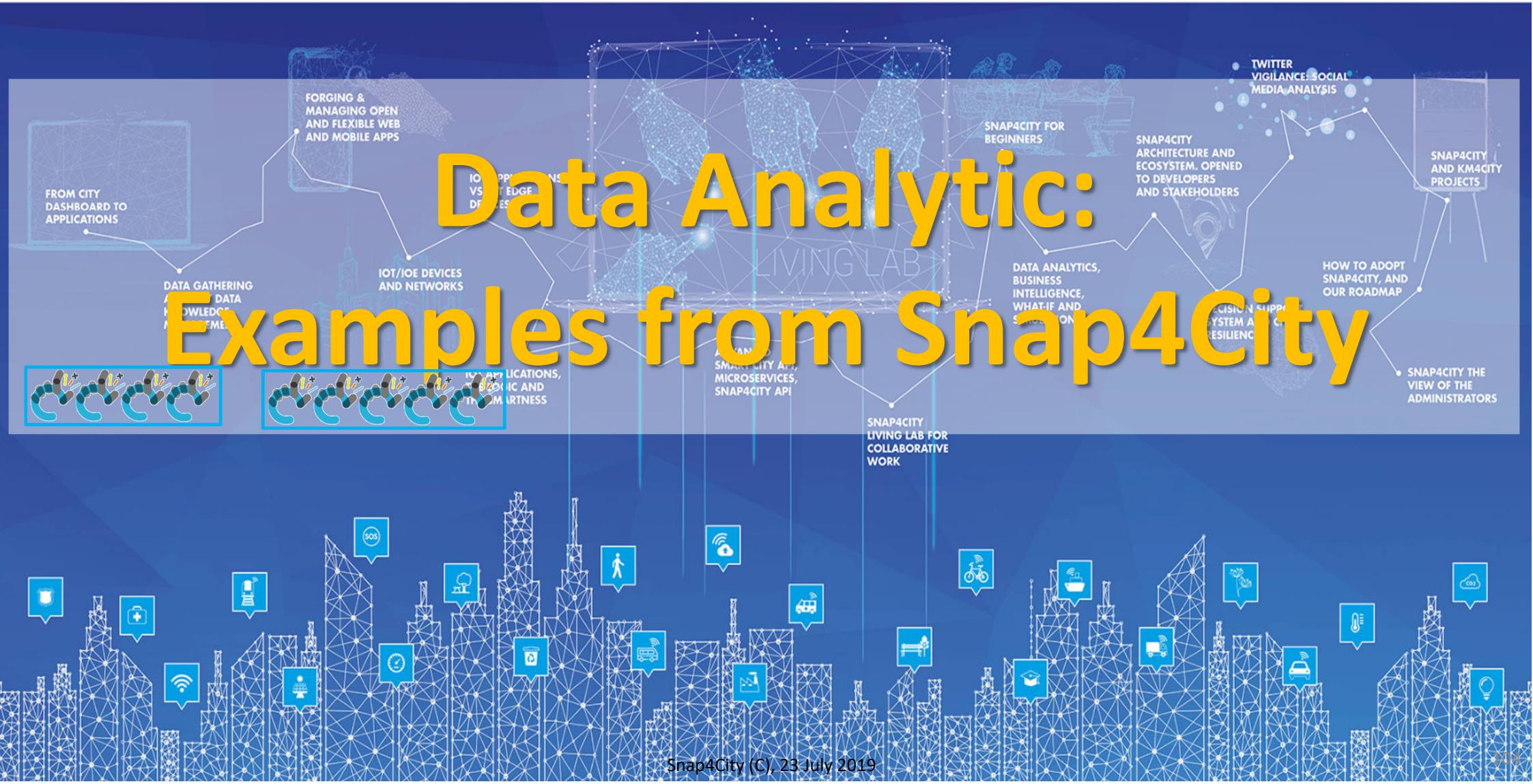




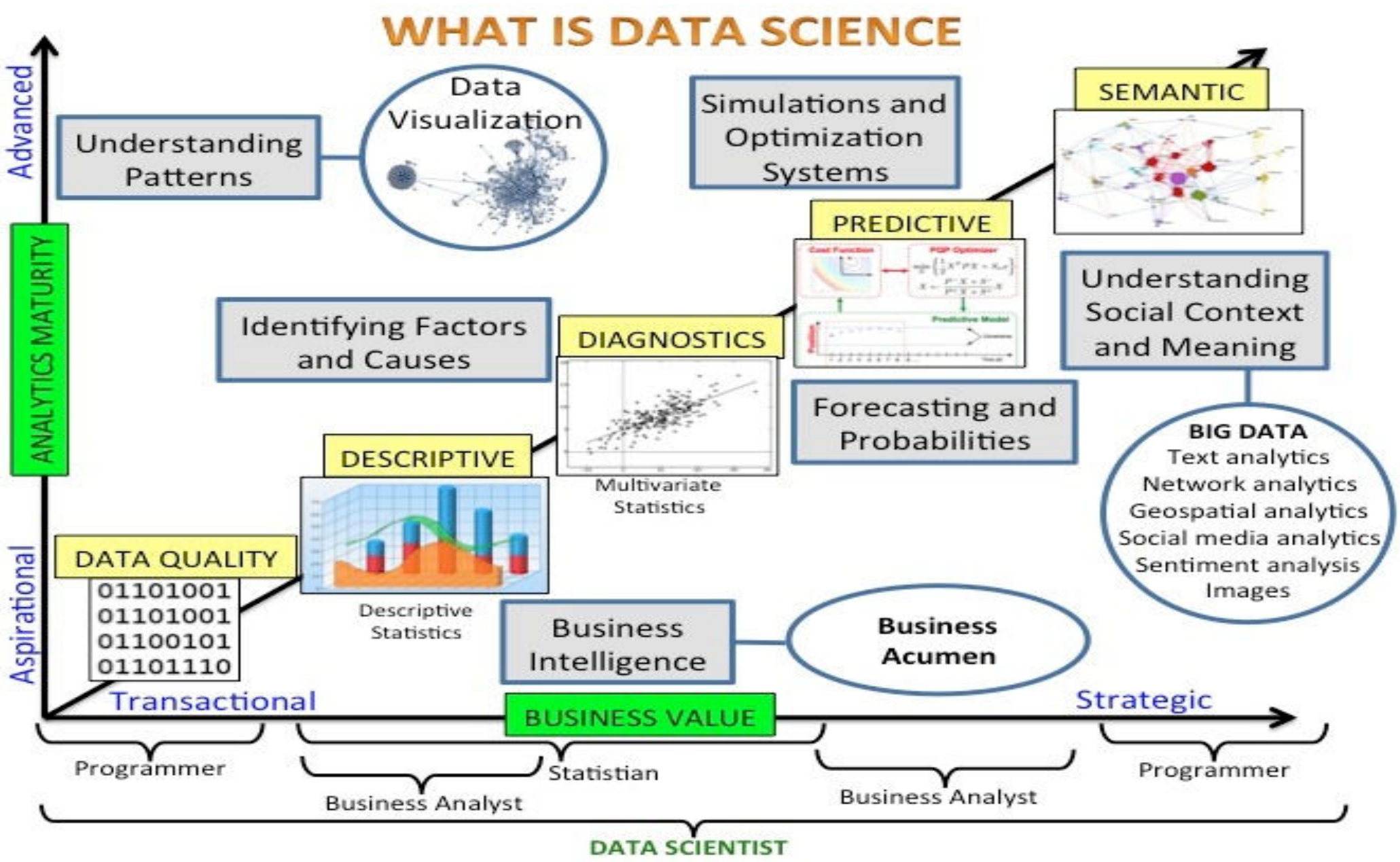
Resulting Dashboard



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



Data Analytic: Examples from Snap4City


WHAT IS DATA SCIENCE
Advanced
ANALYTICS MATURITY
Aspirational
Understanding Patterns
Data Visualization
Simulations and Optimization Systems
SEMANTIC
PREDICTIVE
Identifying Factors and Causes
DIAGNOSTICS
Multivariate Statistics
Understanding Social Context and Meaning
Forecasting and Probabilities
DESCRIPTIVE
DATA QUALITY
01101001
01101001
01100101
01101110
Descriptive Statistics
BIG DATA
Text analytics
Network analytics
Geospatial analytics
Social media analytics
Sentiment analysis
Images
Business Intelligence
Business Acumen
Transactional
BUSINESS VALUE
Strategic
Programmer
Statistian
Business Analyst
Business Analyst
Programmer
DATA SCIENTIST

From Simple Data Analytic to Complex Tools

- **Structural:**
  - **Data Ingestion, Quality Control** on data: data mining, anomaly detection, etc.
  - **Indexing** for fast search and retrieval: Geospatial, textual, temporal, mixt
- **Dynamical:**
  - **Analysis**: heatmap, hot places, distribution, statistical analysis
  - **Predictions** to inform and plan (e.g.: parking, people flow,)
  - **Anomaly detection** for Early Warning, Alerting
- **Special Analytics and Tools** → What-IF Analysis:
  - **Routing** for navigation: modal, multimodal, constrained
  - **Trajectories** of people flow
  - **Traffic Flow** reconstruction
  - **Origin Destination Matrices**
  - **Simulations**: demand vs offer

Snap4City and Data Analytic (summary)

- **Data Analytics** in Snap4City allows to create simple data processing as well as massive computing solutions exploiting statistics, machine learning, operating research, etc. for:
  - predictions, anomaly detection, early warning, OD Matrix construction, simulation, trajectories, what-if analysis, smart routing, heatmaps, etc.
- **can be developed** in:
  - R Studio / Tensor Flow, MapReduce, Java, Python, ETL, IOT Applications
- **can be shared** with other colleagues, and organizations via the Resource Manager

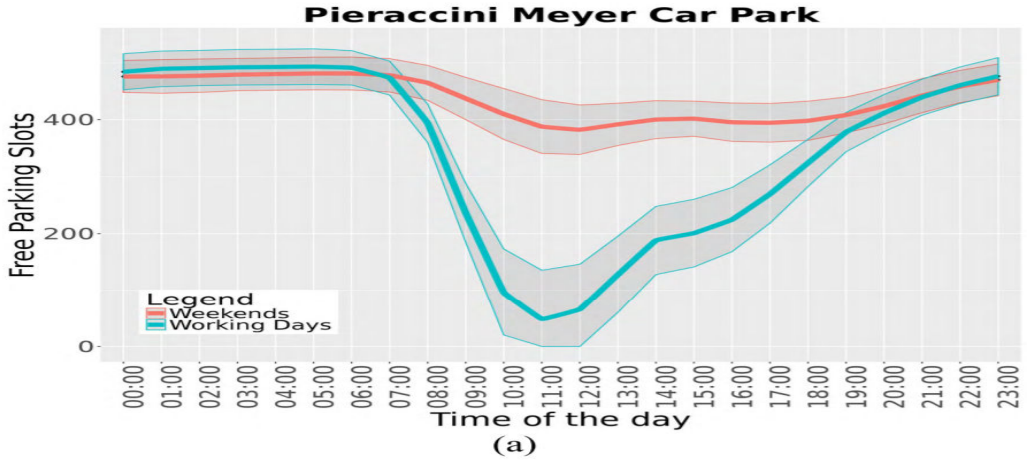
Development in R Studio (self training)

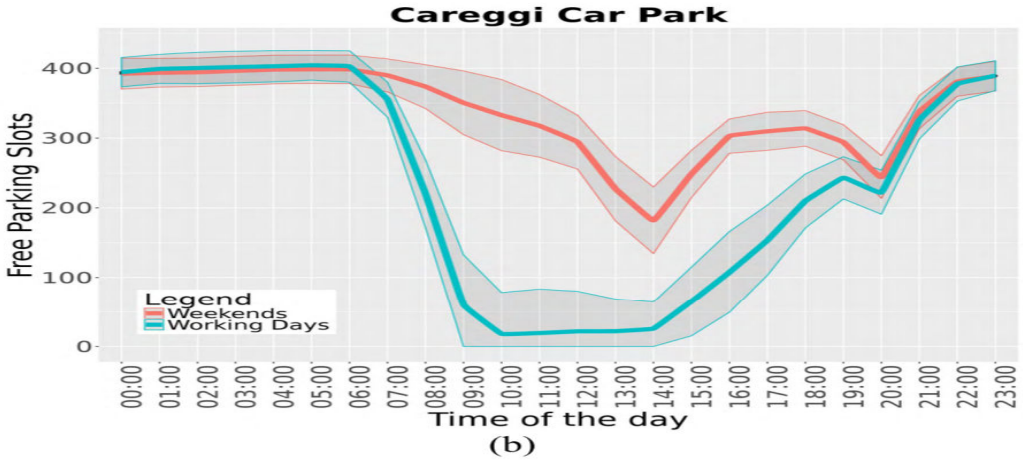
- R Studio Development
- TC7.2 - R Studio for Analytics, exploiting Tensor Flow
- TC7.4 - From R Studio process to MicroService for IOT application, data analytics, machine learning
- TC7.5 - Developing Data Analytics Processes
- US7. Data Analytics and related integration aspects

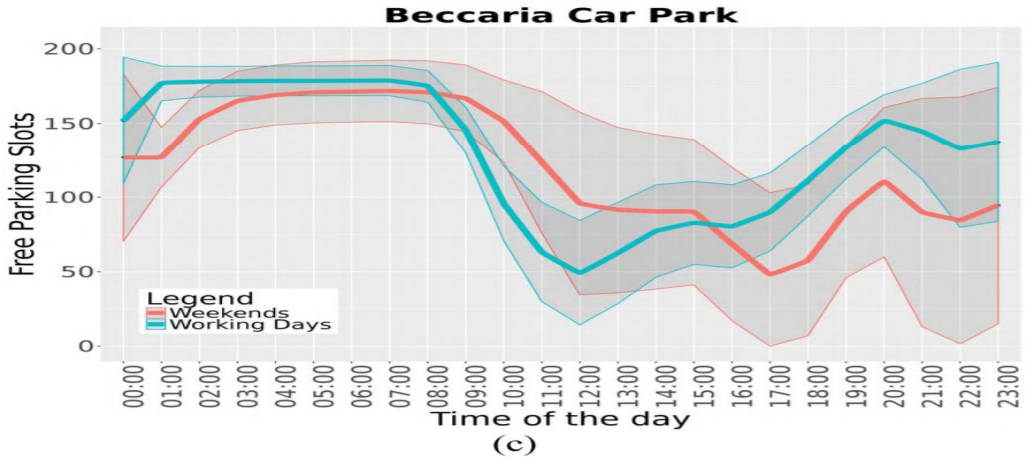
Smart Parking: predictions

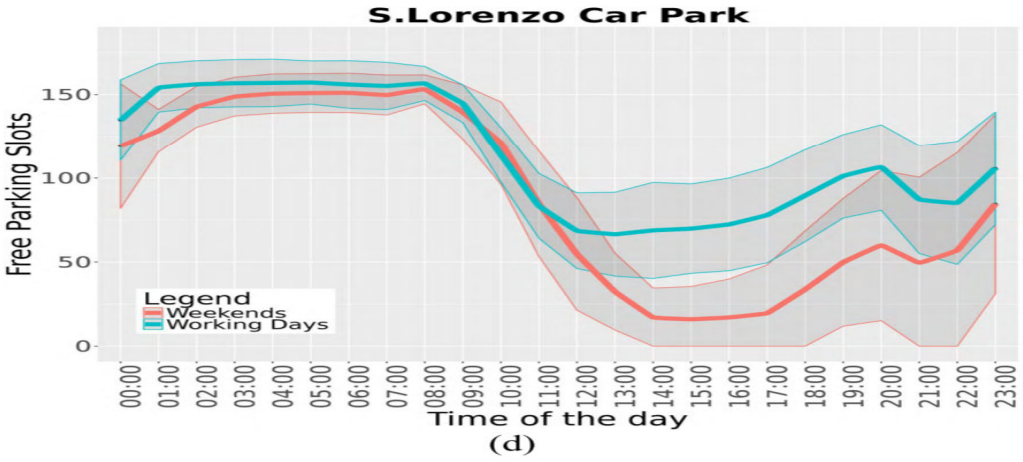
UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

Free Parking space trends









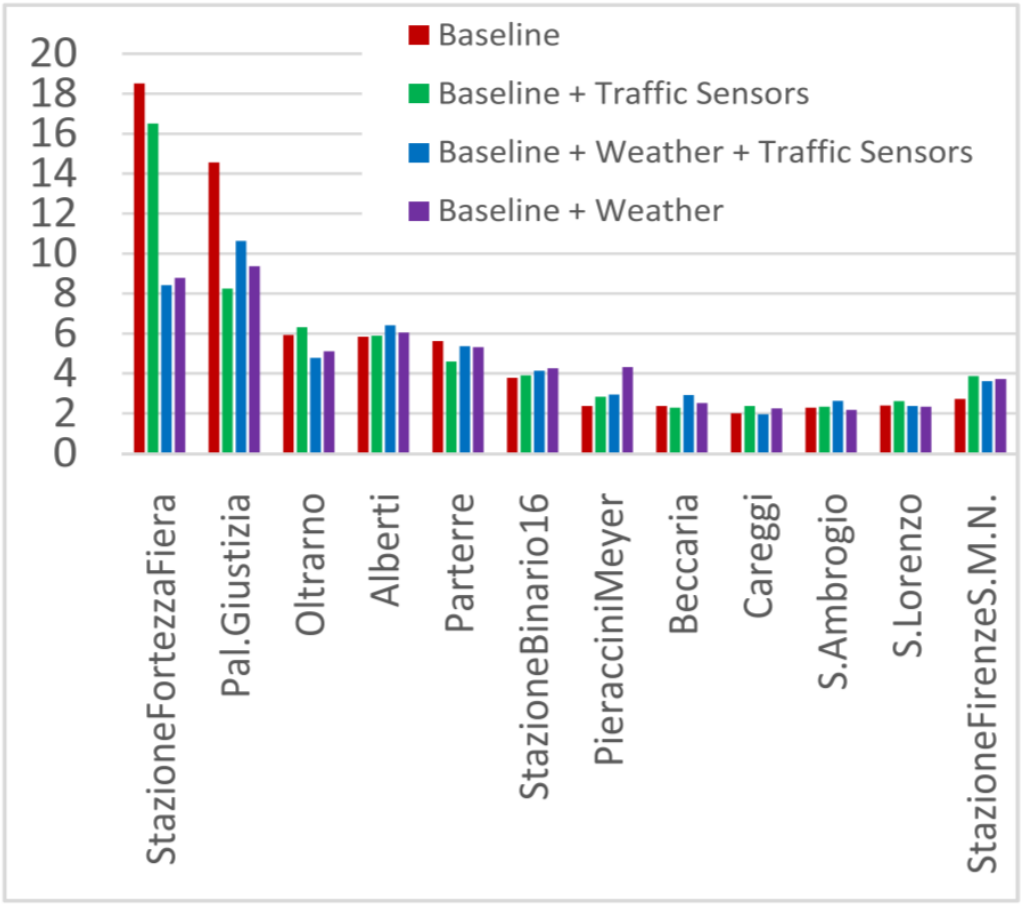
12 parking areas in Florence

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

Free Parking PREDICTIONS

C. Badii, P. Nesi, I. Paoli, "Predicting available parking slots on critical and regular services exploiting a range of open data", IEEE Access, preprint, 2018, https://ieeexplore.ieee.org/abstract/document/8430514/

Comparison Error	Forecasting Techniques		
	BRANN	SVR	RNN
Careggi car park			
MASE Night	34.85	16.29	20.01
MASE Morning	0.76	1.42	2.82
MASE Afternoon	1.89	4.34	3.66
MASE Evening	1.99	1.51	2.33
MASE	1.87	2.34	3.16
Pieraccini Meyer car park			
MASE Night	6.08	12.83	10.03
MASE Morning	0.86	1.27	4.90
MASE Afternoon	1.87	2.91	6.75
MASE Evening	1.36	1.57	10.23
MASE	1.37	2.06	6.67
S. Lorenzo car park			
MASE Night	10.33	11.81	18.34
MASE Morning	2.13	1.91	3.93
MASE Afternoon	2.70	3.15	2.37
MASE Evening	2.15	3.09	3.82
MASE	2.72	3.21	4.19
Beccaria car park			
MASE Night	9.32	7.80	12.47
MASE Morning	0.95	1.25	4.87
MASE Afternoon	2.49	2.14	2.45
MASE Evening	2.96	4.75	5.91
MASE	2.13	2.67	4.85



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

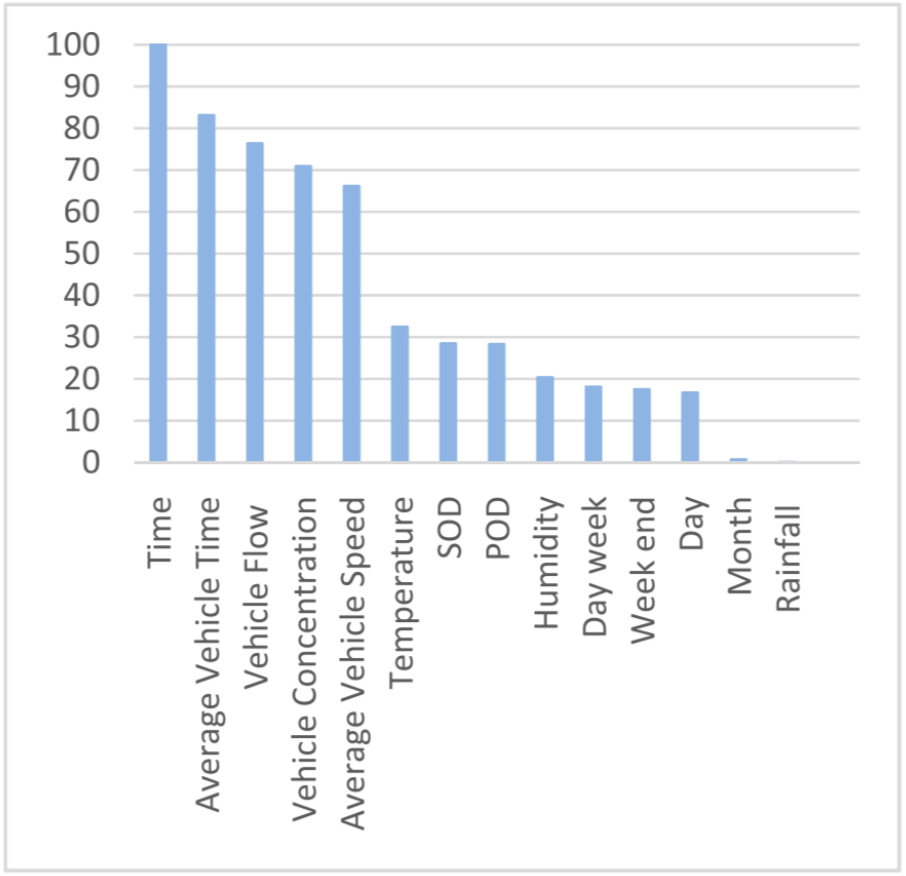
Free Parking PREDICTIONS

KM4CITY

Performances

Training	Forecasting Techniques			
	BRANN	SVR	RNN	ARIMA
Average Training processing time (sec)	76.3	9.1	598.7	9.2
Re-Training frequency	Daily	Daily	Daily	Hourly
Training period	3 months	3 months	3 months	3 months
Estimation	**BRANN**	**SVR**	**RNN**	**ARIMA**
Average Estimation time (sec)	0.0031	0.0052	0.034	0.0015
Estimation frequency	Hourly	Hourly	Hourly	Hourly
Estimation predicted period	1 hour	1 hour	1 hour	1 hour

Relevance of Variable



Free Parking Predictions

Careggi car park			
Model features	BRNN model results		
	R-squared	RMSE	MASE
Baseline	0.974	24	1.87
Baseline + Weather	0.975	24	1.75
Baseline + Traffic sensors	0.975	24	2.04
Baseline + Weather + Traffic sensors	0.975	24	1.87

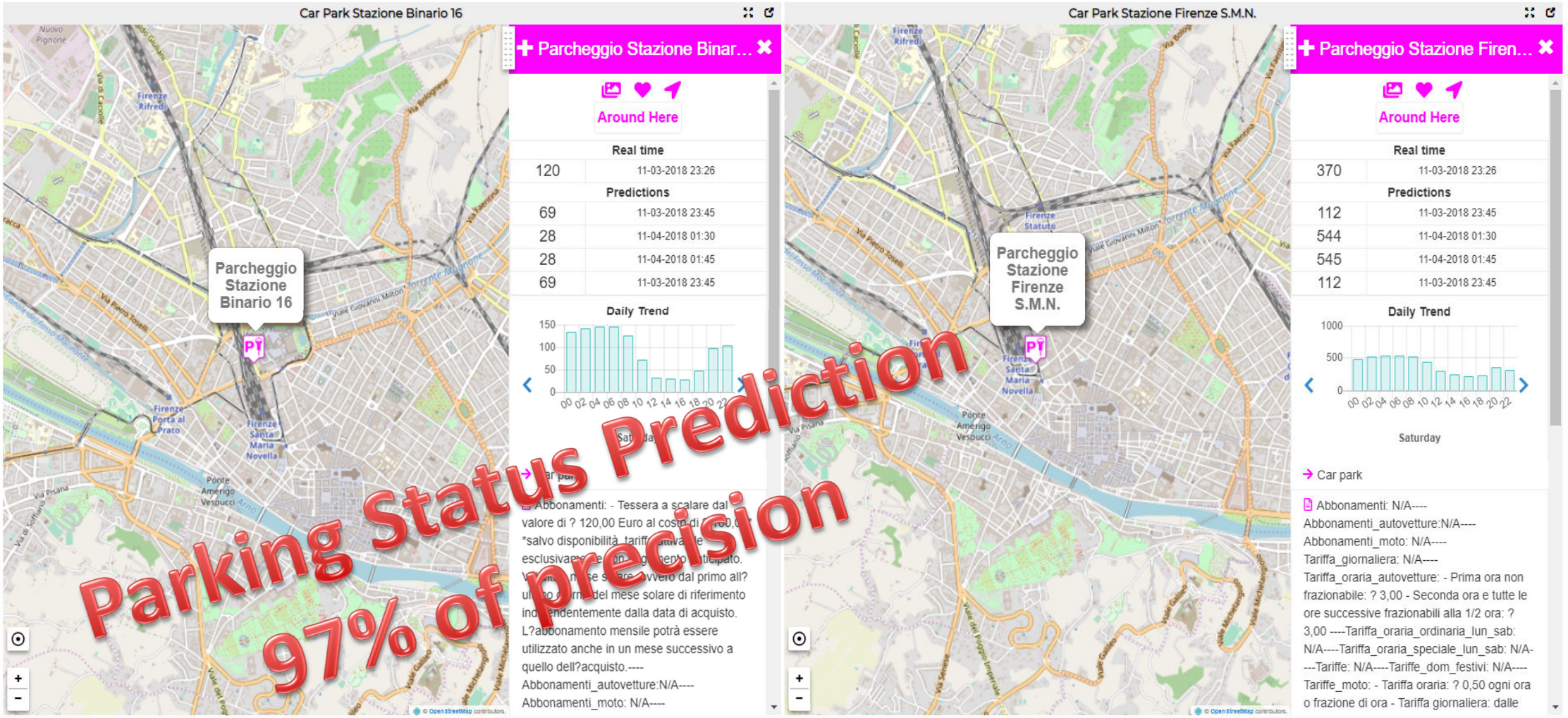
Active on Mobile Apps as:

- «Firenze dove cosa»
- «Toscana dove cosa»

Precision: 97,5%

Monitoring Station for Parking

Sat 3 Nov 23:39:55



Parking Status Prediction

97% of precision

https://www.disit.org/dashboardSmartCity/view/index.php?iddasboard=MjQ2

Predictions on Parking

IEEE Access

- C. Badii, P. Nesi, I. Paoli, "Predicting available parking slots on critical and regular services exploiting a range of open data", IEEE Access, preprint, 2018, https://ieeexplore.ieee.org/abstract/document/8430514/

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

SNAP4CITY KM4CITY

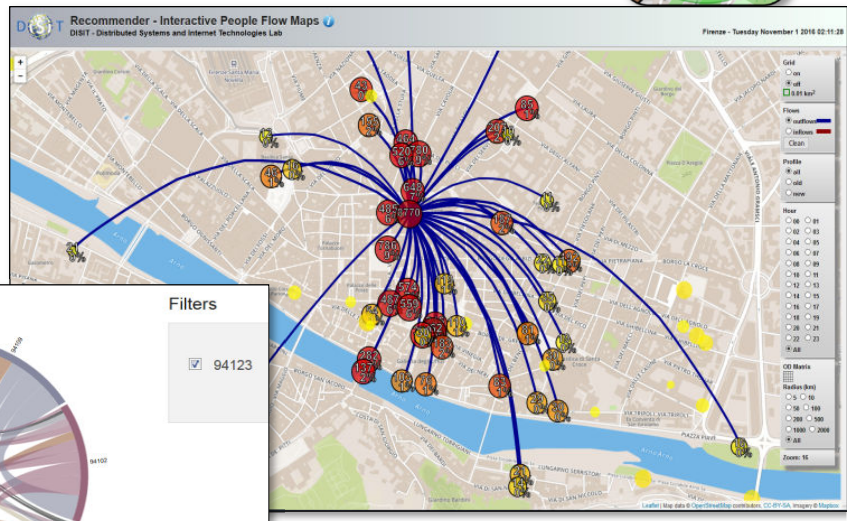
User Behaviour Analysis via Wi-Fi, OD Matrices, Trajectories

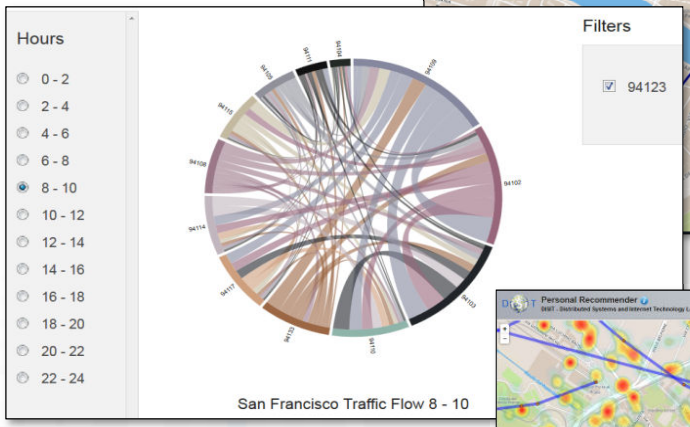
Sii-Mobility

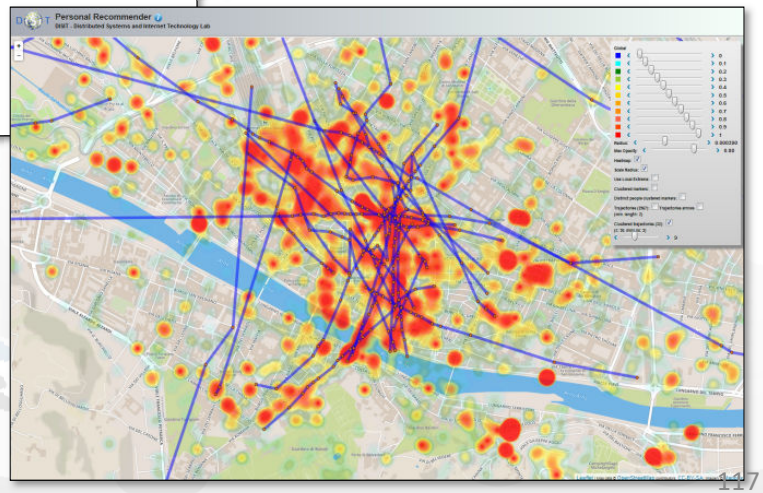
KM 4 CITY

User Behaviour Analysis

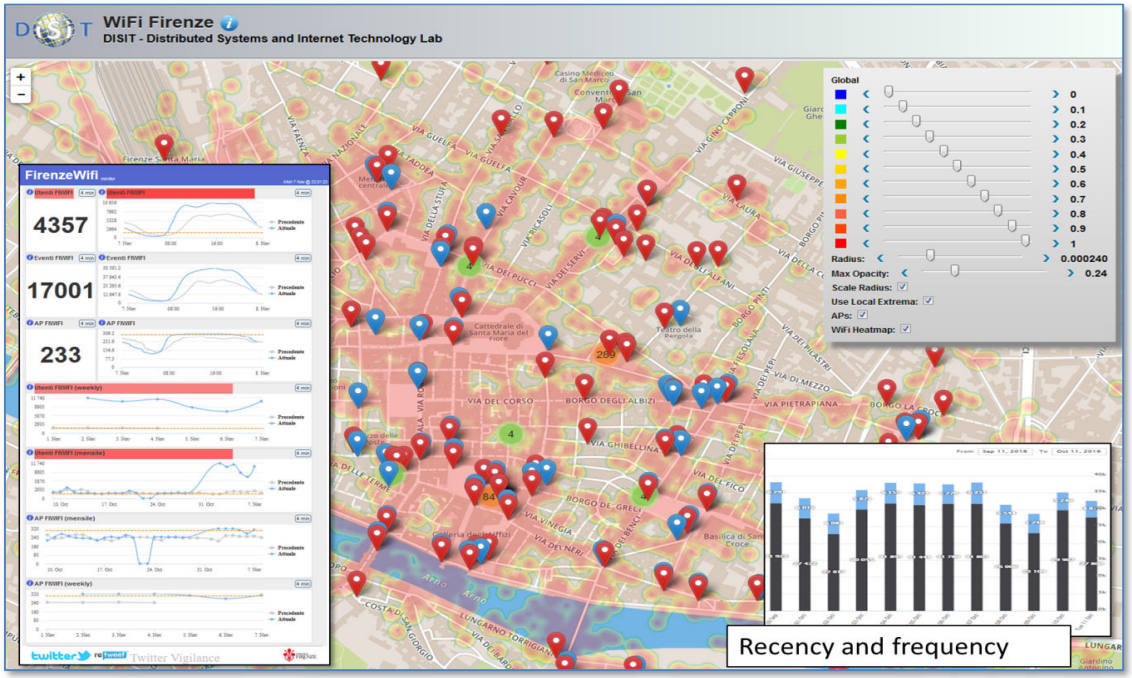
- **Monitoring movements by traffic flow sensors**
  - Spires and virtual spires
- **Monitoring movements from Mobile Cells**
  - Unsuitable for precise tracking and OD production
- **Monitoring movements from Wi-Fi**
- **Monitoring movements** and much more from mobile Apps

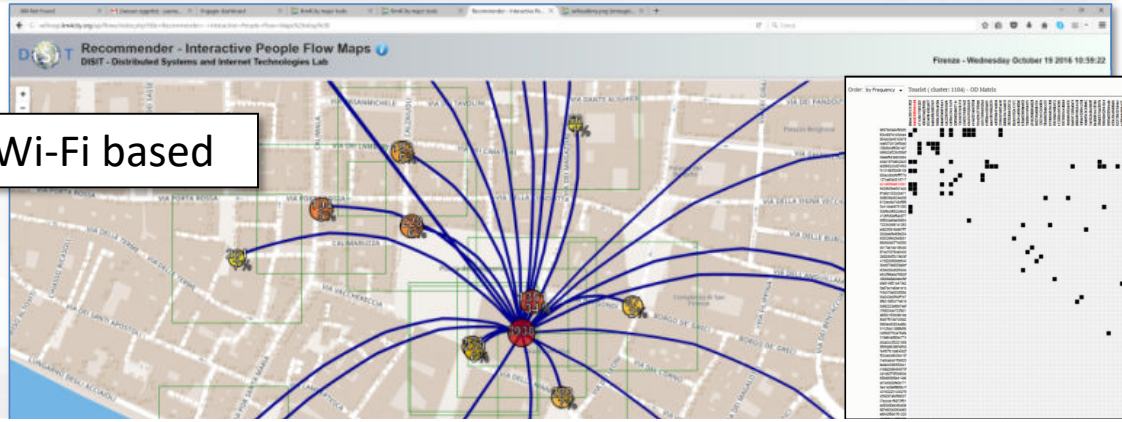




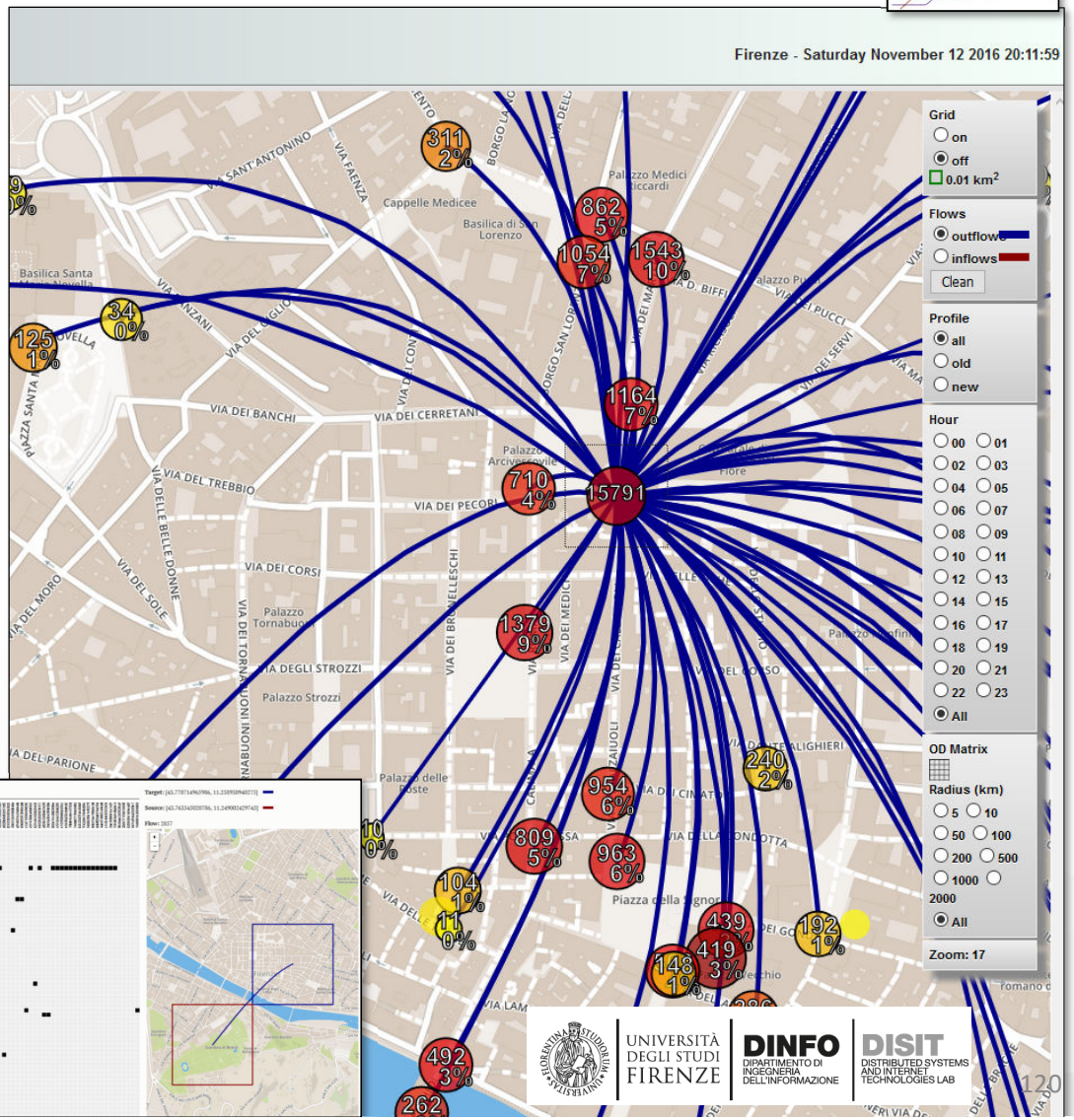


Origin Destination Matrix Estimation





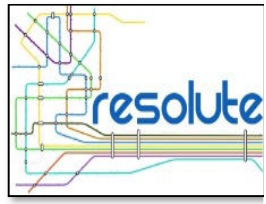
Wi-Fi based



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

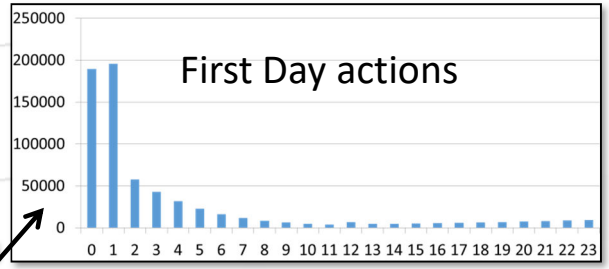
UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

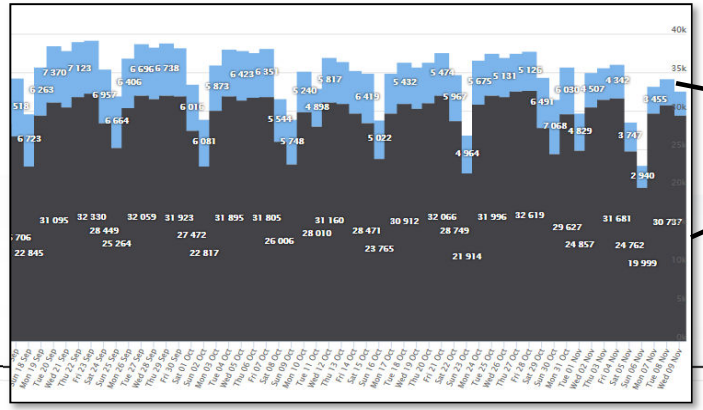
User Behaviour Analysis



Distinct APs: 343
Distinct APs (last 24 hours): 311
Distinct Users (last 180 days): 1102098
Distinct Excursionists (last 180 days, < 24 h): 687025

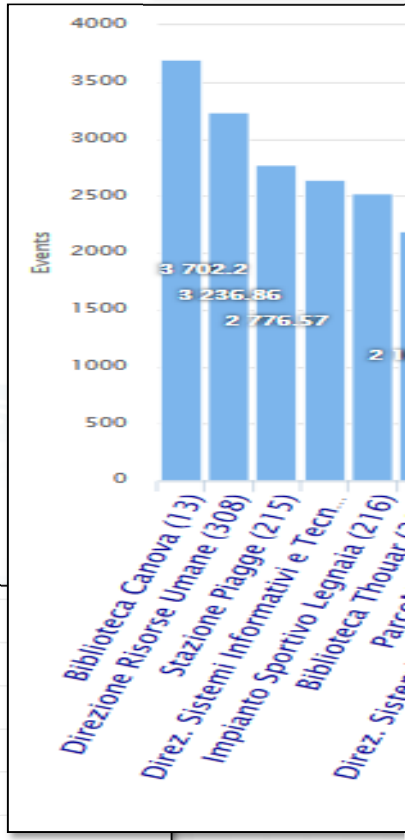
Excursionists

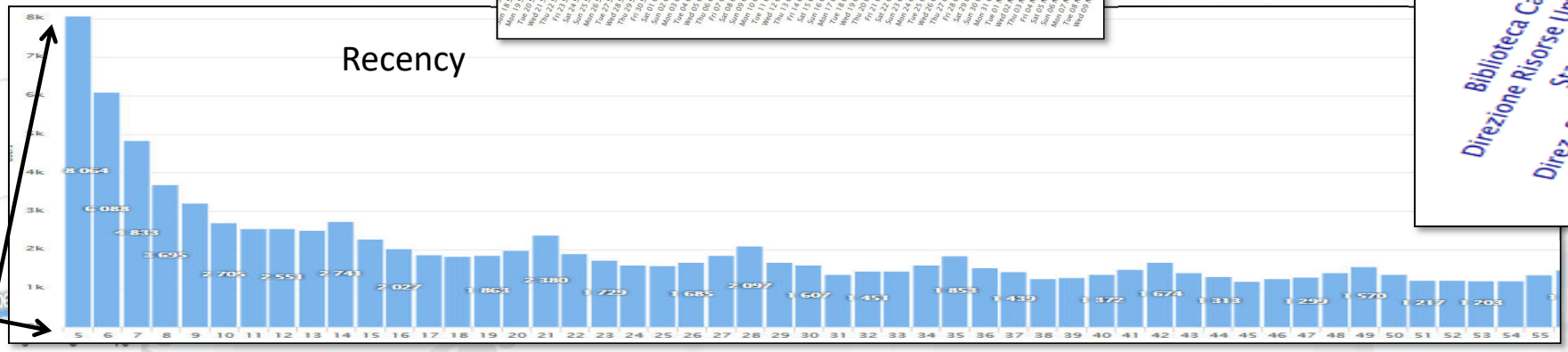




New City Users
VS
Returning

Where

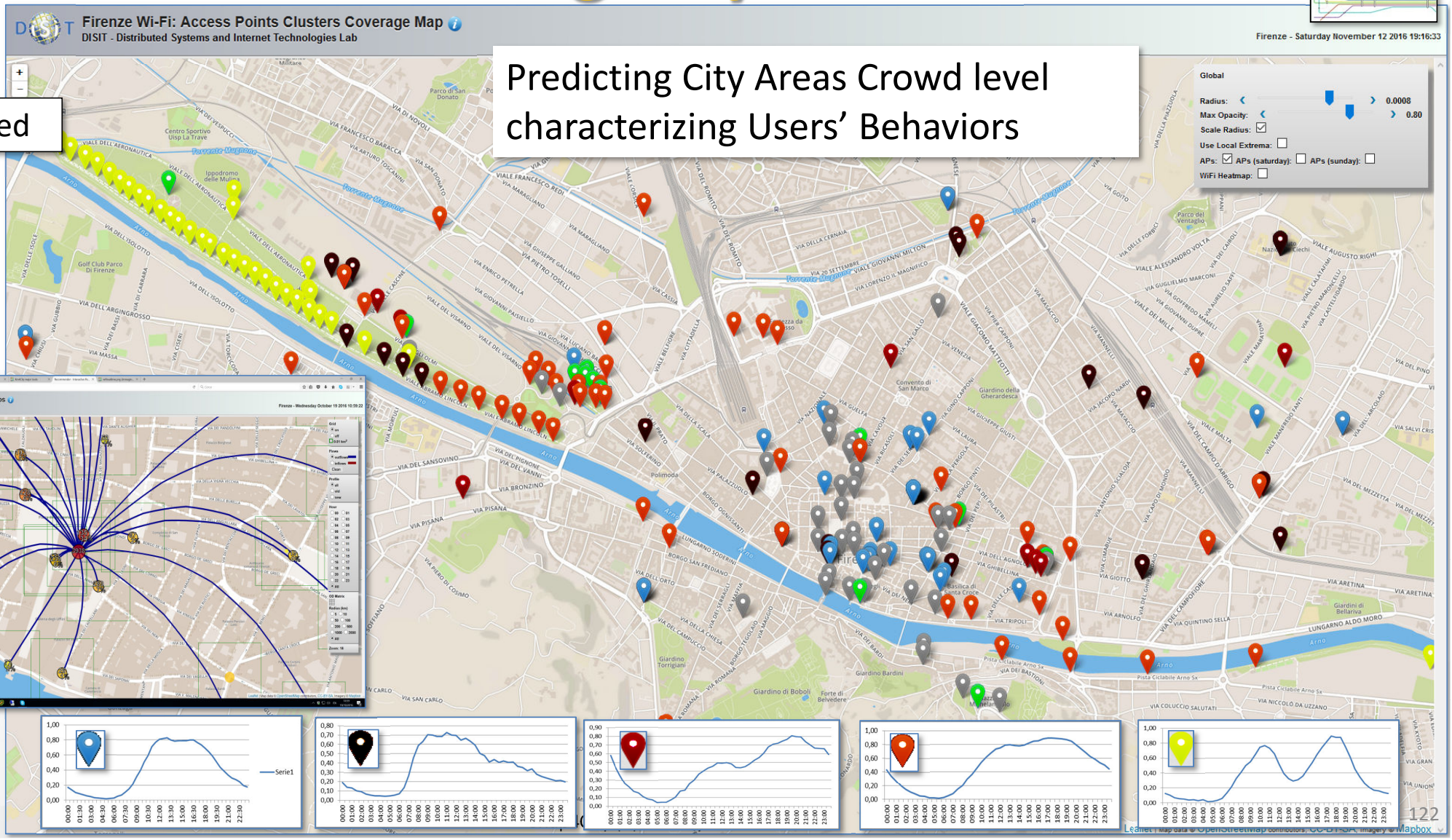




Characterizing City Areas

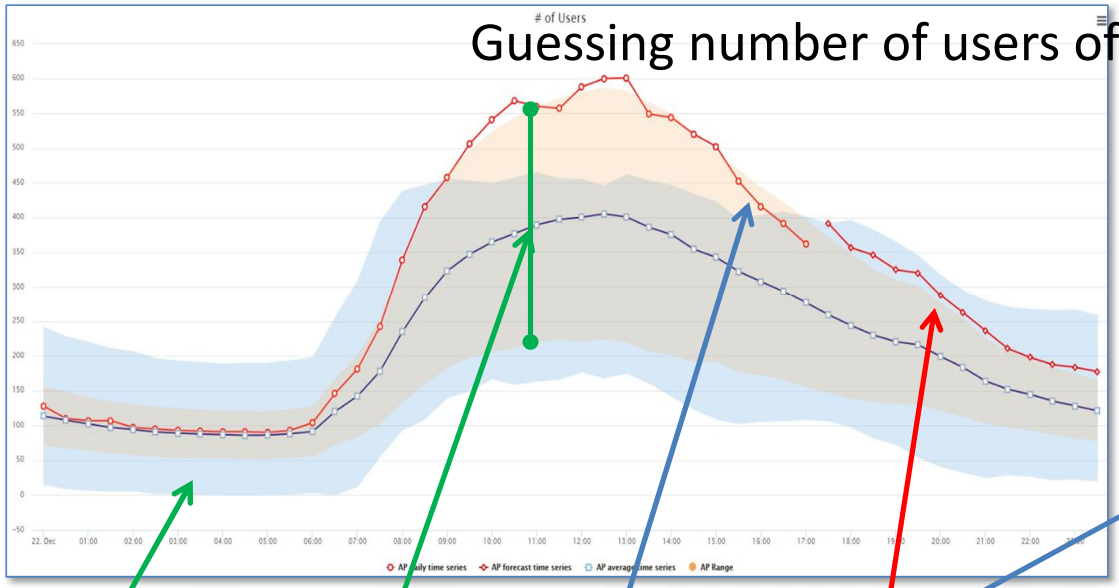
Wi-Fi based

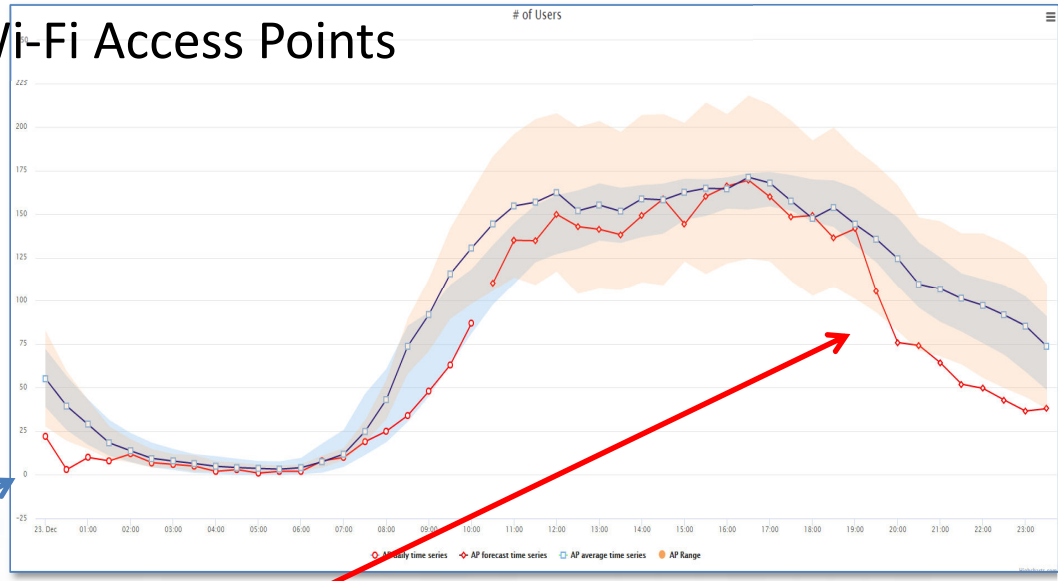
Predicting City Areas Crowd level characterizing Users' Behaviors



Prediction and Identification of Anomalies

Guessing number of users of Wi-Fi Access Points





Cluster confidence

AP average and confidence

Actual AP trend for today

AP prediction for the next time slot in the day on the basis of past weeks

Predictive precision of the 95%

User Behaviour Analysis

- P. Bellini, D. Cenni, P. Nesi, I. Paoli, "Wi-Fi Based City Users' Behaviour Analysis for Smart City", Journal of Visual Language and Computing, Elsevier, 2017. http://www.sciencedirect.com/science/article/pii/S1045926X17300083

ELSEVIER

Traffic Flow Reconstruction from Traffic Sensors Data

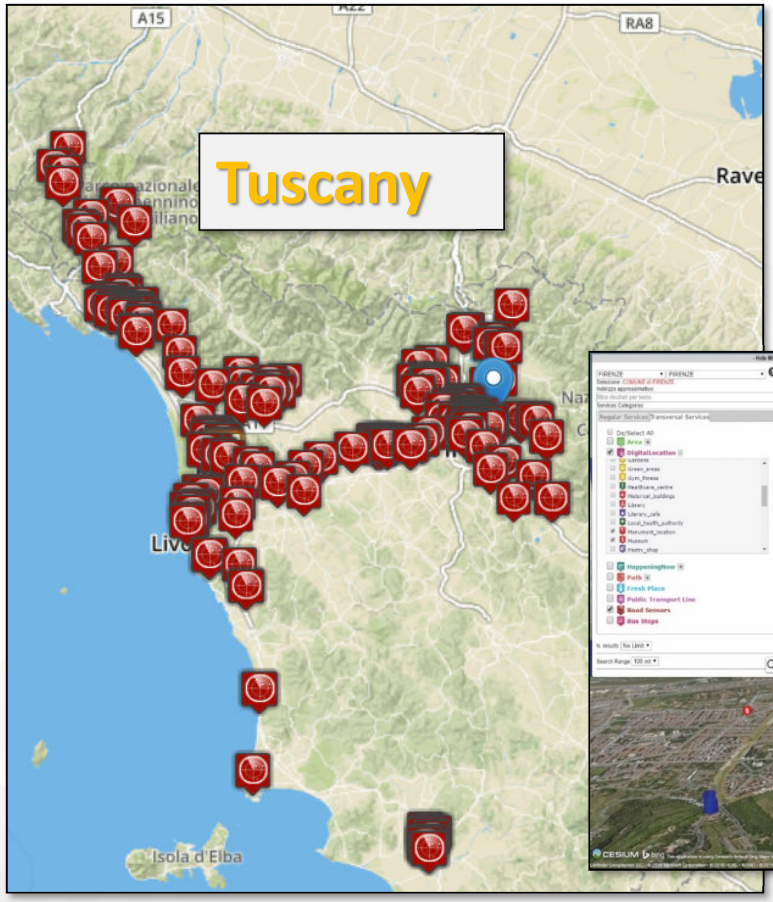
UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

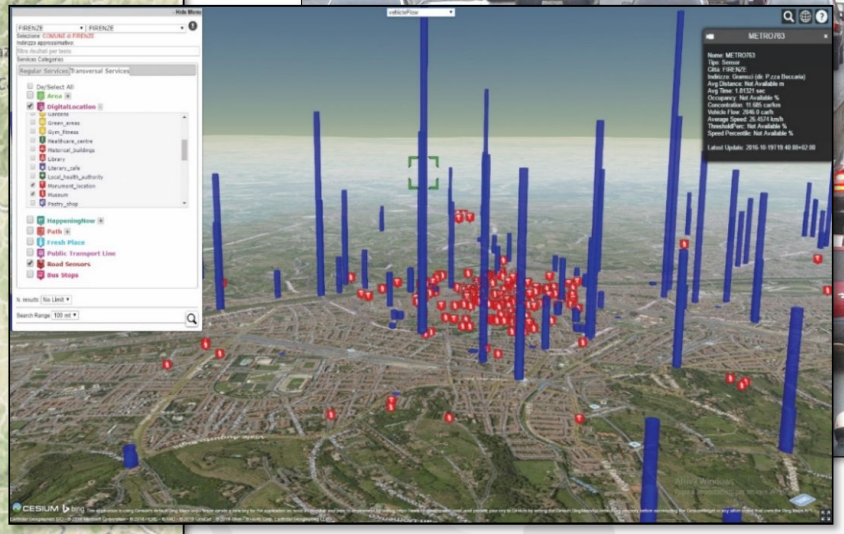
Traffic Flow Tools

Sii-Mobility

Spire and Virtual Spires (cameras), Bluetooth, …

Specifically located: along, around, on gates, on x…

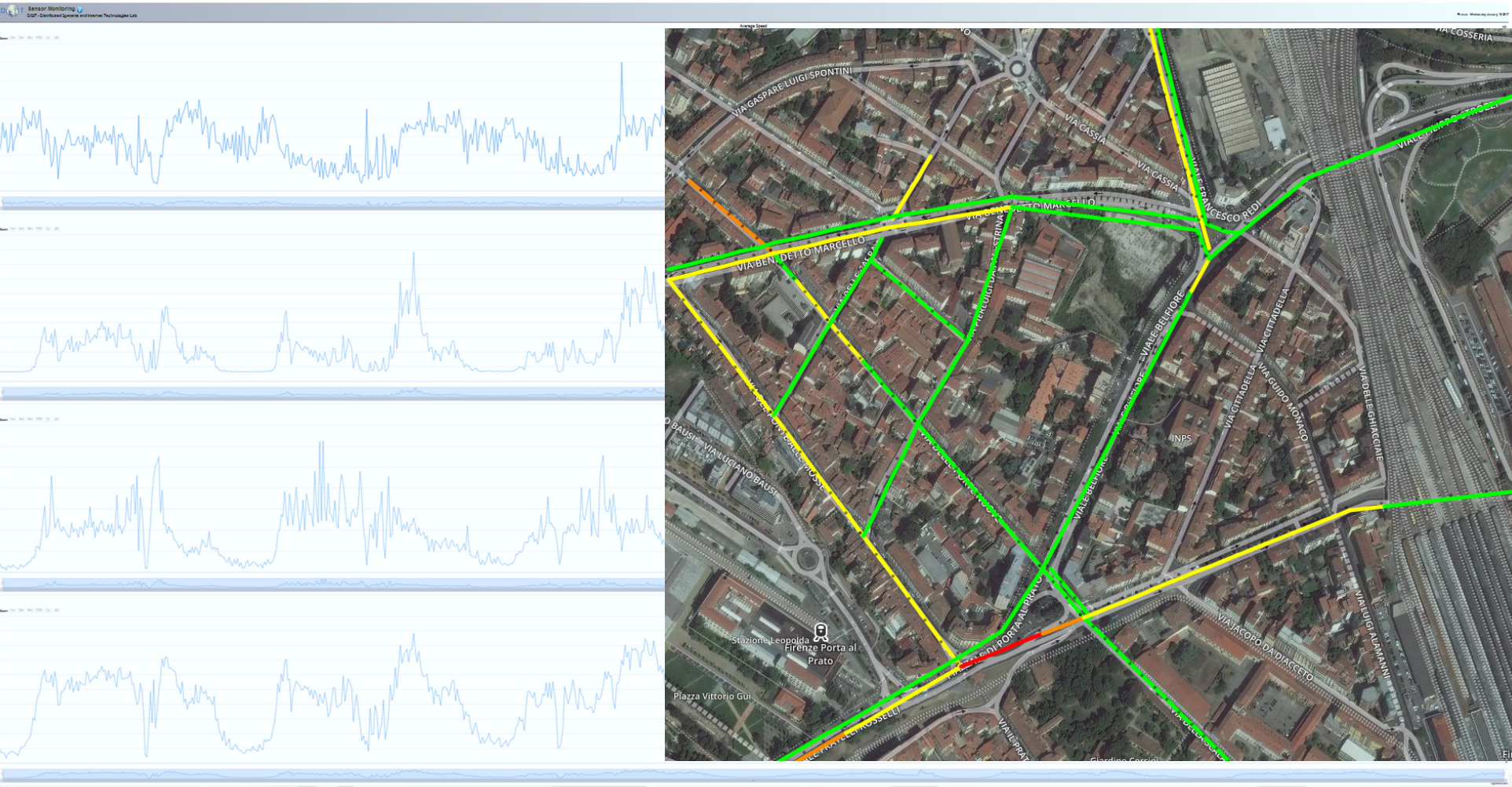


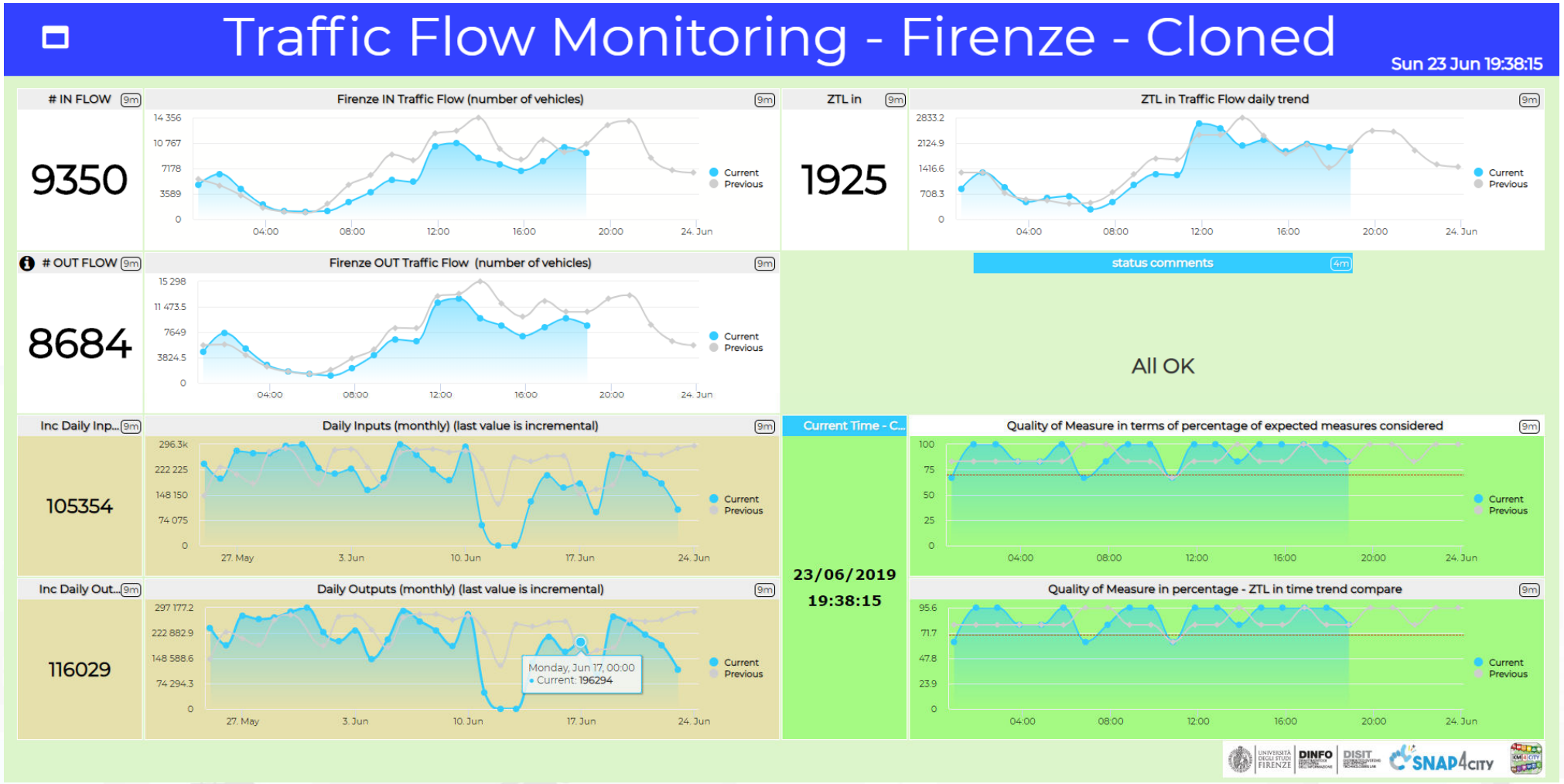


UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

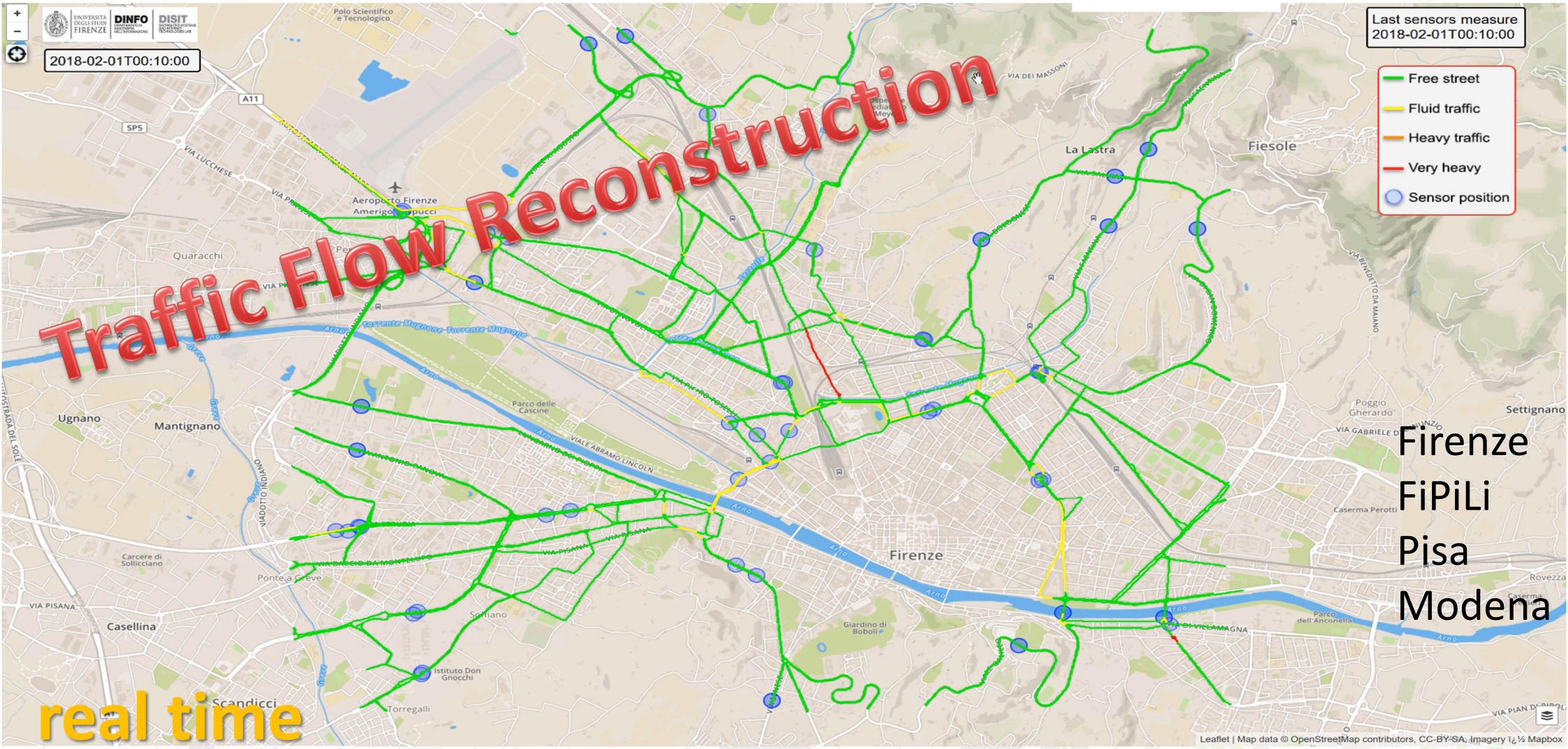
Traffic Flow data

Sii-Mobility





https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc2MQ==



http://firenzetraffic.km4city.org

http://firenzetraffic.km4city.org/new.html

http://firenzetraffic.km4city.org/newSensors.html

Traffic Flow Reconstruction for the cities

Traffic Flow Reconstruction

https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc5NQ==

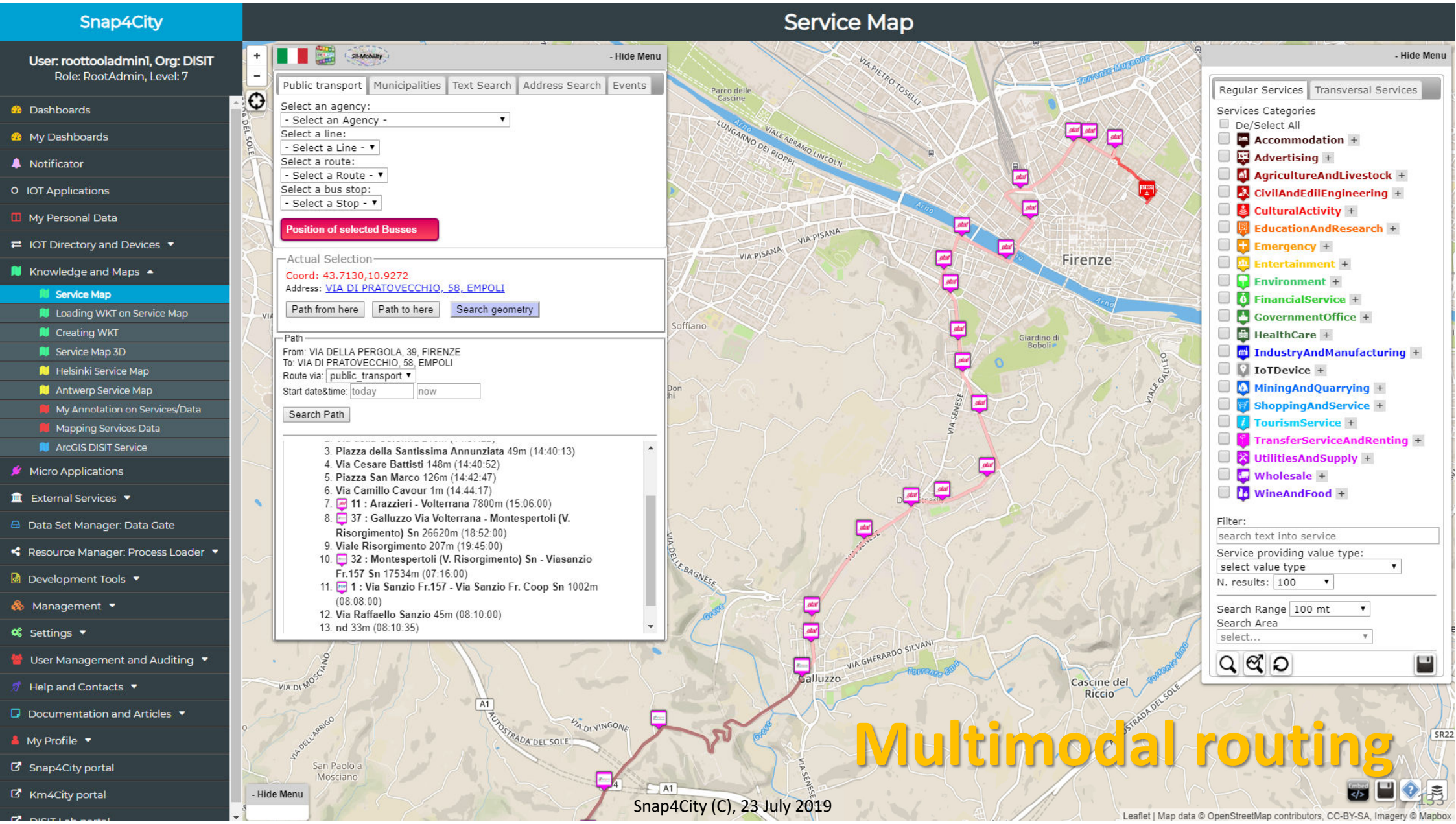
Traffic Flow Reconstruction (self training)

- P. Bellini, S. Bilotta, P. Nesi, M. Paolucci, M. Soderi, "Traffic Flow Reconstruction from Scattered Data", IEEE SMARTCOMP, IEEE international conference on smart computing, 18-20 June, Taormina, Sicily, Italy. 2018
- P. Bellini, S. Bilotta, P. Nesi, M. Paolucci, M. Soderi, "Real-Time Traffic Estimation of Unmonitored Roads", IEEE-DataCom'2018, Athens, 2018

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

SNAP4CITY KM4CITY

Modal & Multimodal Routing for Navigation and Travel Planning


Snap4City
Service Map
User: roottooladmin1, Org: DISIT
Role: RootAdmin, Level: 7
Dashboards
My Dashboards
Notificator
IOT Applications
My Personal Data
IOT Directory and Devices
Knowledge and Maps
Service Map
Loading WKT on Service Map
Creating WKT
Service Map 3D
Helsinki Service Map
Antwerp Service Map
My Annotation on Services/Data
Mapping Services Data
ArcGIS DISIT Service
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
- Hide Menu
Public transport
Municipalities
Text Search
Address Search
Events
Select an agency:
- Select an Agency -
Select a line:
- Select a Line -
Select a route:
- Select a Route -
Select a bus stop:
- Select a Stop -
Position of selected Busses
Actual Selection
Coord: 43.7130,10.9272
Address: VIA DI PRATOVECCHIO, 58, EMPOLI
Path from here
Path to here
Search geometry
Path
From: VIA DELLA PERGOLA, 39, FIRENZE
To: VIA DI PRATOVECCHIO, 58, EMPOLI
Route via: public_transport
Start date&time: today now
Search Path
3. Piazza della Santissima Annunziata 49m (14:40:13)
4. Via Cesare Battisti 148m (14:40:52)
5. Piazza San Marco 126m (14:42:47)
6. Via Camillo Cavour 1m (14:44:17)
7. 11 : Arazzieri - Volterrana 7800m (15:06:00)
8. 37 : Galluzzo Via Volterrana - Montespertoli (V. Risorgimento) Sn 26620m (18:52:00)
9. Viale Risorgimento 207m (19:45:00)
10. 32 : Montespertoli (V. Risorgimento) Sn - Viasanzio Fr.157 Sn 17534m (07:16:00)
11. 1 : Via Sanzio Fr.157 - Via Sanzio Fr. Coop Sn 1002m (08:08:00)
12. Via Raffaello Sanzio 45m (08:10:00)
13. nd 33m (08:10:35)
- Hide Menu
Regular Services
Transversal Services
Services Categories
De/Select All
Accommodation
Advertising
AgricultureAndLivestock
CivilAndEdilEngineering
CulturalActivity
EducationAndResearch
Emergency
Entertainment
Environment
FinancialService
GovernmentOffice
HealthCare
IndustryAndManufacturing
IoTDevice
MiningAndQuarrying
ShoppingAndService
TourismService
TransferServiceAndRenting
UtilitiesAndSupply
Wholesale
WineAndFood
Filter:
search text into service
Service providing value type:
select value type
N. results: 100
Search Range 100 mt
Search Area
select...
Firenze
Galluzzo
Cascine del Riccio
Multimodal routing
Snap4City (C), 23 July 2019
Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox

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

KM 4 CITY

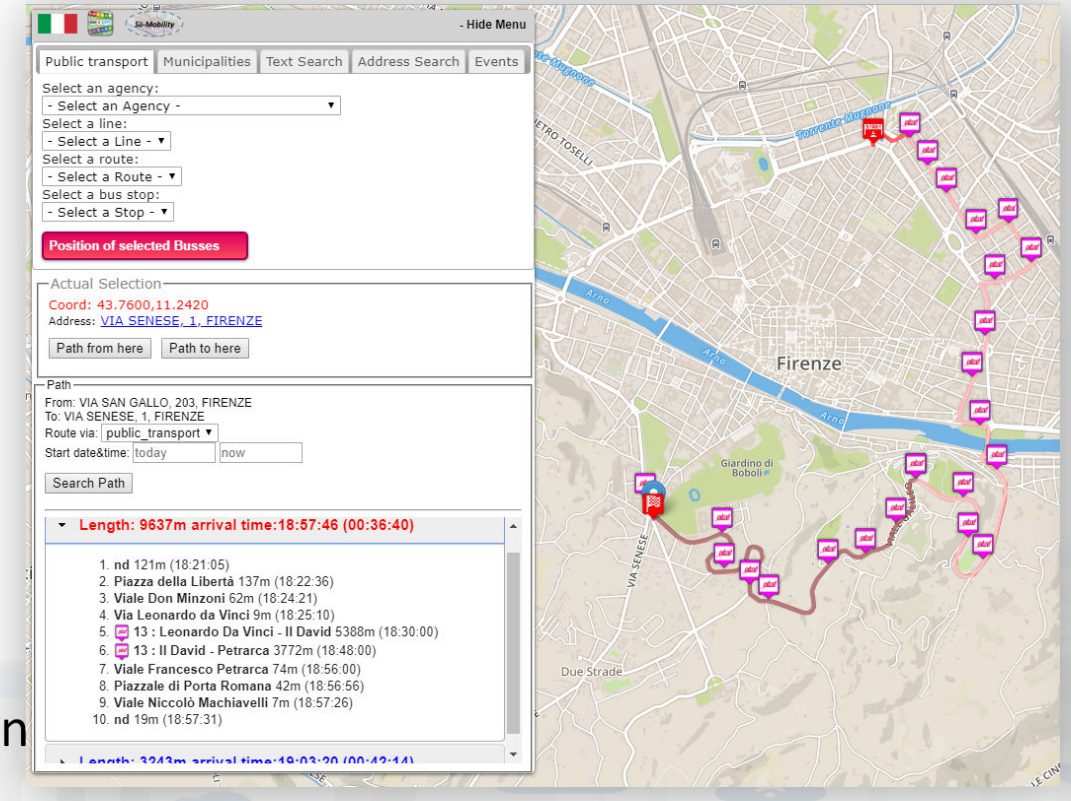
Routing and Multimodal Routing

Modes:

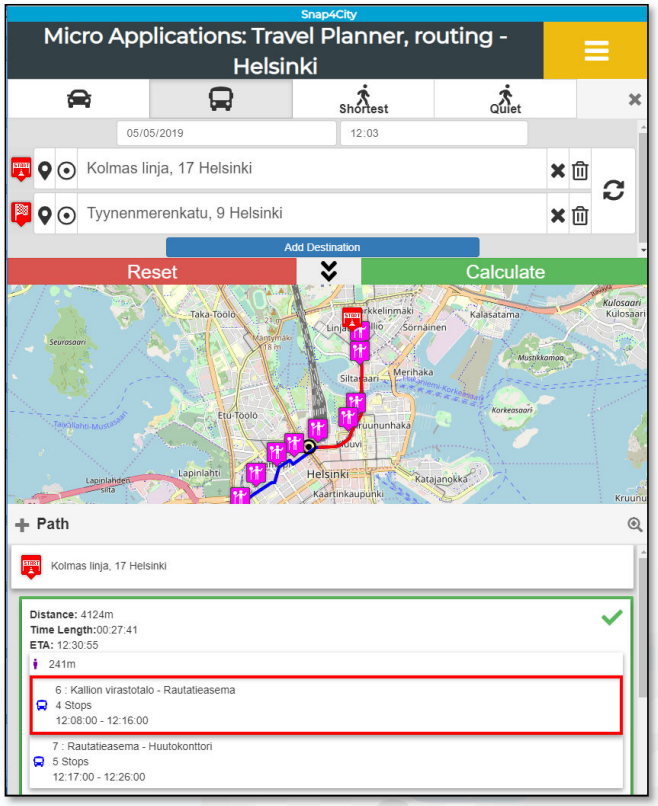
- Pedonal, Vehicles
- Public Multimodal
- Multi Point for Delivering
- Constrained: quite, blocked, etc.

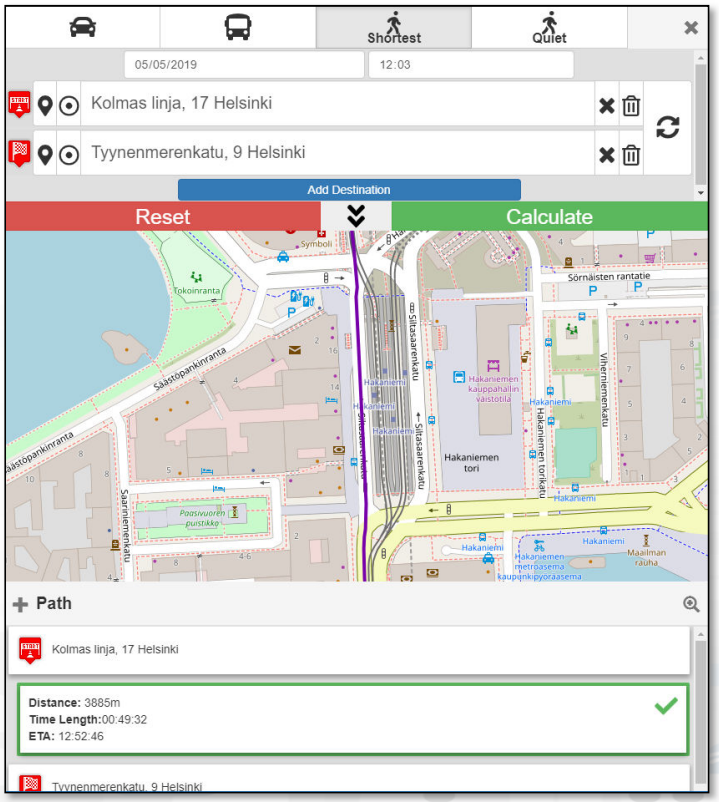
Test it on our:

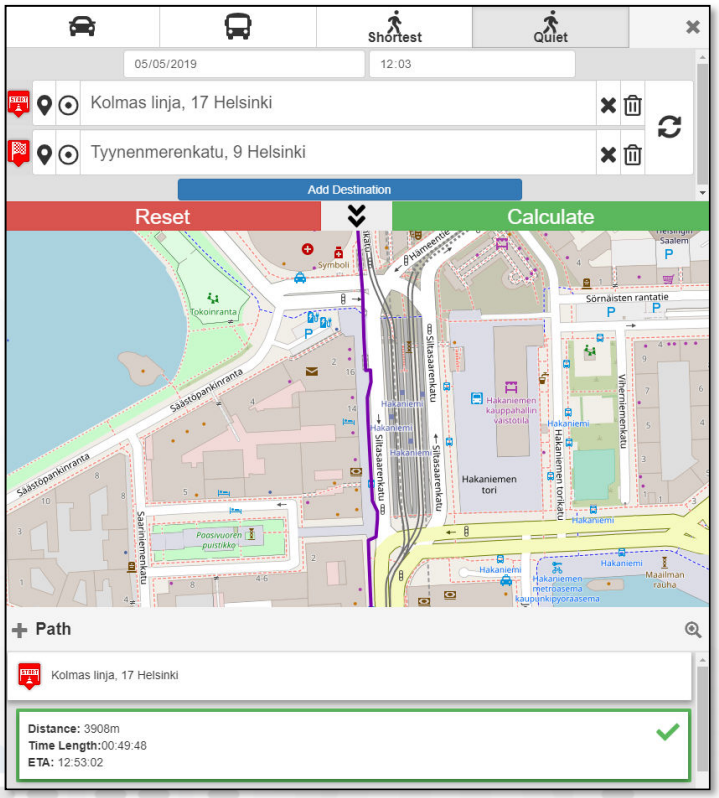
- Mobile Apps
- MicroApplication
- Dashboard
- ServiceMap service on Tuscany in Snap4City



Routing for quite



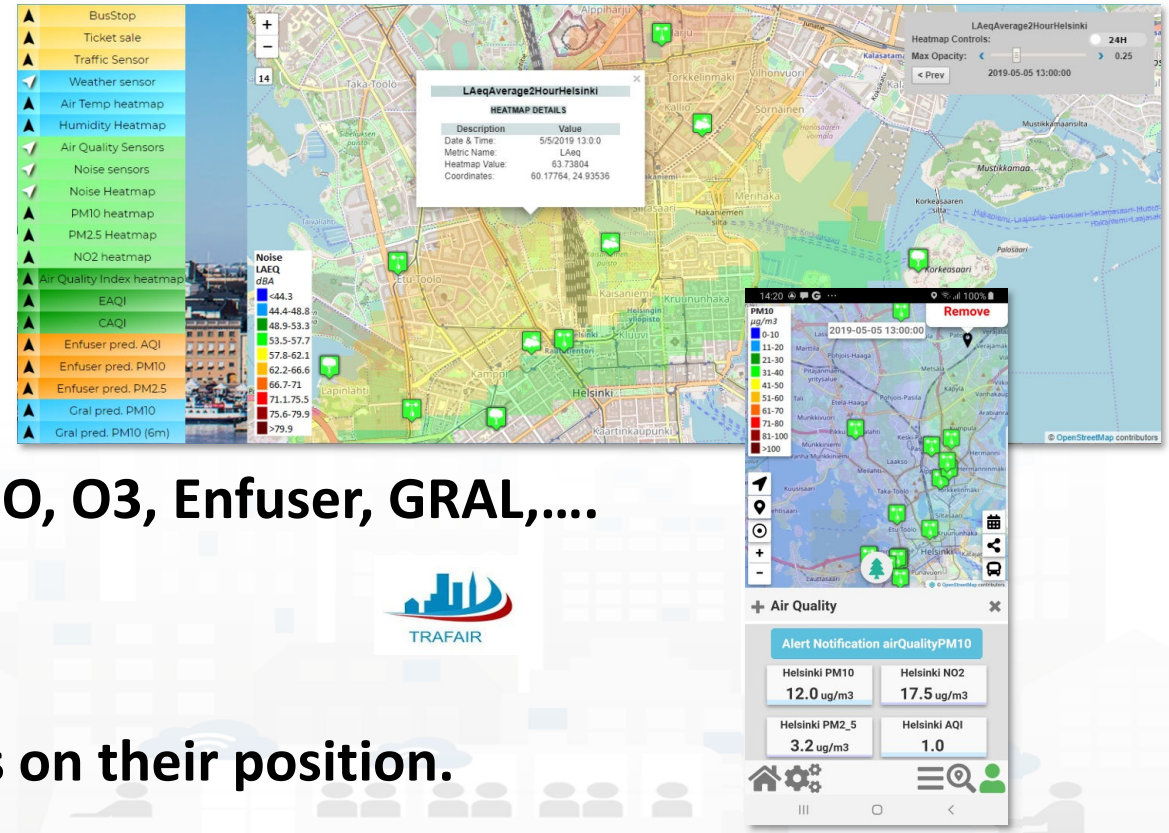




Environmental Data: Predictions, Early Warning

Data Analytics: Heatmaps

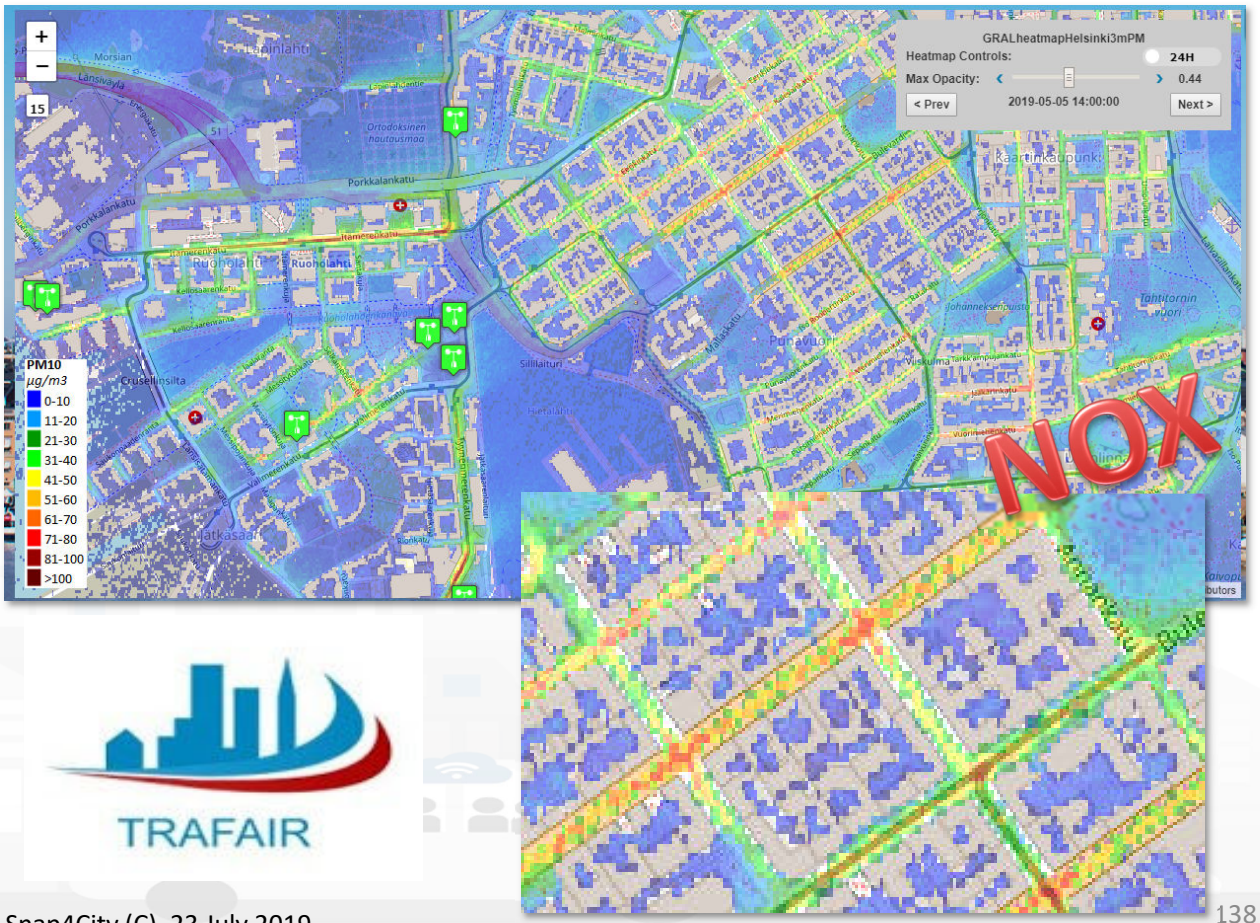
- Over the Gaussian Heatmaps
- Calibrated heatmaps on the basis of Interpolated data for:
  - From 200x200 to 4x4 mt
  - PM10, PM2.5, SO2, NO2, Noise, NO, O3, Enfuser, GRAL,….
  - Any programmed Color map
  - Animations over H24
  - Picking values in any place, values on their position.
  - On Web and Mobile App



TRAFAIR

Environmental Data Predictions: GRAL

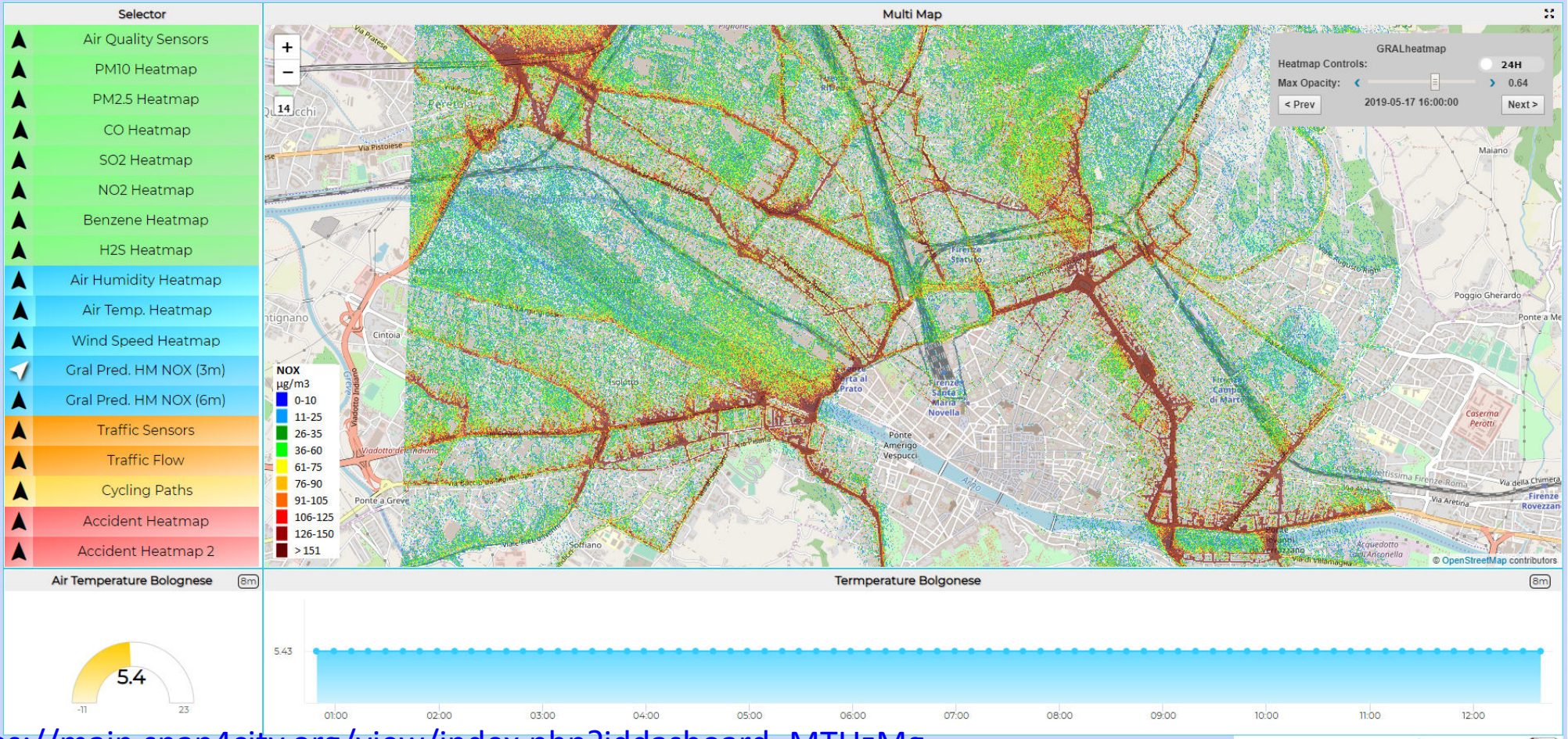
- **GRAL predictions: PM10, NOX, ….**
  - **Comparison wrt real time values in actual value of Sensors**
  - Graz Lagrangian Model.
- GRAL model takes into account:
  - pollution sources (for example the vehicles, their distribution on the streets, the about of pollution they produce according to their distribution over time and space, etc.),
  - structure of the city (streets and shape 3D of the buildings),
  - weather forecast (wind intensity and direction), etc.
- GRAL can be applied on NOX, PM10, PM2.5, … or any other particles



Heatmap Firenze - trafair

different data

Fri 17 May 12:49:34

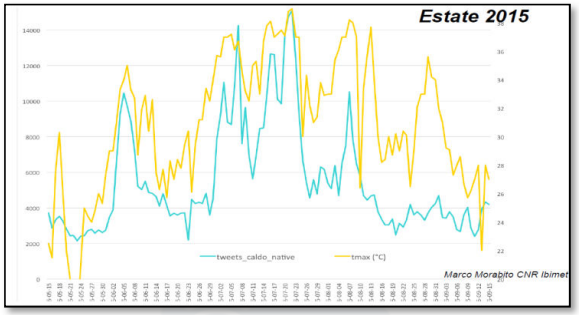


https://main.snap4city.org/view/index.php?iddasboard=MTUzMg==

Social Media Analysis: Early Warning, Predictions

Prediction/Assessment

- Football game results as related to the volume of Tweets
- Number of votes on political elections, via sentiment analysis, SA
- Size and inception of contagious diseases
- marketability of consumer goods
- public health seasonal flu
- box-office revenues for movies
- places to be visited, most visited
- number of people in locations like airports
- audience of TV programmes, political TV shows
- weather forecast information
- Appreciation of services



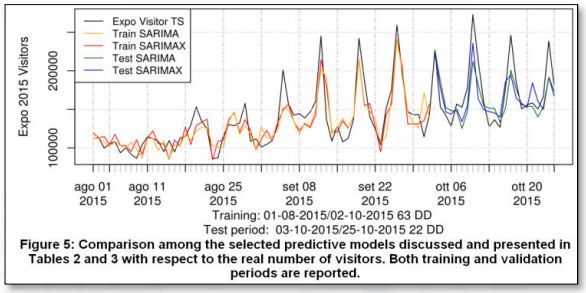
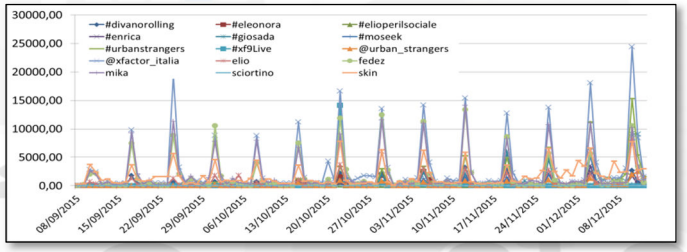


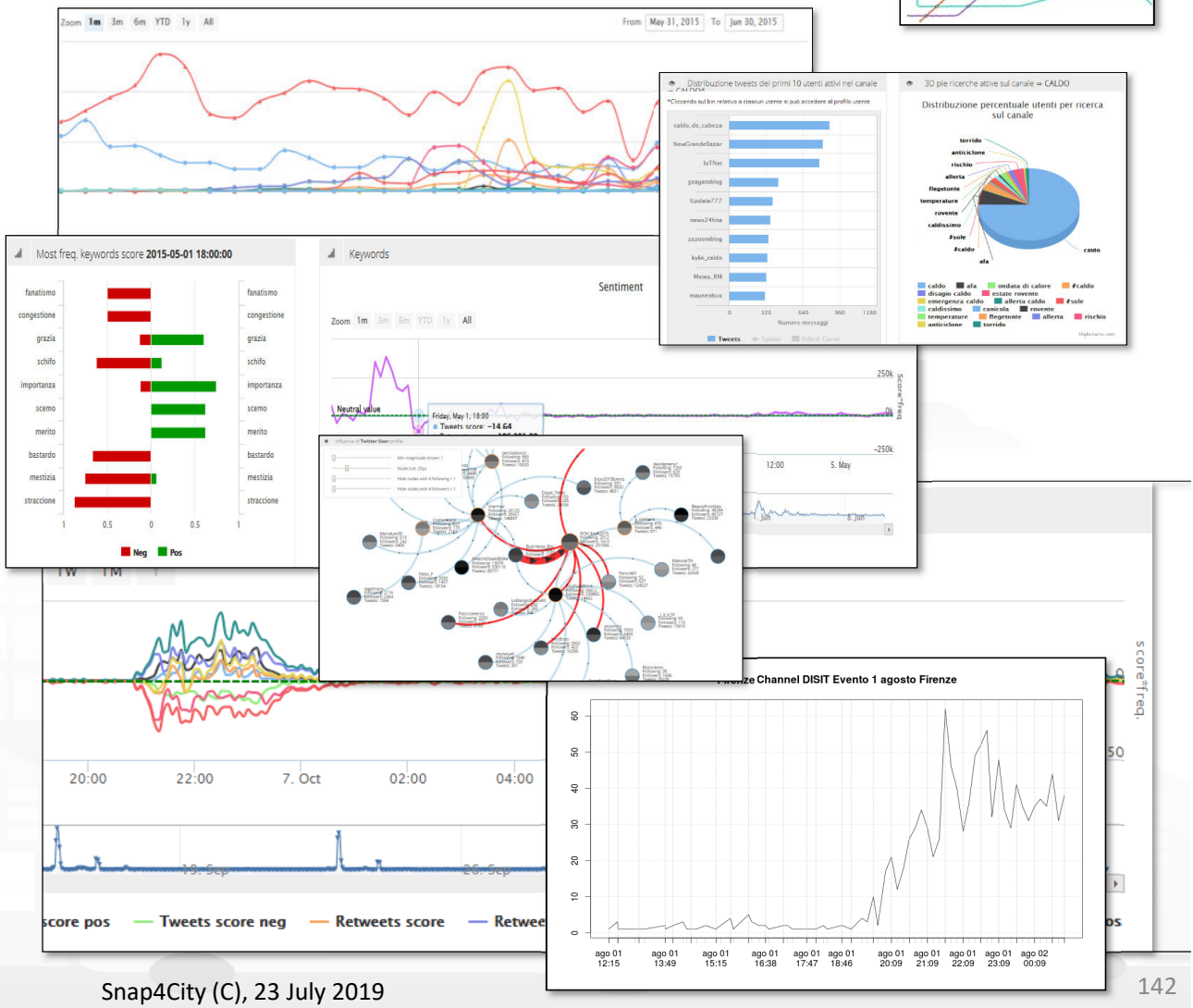
Figure 5: Comparison among the selected predictive models discussed and presented in Tables 2 and 3 with respect to the real number of visitors. Both training and validation periods are reported.



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

Twitter Vigilance

- http://www.disit.org/tv
- http://www.disit.org/rttv
- Citizens as sensors to
  - Assess sentiment on services, events, …
  - Response of consumers wrt, …
  - Early detection of critical conditions
  - Information channel
  - Opinion leaders
  - Communities
  - Formation
  - Predicting volume of visitors for tuning the services



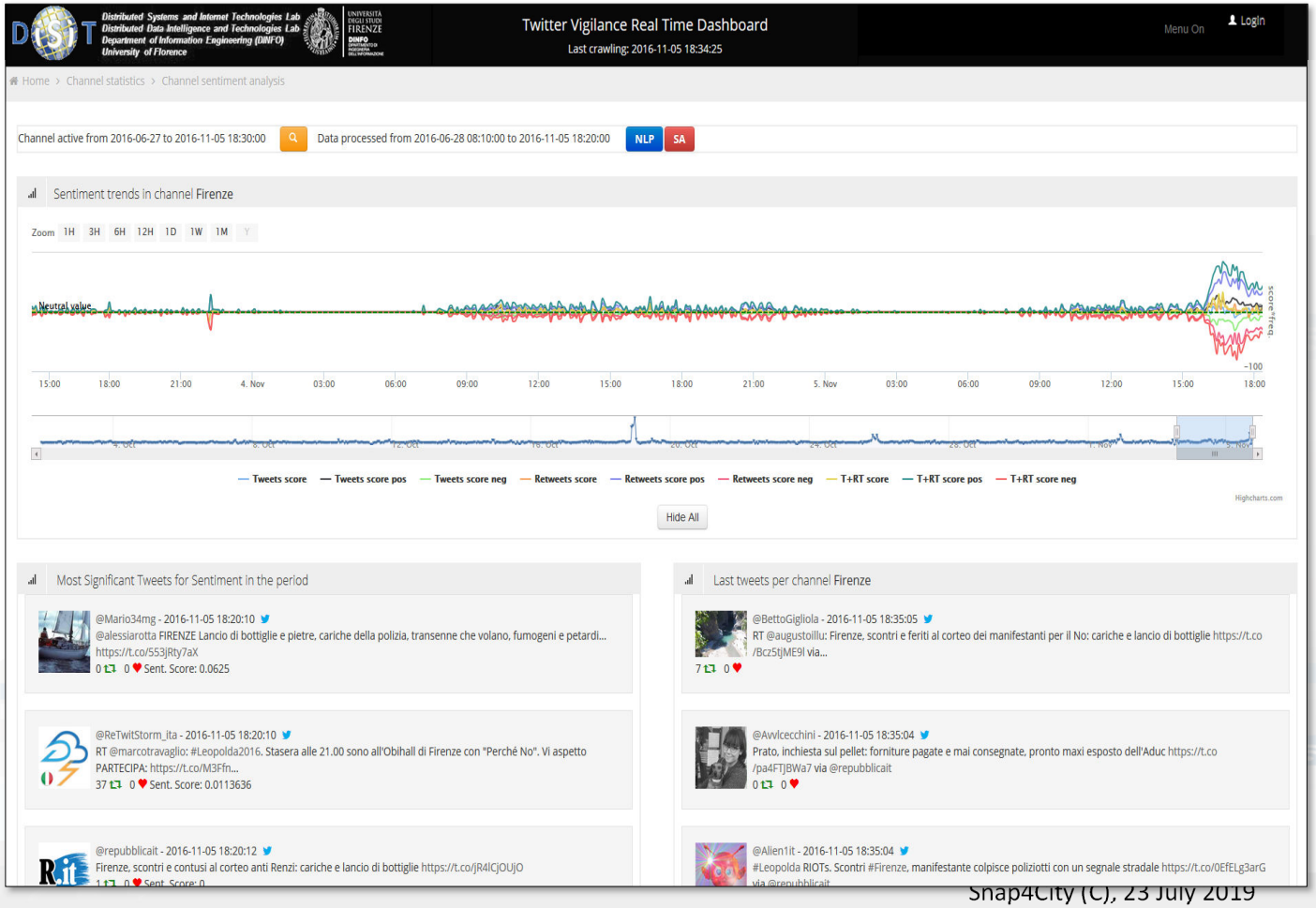
Twitter Vigilance

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

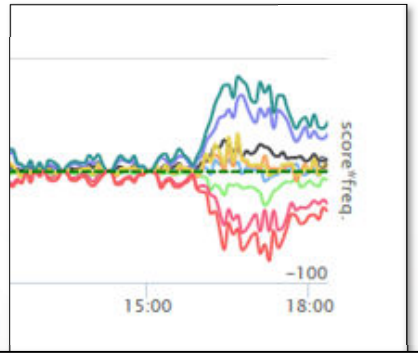
Twitter Vigilance

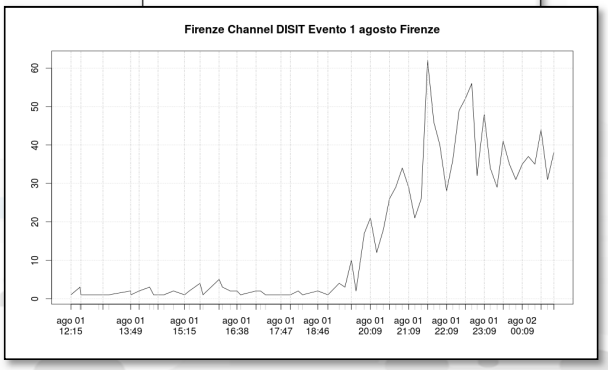
resolute

Twitter Vigilance RT: sentiment analysis



Real time Early Warning

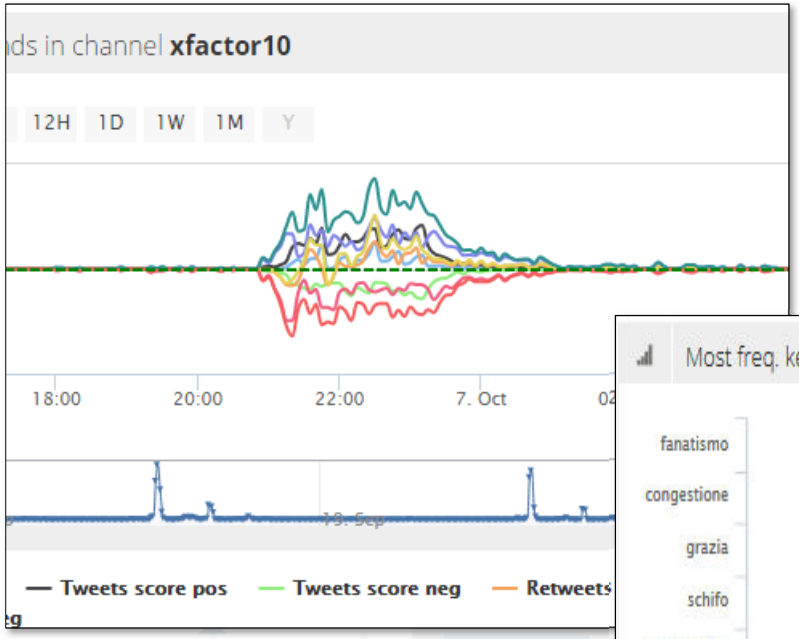


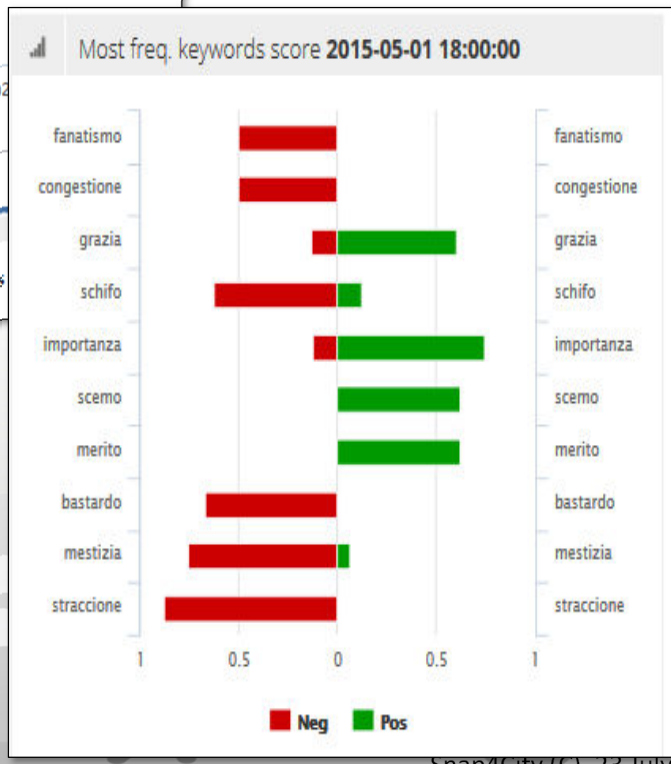


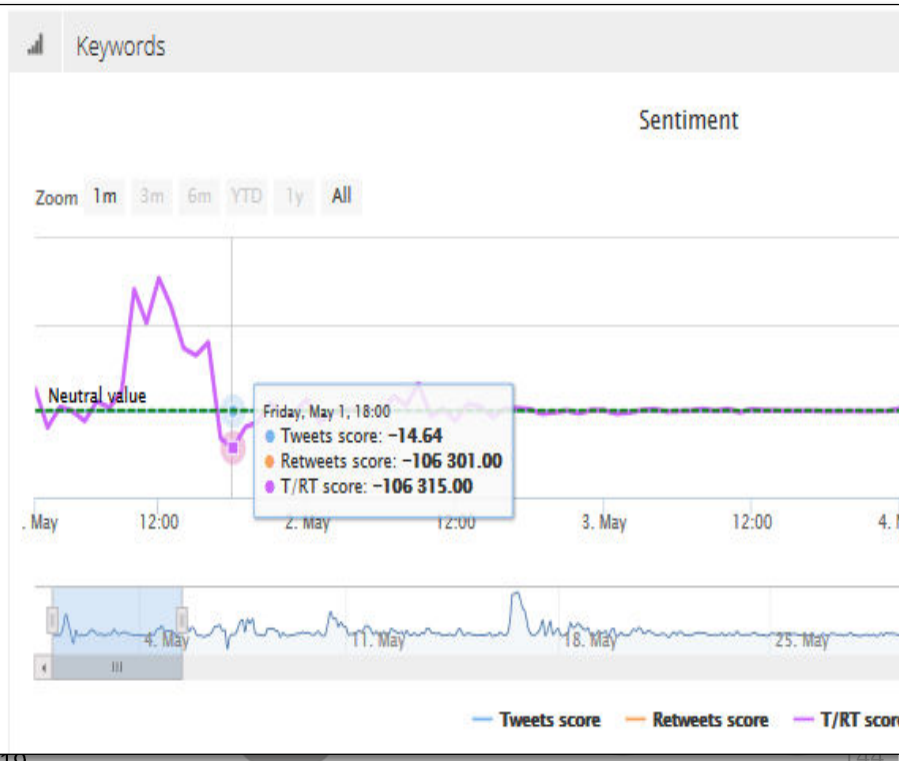
UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB
http://www.disit.org

Sentiment Analysis

Sii-Mobility



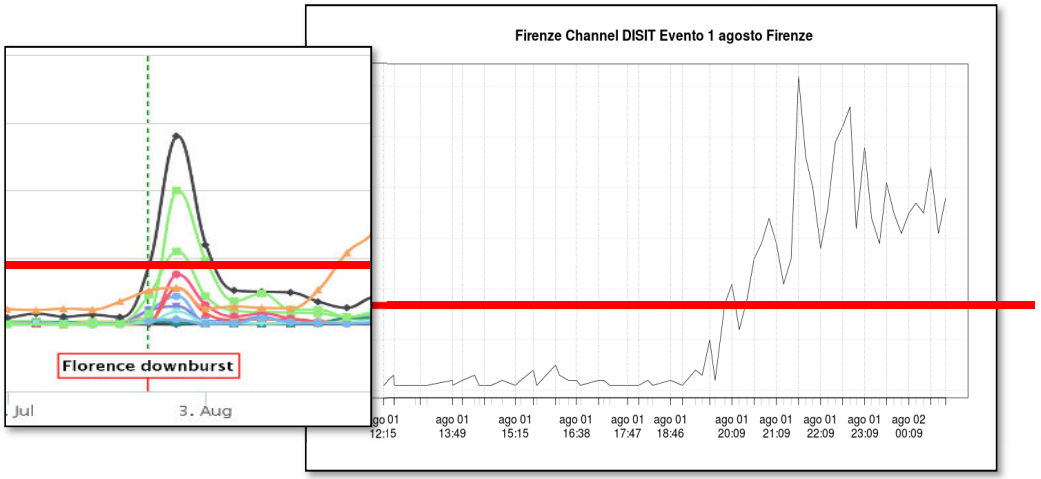


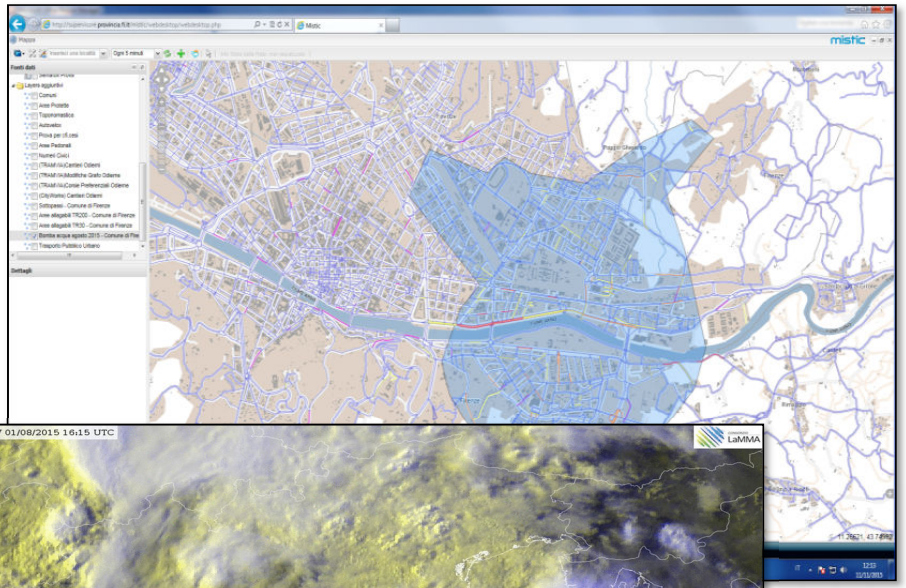


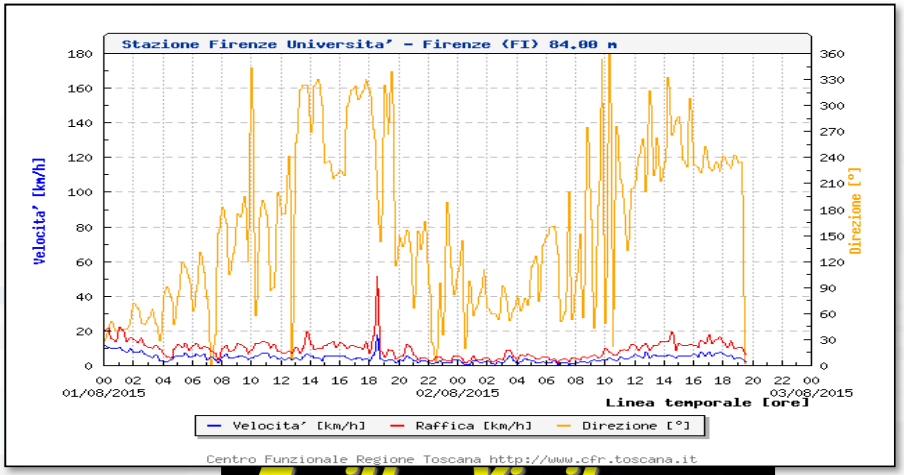
Twitter Vigilance

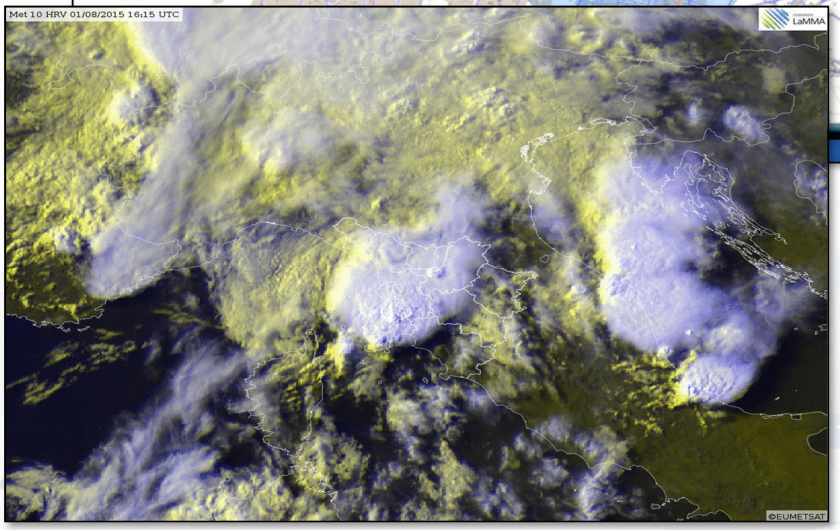
UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB
http://www.disit.org

Early Warning
Twitter Vigilance and Water Bomb





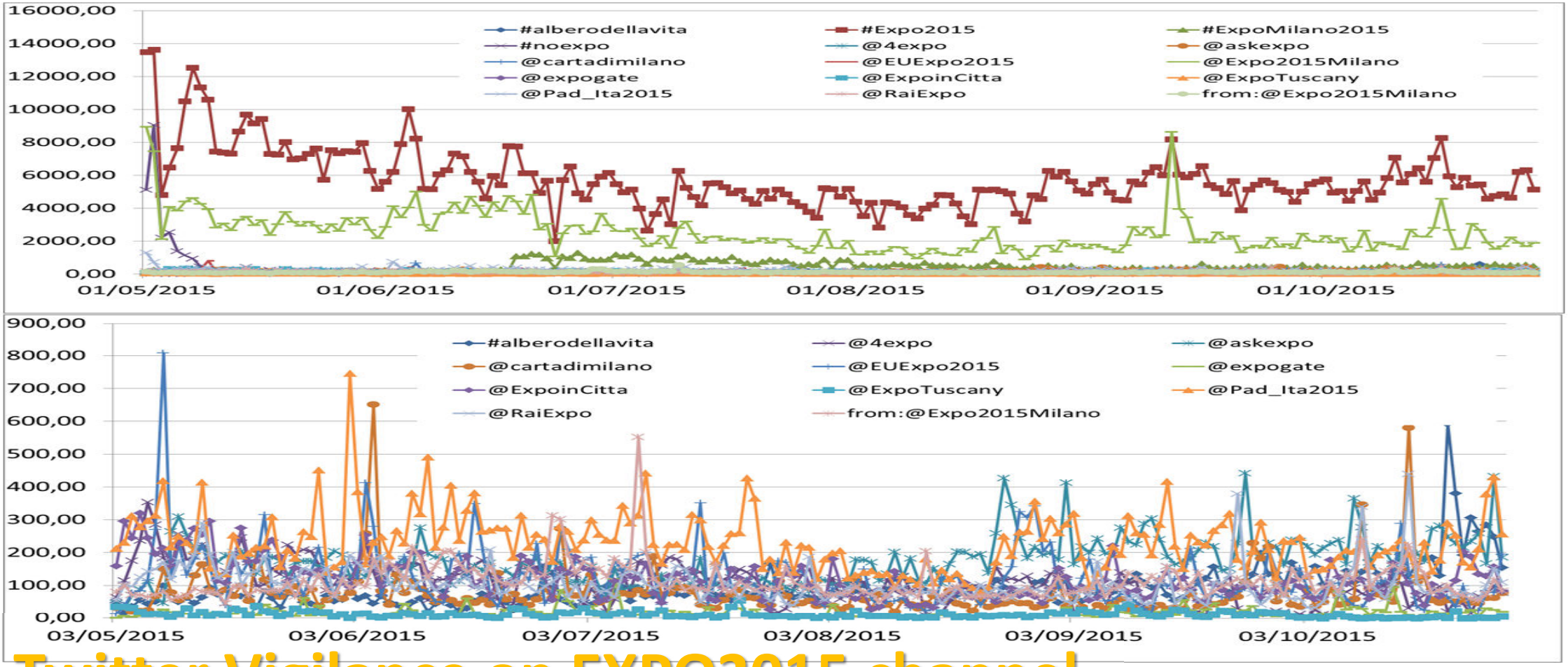




Twitter Vigilance

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

Predicting EXPO2015

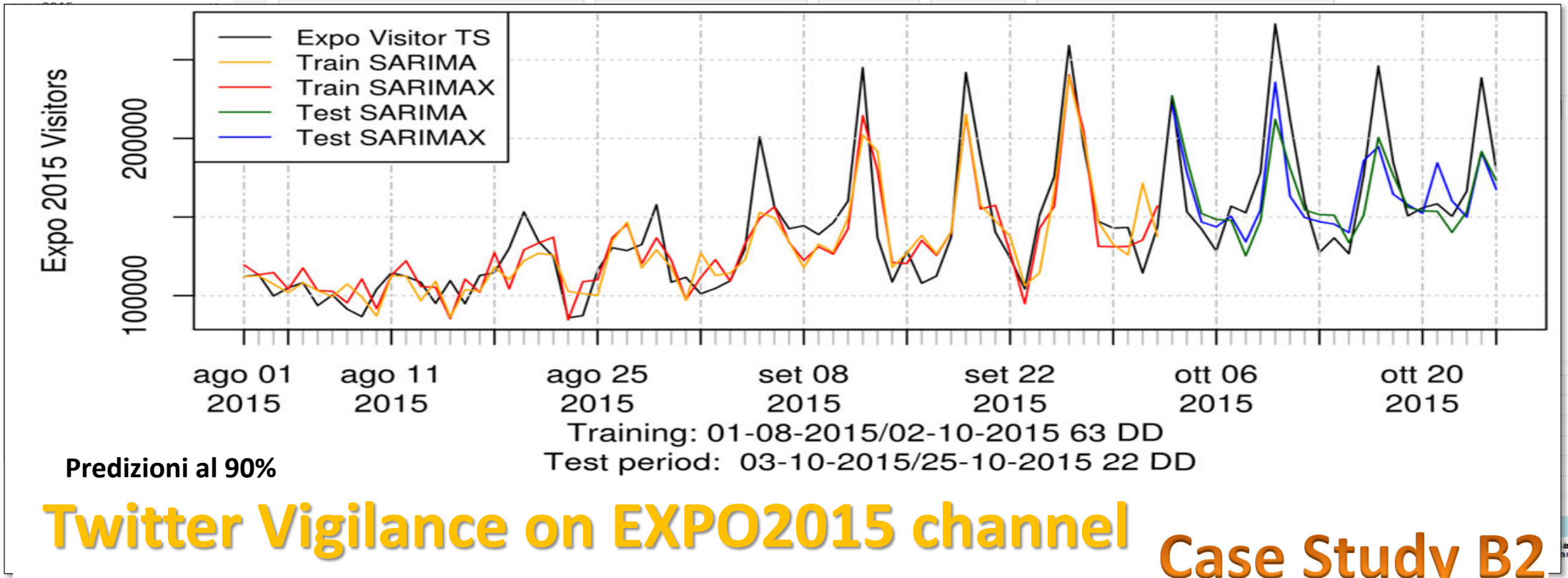


Twitter Vigilance on EXPO2015 channel

Twitter Vigilance

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

Twitter Vigilance monitoring and predictions



Predizioni al 90%

Twitter Vigilance on EXPO2015 channel

Case Study B2

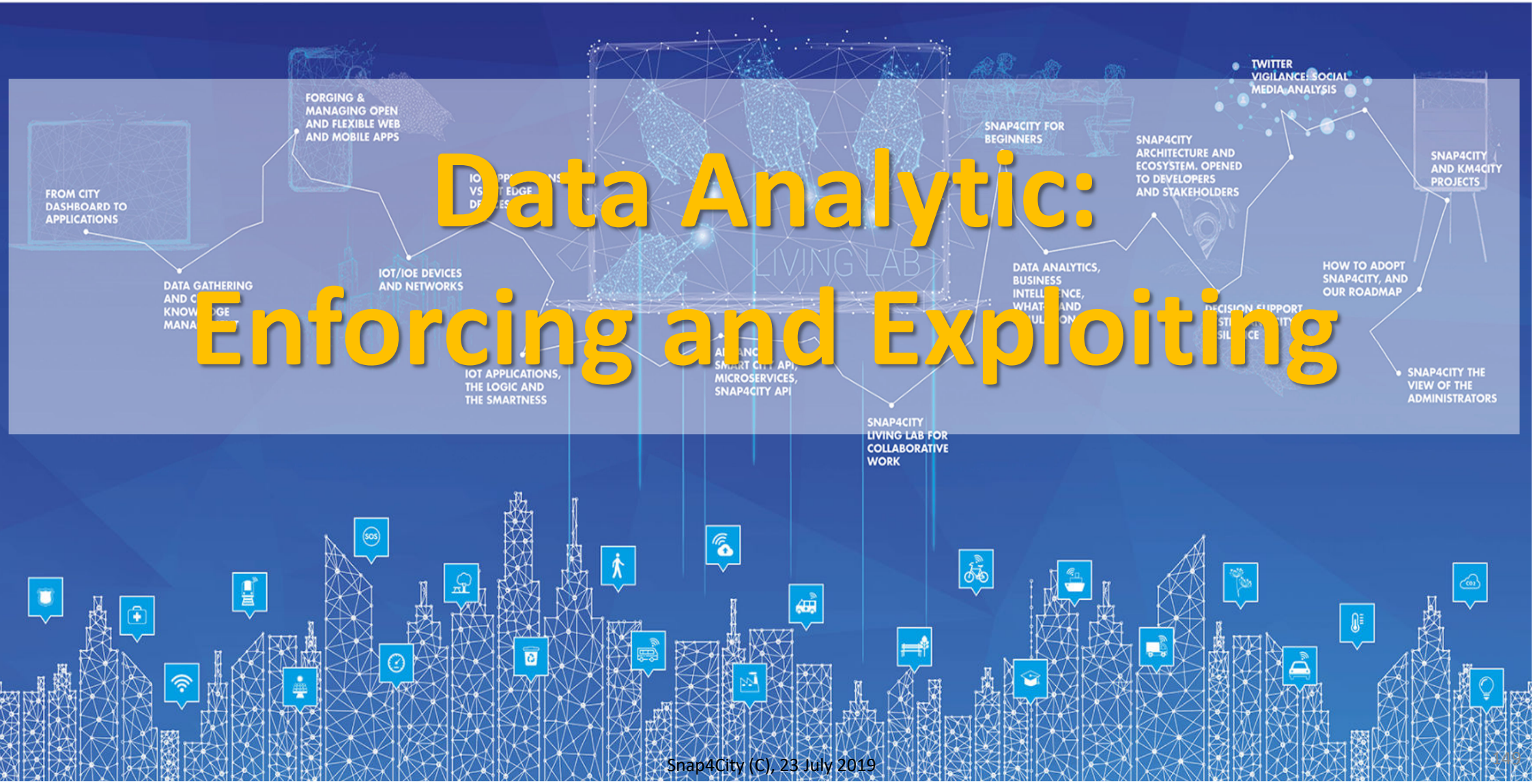
Predicting volume of visitors for tuning the services

Twitter Vigilance

Citations and self training

- P. Nesi, G. Pantaleo, I. Paoli, I. Zaza, "Assessing the reTweet Proneness of tweets: predictive models for retweeting", Multimedia Tools and Applications, Springer, 2018. https://link.springer.com/article/10.1007/s11042-018-5865-0
- A. Crisci, V. Grasso, P. Nesi, G. Pantaleo, I. Paoli, I. Zaza, "Predicting TV programme Audience by Using Twitter Based Metrics", Multimedia Tools and Applications, springer. 10.1007/s11042-017-4880-x, 2017 https://link.springer.com/article/10.1007/s11042-017-4880-x
- V. Grasso, A. Crisci, M. Morabito, P. Nesi, G. Pantaleo, "Public crowdsensing of heat waves by social media data", Adv. Sci. Res., 14, 217-226, https://doi.org/10.5194/asr-14-217-2017, 2017, 10.5194/asr-14-217-2017 . http://www.adv-sci-res.net/14/217/2017/
- V. Grasso, A. Crisci, M. Morabito, P. Nesi, G. Pantaleo, I. Zaza, B. Gozzini, "Italian codified hashtags for weather warning on Twitter–who is really using them?." Advances in Science and Research 14 (2017): 63-69. http://www.adv-sci-res.net/14/63/2017/asr-14-63-2017.pdf

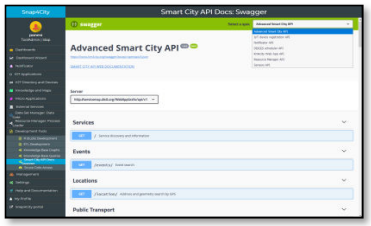
SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT CITIES



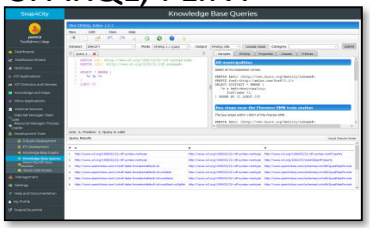
Data Analytic: Enforcing and Exploiting

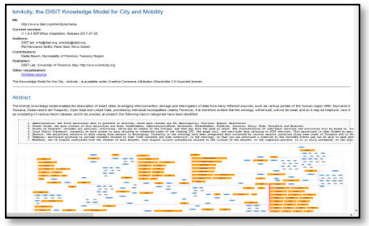
Data Analytics Dev. in R Studio and/or Tensor Flow

Swagger

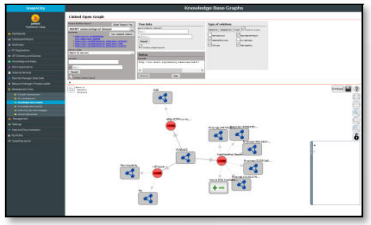


SPARQL, FLINT





Ontology Schema

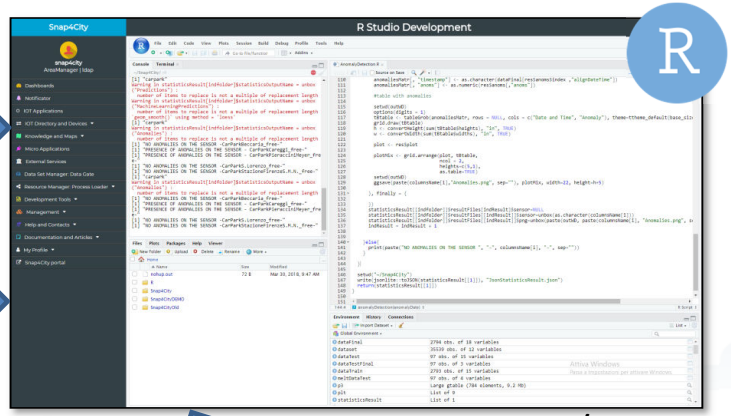


LOG.disit.org

Knowledge Base, Km4City

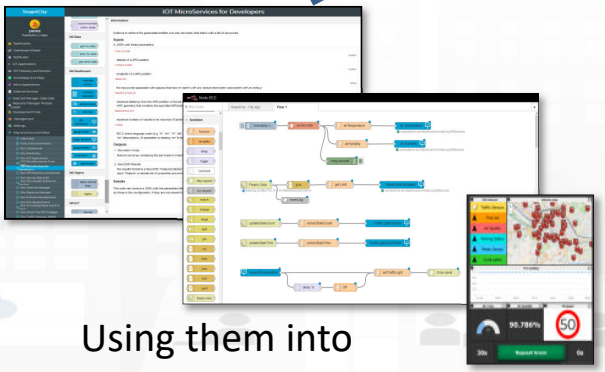
Big Data Store Facility

Smart City API from Knowledge Base and other tools



R Studio

Creating MicroServices

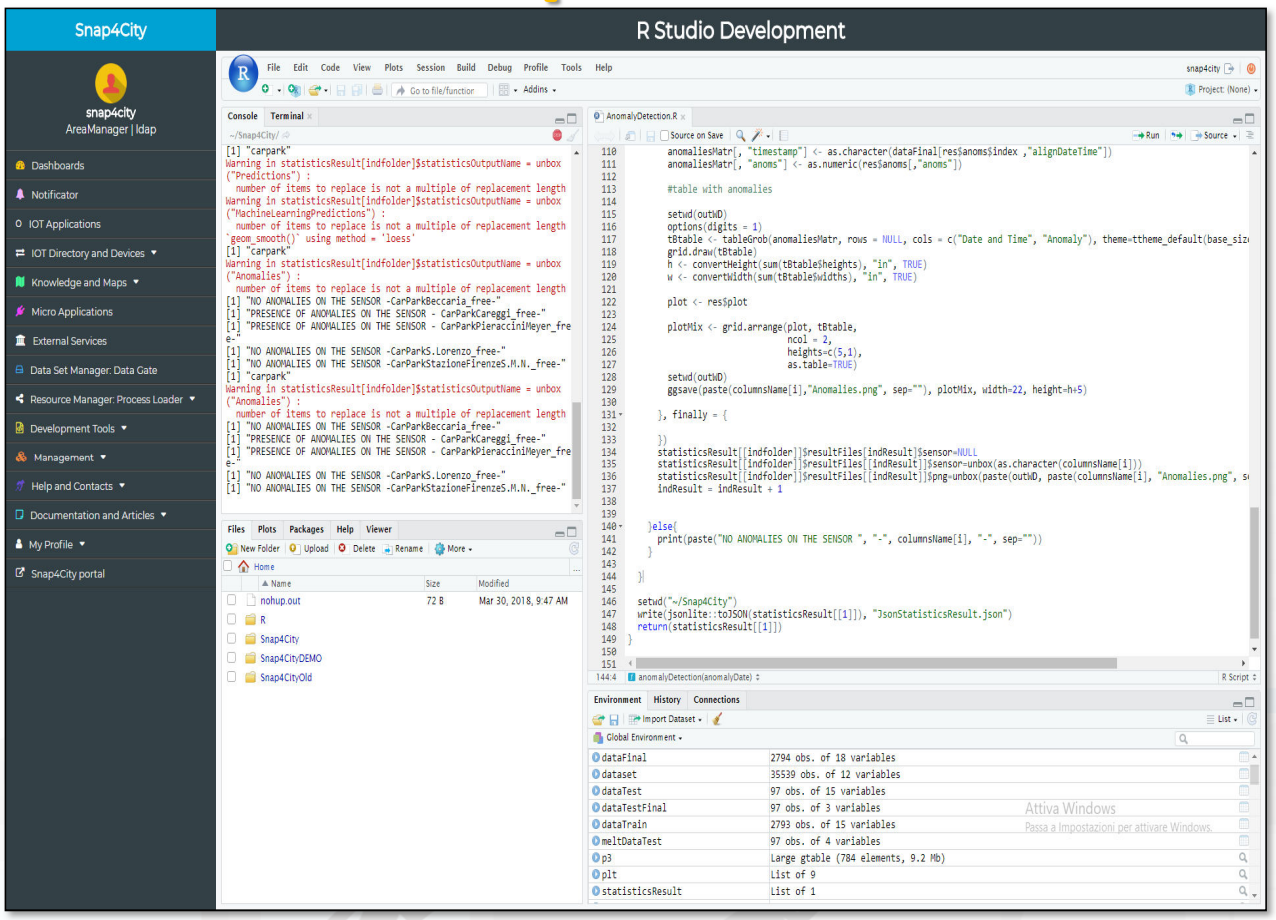


Using them into IOT Applications

Saving / Sharing reusing

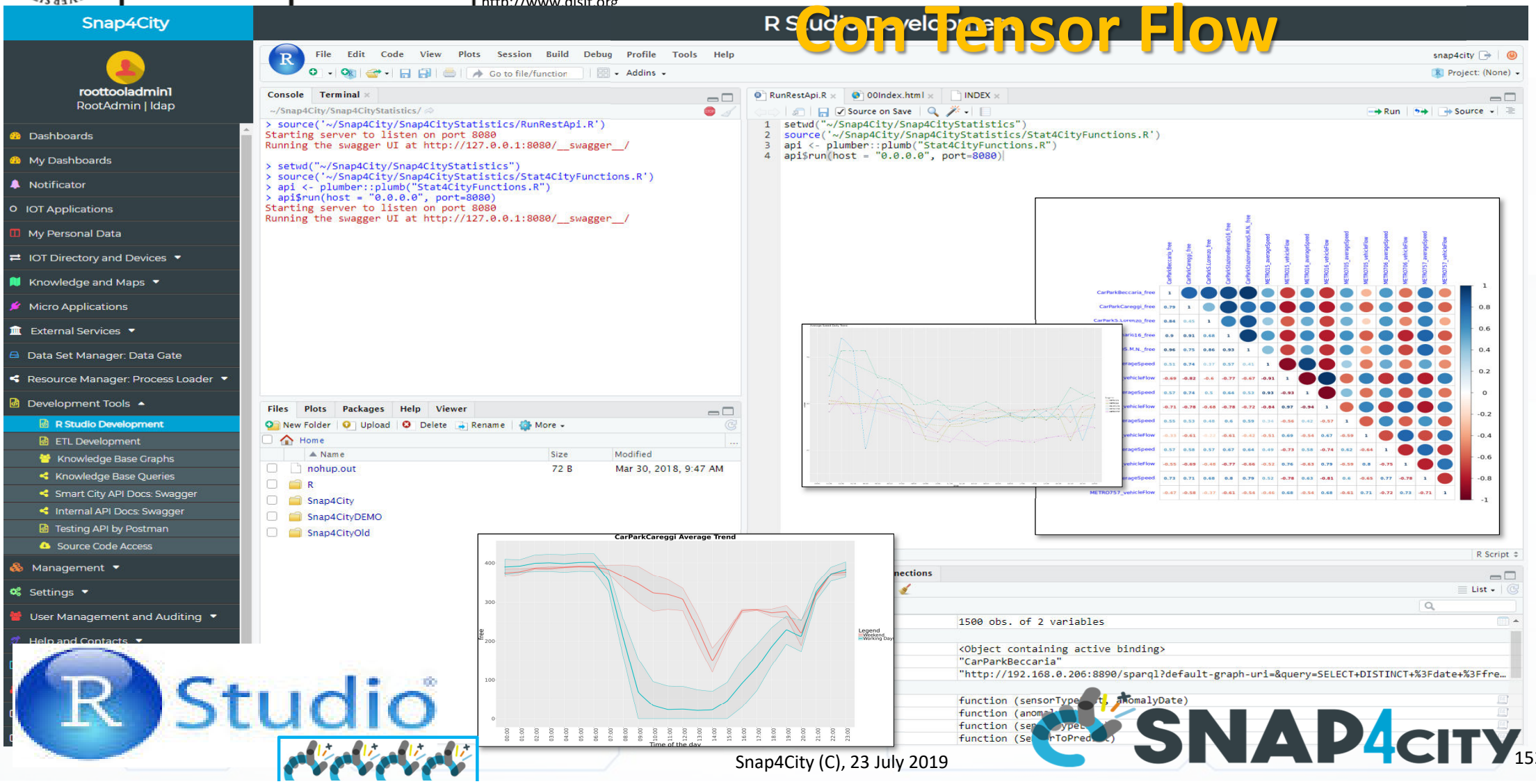
Resource Manager

Developer in R Studio + Tensor Flow

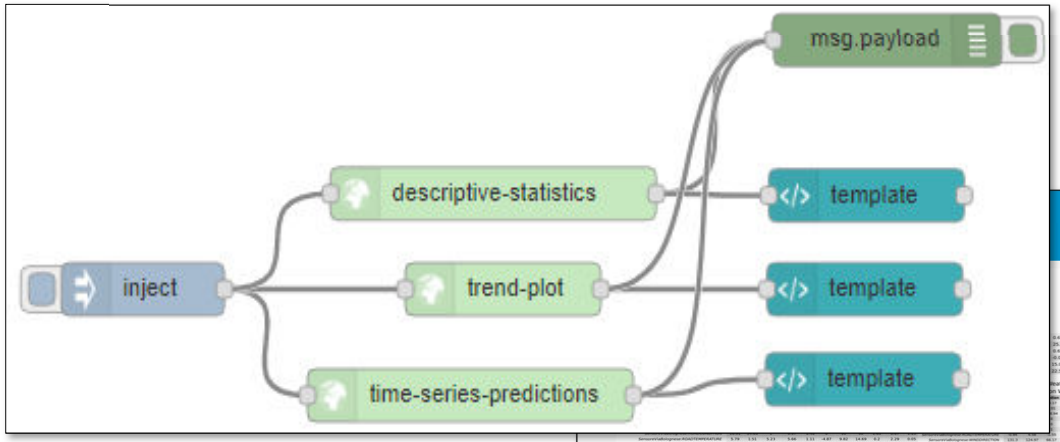


UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB http://www.disit.org

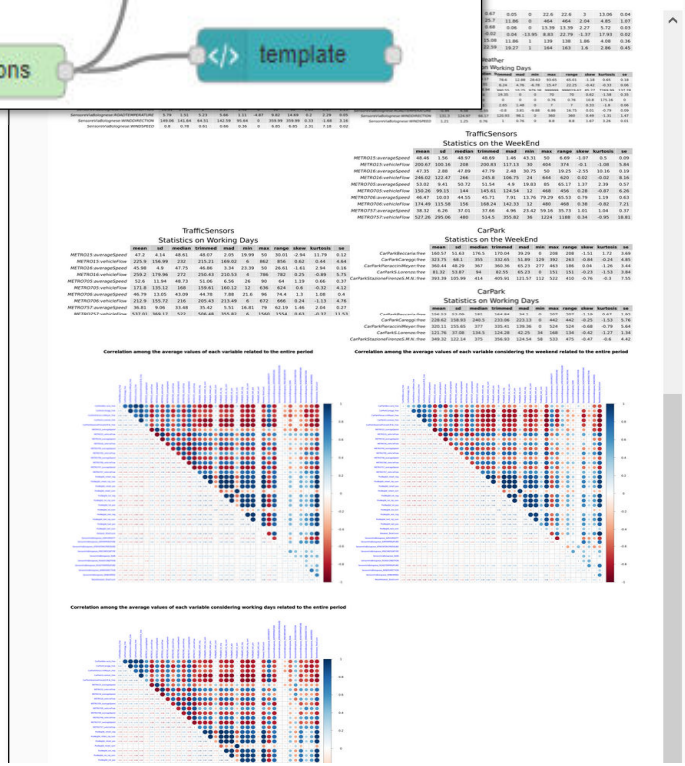
Data Analytics in R Studio Con Tensor Flow

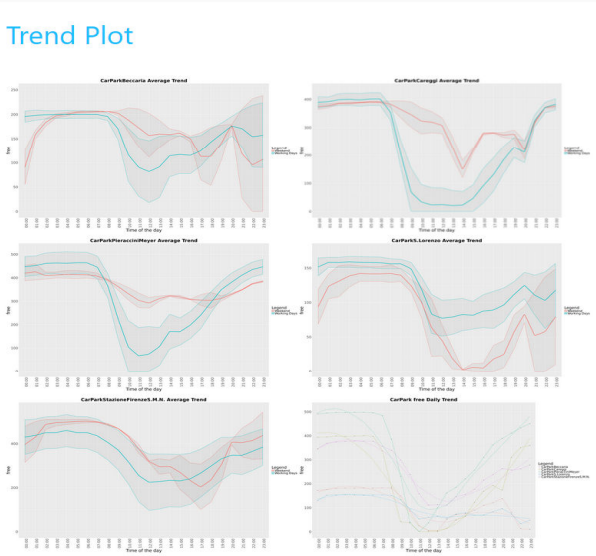


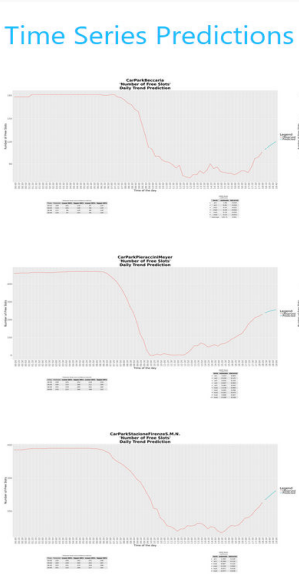
From R studio data analytics to MicroService



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

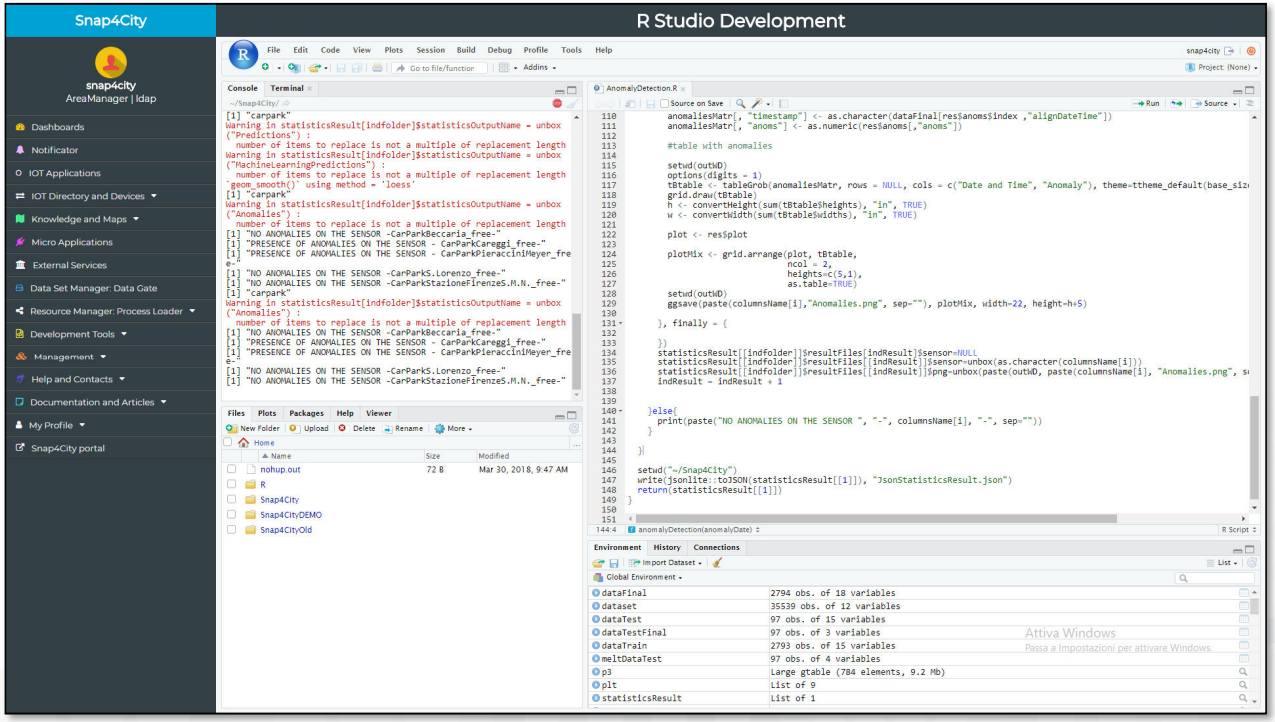


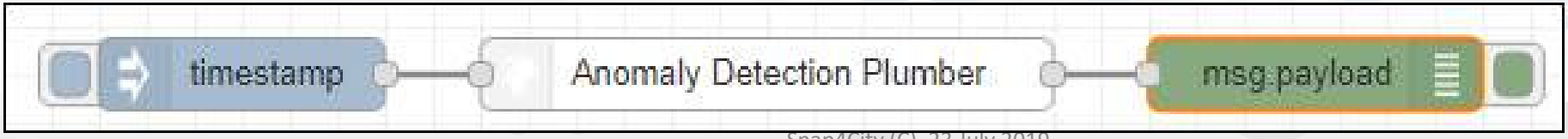




SNAP4CITY

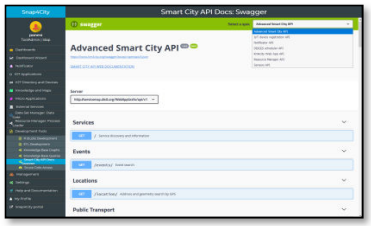
Developing in R Studio and/or Tensor Flow



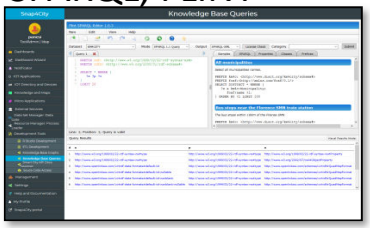


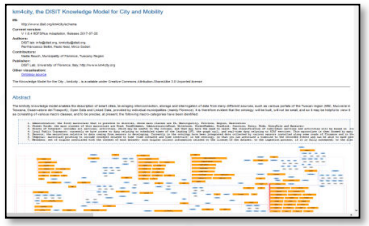
Data Analytics Dev. in Java, Python, ..

Swagger

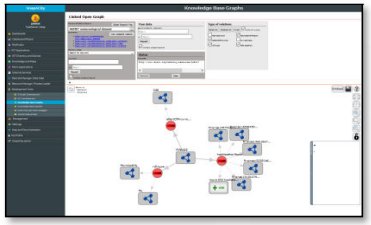


SPARQL, FLINT





Ontology Schema

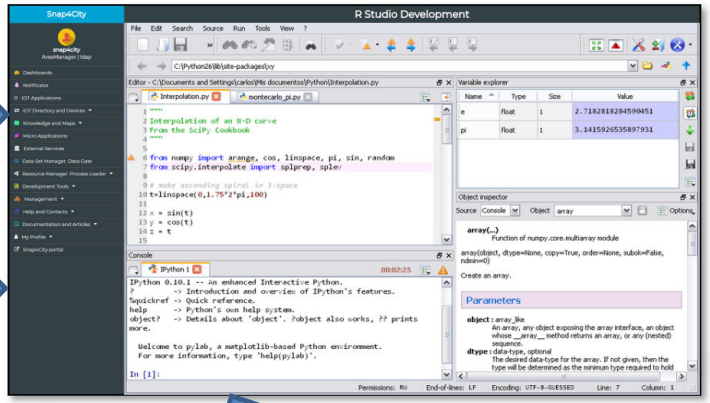


LOG.disit.org

Knowledge Base, Km4City

Big Data Store Facility

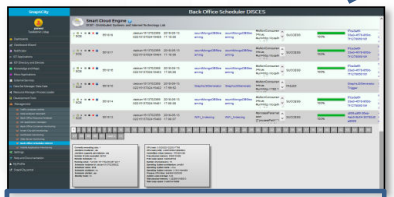
Smart City API from Knowledge Base and other tools



Coding Testing

Saving / Sharing resusing

DISCES scheduler



Process

Distributed Back Office

Monitoring

Data sources

Resource Manager

Real Time Data Analytics using R Studio. Exploitation in IOT Applications (DEMO)

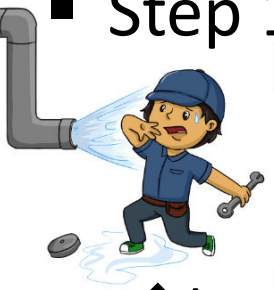
Real Time Data Analytics using RStudio

1. How to create a Data Analytic Node based on R Script (*plumberized*):
   - How to download Real-Time data using Smart City APIs
   - How to save heatmaps using Heatmap APIs
2. How to create an IOT Application for Real-Time Data Analytics:
   - How to upload the R script and create a Data Analytic Node instance
3. How to visualize the created heatmap in a dashboard

Real Time Data Analytics using R Studio

How to create a *plumberized* R script -1

PLUMBER is an **R** package that generates a web API from the **R** code you already have.



- Step 1 - *Plumberize* the code:

```
#' @get /TuscanyHeatmap
#' @serializer unboxedJSON
```

❖In order to send a response from R to an API client, the object must be *serialized* into some format that the client can understand (JSON format).

Note that, **@get** and **@serializer** annotations must to be put on the top of the code. Any comments must not be inserted before the annotations or between them and the R function.

Real Time Data Analytics using R Studio

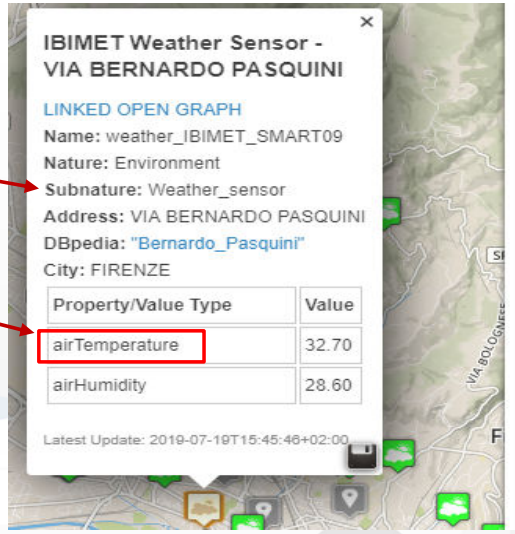
How to create a *plumberized* R script - 2

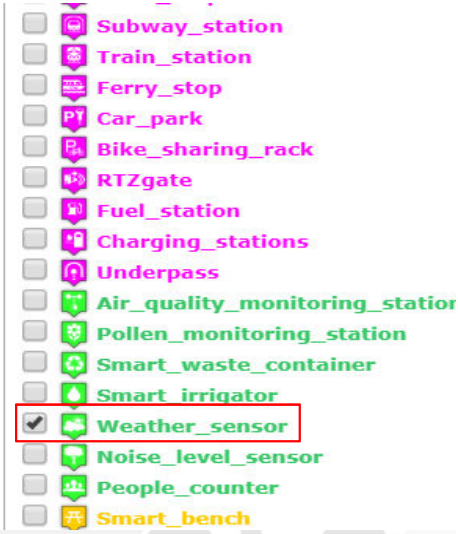
- Step 2 - Create an R function with the same name of the **@get** parameter:

```
TuscanyHeatmap <- function(sensorCategory, varName, fromDateTime, toDateTime, heatmapName){
```

heatmapName = "airTemperatureTuscanyTest"
sensorCategory = "Weather_sensor"
varName = "airTemperature"
toDateTime = "2019-07-23T10:00:00"
fromDateTime = "2-hour"

https://www.snap4city.org/dashboardSmartCity/management/iframeApp.php?linkUrl=https%3A%2F%2Fservicemap.snap4city.org%2F&linkId=map1link2&pageTitle=Service%20Map%20(Toscana)&fromSubmenu=kmlink





Real Time Data Analytics using R Studio

How to download Real-Time data using API - 1

- Step 3 - Upload All Service Uris (sensor stations) from service map in the area of interest:

```
query <- paste("https://servicemap.disit.org/WebAppGrafo/api/v1/?selection=
               42.5024779733486 9;8.19580078125;44.6061127451739;13.4225463867187
               &categories=",sensorCategory,
         "&maxResults=0&maxDists=0.1&format=json",sep="")

sensorCategoryJson <- fromJSON(query) #jsonlite package

suri <- sensorCategoryJson$Services$features$properties$serviceUri #serviceUri
```

Real Time Data Analytics using R Studio

How to download Real-Time data using API - 2

```
https://servicemap.disit.org/WebAppGrafo/api/v1/?selection=42.6789
7316354954;9.954032295814045;44.00523270268637;12.063407295814045&
categories=Weather_sensor&maxResults=0&maxDists=0.1&format=json
```

```
"http://www.disit.org/km4city/resource/IBIMET_SMART11"
"http://www.disit.org/km4city/resource/IBIMET_SMART04"
"http://www.disit.org/km4city/resource/IBIMET_SMART13"
"http://www.disit.org/km4city/resource/IBIMET_SMART06"
"http://www.disit.org/km4city/resource/IBIMET_SMART17"
"http://www.disit.org/km4city/resource/IBIMET_SMART33"
"http://www.disit.org/km4city/resource/IBIMET_SMART33"
"http://www.disit.org/km4city/resource/IBIMET_SMART25"
"http://www.disit.org/km4city/resource/IBIMET_SMART24"
"http://www.disit.org/km4city/resource/IBIMET_SMART30"
[...]
```

Real Time Data Analytics using R Studio

How to download Real-Time data using API - 3

- Step 4 - Upload data related to a specific time interval (fromTime/toTime) for each Service Uri:

```
sensorData <- vector("list", length(suri))
for (i in 1:length(suri)){
  temp=c()
  #api to upload the realtime data
  api <- paste("https://servicemap.disit.org/WebAppGrafo/api/v1/?serviceUri=",
               suri[i],"&fromTime=",fromDateTime,
               "&toTime=",toDateTime,sep="")
  sensorCategoryData <- fromJSON(api)
```

```
https://servicemap.disit.org/WebAppGrafo/api/v1/?serviceU
ri="http://www.disit.org/km4city/resource/IBIMET_SMART11"
&fromTime=2-hour&toTime=2019-07-23T10:00:00
```

Real Time Data Analytics using R Studio

How to download Real-Time data using API - 4

- Step 5 – Data manipulation and data Interpolation…

… After data manipulation and interpolation we obtain something like this:

long	lat	value
11.24686	42.76616	39.87238
11.30287	42.76616	39.54115
11.35888	42.76616	39.20993
11.41489	42.76616	38.87870
11.47090	42.76616	38.54747
11.52691	42.76616	38.21624
11.58292	42.76616	37.88501

[…]

Interpolated values

Real Time Data Analytics using R Studio

How to save heatmaps using API - 1

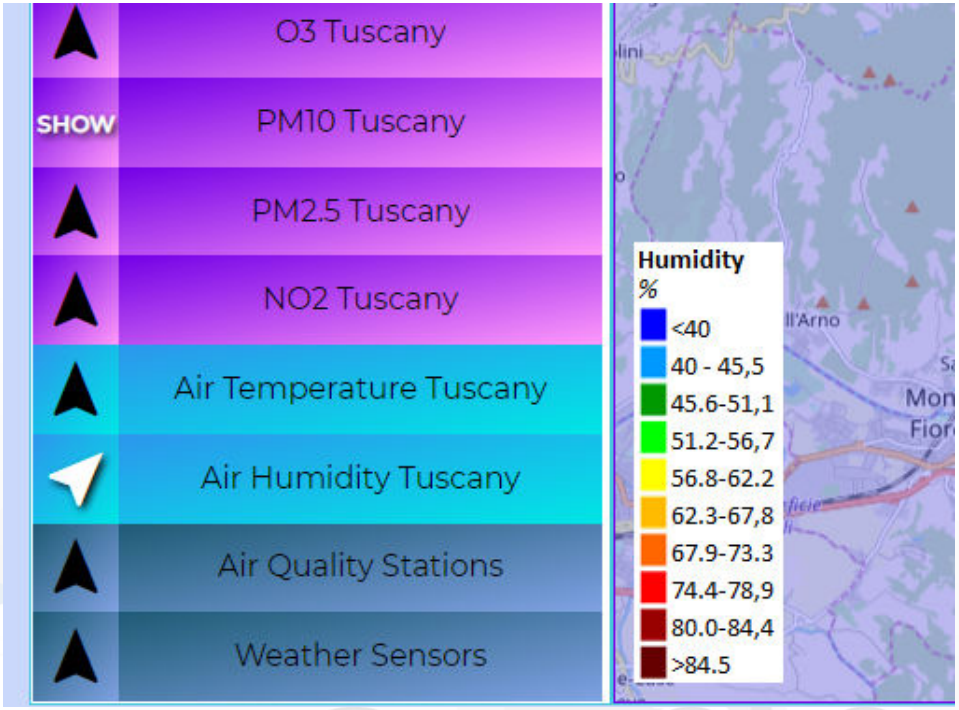
- Step 6 - Create a R list:

```
interpolatedHeatmap=list()
interpolatedHeatmap$attributes=vector("list", dim(interpolatedData)[1])
interpolatedHeatmap$saveStatus=list()

for(i in 1:dim(interpolatedData)[1]) {

  #list
  lat = as.numeric(interpolatedData[i, "lat"])
  long = as.numeric(interpolatedData[i, "long"])
  meanObs = interpolatedData[i, "value"]

  listAttribTemp = list("mapName"=heatmapName,  "metricName"= metricName,
                        "description"= paste("Average from",fromDateTime,"to",toDateTime,sep=" "),
                        "clustered"= 0, "latitude"=lat, "longitude"=long,
                        "value"= meanObs, "date"= paste(toDateTime, "Z", sep=""), "org"="DISIT")

  interpolatedHeatmap$attributes[[i]]=listAttribTemp
```

Real Time Data Analytics using R Studio

How to save heatmaps using API - 2

Note that, the "metricName" identifies the legend for each heatmap and the colour scale to be used.



It corresponds to the *varName* of the R function except for PM10 and PM2.5 measurements:

- "HighDensityPM10"
- "HighDensityPM25"

Real Time Data Analytics using R Studio

How to save heatmaps using API - 3

- Step 7 - Transform the R list in a Json and save heatmap data using API:

```
request_body_json <- toJSON(interpolatedHeatmap$attributes, auto_unbox = TRUE, digits = 10)

resultPOST <- POST(url = "http://snap4city:disit2019@192.168.0.59:8000/insertArray",
                   body = request_body_json,
                   encode = "json", add_headers("Content-Type" = "application/json"))
```

[...]

```
  apiFinal <- paste("http://192.168.0.59/setMap.php?mapName=",heatmapName,
                    "&metricName=",metricName,
                    "&date=",paste(toDateTime, "Z", sep=""),
                    "&completed=", completed,  sep="")
resultPOST <- GET(url = apiFinal)
```

Real Time Data Analytics using R Studio

How to save heatmaps using API - 4

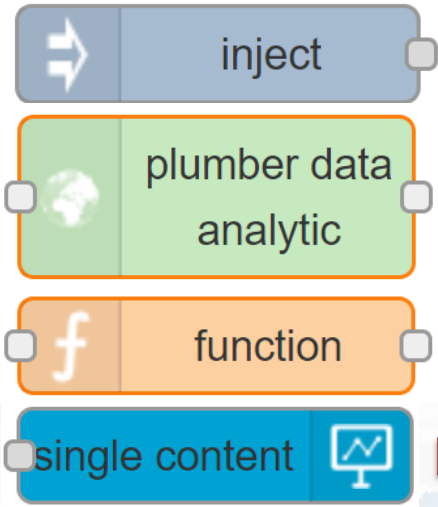
JSON Array Format example


```
[
{
"mapName": "airTemperatureTuscanyTest",
"metricName": "airTemperature",
"description": " Air Temperature heatmap … ",
"clustered": 0,
"latitude": 43.1,
"longitude": 11.1,
"value": 25.5,
"date": "2019-07-23T10:00:00Z"
"org": "DISIT"
}, { […] }]
```

IOT App for Real Time Data Analytics

How to create a Data Analytics IOT Application

What we need:



To insert the R function parameter

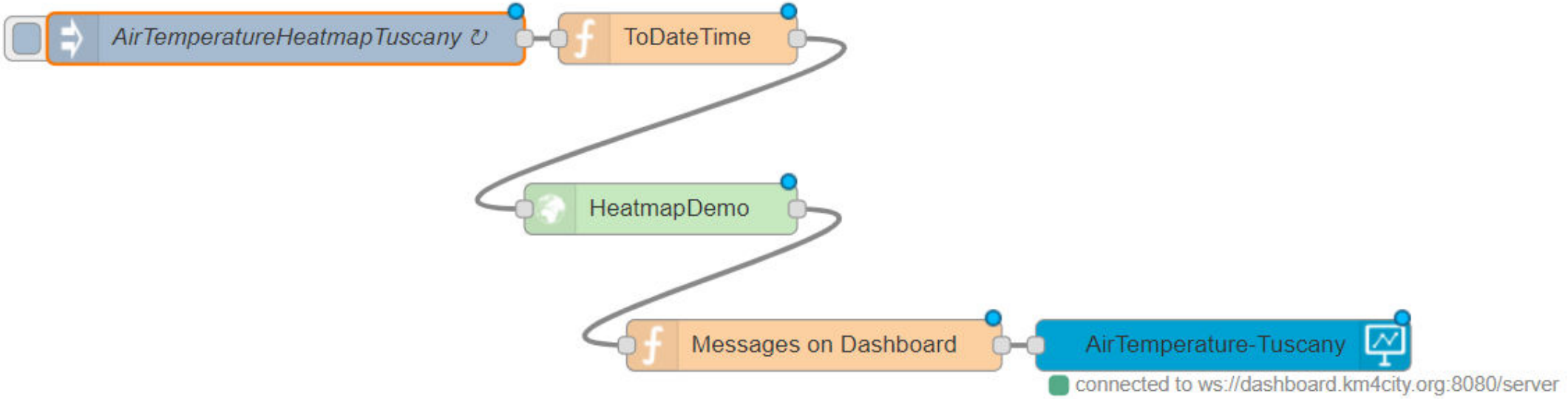
To upload the R script and create a plumber instance

To visualize strings/numbers/html on a dashboard

To execute JavaScript code on output messages

IOT App for Real Time Data Analytics

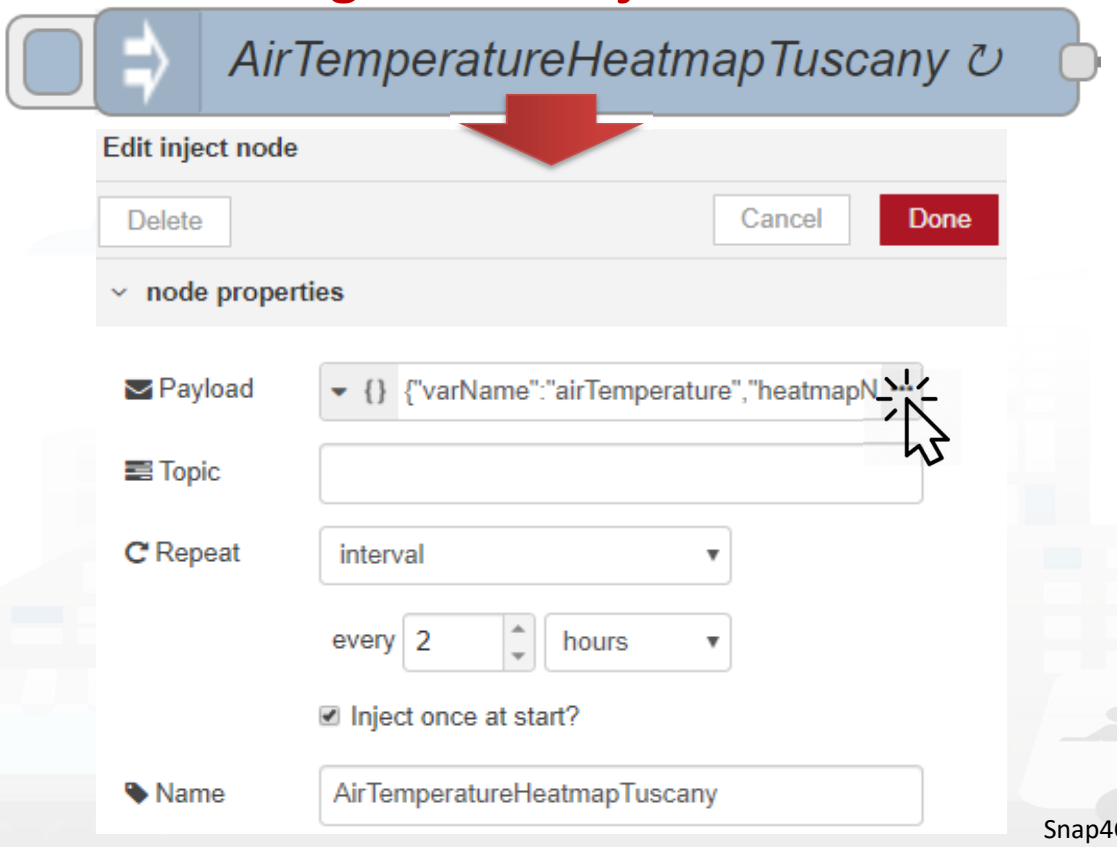
How to create a Data Analytics IOT Application



IOT App for Real Time Data Analytics

Nodes Configuration – Inject Node

How to configure the **inject** node:



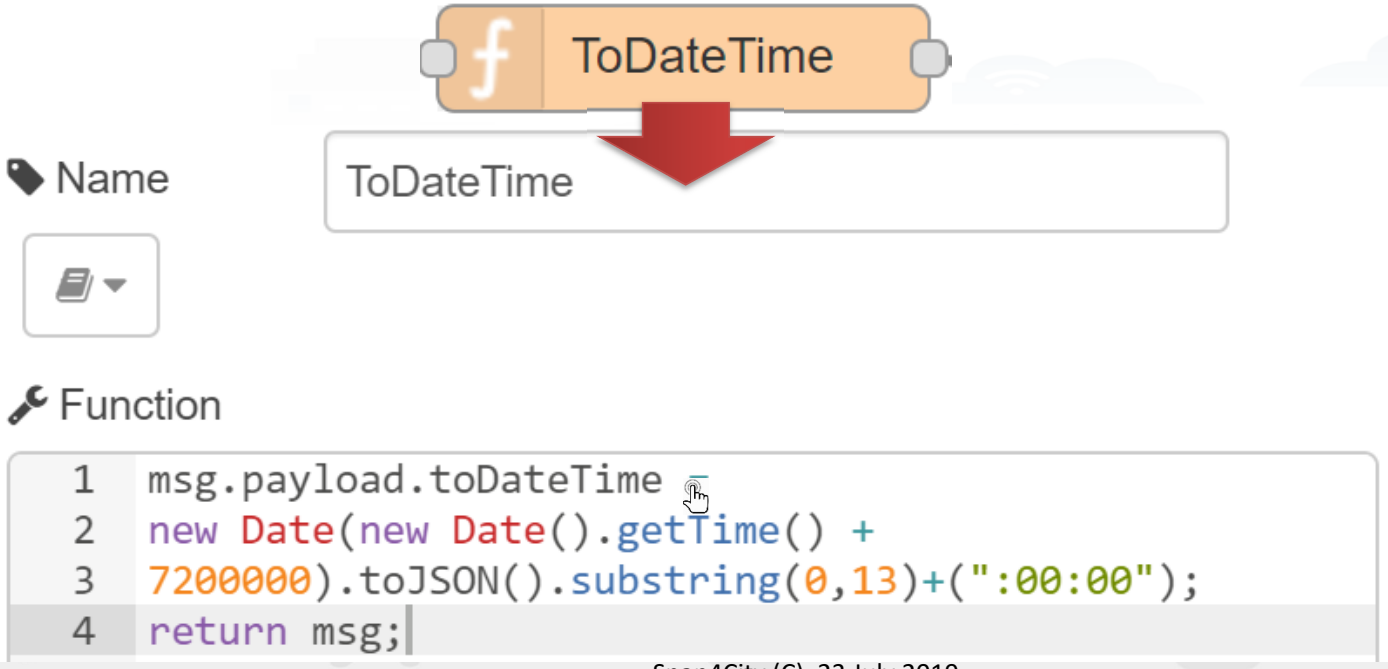
The JSON Format of the Payload property has the same notation of the R function parameters:

```
{   "varName": "airTemperature",
"heatmapName":
"airTemperatureTuscanyTest",
"fromDateTime": "2-hour",
"sensorCategory": "Weather_sensor"
}
```

IOT App for Real Time Data Analytics

Nodes Configuration – Function Node for Date and Time

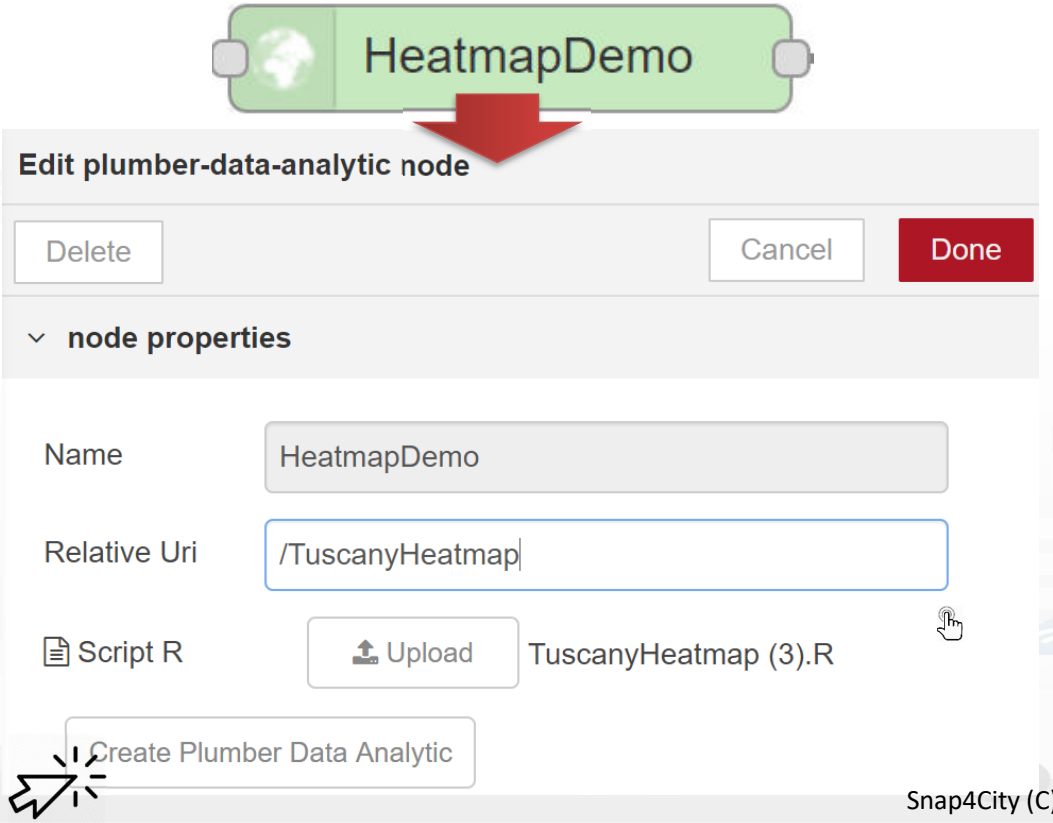
- Before configure the plumber data analytic node is necessary to execute a JavaScript code to dynamically update the date ("toDateTime" parameter):



IOT App for Real Time Data Analytics

Nodes Configuration – Plumber Data Analytic Node

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



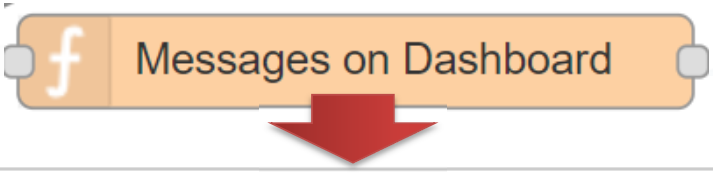
Relative Uri is the same of the R @get annotation:

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

IOT App for Real Time Data Analytics

Nodes Configuration – Function Node for Messages on Dashboard

- Before configure the single content node is necessary to execute a JavaScript code to visualize the status of the heatmap:



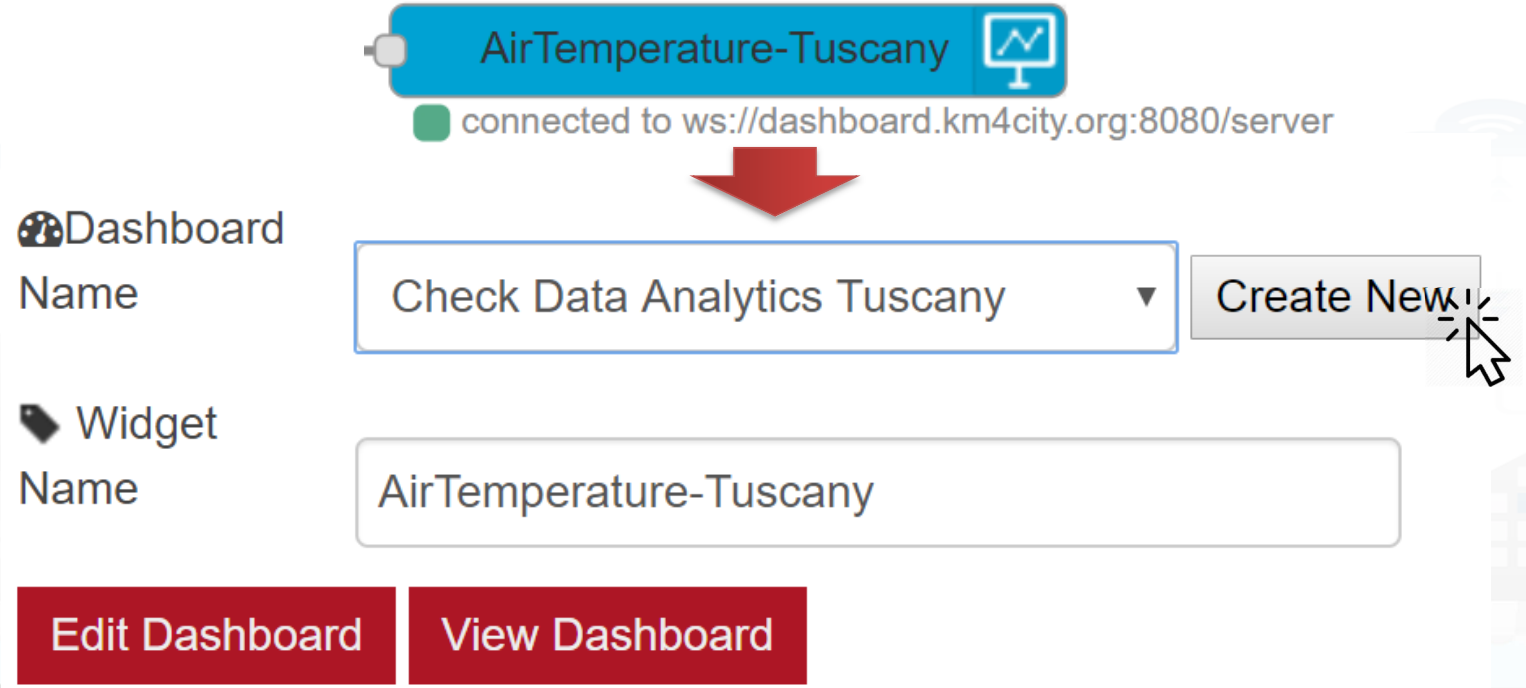
Function

```
1  msg.payload=msg.payload.message+" "+msg.payload.dateTime;
2  if(msg.payload.indexOf("Completed")!= -1){
3      msg.payload ="<span style='color:green;'>"+
4      msg.payload + "</span>"
5  } else  if (msg.payload.indexOf("No Availabe Data") != -1){
6      msg.payload ="<span style='color:orange;'>"+
7      msg.payload + "</span>"
8  }
9  return msg;
```

IOT App for Real Time Data Analytics

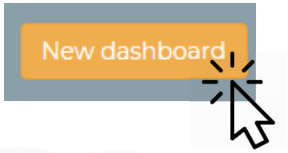
Nodes Configuration – Single Content Node

How to configure the **single content** node:

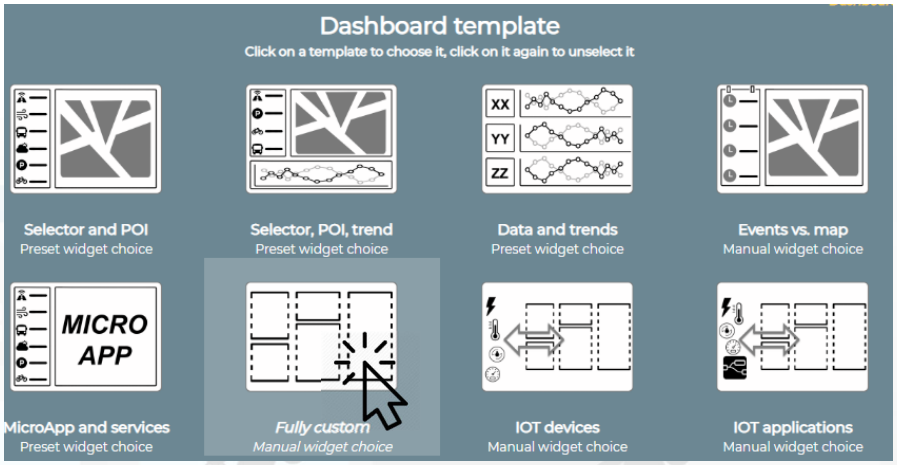


Wizarded Heatmap Visualization

1. Create a New Dashboard from Dashboard (Public)

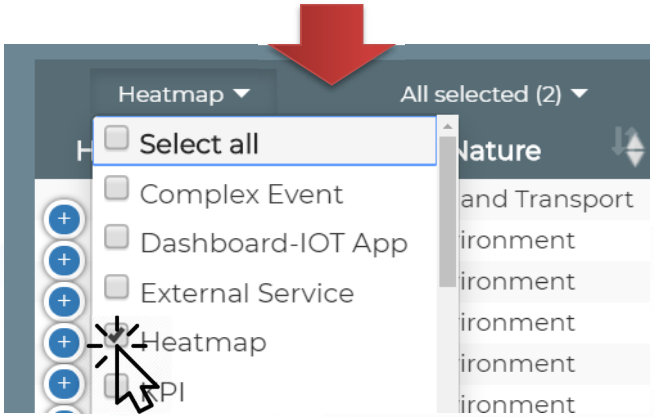


2. Insert a Dashboard Title and select a Dashboard Template

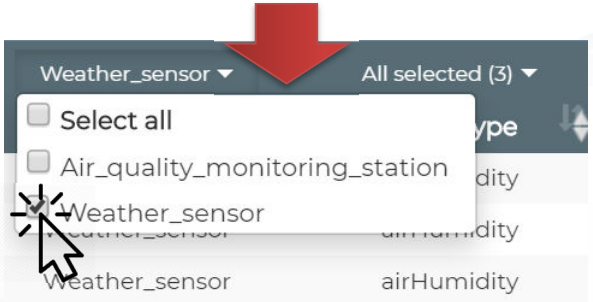


Wizarded Heatmap Visualization

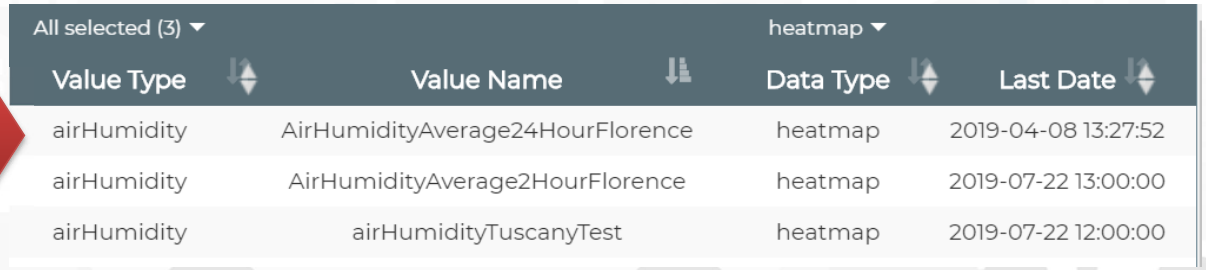
3. Select the Heatmap box as High-Level-Type



4. Select the Sensor Category (Subnature)



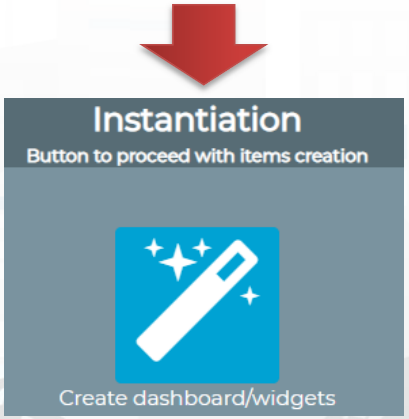
5. Select the measure (Value Type) and the Heatmap Name (Value Name)

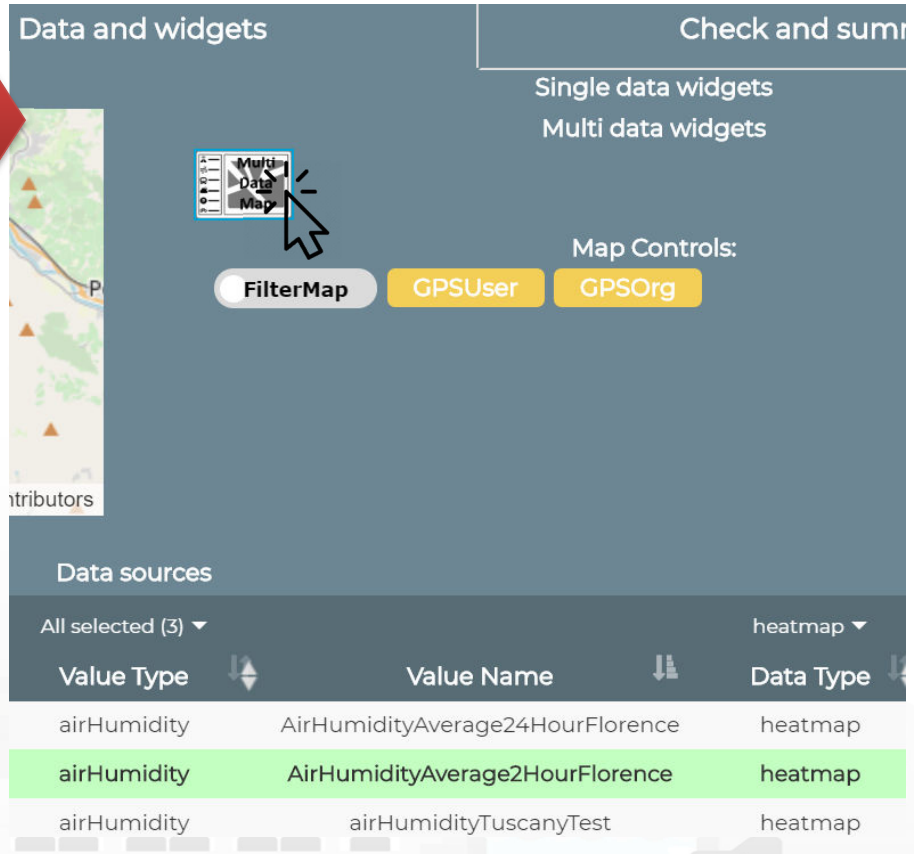


Wizarded Heatmap Visualization

6. After the Heatmap selection, select the Multi Data Map button and click on next

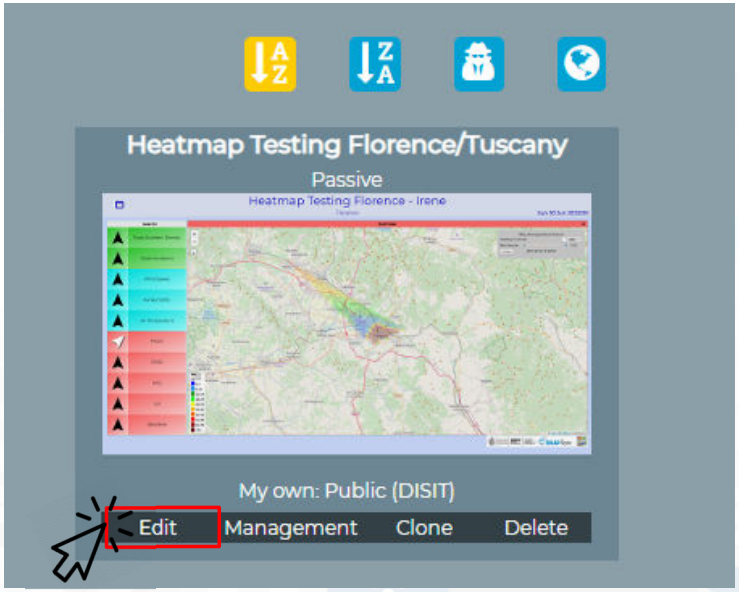
7. Select the instantiation button to proceed with items creation



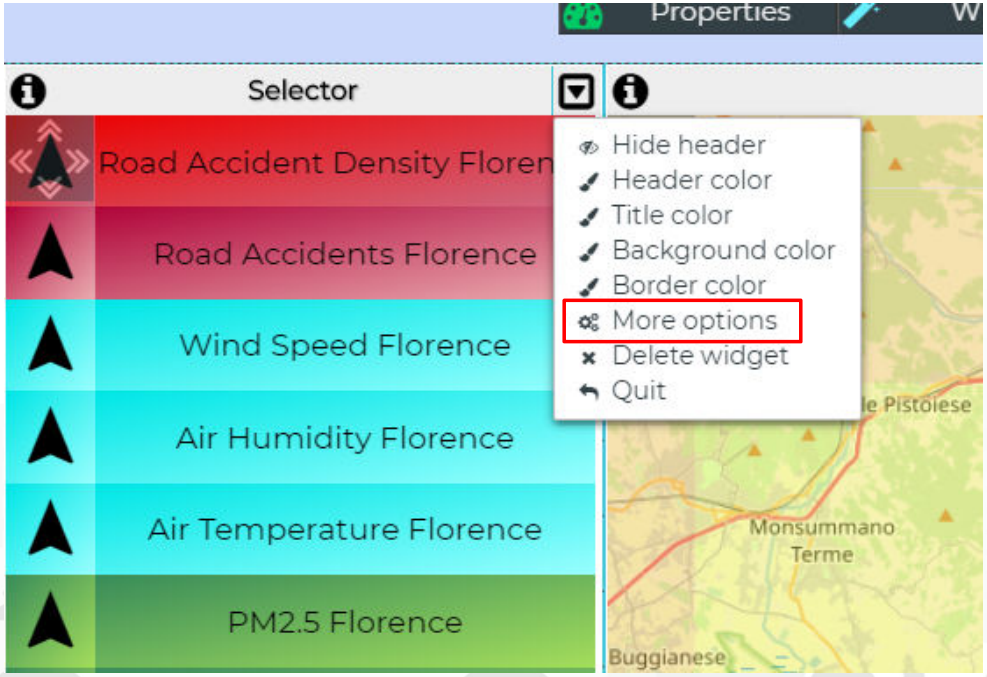


Manually Heatmap Visualization

1. Select a Dashboard and click on Edit

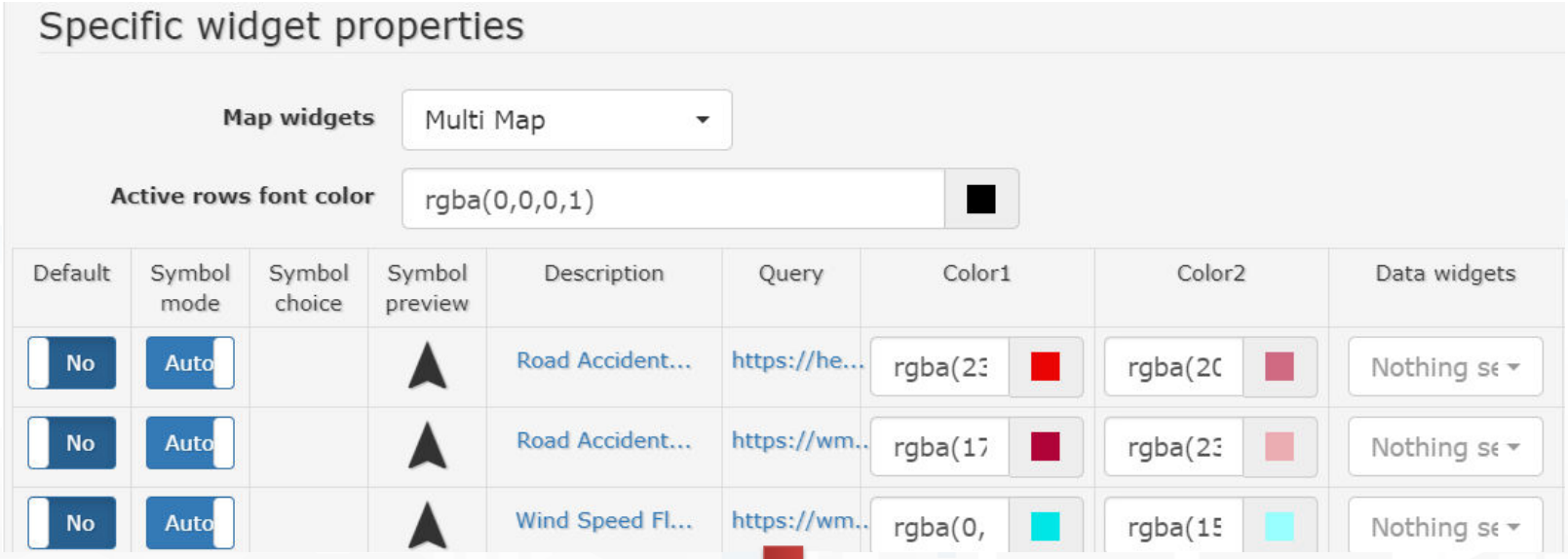


2. Select on More Options to modify the widget properties



Manually Heatmap Visualization

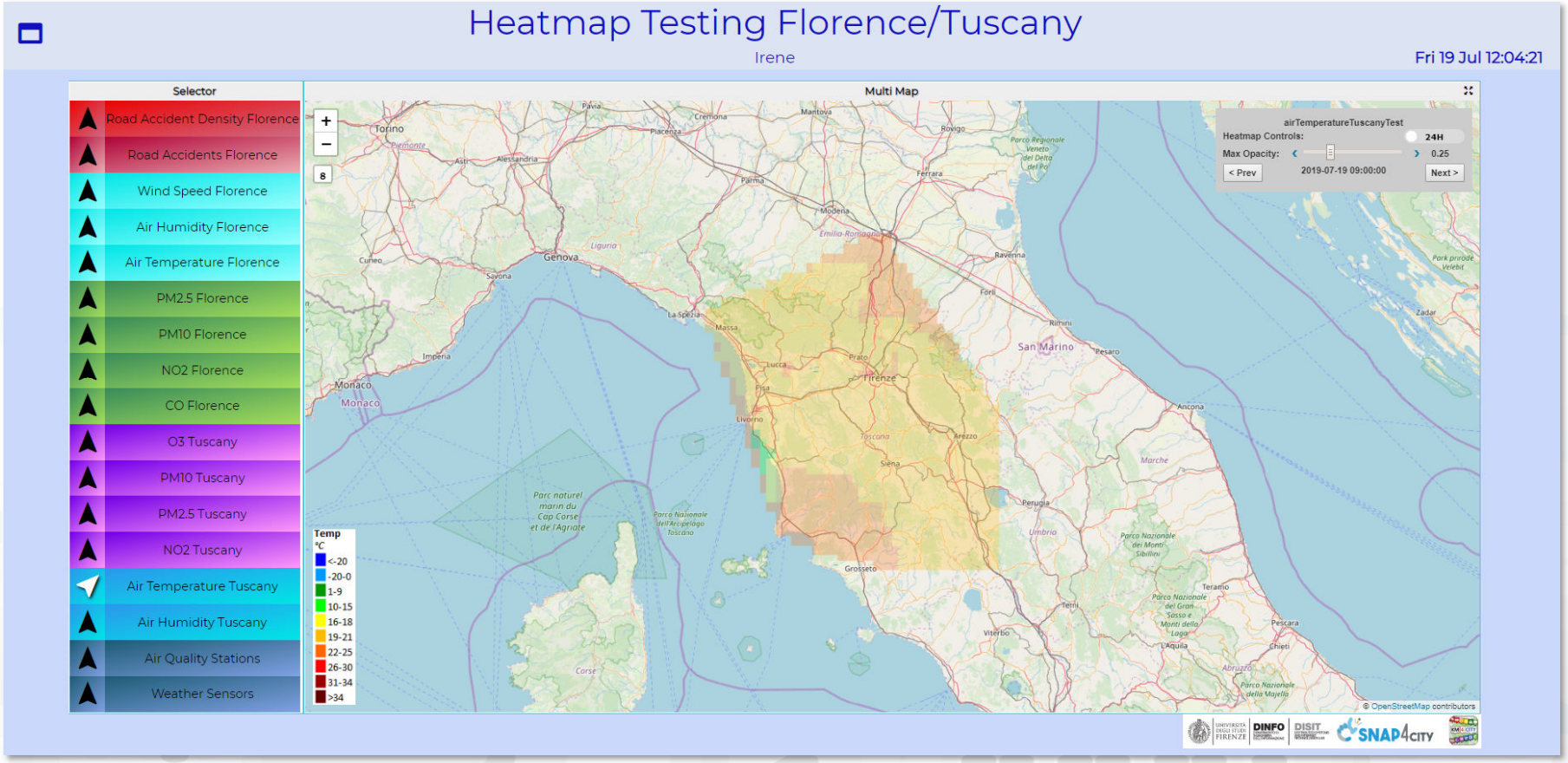
3. Change the Query to visualize the new heatmap



https://wmsserver.snap4city.org/geoserver/Snap4City/wms?service=WMS&layers=heatmapName

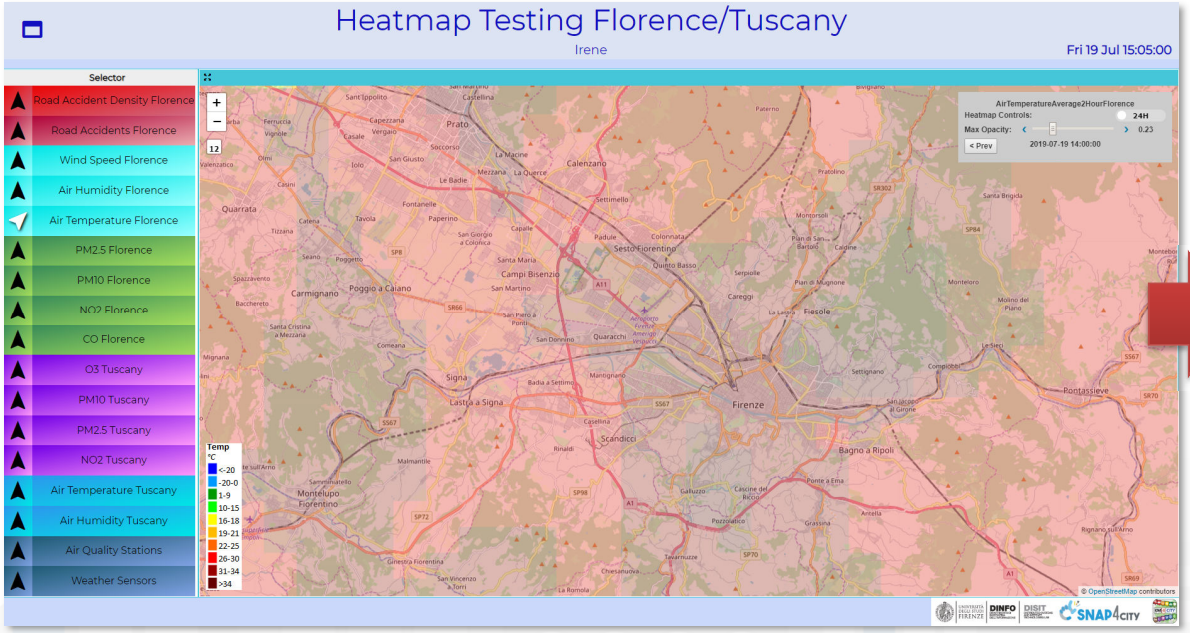
https://wmsserver.snap4city.org/geoserver/Snap4City/wms?service=WMS&layers=airTemperatureTuscanyTest

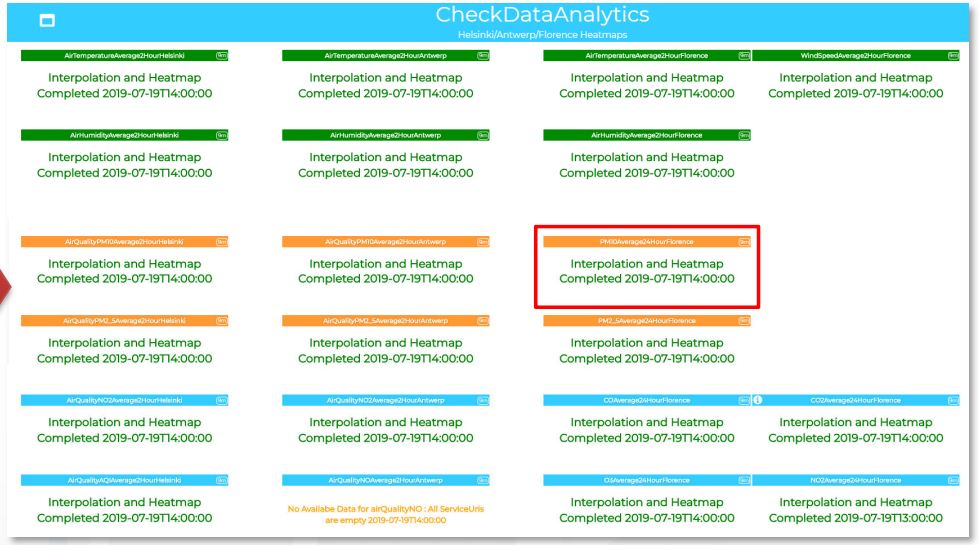
Heatmap Visualization



https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTI2OA==

Heatmap Visualization and Heatmap Status Check on Dashboards





https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc3MQ==

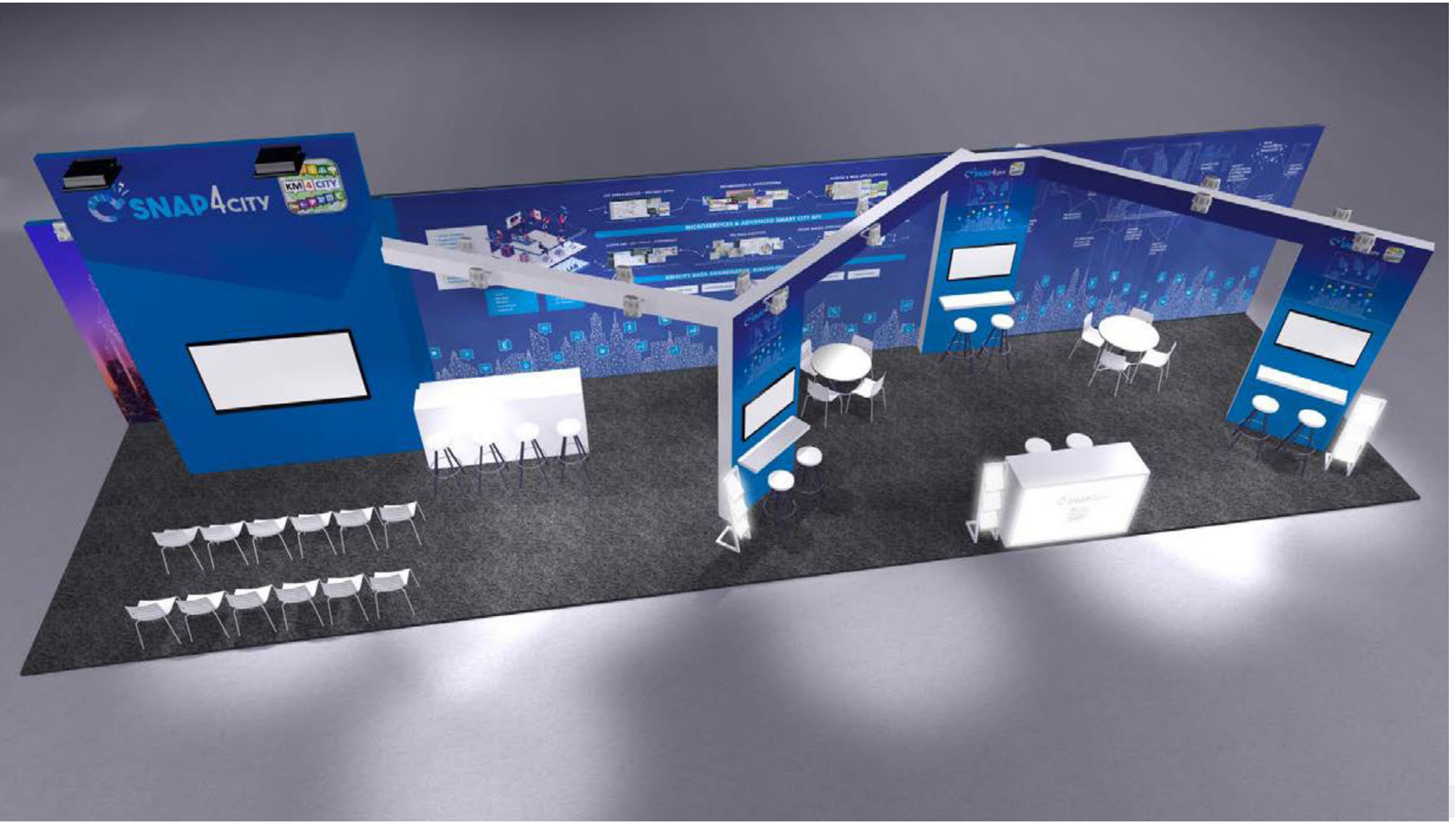
R studio Development documentation (self training)

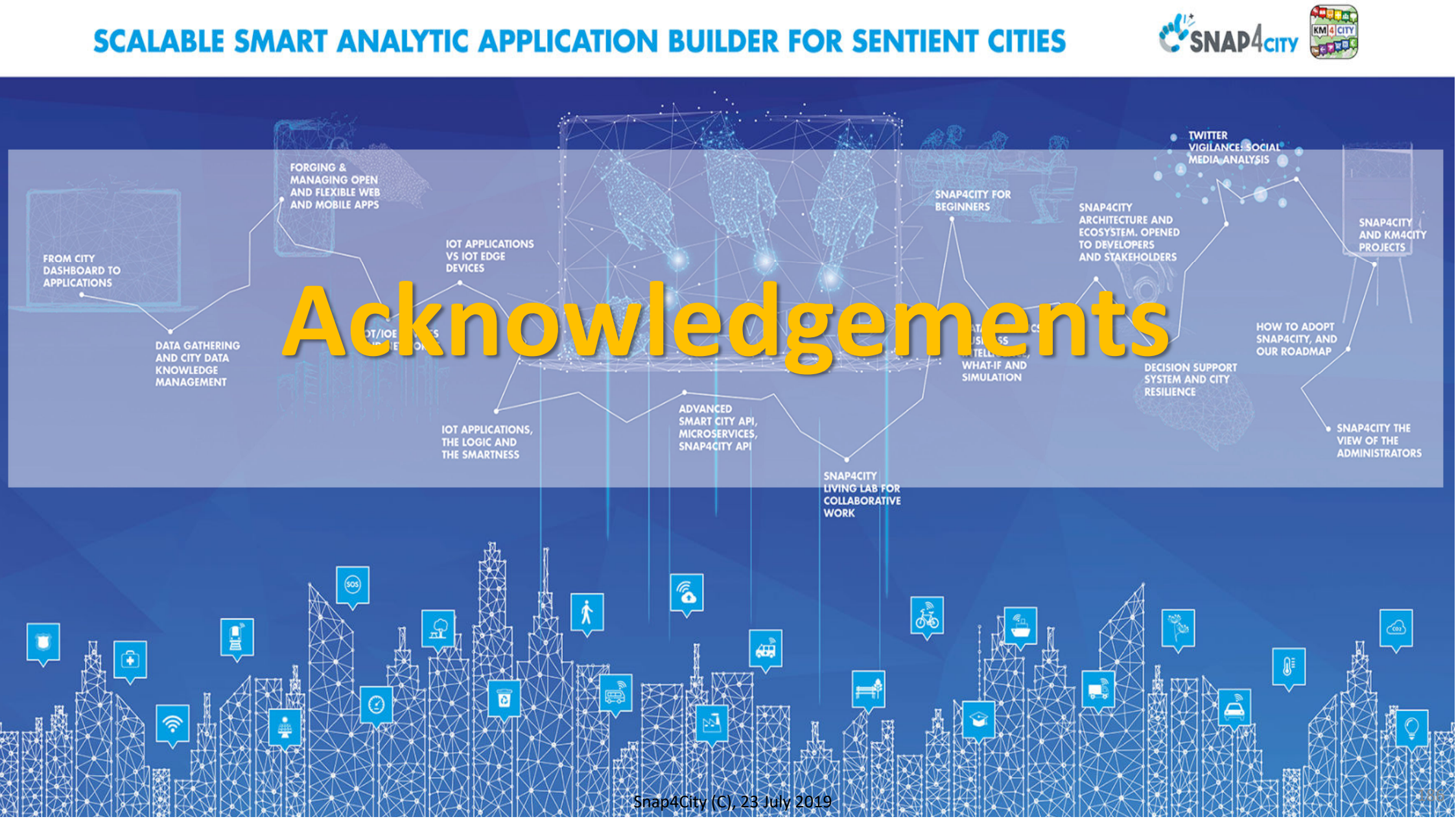
https://www.snap4city.org/dashboardSmartCity/management/iframeApp.php?linkUrl=https%3A%2F%2Fwww.snap4city.org%2Fdrupal%2Fnode%2F25&linkId=25link&pageTitle=Doc:%20R%20Studio%20Development&fromSubmenu=handddocLink

- TC7.1. Exploiting data analytics and machine learning in IOT Applications as MicroService
- TC7.2. R Studio for Analytics, exploiting Tensor Flow
- TC7.3. Download data from AMMA (Application and MicroService Monitor and Analyser), ResDash (Resource Dashboard) and DevDash (Development Dashboard) tools
- TC7.4. From R Studio process to MicroService for IOT application, data analytics, machine learning
- TC7.5. Developing Data Analytics Processes
- TC7.6. How to get data from API into R studio
- TC7.7. How to Save resulting data via API from R studio
- TC7.8. Example of how to CreateLastValuesMean.R
- TC7.9. CreateHourlyAvgTrendPerDay.R
- TC7.10. CreateHeatmap.R
- TC2.31 - Create Data Analytic Flow
- TC2.32 - Make Your Data Analytic Flow Public

Smart City Expo World Conference 2019, Barcellona


SNAP4CITY
KM4CITY


SNAP4CITY
KM4CITY



Acknowledgements

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.*

SNAP4CITY

KM 4 CITY

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

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