



*Be smart in a SNAP!*

LIVING LAB

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

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

**SCALABLE SMART ANALYTIC APPLICATION BUILDER FOR SENTIENT CITIES**



UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND CULTURE  
TECHNOLOGIES LAB





UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB



**SNAP4city**



Powered by

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

<https://www.Snap4City.org>

100%  
OPEN  
SOURCE

23 July 2019, 3° Training day of 3, Florence

<https://www.snap4city.org/drupal/node/485>

Paolo Nesi, [paolo.nesi@unifi.it](mailto:paolo.nesi@unifi.it)

<https://www.Km4City.org>

<https://www.disit.org>



Snap4City (C), 23 July 2019





# 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.
- Moderate.
- Good.
- Golden.
- Professional.
- Excellent.



non programmer level

Some JavaScript rudiment coding

JavaScript programming





Programming in R Studio

Exploiting Smart City API

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](#) 



# Dashboard Management and GDPR

FROM CITY  
DASHBOARD TO  
APPLICATIONS

DATA GATHERING  
AND CITY DATA  
KNOWLEDGE  
MANAGEMENT

FORGING &  
MANAGING OPEN  
AND FLEXIBLE WEB  
AND MOBILE APPS

IOT/IOE DEVICES  
AND NETWORKS

IOT APPLICATIONS,  
THE LOGIC AND  
THE SMARTNESS

ADVANCED  
SMART CITY API,  
MICROSERVICES,  
SNAP4CITY API

SNAP4CITY  
LIVING LAB FOR  
COLLABORATIVE  
WORK

SNAP4CITY FOR  
BEGINNERS

DATA ANALYTICS,  
BUSINESS  
INTELLIGENCE,  
WHAT-IF AND  
SIMULATION

SNAP4CITY  
ARCHITECTURE AND  
ECOSYSTEM. OPENED  
TO DEVELOPERS  
AND STARTUPS

DECISION SUPPORT  
SYSTEM AND CITY  
RESILIENCE

TWITTER  
VIGILANCE, SOCIAL  
MEDIA ANALYSIS

HOW TO ADOPT  
SNAP4CITY, AND  
OUR ROADMAP

SNAP4CITY  
AND KM4CITY  
PROJECTS

SNAP4CITY THE  
VIEW OF THE  
ADMINISTRATORS



# *Dashboards Listing, Managing, Sharing, Delegation*







UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB



# Dashboard List and Editor

Snap4City

User: roottooladmin1, Org: none  
Role: RootAdmin, Level: 7

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

Dashboards

Cards

1 2 Next

data

New dashboard

DataCenter

Passive

disit: Public

Edit Management Clone Delete

Datacenter Energy Consumption

Passive

disit: Public

Edit Management Clone Delete

DataCenter gas and smoke (desktop)

Passive

disit: Public

Edit Management Clone Delete

DataCenter gas and smoke (mobile)

Passive

disit: Public

Edit Management Clone Delete

FirenzeWiFi

Passive

disit: Private

Edit Management Clone Delete

Florence data overview

Passive

disit: Public

Edit Management Clone Delete

Leonardo - Smart city data 2

Passive

Leonardo: Public

Edit Management Clone Delete

My data and trends

Passive

nicola.mitolo: Public

Edit Management Clone Delete

My data trends

Passive

nicola.mitolo: Public

Edit Management Clone Delete

Notificator monitoring

Passive

disit: Public

Edit Management Clone Delete

Pisa Real Time Data

Passive

mitolo: Public

Edit Management Clone Delete

Real Time Sensors via ServiceMap3D

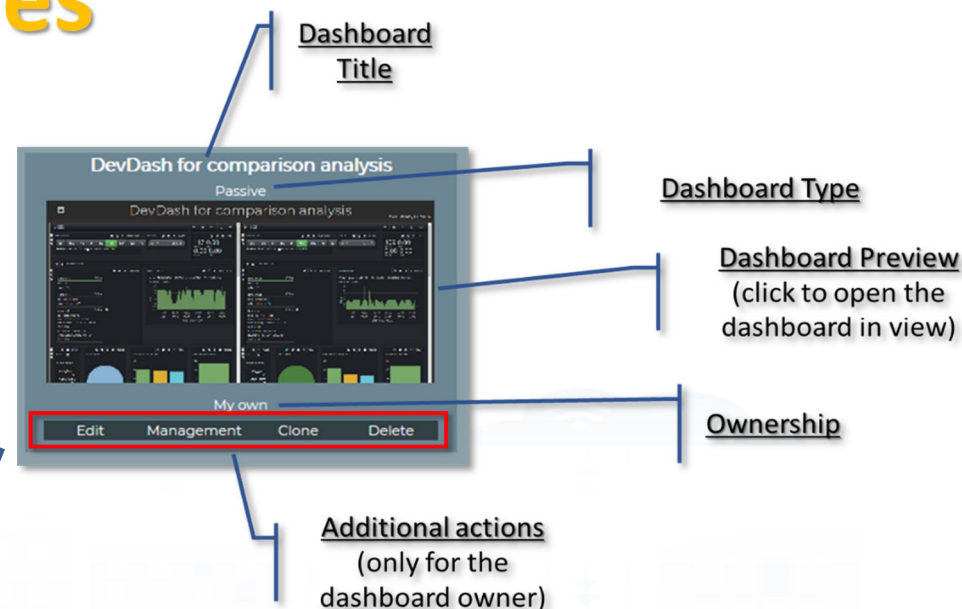
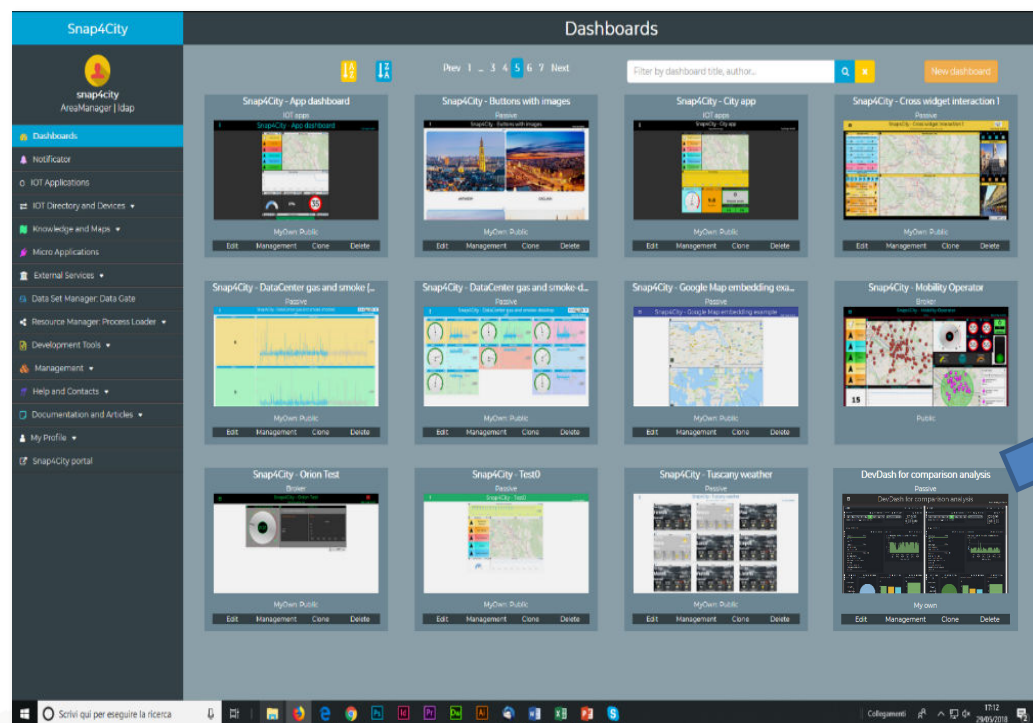
Passive

disit: Public

Edit Management Clone Delete



# 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]

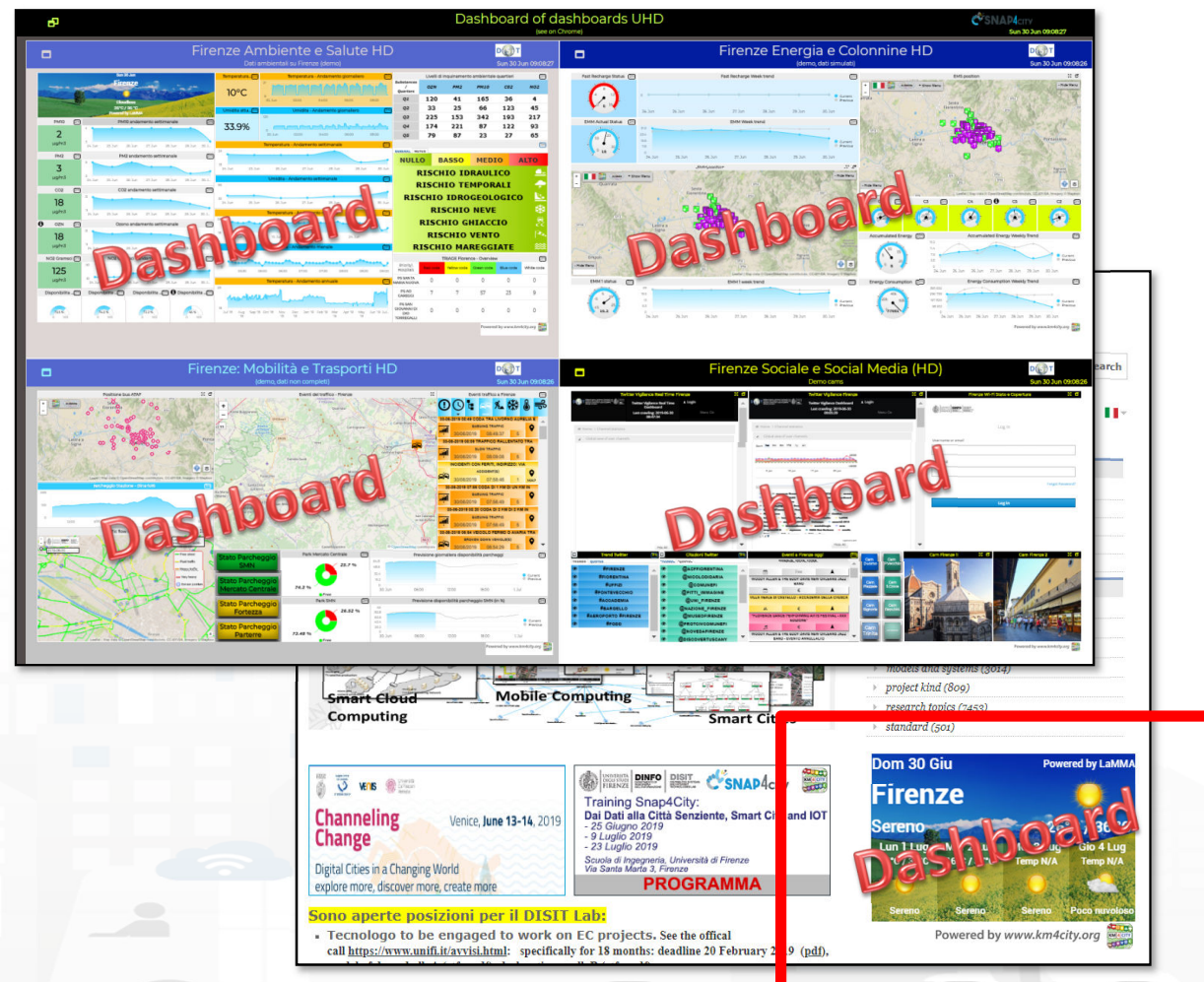
The image displays four overlapping screenshots of the Snap4City Management dashboard, illustrating various configuration options:

- Top Screenshot (Ownership):** Shows the "Ownership" tab for "Air Quality Sensor in Jätkäsaari - Helsinki". It includes a map and a "Change ownership" form with a "New owner username" field and a "Confirm" button. A warning message states: "New owner username can't be empty".
- Second Screenshot (Visibility):** Shows the "Visibility" tab for the same sensor. It includes a "Change visibility" dropdown menu set to "Public" and a "Confirm" button.
- Third Screenshot (Delegations):** Shows the "Delegations" tab for "Helsinki vs Antwerp comparison". It includes an "Add new delegation" form with a "Delegated username" field and a "Confirm" button. A warning message states: "Delegated username can't be empty". Below, it shows "Current user delegations" with a table listing "Delegated user" (paolo.ant2) and a "Remove" button.
- Bottom Screenshot (Group Delegations):** Shows the "Group Delegations" tab for "Helsinki vs Antwerp comparison". It includes an "Add new group delegation" form with dropdowns for "Helsinki" and "Citizens", and a "Confirm" button. Below, it shows "Current group delegations" with a table listing "Delegated group" (Helsinki - All Groups) and a "Remove" button.



# 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..



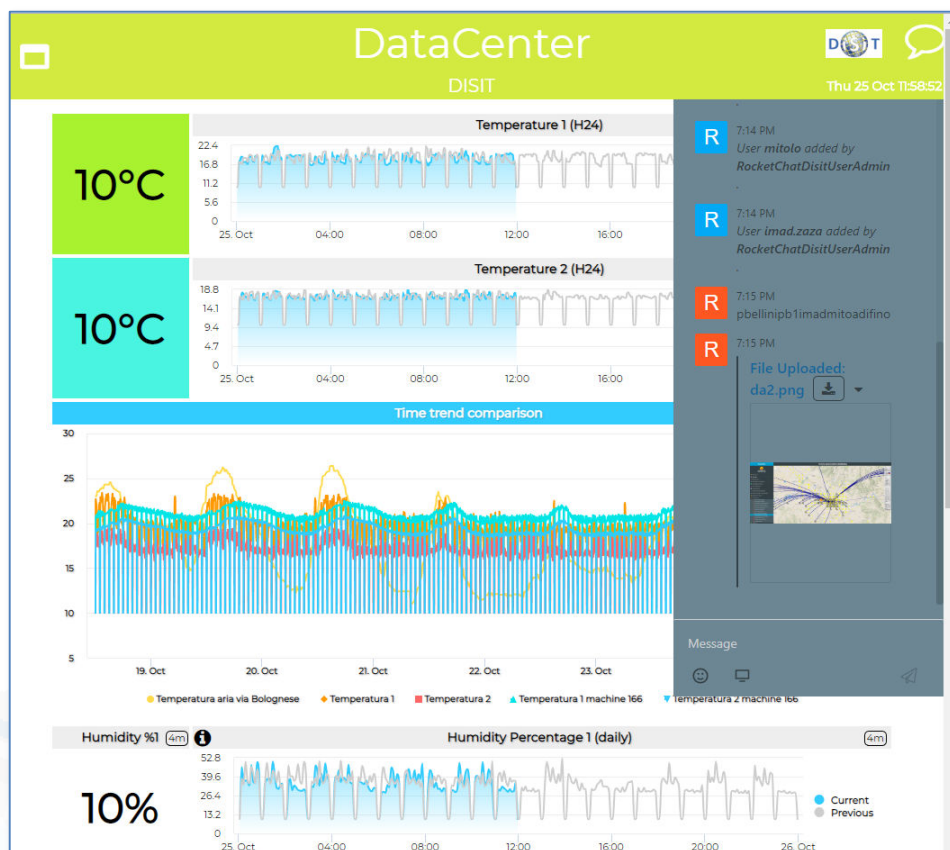


# *Dashboard Chatroom and Notifications*





# ChatRoom Per Dashboard



## Chat Management

The screenshot displays a RocketChat interface for a user named 'rootooladmin' with the role 'RootAdmin, Level: 7'. The left sidebar shows a navigation menu with various options including 'Dashboards', 'My Dashboards', 'Notifier', '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', and 'User Chats Management'. The main area shows a chat room titled 'Dashboard' with a list of channels and a chat history. The chat history includes messages from 'rootooladmin' and 'imad.zaza' added by 'RocketChatDisitUserAdmin'. A file named 'da2.png' has been uploaded to the chat.





## 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

**Notificator**  
Events generators management

Dashboard title	Widget title	Metric type	Dashboard link	Add/edit/delete notifications
DataCenter	Temp 1 edited	DCTemp1	<a href="#">Link</a>	<a href="#">Settings</a>
FirenzeWiFi	AP FWIFI	WifiStream_Aps	<a href="#">Link</a>	<a href="#">Settings</a>
Mugnone 2016	Utenti attivi (12 ore)	N_Active_Users	<a href="#">Link</a>	<a href="#">Settings</a>
Mugnone 2016	Utenti attivi (giorno)	N_Active_Users	<a href="#">Link</a>	<a href="#">Settings</a>
DataCenter	Temperature 2 (last 12 hours)	DCTemp2	<a href="#">Link</a>	<a href="#">Settings</a>
DataCenter	Temp 2	DCTemp2	<a href="#">Link</a>	<a href="#">Settings</a>

**Events log**

Generator container	Generator name	Generator type	User	Event time	Event type	Application	Link
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini	2017-08-09 17:38:03	Value <= 50 - Bad	Dashboard Manager	<a href="#">Link</a>
ToolAdmin - Public	Park Free - Speedo	Park_Free	marazzini	2017-08-09 17:37:32	27 < Value <= 54 - Ok	Dashboard Manager	<a href="#">Link</a>
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini	2017-08-09 17:37:02	Value <= 50 - Bad	Dashboard Manager	<a href="#">Link</a>
ToolAdmin - Public	Park Free - Speedo	Park_Free	marazzini	2017-08-09 17:36:28	27 < Value <= 54 - Ok	Dashboard Manager	<a href="#">Link</a>
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini	2017-08-09 17:36:02	Value <= 50 - Bad	Dashboard Manager	<a href="#">Link</a>
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini	2017-08-09 17:35:58	Value <= 11 - Low	Dashboard Manager	<a href="#">Link</a>
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini	2017-08-09 17:35:42	6 < Value <= 30 - Ok	Dashboard Manager	<a href="#">Link</a>
ToolAdmin - Public	Park Free - Speedo	Park_Free	marazzini	2017-08-09 17:35:25	27 < Value <= 54 - Ok	Dashboard Manager	<a href="#">Link</a>



# Data Type Management GDPR Compliant

FROM CITY  
DASHBOARD TO  
APPLICATIONS

DATA GATHERING  
AND CITY DATA  
KNOWLEDGE  
MANAGEMENT

FORGING &  
MANAGING OPEN  
AND FLEXIBLE WEB  
AND MOBILE APPS

IOT/IOE DEVICES  
AND NETWORKS

IOT APPLICATIONS,  
THE LOGIC AND  
THE SMARTNESS

SMART APPLICATIONS,  
IOT AND  
GEOSPATIAL DEVICES

SMART CITY API,  
MICROSERVICES,  
SNAP4CITY API

SNAP4CITY  
LIVING LAB FOR  
COLLABORATIVE  
WORK

SNAP4CITY FOR  
BEGINNERS

DATA ANALYTICS,  
BUSINESS  
INTELLIGENCE,  
WHAT-IF AND  
PREDICTIONS

SNAP4CITY  
ARCHITECTURE AND  
ECOSYSTEM. OPENED  
LEVELS AND CAPABILITIES

DECISION SUPPORT  
SYSTEM AND CITY  
RESILIENCE

TWITTER  
VIGILANCE, SOCIAL  
MEDIA ANALYSIS

HOW TO ADOPT  
SNAP4CITY, AND  
OUR ROADMAP

SNAP4CITY  
AND KM4CITY  
PROJECTS

SNAP4CITY THE  
VIEW OF THE  
ADMINISTRATORS



# 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**

- Correctness
- Transparency
- Security
- Integrity
- Privacy
- Auditing
- ...

**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**)



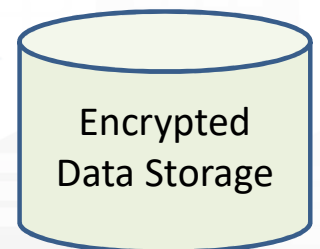
# GDPR: General Data Protection Regulation

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





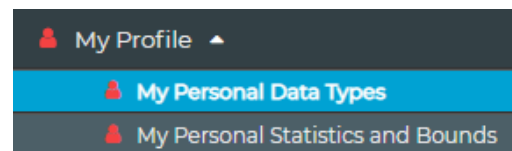
# GDPR Compliant

## 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 received 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 bas
- **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.



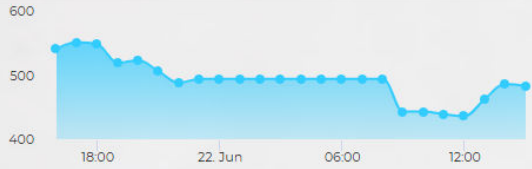


# Smart Bench

Sat 22 Jun 15:45:23

airQualityCO2 (8m)

483  
μg/m3



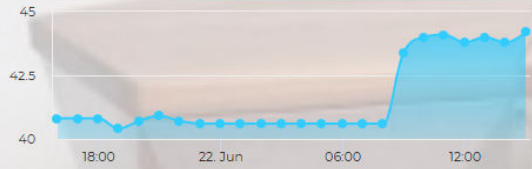
Temperature (8m)

27.1  
°C



Humidity (8m)

44.2  
%



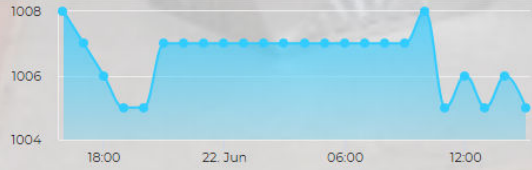
light (8m)

0



Pressure (8m)

1005  
hPa



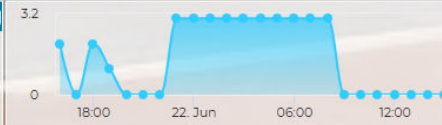
totalSittings (8m)

0



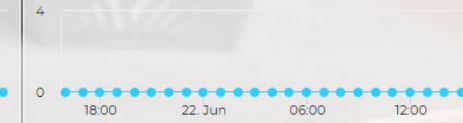
totalTransits (8m)

0



sittings (8m)

0



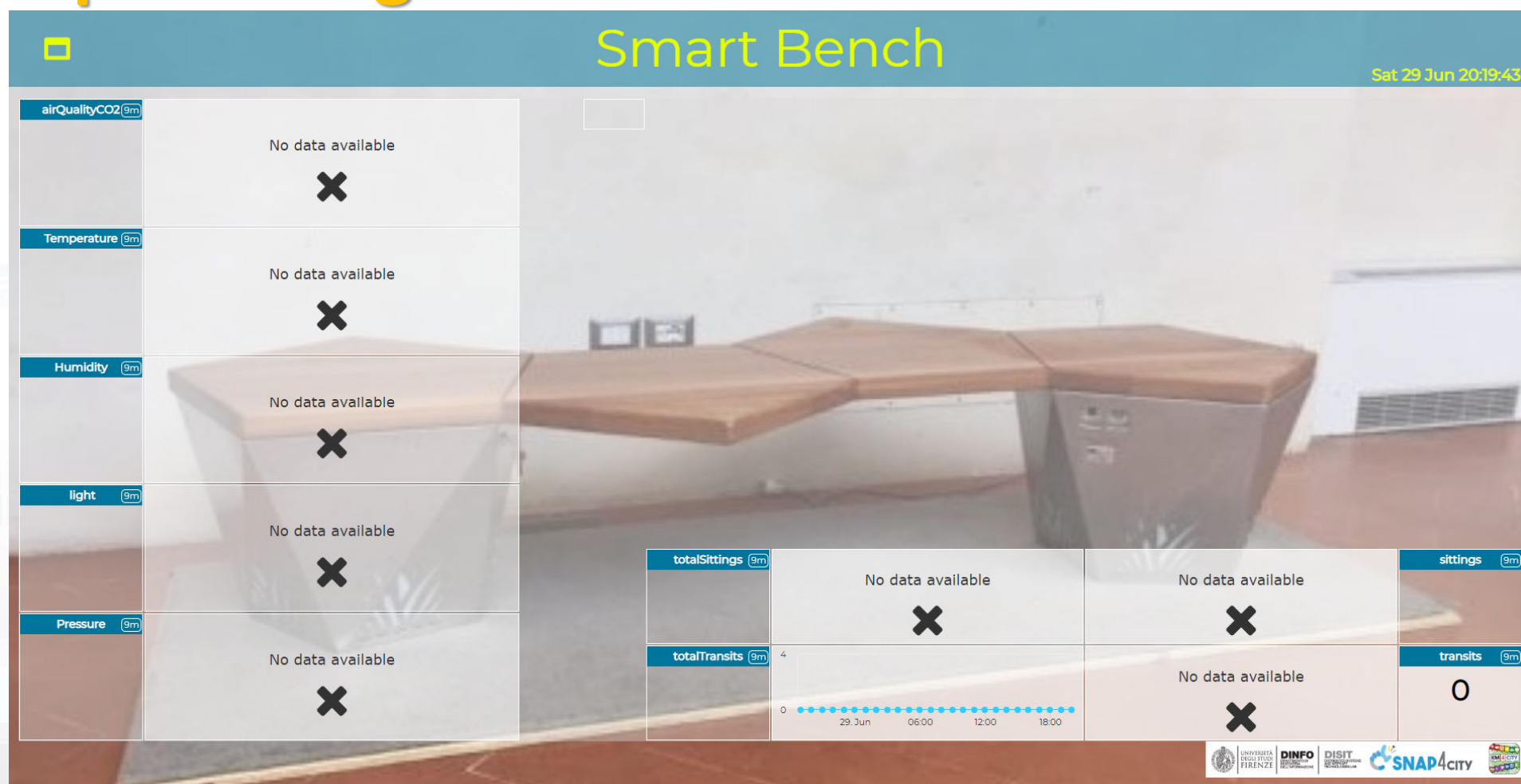
transits (8m)

0





# Example: Delegated Dashboard but not all data

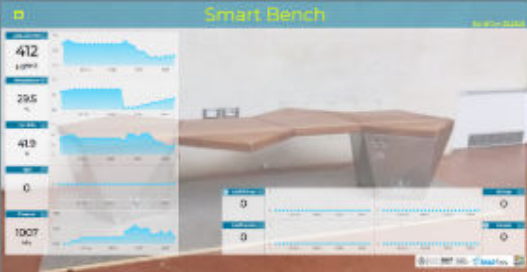




# Dashboard Delegation

Management

Ownership
Visibility
Delegations
Group Delegations

Smart Bench


Add new delegation


*Delegated username can't be empty*

Current user delegations

Delegated user	Remove
comunedashres	



# 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**

Snap4City

User: roottoolsadmin, Org: DIST

Role: RootAdmin, Level: 7

Add New

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

DIST Lab portal

My Personal Data

10 ▾																	
Datetime of insert ▾	Last Date ▾	Nature ▾	Sub Nature ▾	Value Type ▾	Value Name ▾	Data Type ▾	Last Value ▾	Ownshi p ▾	Descriptio n ▾	Info ▾	latitudes ▾	longitude ▾	paramete rs ▾	Edit	Value	View	
2018-10-05 17:47:51	2018-10-12 17:52:53	UtilitiesAndSupply	Energy_supply	MWh per persona per anno	Consumo di energia	float	12	private	Media consumo di energia per persona per anno					EDIT	ADD	VIEW	
2018-10-05 17:46:16	2018-10-05 17:43:15	Mobility and Transport	Accomodation	Percentage	Percentuale Km piste ciclabili sui km totali	percentage	19.7	private	Percentuale Km piste ciclabili sui km totali a Firenze					EDIT	ADD	VIEW	
2018-10-05 17:44:02	2018-10-05 17:43:24	Generic	Decision Support	Percentage	Tasso di disoccupazione	percentage	6.80	private	Tasso di disoccupazione a Firenze					EDIT	ADD	VIEW	
2018-10-05 17:43:12	2018-10-05 17:42:10	Environmen t	Accomodation	Euro a persona	PLI_residenti	integer	23606	private						EDIT	ADD	VIEW	
2018-10-05 17:41:32	2018-10-05 17:40:29	Environmen t	Pollution data	Number	Superamenti anno	integer	26	private	Superamenti PM10 anno a Firenze					EDIT	ADD	VIEW	
2018-10-05 17:39:01	2018-10-05 17:37:58	Environmen t	Waste_collection_and_treatment	Percentage	Percentuale di riciclo rifiuti	percentage	56	private	Percentuale di riciclo rifiuti in Firenze					EDIT	ADD	VIEW	
2018-10-05 17:36:42	2018-10-05 17:35:44	Environmen t	Waste_collection_and_treatment	tpes/anno	Quantità di rifiuto per abitante	float	0.629	private						EDIT	ADD	VIEW	
2018-08-31 17:55:36		Environmen t	Pollen data	Polline	Ambrosia	float		private	Misurazione polline famiglia Ambrosia		43.7847	11.2310		EDIT	ADD	VIEW	
	2018-04-03 00:05:43	Infrastruktur e			Bolognese_dewPoint	float	5.479	public						EDIT	ADD	VIEW	
	2018-04-03 00:05:46	Infrastruktur e			Bolognese_windGust	float	1.035	public						EDIT	ADD	VIEW	





UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB



# For example: the Smart Bench Data Delegation

My Data, KPI, POI

10 ☐ My ☐ Public ☐ Delegated Add My KPI Add My POI Add My Data

No. +	High Level Type	Nature	Sub Nature	Value Name	Value Type	Data Type	Last Date	Last Value	Ownership	Username	Controls	Data	Visibility
17056232	MyKPI	Entertainment	Smart_bench	Smart Bench Transits	transits	integer	6/29/2019, 7:59:52 PM	0	private <a href="#">MAKE PUBLIC</a>	disit_comunefi	<a href="#">VIEW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>	<a href="#">VALUES</a> <a href="#">METADATA</a>	<a href="#">DELEGATE USERS</a> <a href="#">CHANGE OWNERSHIP</a>
17056231	MyKPI	Entertainment	Smart_bench	Smart Bench Light	light	float	6/29/2019, 7:59:53 PM	0	private <a href="#">MAKE PUBLIC</a>	disit_comunefi	<a href="#">VIEW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>	<a href="#">VALUES</a> <a href="#">METADATA</a>	<a href="#">DELEGATE USERS</a> <a href="#">CHANGE OWNERSHIP</a>
17056230	MyKPI	Entertainment	Smart_bench	Smart Bench CO2	airQualityCO2	float	6/29/2019, 7:59:53 PM	412	private <a href="#">MAKE PUBLIC</a>	disit_comunefi	<a href="#">VIEW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>	<a href="#">VALUES</a> <a href="#">METADATA</a>	<a href="#">DELEGATE USERS</a> <a href="#">CHANGE OWNERSHIP</a>
17056229	MyKPI	Entertainment	Smart_bench	Smart Bench Total Sittings	totalSittings	integer	6/29/2019, 7:59:53 PM	0	private <a href="#">MAKE PUBLIC</a>	disit_comunefi	<a href="#">VIEW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>	<a href="#">VALUES</a> <a href="#">METADATA</a>	<a href="#">DELEGATE USERS</a> <a href="#">CHANGE OWNERSHIP</a>
17056228	MyKPI	Entertainment	Smart_bench	Smart Bench Sittings	sittings	integer	6/29/2019, 7:59:53 PM	0	private <a href="#">MAKE PUBLIC</a>	disit_comunefi	<a href="#">VIEW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>	<a href="#">VALUES</a> <a href="#">METADATA</a>	<a href="#">DELEGATE USERS</a> <a href="#">CHANGE OWNERSHIP</a>
17056227	MyKPI	Entertainment	Smart_bench	Smart Bench Total Transits	totalTransits	integer	6/29/2019, 7:59:52 PM	0	private <a href="#">MAKE PUBLIC</a>	disit_comunefi	<a href="#">VIEW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>	<a href="#">VALUES</a> <a href="#">METADATA</a>	<a href="#">DELEGATE USERS</a> <a href="#">CHANGE OWNERSHIP</a>
17056226	MyKPI	Entertainment	Smart_bench	Smart Bench Pressure	Pressure	float	6/29/2019, 7:59:53 PM	0	private <a href="#">MAKE PUBLIC</a>	disit_comunefi	<a href="#">VIEW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>	<a href="#">VALUES</a> <a href="#">METADATA</a>	<a href="#">DELEGATE USERS</a> <a href="#">CHANGE OWNERSHIP</a>
17056225	MyKPI	Entertainment	Smart_bench	Smart Bench Humidity	Humidity	float	6/29/2019, 7:59:53 PM	0	private <a href="#">MAKE PUBLIC</a>	disit_comunefi	<a href="#">VIEW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>	<a href="#">VALUES</a> <a href="#">METADATA</a>	<a href="#">DELEGATE USERS</a> <a href="#">CHANGE OWNERSHIP</a>
17056224	MyKPI	Entertainment	Smart_bench	Smart Bench Temperature	Temperature	float	6/29/2019, 7:59:53 PM	0	private <a href="#">MAKE PUBLIC</a>	disit_comunefi	<a href="#">VIEW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>	<a href="#">VALUES</a> <a href="#">METADATA</a>	<a href="#">DELEGATE USERS</a> <a href="#">CHANGE OWNERSHIP</a>

Showing 1 to 9 of 9 My KPI Data First < - - 1 - - > Last

My Data, KPI, POI

[Return to My KPI Data List](#)

Values of KPIData: No. 17056225 Value Smart Bench Light

KPI Delegation Details

Username

Delegated \*

[Close](#) [SAVE](#)

No. +	Username Delegated	Insert Time	Controls
No data available in table			

Showing 1 to 0 of 0 My KPI Delegation First < - - 1 - - > Last Page Number  [Go](#)



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







# Management of MyKPI, MyPOI, ...

**Snap4City** My Data, KPI, POI

User: paolo.hel2, Org: Helsinki  
Role: AreaManager, Level: 2

10 | My | Public | Delegated | Add My KPI | Add My POI | Add My Data | Filter Table | Search

No.	High Level Type	Nature	Sub Nature	Ownership	Username	Controls	Data	Visibility
17056205	MyKPI	UtilitiesAndSupply	Agents	private	paolo.hel2	VIEW MAKE PUBLIC	VALUES METADATA	DELEGATE USERS CHANGE OWNERSHIP
17056204	MyKPI	UtilitiesAndSupply	Agents	private	paolo.hel2	VIEW MAKE PUBLIC	VALUES METADATA	DELEGATE USERS CHANGE OWNERSHIP
17056197		EducationAndResearch	CulturalEc	private	paolo.hel2	VIEW MAKE PUBLIC	VALUES METADATA	DELEGATE USERS CHANGE OWNERSHIP
17056190	MyKPI	UtilitiesAndSupply	Agents	map private	paolo.hel2	VIEW MAKE PUBLIC	VALUES METADATA	DELEGATE USERS CHANGE OWNERSHIP
17056186	MyKPI	UtilitiesAndSupply	Agents	private	paolo.hel2	VIEW MAKE PUBLIC	VALUES METADATA	DELEGATE USERS CHANGE OWNERSHIP
17056185	MyKPI	UtilitiesAndSupply	Agents	private	paolo.hel2	VIEW MAKE PUBLIC	VALUES METADATA	DELEGATE USERS CHANGE OWNERSHIP
17056173	MyKPI	UtilitiesAndSupply	Agents	private	paolo.hel2	VIEW MAKE PUBLIC	VALUES METADATA	DELEGATE USERS CHANGE OWNERSHIP
17055924	MyPOI	Accommodation	Hotel	private	paolo.hel2	VIEW MAKE PUBLIC	VALUES METADATA	DELEGATE USERS CHANGE OWNERSHIP

Showing 1 to 8 of 8 My KPI Data | First | 1 | Last | Page Number | Go

**My POI Details**

Nature \*  
Subnature \*  
Latitude \*  
Longitude \*

Value Name  
Value Type  
Data Type: integer

**Micro Applications: My Points of Interest, MyPOI**

Services: 8

**Add POI**

**Points Of Interest**

- Eccolo  
2038531 m 289 m
- My Hotel  
2039116 m 519 m
- S4chelsinkitrackerlocation  
2041652 m 3180 m

**S4chelsinkitrackerlocation**

Cancel Edit Delegate

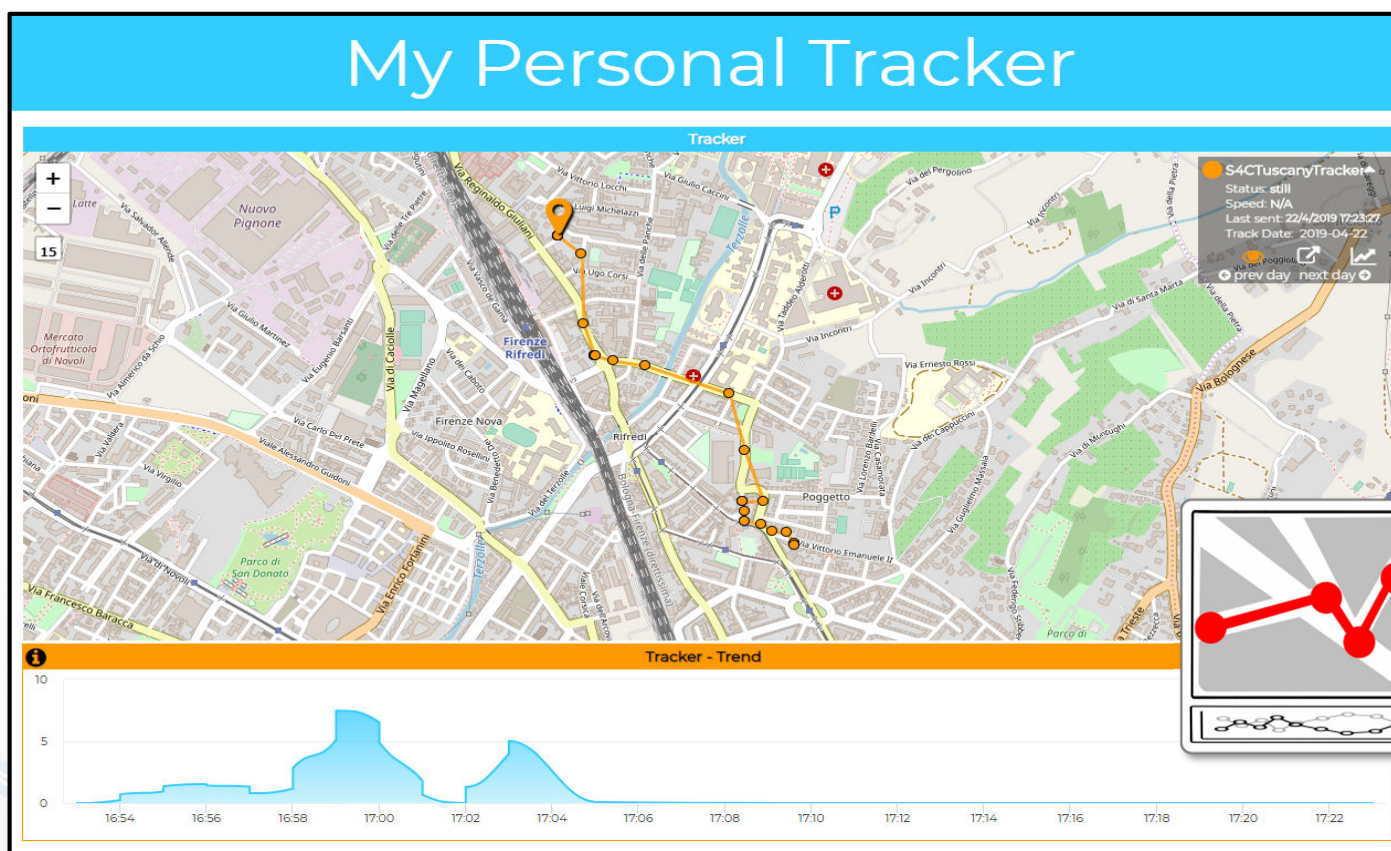
2019-05-08

Show 10 entries

Values	DataTime	Latitude	Longitude
0	08/05/2019	43.792	11.25

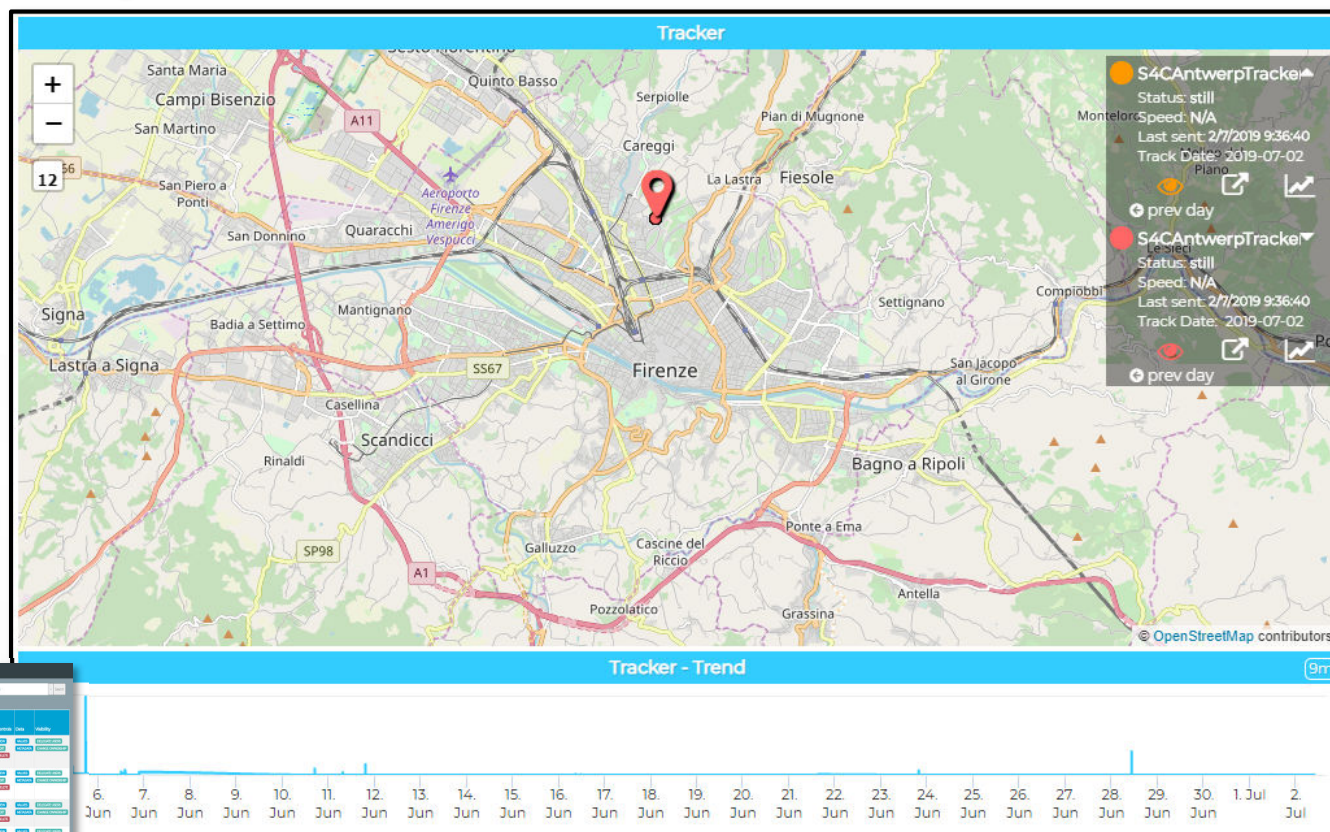
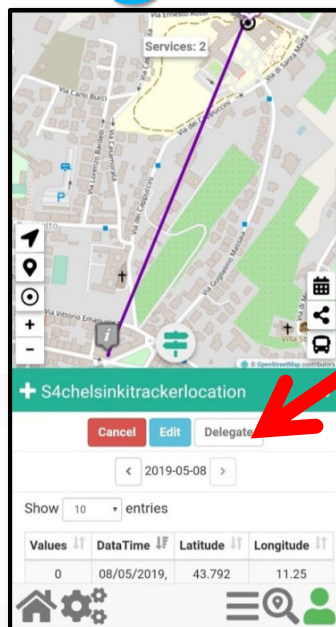


# Recalling Exercise number 3





# Accessing Multiple Trajectories



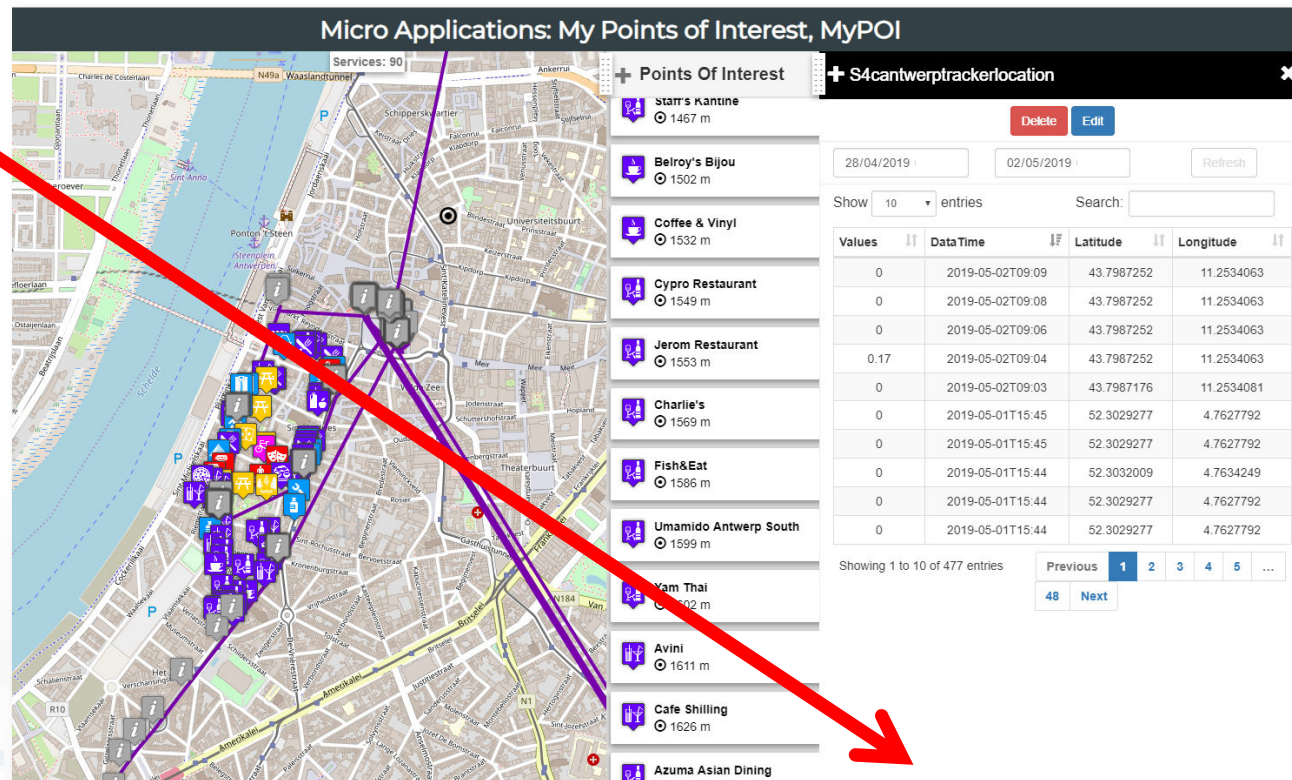
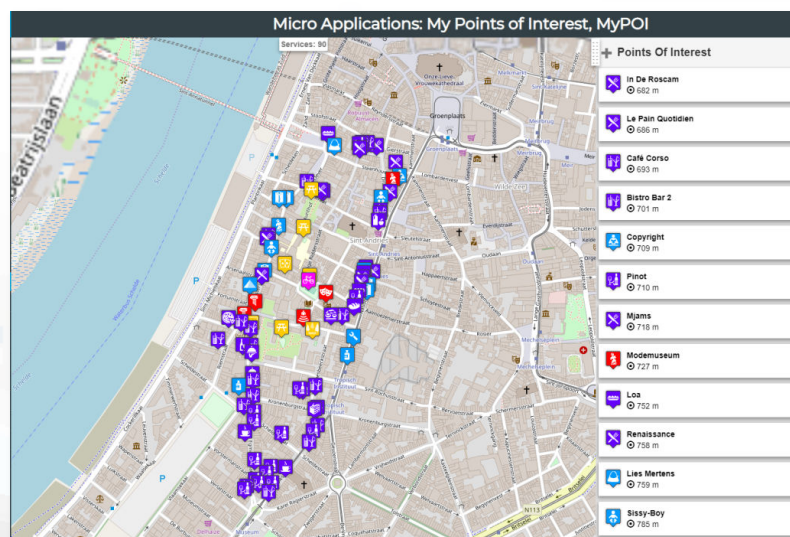
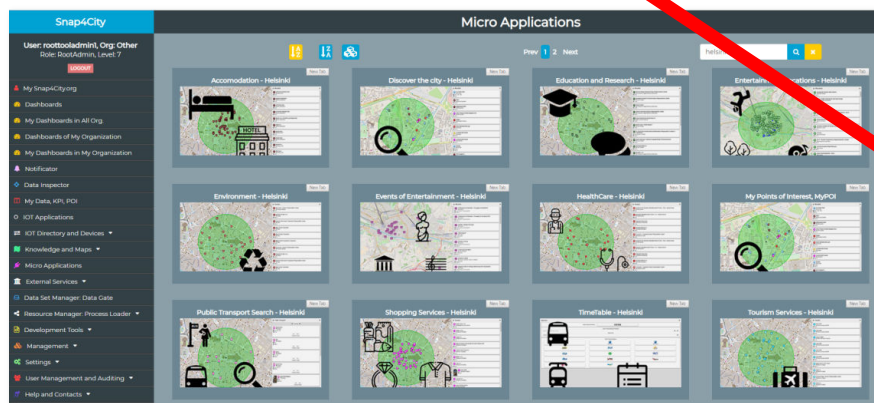
My Data, KPI, POI

ID	Name	Lat	Lon	Alt	Info	Created	Updated	Deleted	Info	Info
1	San Donato	43.792	11.25	11.25	San Donato	2019-05-08	2019-05-08	2019-05-08	San Donato	San Donato
2	San Donato	43.792	11.25	11.25	San Donato	2019-05-08	2019-05-08	2019-05-08	San Donato	San Donato
3	San Donato	43.792	11.25	11.25	San Donato	2019-05-08	2019-05-08	2019-05-08	San Donato	San Donato
4	San Donato	43.792	11.25	11.25	San Donato	2019-05-08	2019-05-08	2019-05-08	San Donato	San Donato
5	San Donato	43.792	11.25	11.25	San Donato	2019-05-08	2019-05-08	2019-05-08	San Donato	San Donato

You can  
revoke at any  
instant from  
Web and  
mobile App



# MicroApplications: several new



- Personal POI, KPI, Trajectories/trips



# IOT Applications, Devices and Dashboards

FROM CITY DASHBOARD TO APPLICATIONS

DATA GATHERING AND CITY DATA KNOWLEDGE MANAGEMENT

FORGING & MANAGING OPEN AND FLEXIBLE WEB AND MOBILE APPS

IOT APPLICATIONS VIRTUAL LOGIC DEVICES

IOT/IOE DEVICES AND NETWORKS

IOT APPLICATIONS, THE LOGIC AND THE SMARTNESS

SNAP4CITY API, MICROSERVICES, SNAP4CITY API

SNAP4CITY

SNAP4CITY FOR BEGINNERS

DATA ANALYTICS, BUSINESS INTELLIGENCE, WHAT-IF AND SIMULATION

SNAP4CITY ARCHITECTURE AND ECOSYSTEM, OPENED TO DEVELOPERS AND DATA HOLDERS

DECISION SUPPORT SYSTEM AND CITY RESILIENCE

HOW TO ADOPT SNAP4CITY, AND OUR ROADMAP

SNAP4CITY THE VIEW OF THE ADMINISTRATORS

TWITTER VIGILANCE: SOCIAL MEDIA ANALYSIS

SNAP4CITY AND KM4CITY PROJECTS

LIVING LAB



# Dashboard with intelligence App

Dashboards with IOT Applications for enforcing smart and intelligence into them

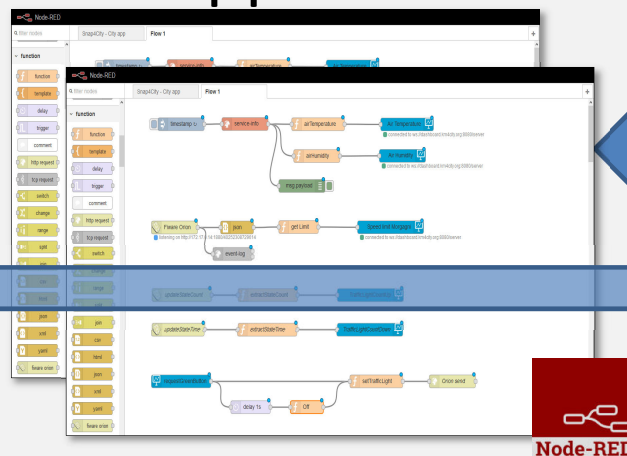
**Dashboard-IOT App**

IOT and City data World

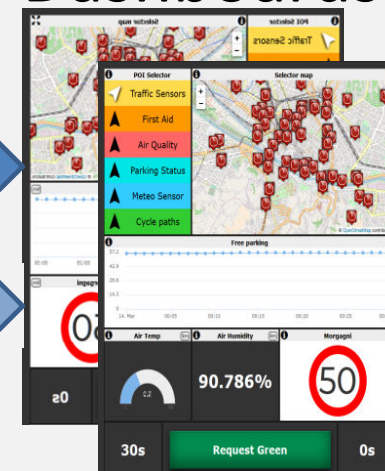


My IOT Devices

IOT Applications

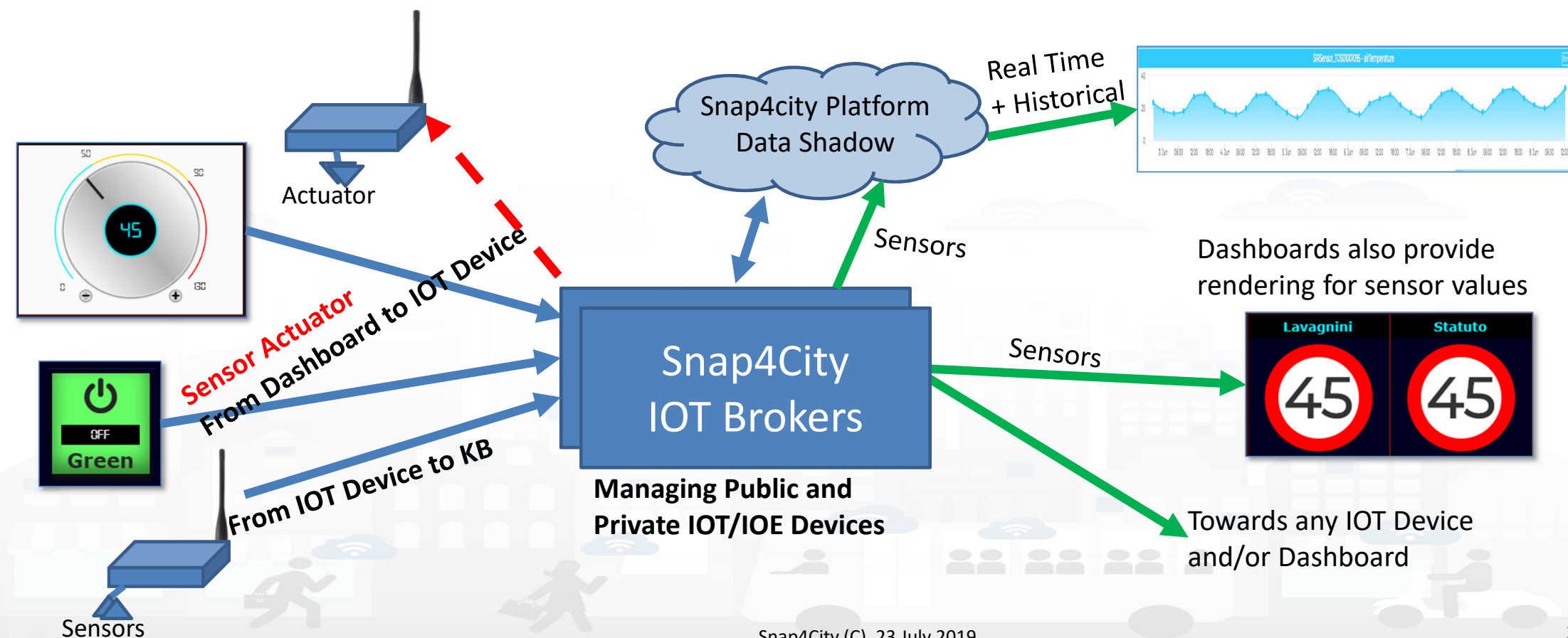


Dashboards





# IOT Data Driven

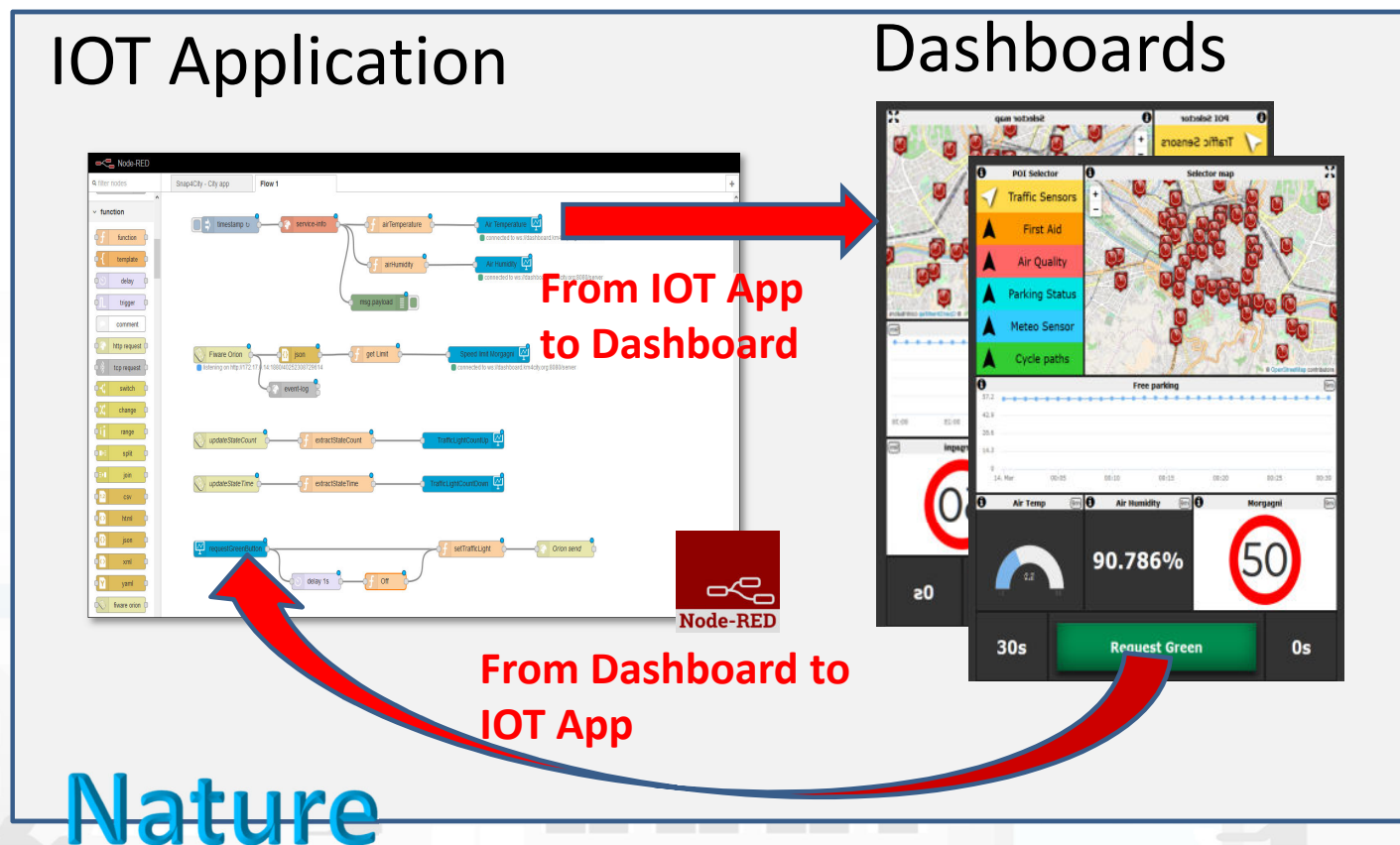




# HLT: Sensors-Actuators

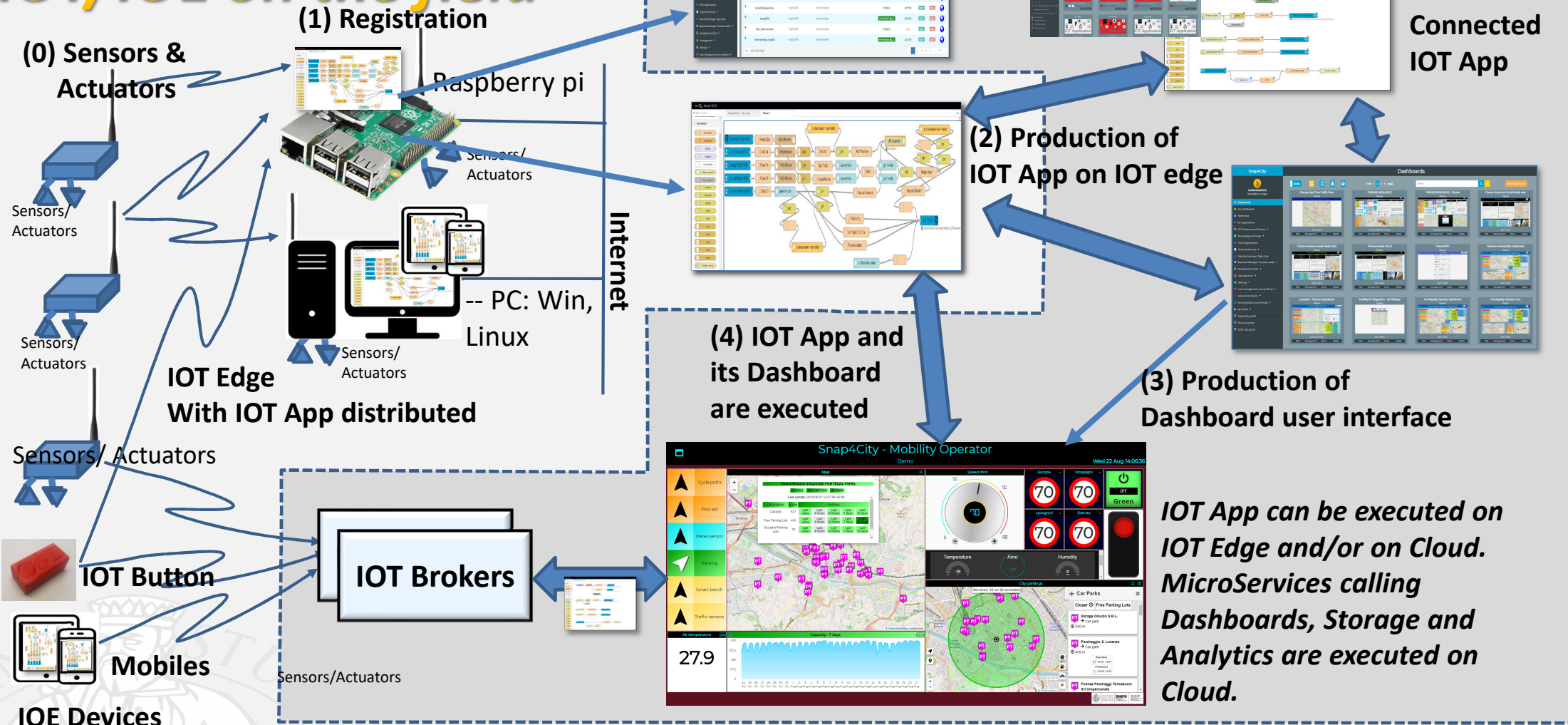
High Level Types

- 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







UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB



# *Managing IOT Applications*







UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB



# IOT Application Listing, they can be

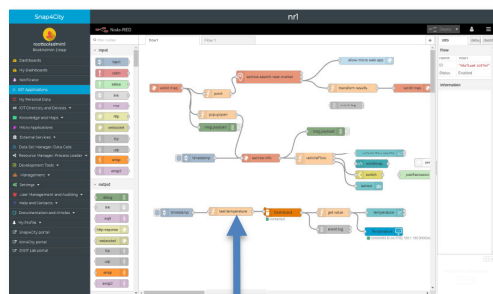
- Basic (white)
- Advanced (red)
- IOT Edge
  - Raspberry Pi
  - Android
  - Win/Linux
- Data Analytic (Plumber)
- Web Scraper (Portia)

The screenshot displays the Snap4City web interface. On the left is a dark sidebar menu with options like Dashboards, My Dashboards, Notificator, IOT Applications (highlighted), My Personal Data, IOT Directory and Devices, Knowledge and Maps, Micro Applications, External Services, Data Set Manager, Resource Manager, Development Tools, Management, Settings, User Management and Auditing, Help and Contacts, Documentation and Articles, My Profile, Snap4City portal, Km4City portal, and DISIT Lab portal. The main area is titled 'IOT Applications' and shows a grid of application tiles. Each tile includes a timestamp, a set of icons representing different IoT capabilities, the application name, the owner's name, and a 'Management' button. The tiles shown are: 'IOT Edge App' (owner: badii), 'IOT Edge App' (owner: panesi), 'IOT Edge App' (owner: pb3), 'Data Analytic' (owner: snap4city), 'IOT Edge App' (owner: semolarudy), 'IOT Application' (owner: tester5), 'IOT Application' (owner: semolarudy), 'IOT Application' (owner: comunedashres), 'Deprecated - SilMobilityControlRoom' (owner: badii), 'SamsungGalaxyS4Barcode' (owner: badii), 'IOT Application' (owner: tester2), and 'Web Scraper Portia' (owner: My own).

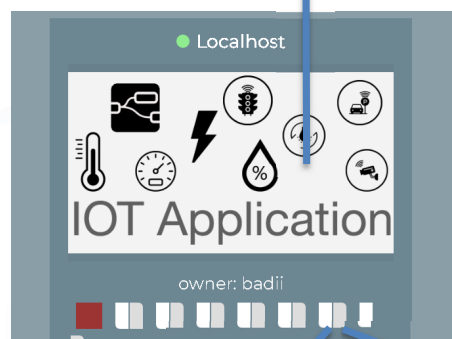


# IOT Applications Listing

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

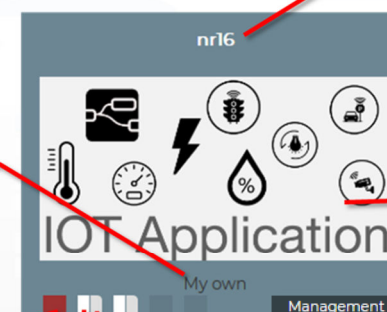
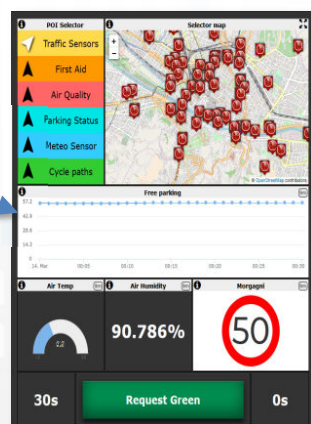
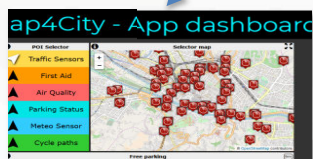


EDIT IOT APP



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

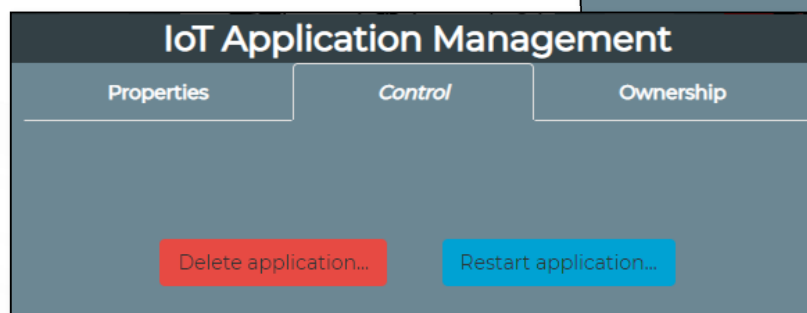
Click to view the Snap4City Dashboard

Click to edit the Snap4City Dashboard

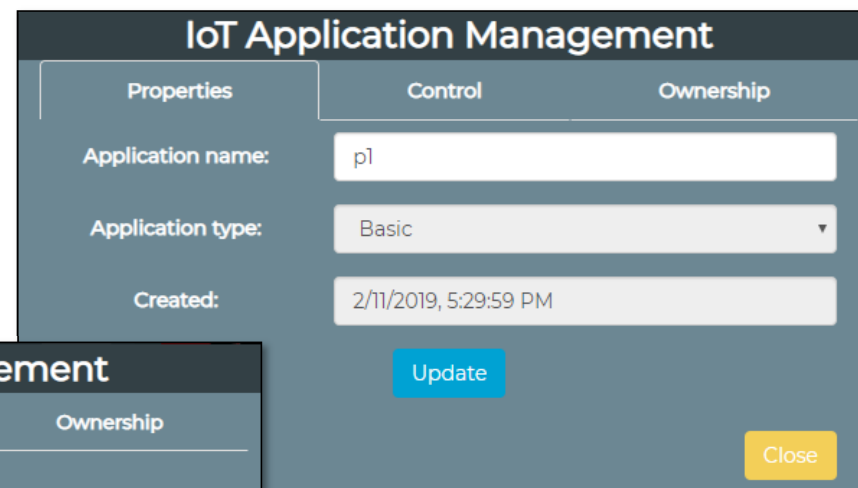


# IOT Application Management

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



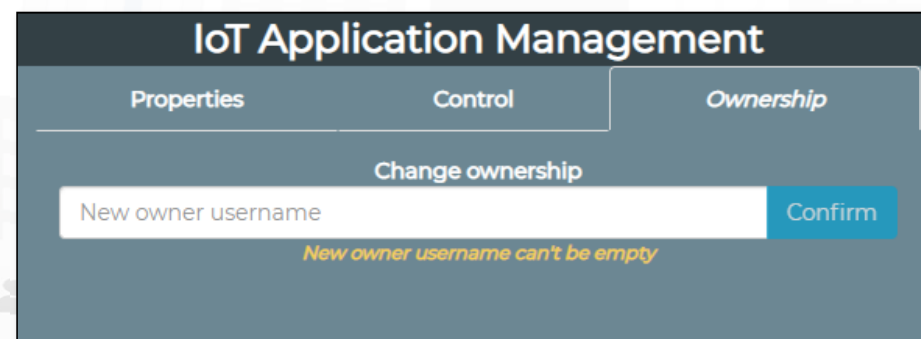
The screenshot shows the 'Control' tab of the IoT Application Management interface. It features two buttons at the bottom: a red 'Delete application...' button and a blue 'Restart application...' button. The 'Properties' and 'Ownership' tabs are visible in the background.



The screenshot shows the 'Properties' tab of the IoT Application Management interface. It displays the following information:

- Application name: p1
- Application type: Basic
- Created: 2/11/2019, 5:29:59 PM

There are 'Update' and 'Close' buttons at the bottom right.



The screenshot shows the 'Ownership' tab of the IoT Application Management interface. It displays the 'Change ownership' section with a text input field for 'New owner username' and a 'Confirm' button. A yellow warning message below the input field reads: 'New owner username can't be empty'.





UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB



# *Authoring IoT Applications*



Snap4City (C), 23 July 2019

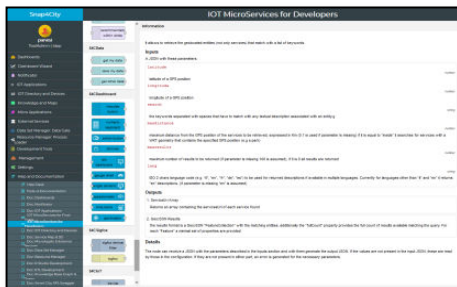




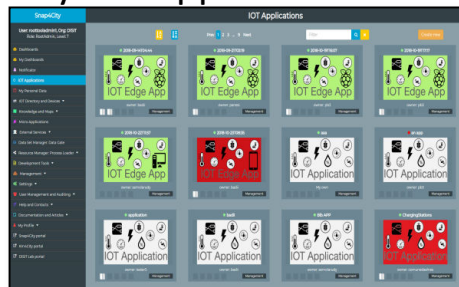
# IOT Applications Development

IOT Discovering

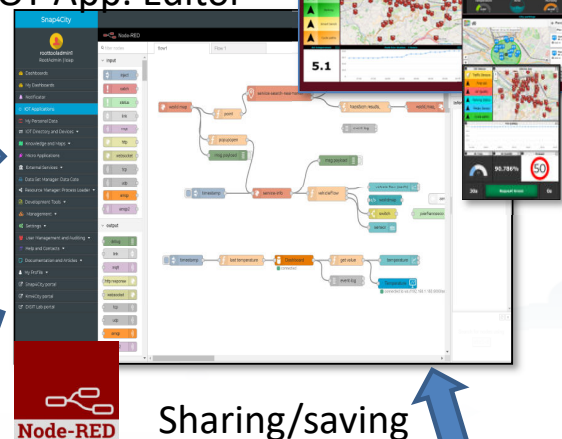
MicroServices collections



My IOT Applications



IOT App. Editor



Generating IOT App  
With Dashboard



Sharing/saving  
reusing IOT App



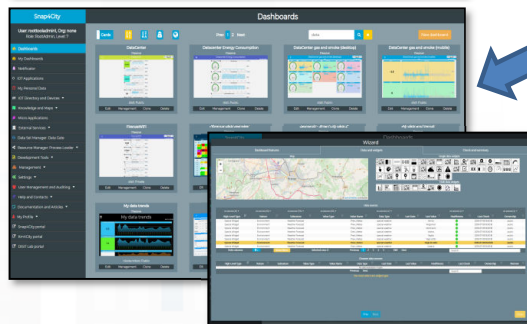
Resource Manager



ServiceMap Discovery  
Knowledge Base, Km4City



Dashboard Collection,  
Editor and Wizard





roottooladmin1  
RootAdmin | Idap

- Dashboards
- My Dashboards
- Notificator
- IOT Applications**
- My Personal Data
- IOT Directory and Devices
- Knowledge and Maps
- Micro Applications
- External Services
- Data Set Manager: Data Gate
- Resource Manager: Process Loader
- Development Tools
- Management
- Settings
- User Management and Auditing
- Help and Contacts
- Documentation and Articles
- My Profile
- Snap4City portal
- Km4City portal
- DISIT Lab portal

Node-RED

Deploy



filter nodes

flow1

Flow 1

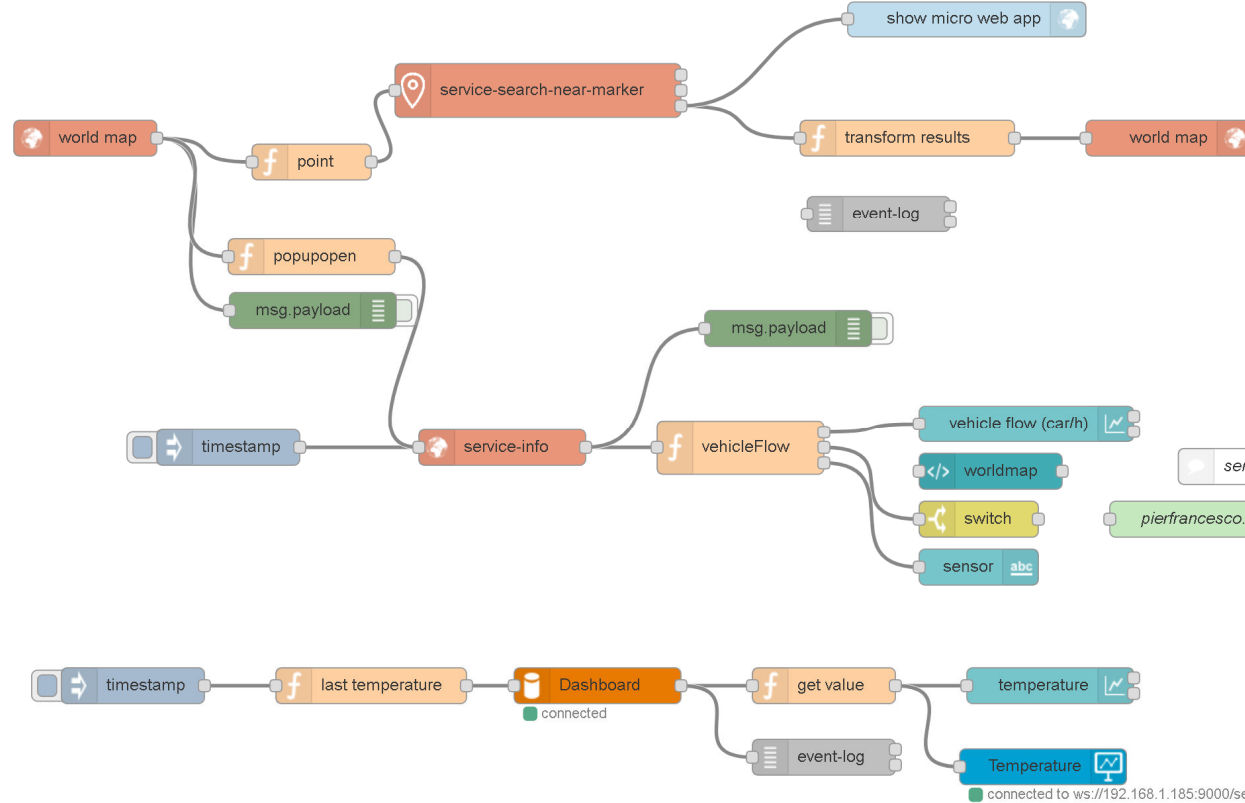


## input

- inject
- catch
- status
- link
- mqtt
- http
- websocket
- tcp
- udp
- amqp
- amqp2

## output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp
- amqp
- amqp2



Node-RED

info

debug

Flow

Name	flow1
ID	"49a71aa0_b297b4"
Status	Enabled

Information

Search for nodes using

ctrl-f





# 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

**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

- Clipboard
- Library
- Import S4C**
- Examples

- Deploy
- View
- Import
- Export
- Search flows
- Configuration nodes
- Flows
- Subflows
- Manage palette
- Settings
- Keyboard shortcuts
- Node-RED website

msg.payload

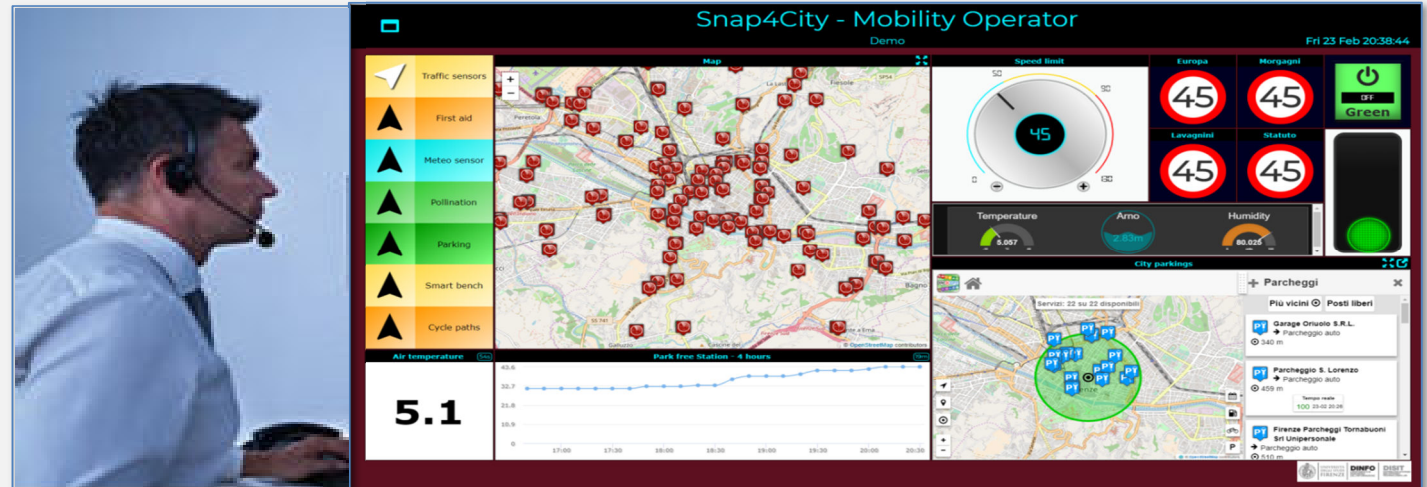




## Control Room Operator

Would like to:

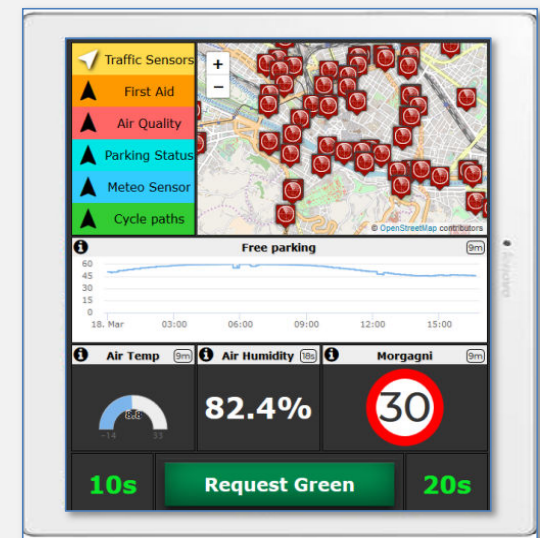
- **Monitor** traffic flow, Environment, Car parking, Cycling, First aid, temp., ..
- **Act and** monitor Dynamic Plates
- **Act and** monitor red lights



## 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

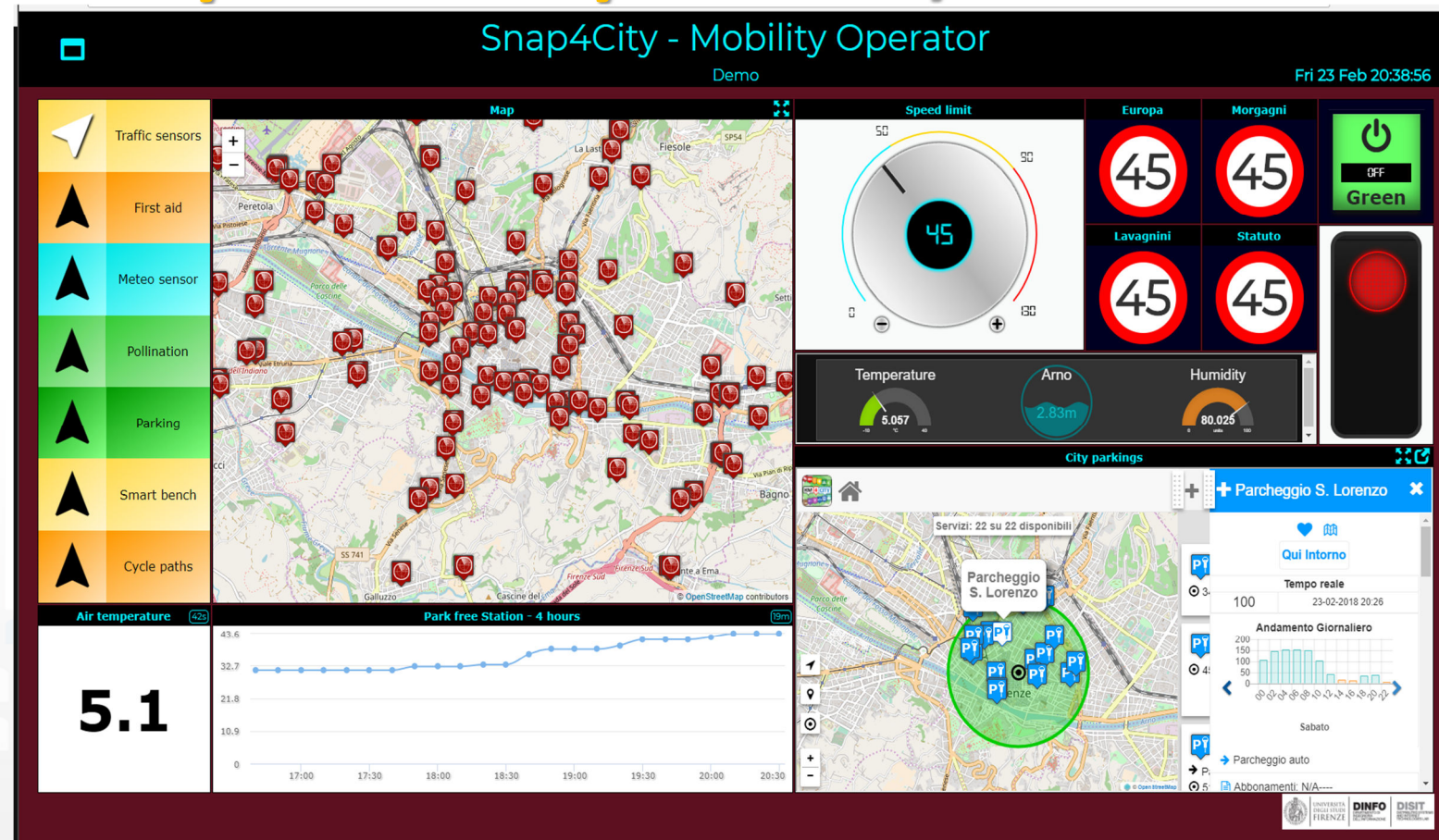
## Sensors:

- Values
- Status

## Actuators:

- Buttons
- Dimers
- Etc.

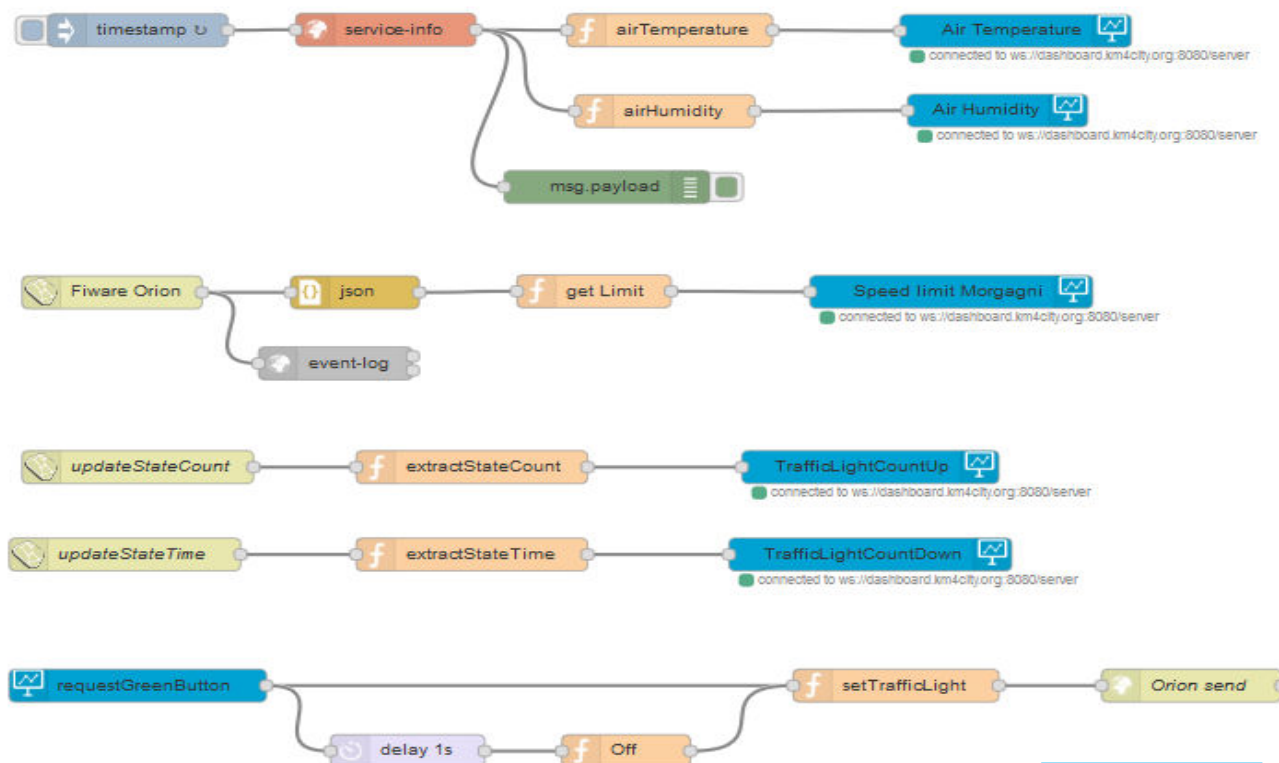
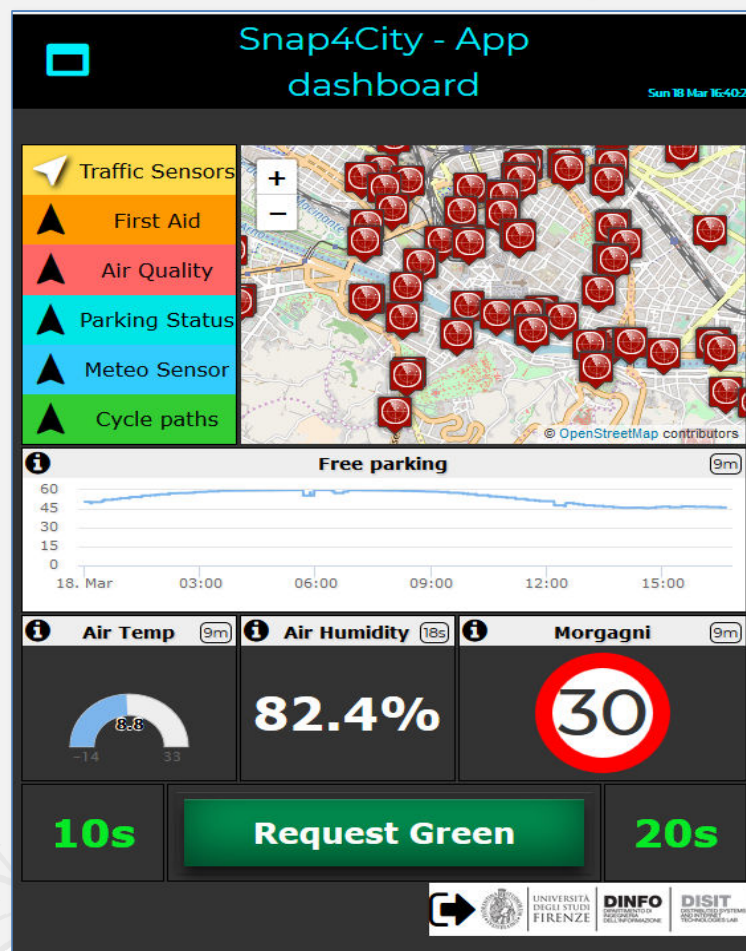
## Virtual Sensors and Actuators



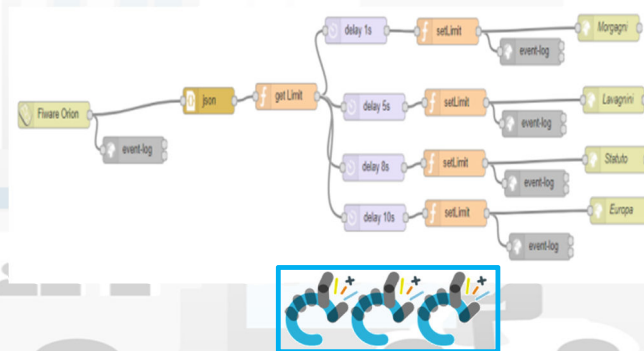
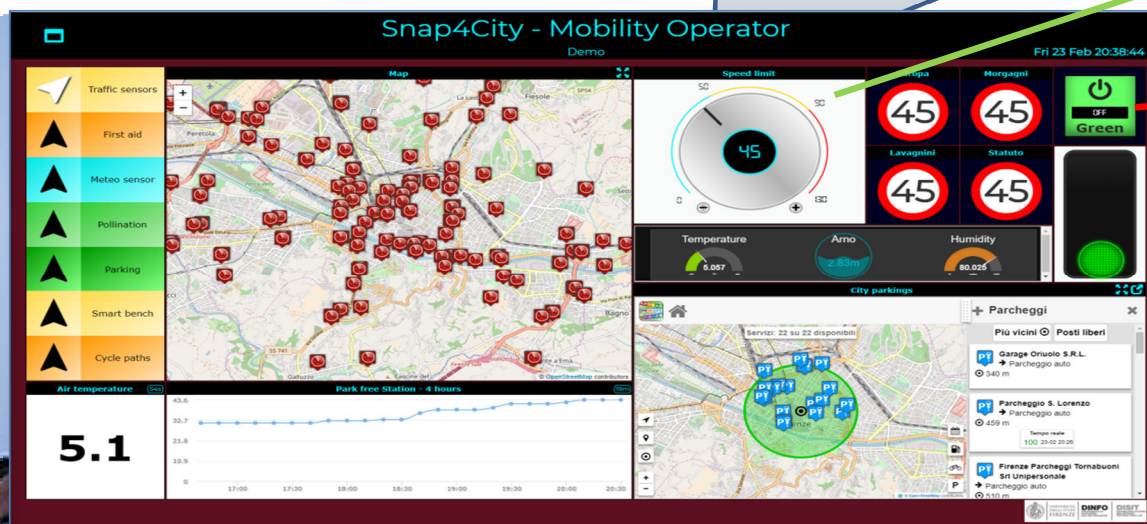
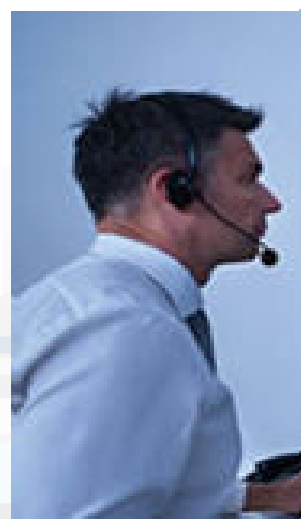
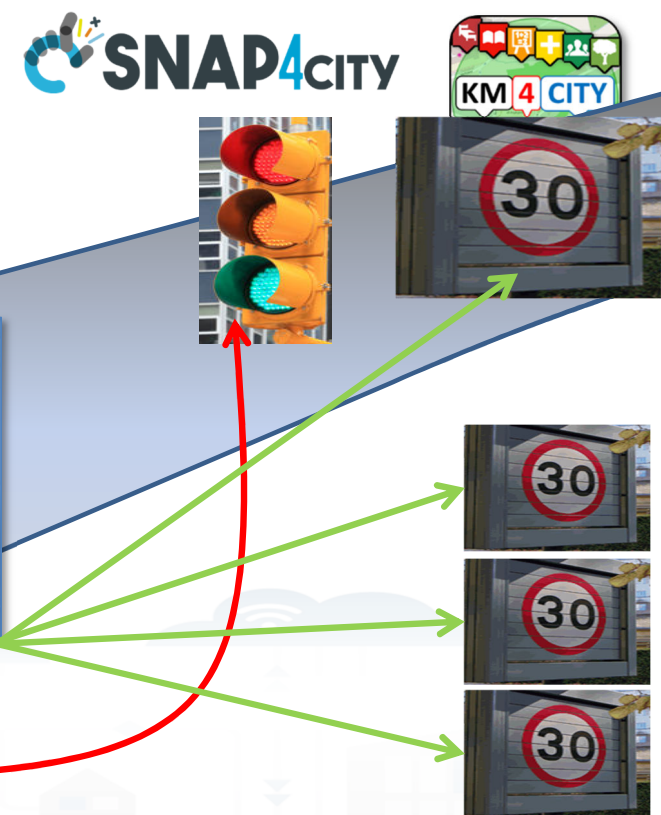


# IOT Application with City Dashboard

## simple development











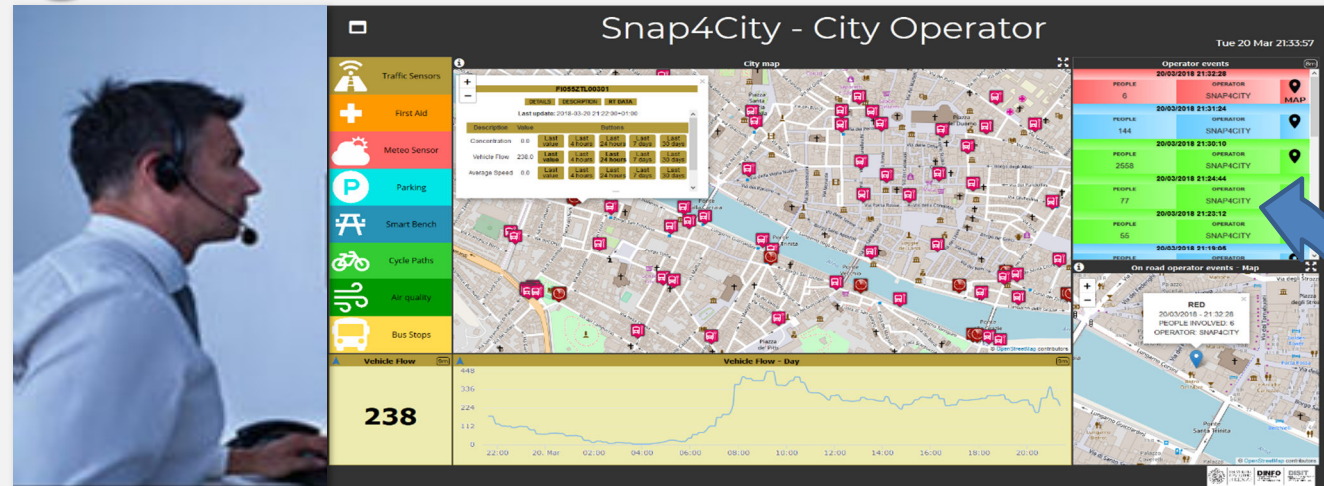
# Reporting Critical Events



## 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:

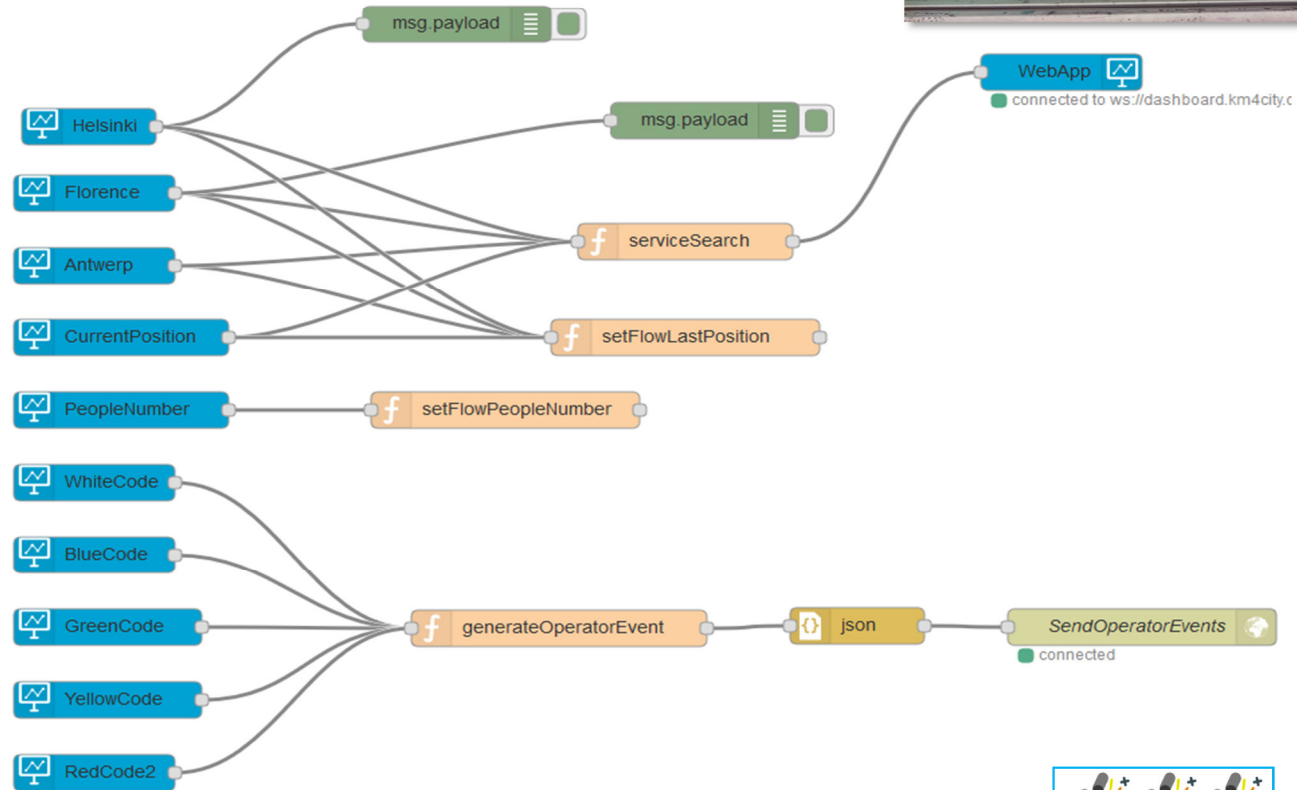
- Monitor data of traffic, Parking, environment, speed limit, services,
- **Send critical event notifications via coded description**



Snap4City (C), 23 July 2019

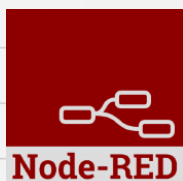


# IOT Application with City Dashboard simple development



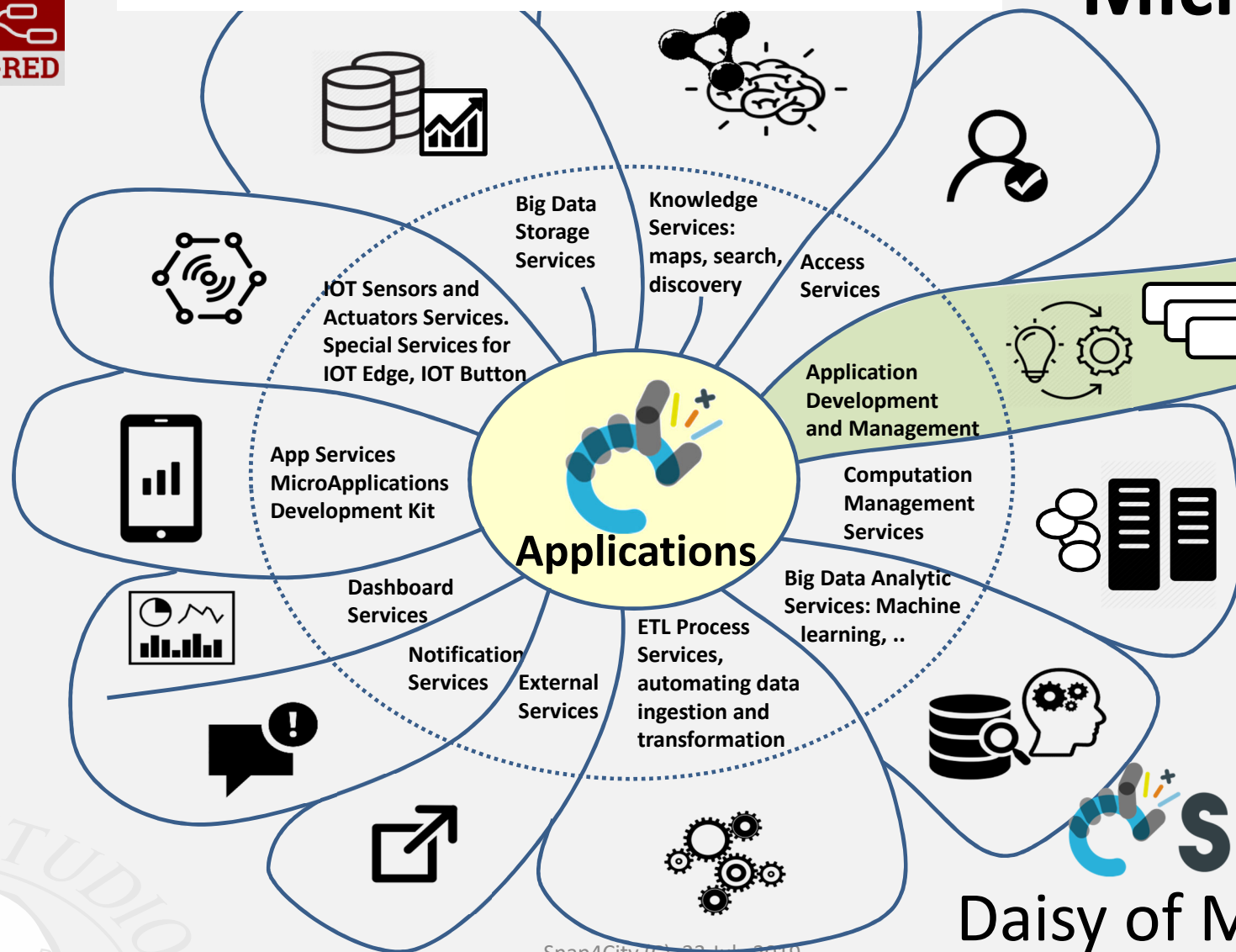


- > input
- > output
- > function
- > social
- > storage
- > analysis
- > advanced
- > lwm2m
- > **S4C SearchDev**
- > S4CMapping
- > S4CManagement
- > S4CDataAnalytic
- > S4CBigData
- > S4C Search
- > S4CData
- > S4CDashboard
- > S4C Sigfox
- > S4CIoT
- > S4CLogDev
- > S4CView
- > S4C Social
- > location
- > dashboard



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

# MicroServices Areas



**SNAP4CITY**  
Daisy of MicroServices

Snap4City (C), 23 July 2019



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

DISIT Lab, Distributed Data Intelligence and  
Information Engineering  
http://www.d...



The screenshot displays the Node-RED block palette with the following categories and blocks:

- input**: inject, catch, status, link, mqtt, http, websocket, tcp, udp, amqp, amqp2, stomp.
- output**: debug, link, mqtt, http response, websocket, tcp, udp, amqp, amqp2, stomp.
- function**: function, template, delay, trigger, comment, http request, tcp request, switch, change, range, split, join, csv, html, json, xml, yaml, soap request, base64, msgpack, random, rbe.
- social**: e mail, twitter, e mail, twitter.
- storage**: tail, file, ftp, mysql, file.
- analysis**: sentiment.
- advanced**: watch, feedparse, sunrise, exec.
- location**: turf, worldmap, worldmap, tracks.
- lwm2m**: lwm2m client, lwm2m client.
- dashboard**: button, dropdown, switch, slider, numeric, text input, date picker, colour picker, form, text, gauge, chart, audio out, notification, ui control, template.

+ on IOT Edge Raspberry

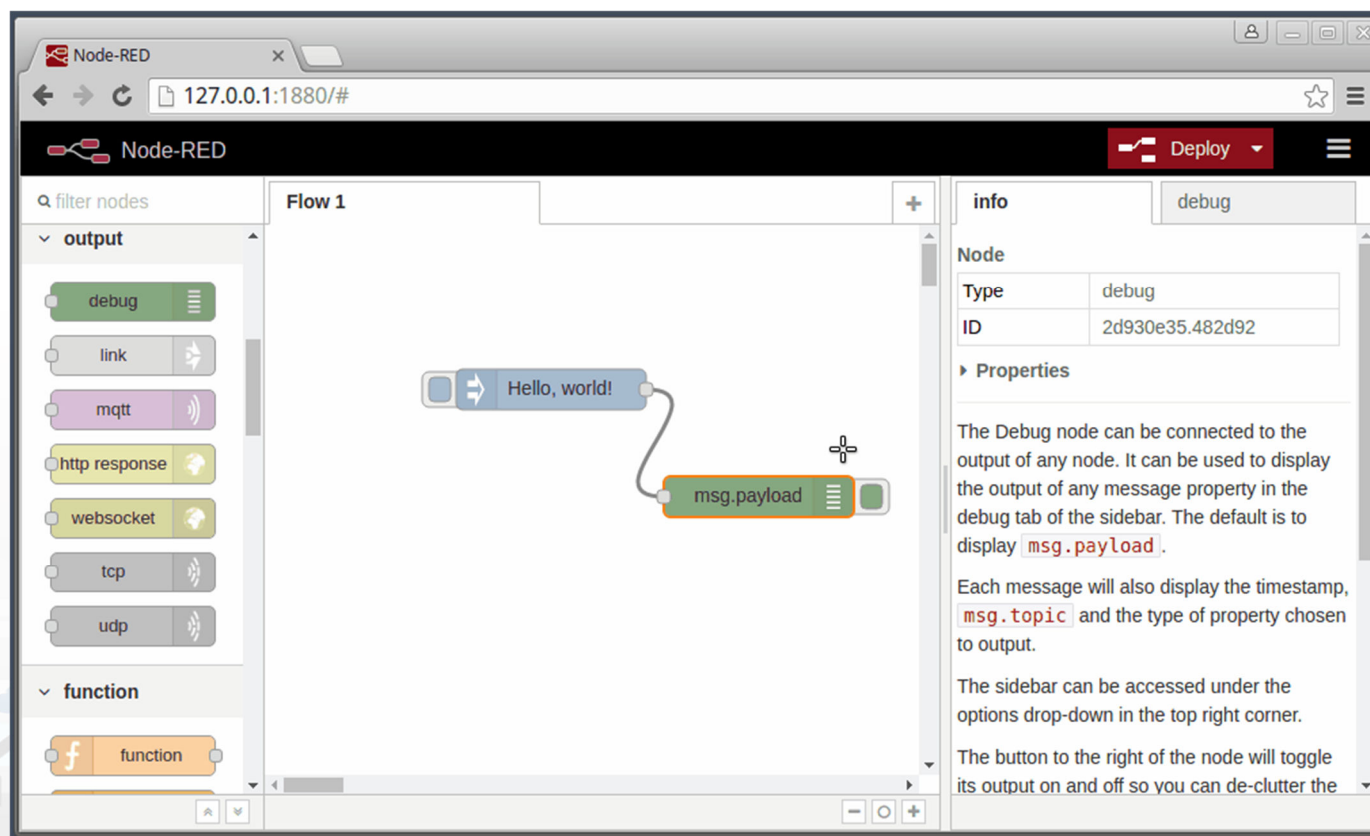
The screenshot displays the Node-RED block palette with the following categories and blocks:

- social**: e mail, twitter, irc, e mail, twitter, irc, google plus, google places, google calendar.
- storage**: tail, file, mongodb, file, mongodb.
- Raspberry Pi**: rpi gpio, rpi gpio, rpi mouse, rpi keyboard, camerapi takephoto, rpi dht22, imagecapture, ledborg, Sense HAT, Sense HAT.
- network**: ping.

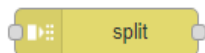
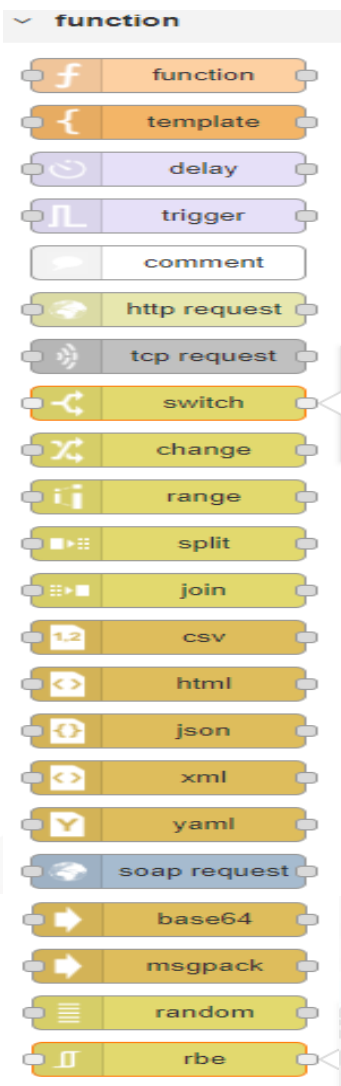


# Hello World of Node-RED

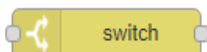
- <http://developer.opto22.com/nodered/general/getting-started/node-red-hello-world/>



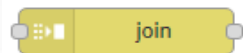




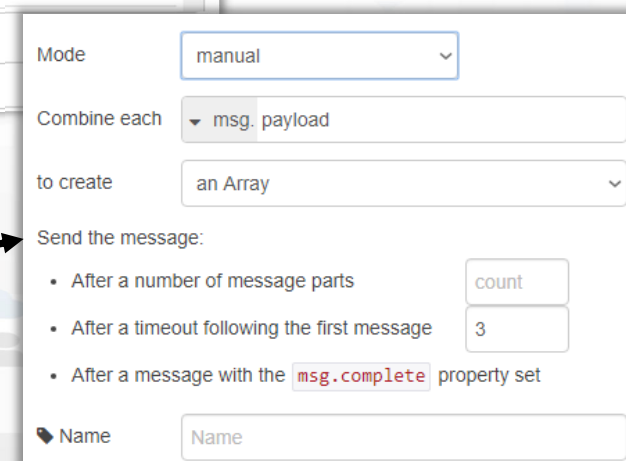
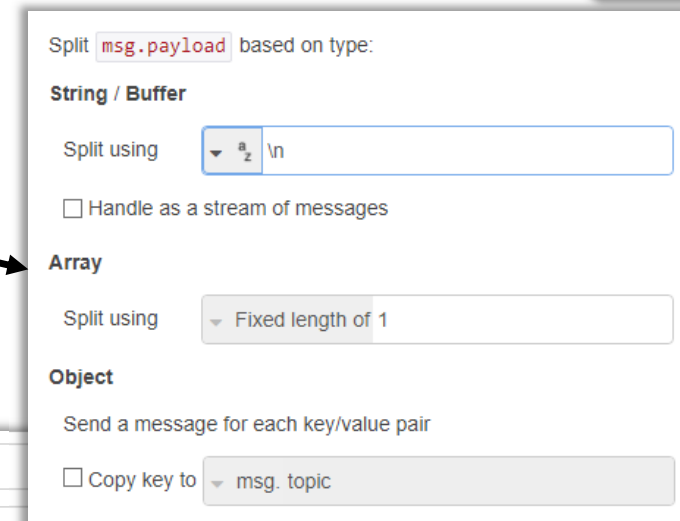
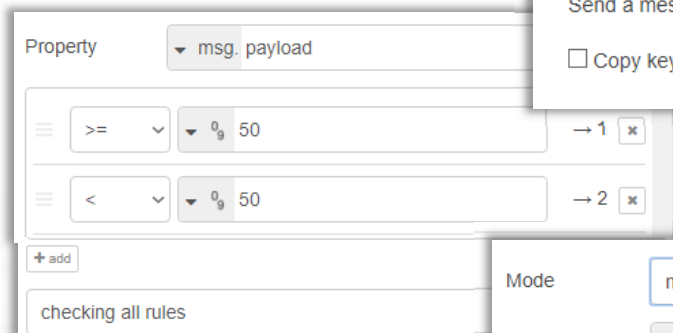
Divides the input message into multiple messages as indicated in the configuration. If you have an array at the input, you can configure it to send each element of the array individually at the output.



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
- S4C Mapping
- S4C Management
- S4C Data Analytic
- S4C Big Data
- S4C Search
- S4C Data
- S4C Dashboard
- S4C Sigfox
- S4C IoT
- S4C LogDev
- S4C View
- S4C Social
- location
- dashboard

The screenshot displays the Node-RED web interface with a large 'SNAP4CITY' watermark and a URL overlay: <https://flows.nodered.org/?term=snap4city>. The interface is organized into several panels:

- S4C SearchDev**: Contains modules for service search, search near GPS position, search near service, search within GPS area, search within WKT area, search within stored WKT area, search by municipality, 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, and address poi search by text exp.
- S4C Management**: Includes 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, trigger job, pause jobs, resume all, resume job, resume jobs, resume trigger, resume triggers, notifiator last events, notifiator history events, and event log.
- S4C Data Analytic**: Features descriptive statistics, trend plot, time series predictions, machine learning predictions, anomaly detection, and number data analytic.
- S4C Mapping**: Contains datagate search, datagate create, and portia crawler.
- S4C Search**: Offers 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, full text search use, 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, and bus routes search within polygon.
- S4C Dashboard**: Includes impulse button, numeric keyboard, switch button, dimmer, gauge chart, single content, speedometer, time trend, geolocator, Bar content, Column content, and web content.
- S4C View**: Contains show micro web app and show general iframe.
- S4C Social**: Features twitter last channel and twitter last tweet.
- S4C KPIData**: Includes get my kpdata, get my kpdata values, get public kpdata values, get delegated kpdata values, and save my kpdata values.
- S4C Sigfox**: Contains sigfox device filter and sigfox.
- S4C IoT**: Includes 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, and snap4all button.



BASIC

ADV  
2019



**S4C Search**

service search near marker	event search near marker	bus routes search within polygon
service search within circle	event search within circle	tpl agencies
service search within polygon	event search within polygon	tpl lines
service search along path	event search along path	tpl routes by agency
distance from coordinates	event search usr	tpl routes by line
service info	address search near marker	tpl stops by route
full text search near marker	geometry search near marker	tpl stop timeline
full text search within circle	address poi search by text usr	recommendatic within circle
full text search within polygon	address poi search by text near marker	value type search near marker
full text search along path	address poi search by text within circle	value type search within circle
full text search usr	bus routes search near marker	value type search within polygon
	bus routes search within circle	value type search along path

## • 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

service info

distance from coordinates



## S4CSearchDev

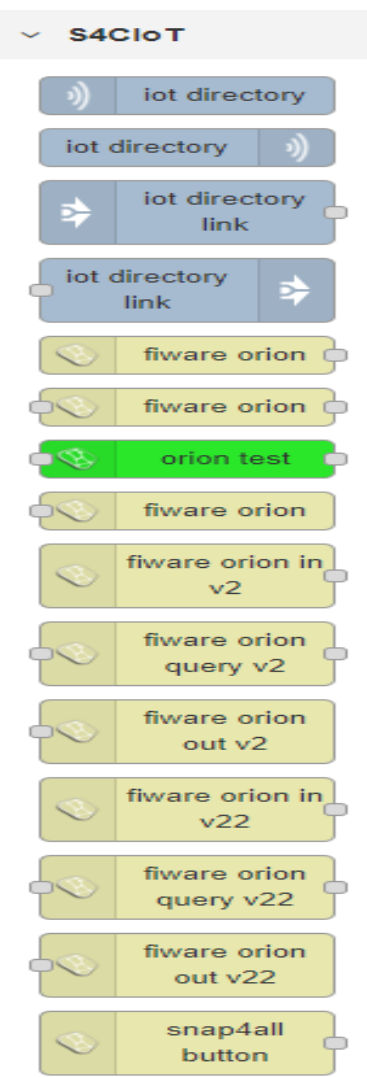


- Similar to basic Search functions but with more flexibility of the function for programming the search
- Adding Dynamic behavior:
  - Getting in input JSON with parameters
- **To Get DATA of a Service / POI /sensor**
  - Historical and real time
  - ANY kind of sensors



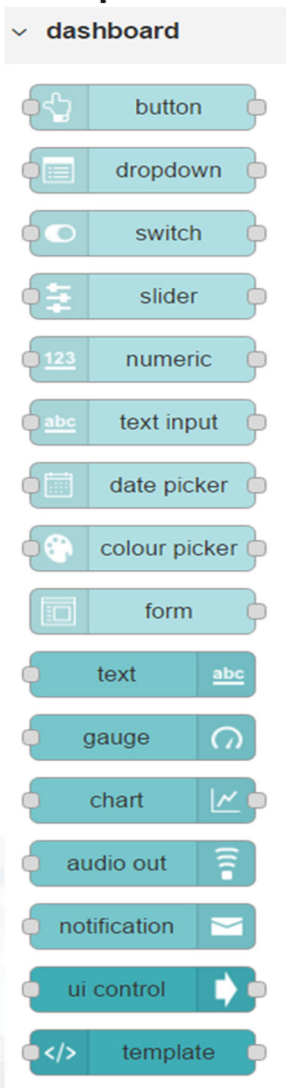


- Search for IOT Devices in a given area, or for kind (temperature, model, location, producer, Broker, ...)
- Subscribe to one or more IOT Devices independently on their protocol, broker, owner, etc.
- Send data to IOT devices
- Establish with IOT Devices Secure certified Connections
- Please note that many other protocols can be also added, adding mode nodes, or registering IOT brokers to the Snap4City IOT Directory





# Dashboard



## Native Local

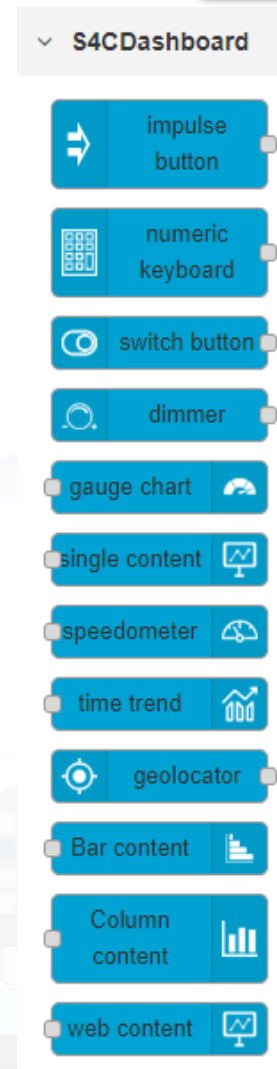
- Input/output
- non secure
- Limited in graphics
- No authentication
- No HLT
- No integration
- Etc..

- Local on IOT Edge

## or Snap4City

- 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

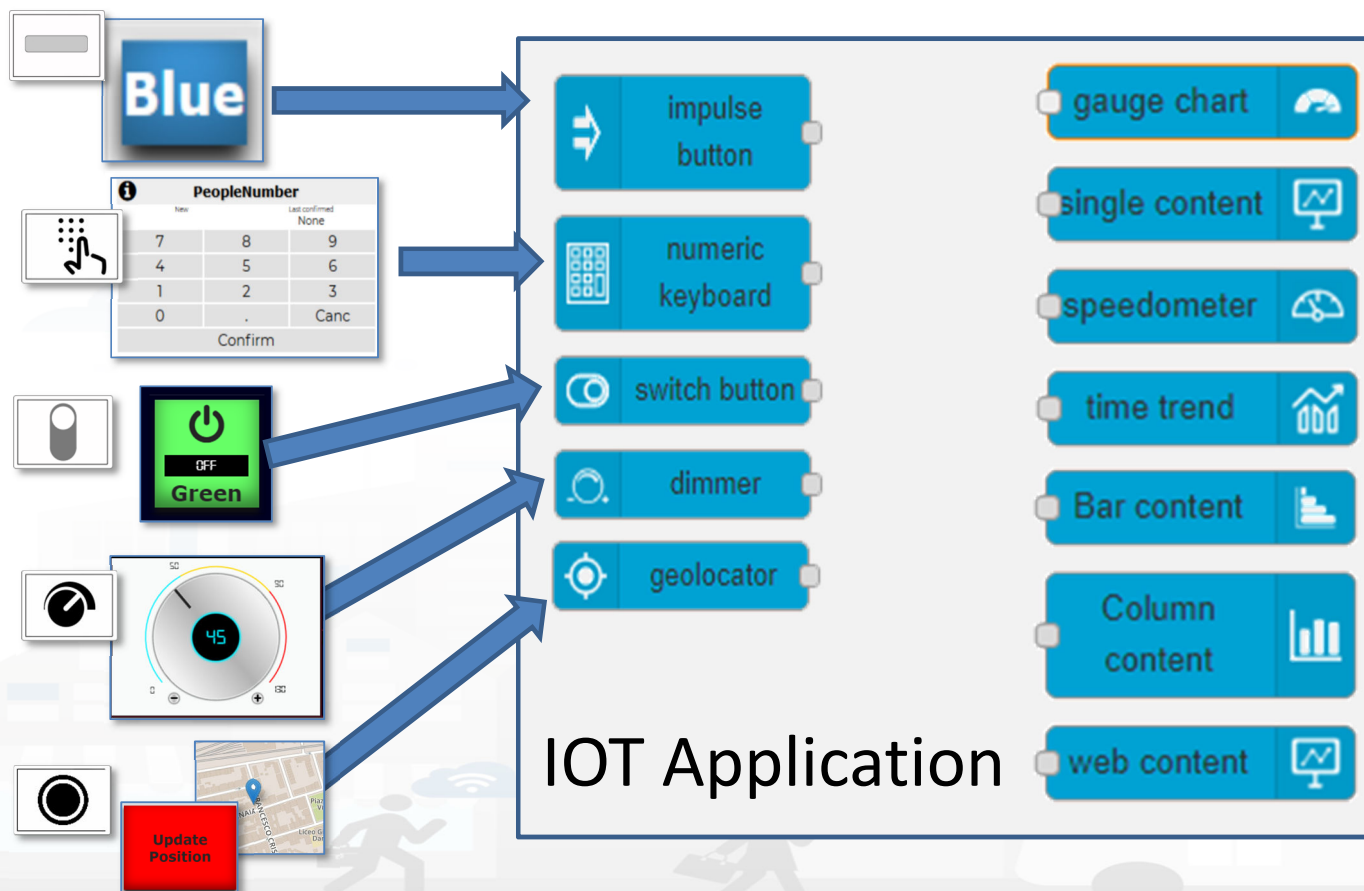
**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB

# Dashboard-IOT App

## Nature

From Dashboard to IOT App

From IOT App to Dashboard



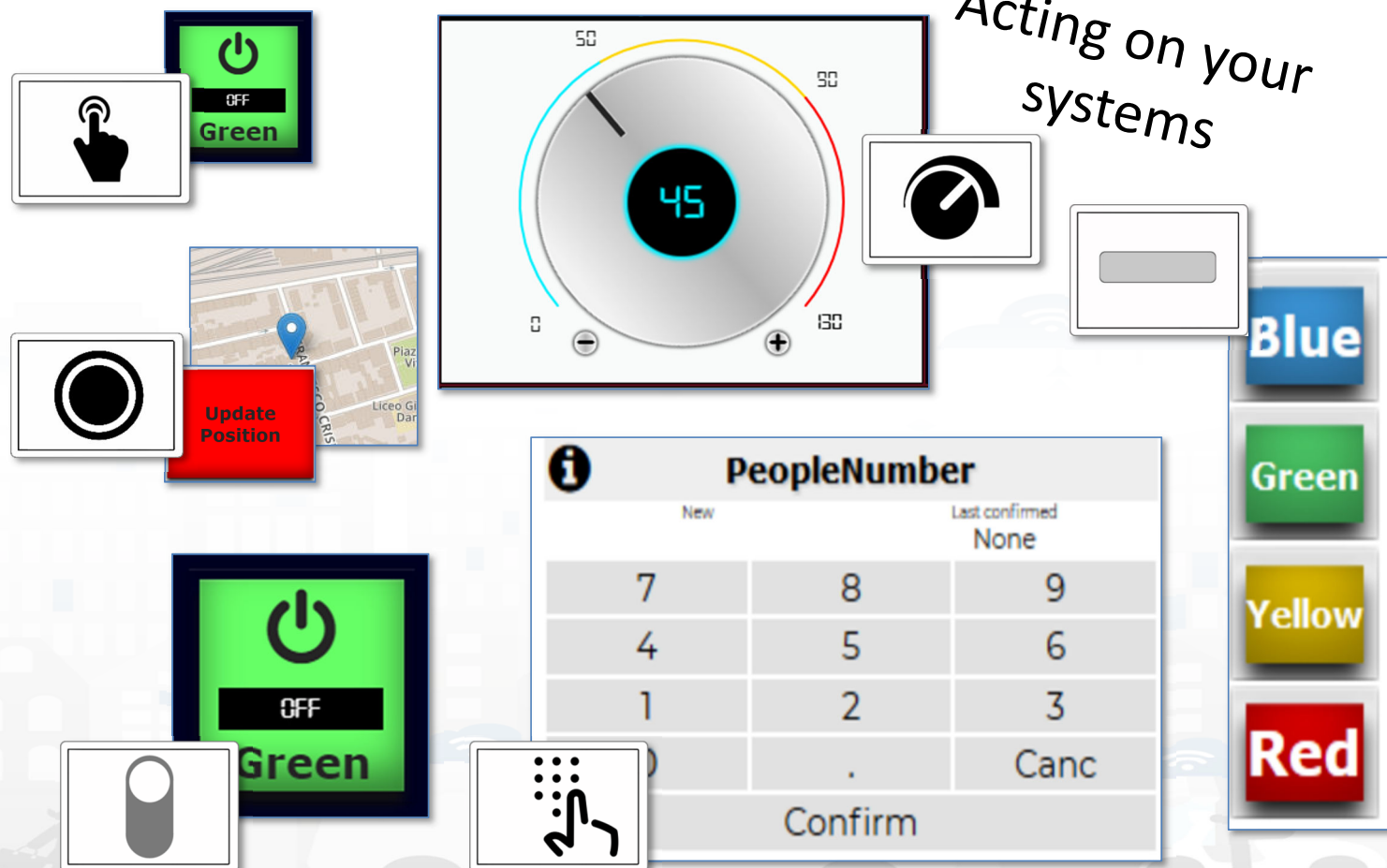


# From Dashboard to IOT Devices

- **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:

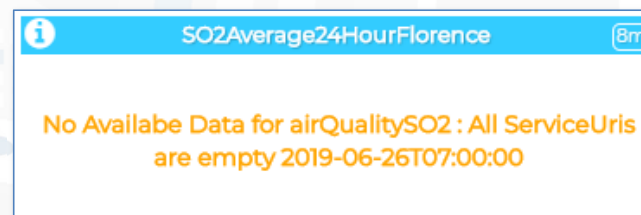
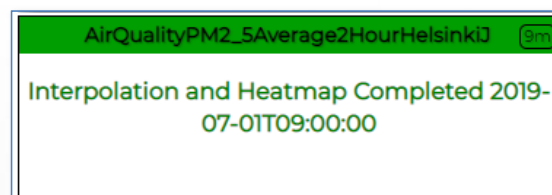
- Numbers
- String
- HTML code

XX

single content



11440 Utenti WiFi



246 TOT. EVENTI SULLA RETE

COLONNINE RICARICA (8m)

176 INSTALLATE

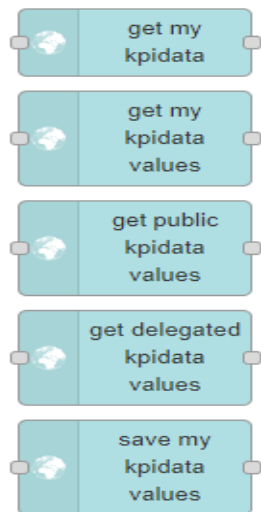
Messages (9m)

Position Updated,  
press Show My  
Position





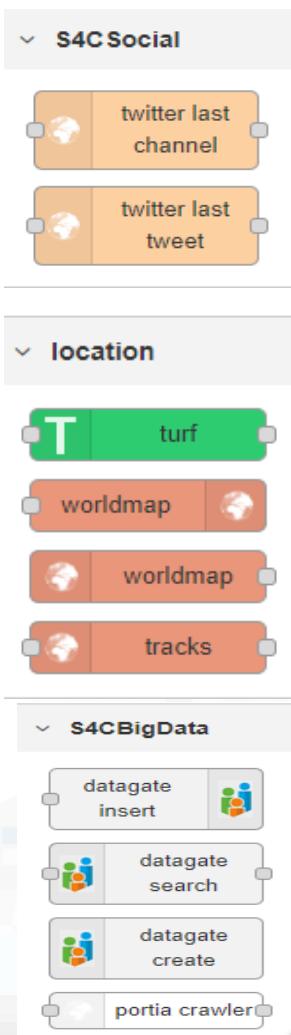
## ✓ S4CKPIData



- 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 change 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



- 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



The screenshot shows the Snap4City interface with three main sections:

- S4C Social:** Contains two orange blocks labeled "twitter last channel" and "twitter last tweet".
- location:** Contains four blocks: a green "T turf" block, and three orange "worldmap" and "tracks" blocks.
- S4C BigData:** Contains four blocks: "datagate insert", "datagate search", "datagate create", and "portia crawler".



# 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](#)



# From Simple to Data processing IOT Applications

FROM CITY DASHBOARD TO APPLICATIONS

DATA GATHERING AND CITY DATA KNOWLEDGE MANAGEMENT

FORGING & MANAGING OPEN AND FLEXIBLE WEB AND MOBILE APPS

IOT/IOE DEVICES AND NETWORKS

IOT APPLICATIONS, THE LOGIC AND THE SMARTNESS

ADVANCED SMART CITY API, MICROSERVICES, SNAP4CITY API

SNAP4CITY LIVING LAB FOR COLLABORATIVE WORK

SNAP4CITY FOR BEGINNERS

DATA ANALYTICS, BUSINESS INTELLIGENCE, WHAT IF AND SIMULATION

SNAP4CITY ARCHITECTURE AND ECOSYSTEM, OPENED TO ALL STAKEHOLDERS

TWITTER VIGILANCE, SOCIAL MEDIA ANALYSIS

HOW TO ADOPT SNAP4CITY, AND OUR ROADMAP

SNAP4CITY AND KM4CITY PROJECTS

SNAP4CITY THE VIEW OF THE ADMINISTRATORS



# 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

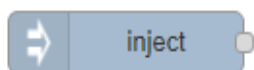
In this demo let's create an IOT Application that:

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

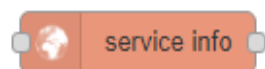




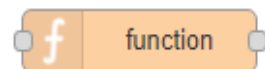
# Nodes for flow



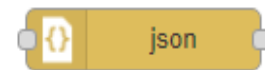
Generates an input for the other nodes. It can be repeated at predefined intervals, entered manually and of various types (string, number, Boolean, json etc.)



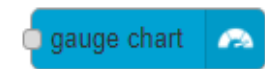
Requests detailed information for a specific service on the platform (such as a car park, hotel, etc.)



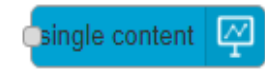
Executes a Javascript code once the input message is received



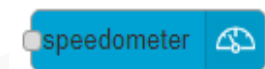
Transforms the incoming message into a JSON



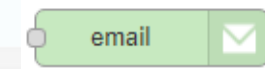
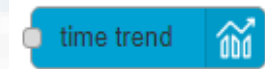
Display values in different modalities on a Dashboard (or on different Dash)



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

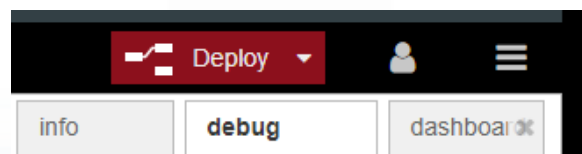


- Inject and Debug



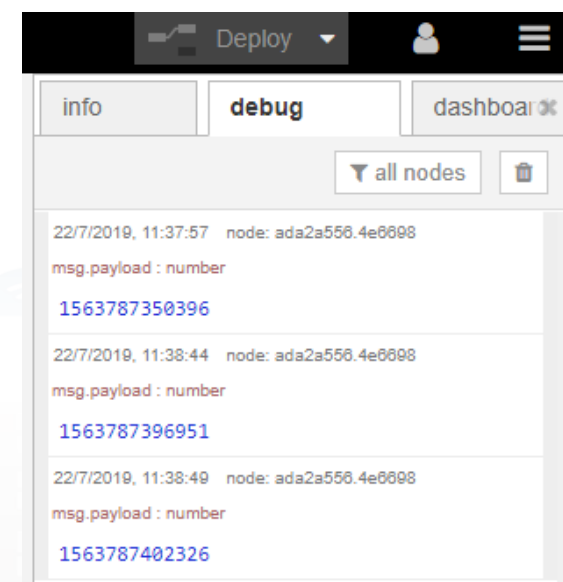
- Connect

- Deploy



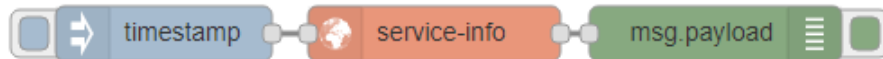
- Click and Observe

- Play with results





## Step 2



- Service Info

- Connect

- Configure

Name

ServiceUri

Language

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

- Deploy

- Click and Observe

- Play with results

```

22/7/2019, 11:54:10 node: ada2a556.4e0698
msg.payload : Object
  object
    Service: object
      predictions: array[0]
    realtime: object
      head: object
      results: object
        bindings: array[1]
          0: object
            capacity: object
            freeParkingLots: object
              value: "77"
            measuredTime: object
            occupancy: object
            occupiedParkingLots: object
            status: object
            updating: object
            trends: array[84]
  
```

Copy the path

Copy the value





UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB

## Step 3

 **SNAP4CITY**



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



Name

Function

```
1 msg.payload = msg.payload.realtime.results.bindings[0].freeParkingLots.value
2 return msg;
```

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

info **debug** dashboard

all nodes

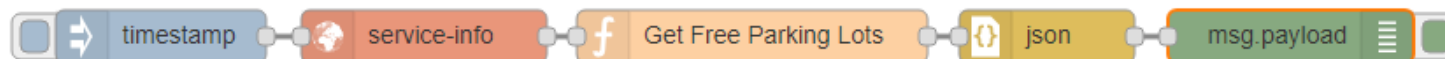
22/7/2019, 12:29:07 node: ada2a558.4e8698

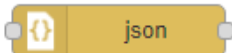
msg.payload : string[2]

"85"



## Step 4



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



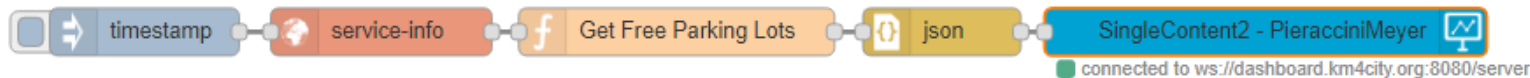
info debug dashboard

all nodes

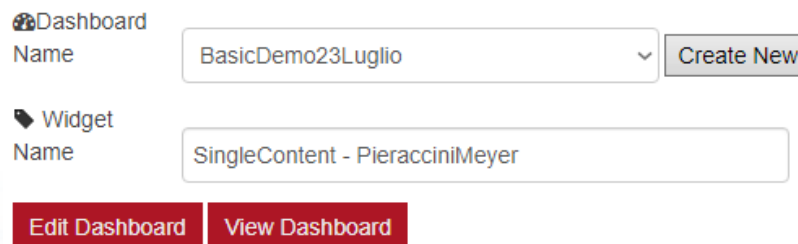
```
22/7/2019, 12:31:00 node: ada2a556.4e8698  
msg.payload : number  
85
```



## Step 5



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

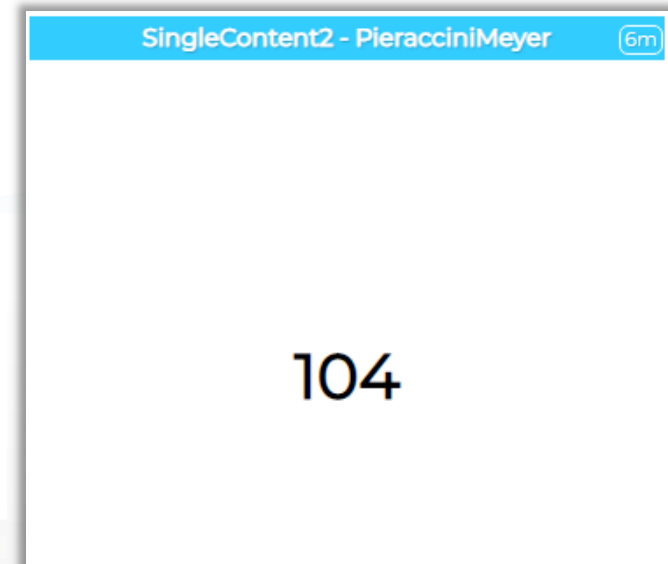


single content

Dashboard Name: BasicDemo23Luglio Create New

Widget Name: SingleContent - PieracciniMeyer

Edit Dashboard View Dashboard

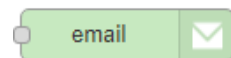




## Step 6

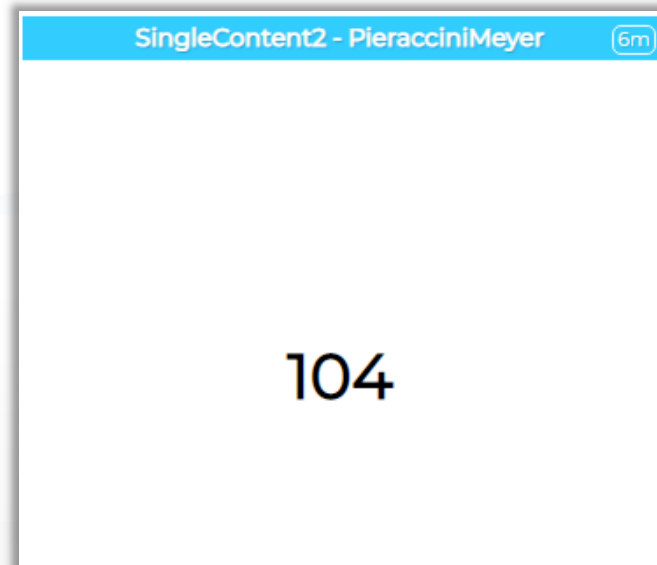


- Email
- Connect
- Configure
- Deploy
- Click and Observe
- Play with results



To: Destination Mail  
 Server: smtp.gmail.com  
 Port: 465 ☒ Use secure connection.  
 Userid: Userid of your mail  
 Password: Password of your mail

Change if not GMAIL





# Nodes configuration

**inject**

**Payload** timestamp

**Topic**

**Repeat** interval

every  minutes

☒ Inject once at start?

**service info**

**Name**

**ServiceUri**

**Language** Italian

**function**

**Name**

**Function**

```

1 msg.payload = msg.payload.realtime.results.
2   bindings[0].freeParkingLots.value
3 return msg;

```

**gauge chart**

**single content**

**speedometer**

**time trend**

**Dashboard** BasicDemo23Luglio Create New

**Widget** SingleContent - PieracciniMeyer

Edit Dashboard View Dashboard



# Nodes connections





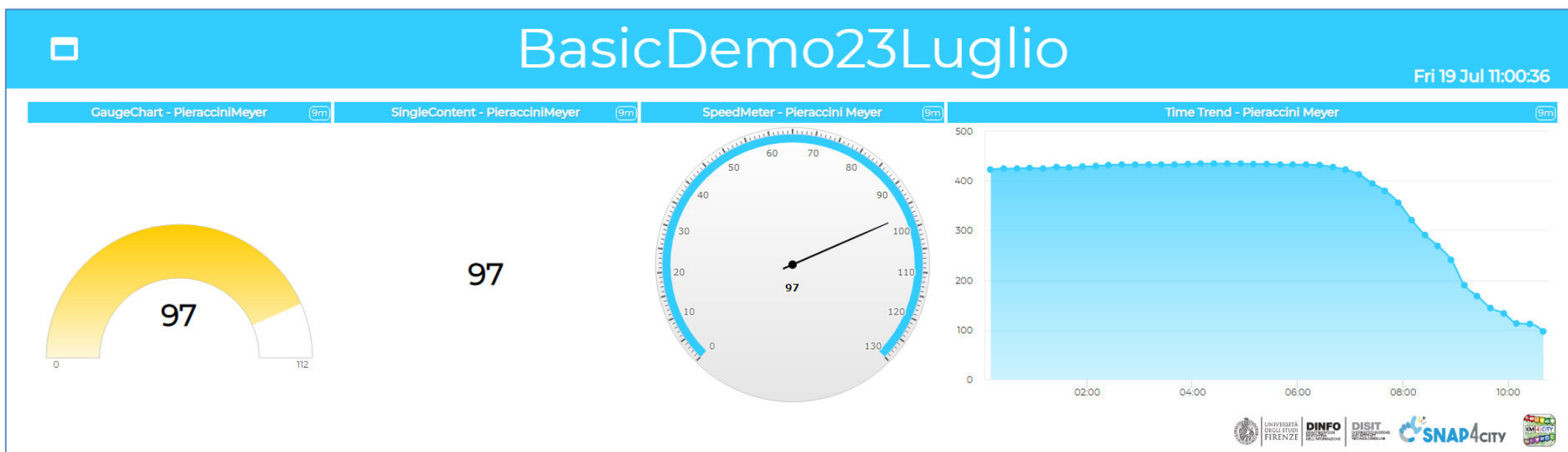
# Explaining: IOT Application Flow

- 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?iddashboard=MTk10Q==>



# *Production of IOT Applications Exercitation*



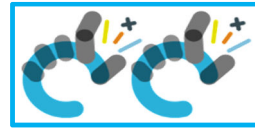




UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB



 **SNAP4CITY**



# 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/CarParkPieracciniMeyer>

## 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: .....

input

inject

output

debug

function

function

template

delay

trigger

comment

http request

tcp request

switch

change

range

split

join

csv

html

json

S4C Search

service search near marker

service search within circle

service search within polygon

service search along path

service info

full text search near marker

full text search within circle

full text search within polygon

full text search along path

full text search usr

S4CDashboard

impulse button

numeric keyboard

switch button

dimmer

gauge chart

single content

speedometer

time trend

geolocator

Bar content

Column content

web content

S4CKPIData

get my kpdata

get my kpdata values

get public kpdata values

get delegated kpdata values

save my kpdata values



# One Possible Solution





# Nodes configuration

**switch** Property

<input type="checkbox"/>	<input type="text" value="≥"/>	<input type="text" value="50"/>	→ 1	<input type="button" value="x"/>
<input type="checkbox"/>	<input type="text" value="&lt;"/>	<input type="text" value="50"/>	→ 2	<input type="button" value="x"/>

**Free Park**

Name

Function

```

1 msg.payload =
2 "<b style='color: green' >Free " + msg.payload + "</b>"
3 return msg;

```

**Busy Park**

Name

Function

```

1 msg.payload =
2 "<b style='color: red' >Full " + msg.payload + "</b>"
3 return msg;

```





UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB



# Resulting Dashboard



<https://main.snap4city.org/view/index.php?iddashboard=MTk2MQ==>







UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB



# *Data Processing with IoT Application (DEMO)*





## Example of more Complex IOT Application

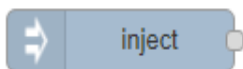
In this demo let's create an IoT Application that:

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

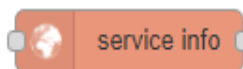




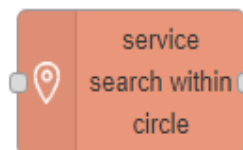
# Nodes for flow 1/2



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)

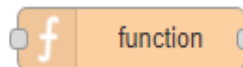


Requests detailed information for a specific service on the platform (such as a car park, hotel, etc.)

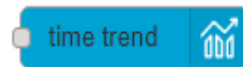
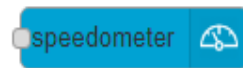
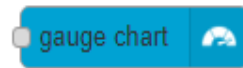


Search in around a certain point of the indicated service. It returns:

- 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

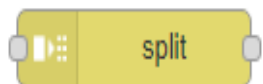


Display values in different modes on a dashboard. The node called single content accepts strings, numbers and html. The others only accept numbers.



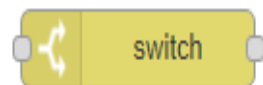


## Nodes for flow 2/2

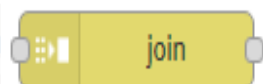


Divides the input message into multiple messages as indicated in the configuration.

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

**inject**

**Payload**

**Topic**

**Repeat**

every

☒ Inject once at start?

**service info**

**Name**

**ServiceUri**

**Language**

**split**

**Array**

Split using

**gauge chart**

**single content**

**speedometer**

**time trend**

**Dashboard**

**Name**

**Widget**

**Name**

**Sum Of Free Park**

**Name**

**Function**

```

1 var sum = 0;
2 for (var i = 0; i < msg.payload.length; i++){
3   sum = sum + parseInt(msg.payload[i].realtime.results.bindings[0].freeParkingLots.value);
4 }
5 msg.payload = sum;
6 return msg;

```



# Nodes configuration 2/2

location pin icon

service search within circle

Max Results

100

Language

French

Latitude

43.775246

Longitude

11.250564

Max Distance

6.534

Categories

Car\_park

car park

Transfer Service And Renting

Car Park

switch

Name

Name

Property

msg.payload.realtime.results

is not null

→ 1

join

Mode

manual

Combine each

msg.payload

to create

an Array

Send the message:

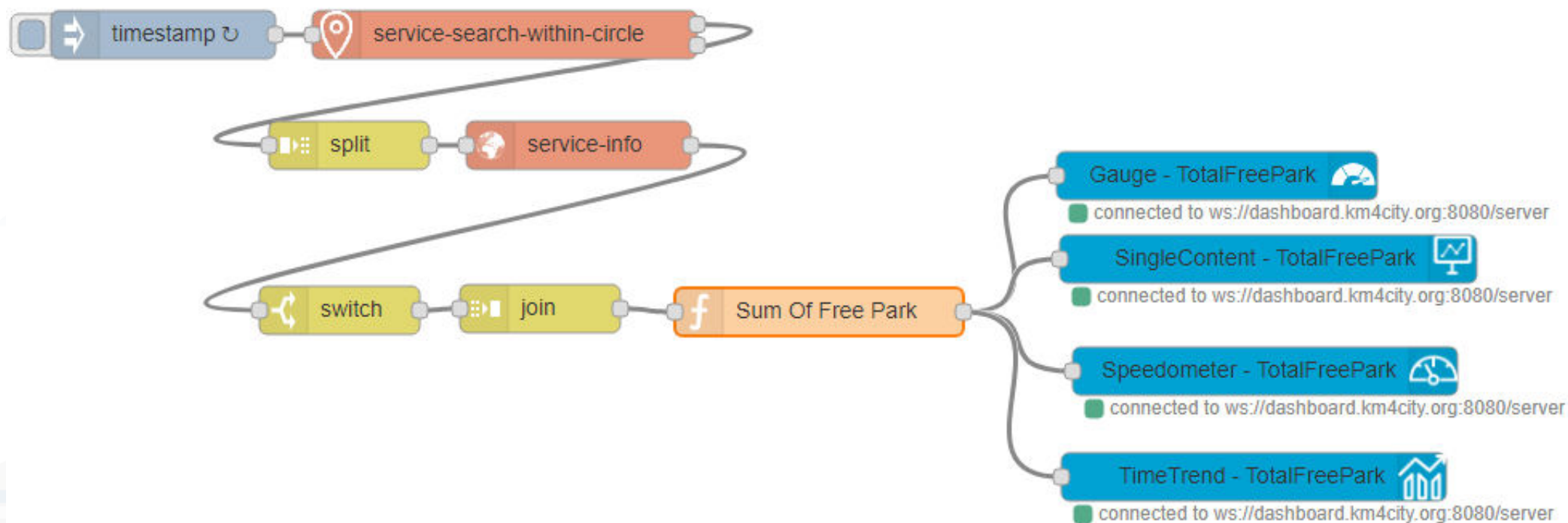
- After a number of message parts 

count
- After a timeout following the first message 

3



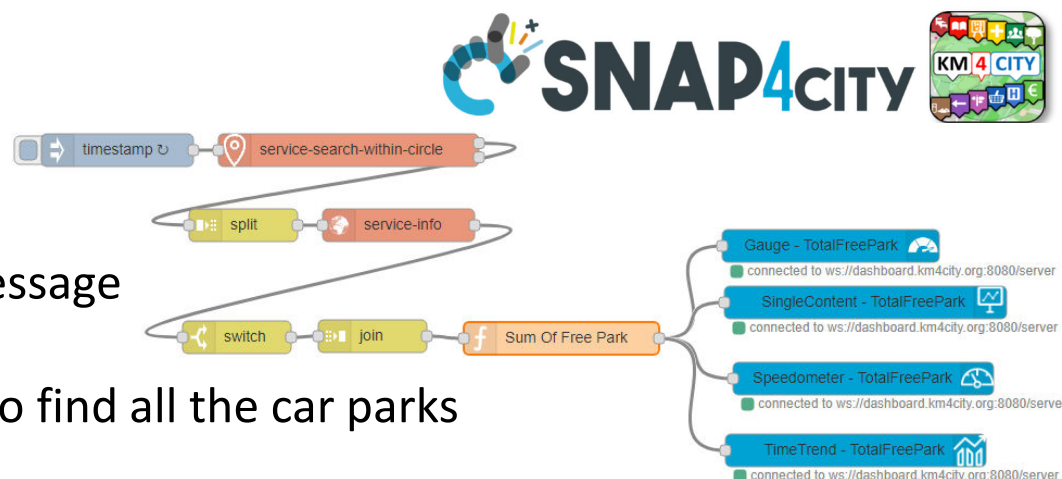
# Nodes connections





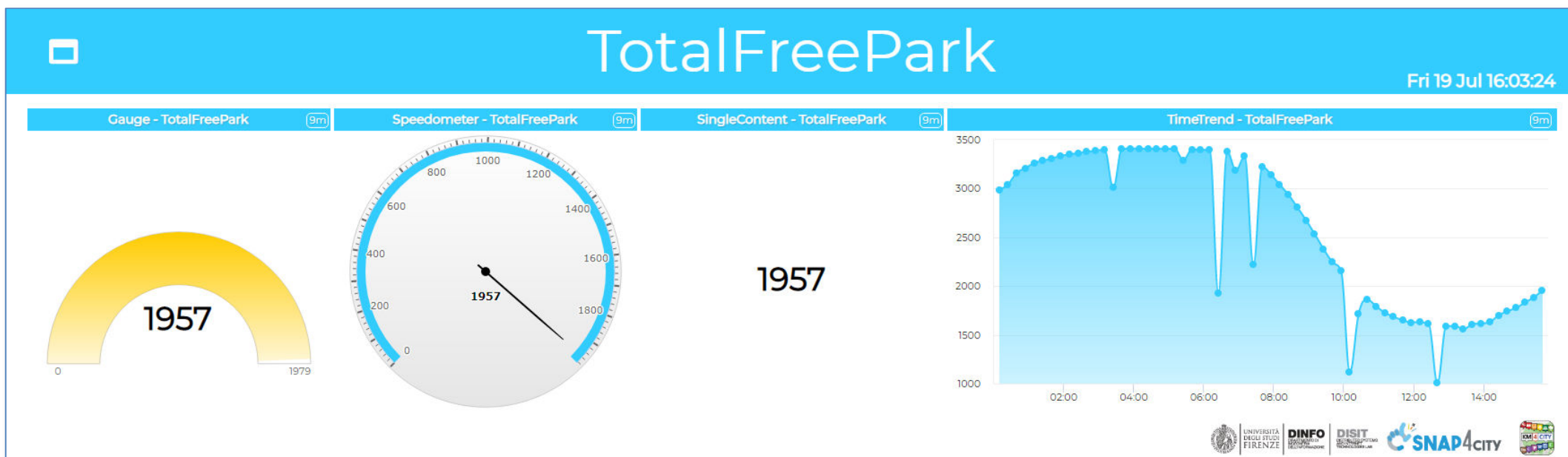
# Nodes explanation 1/2

- 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: .....

input

inject

output

debug

function

function

template

delay

trigger

comment

http request

tcp request

switch

change

range

split

join

csv

html

json

S4C Search

service search near marker

service search within circle

service search within polygon

service search along path

service info

full text search near marker

full text search within circle

full text search within polygon

full text search along path

full text search usr

S4CDashboard

impulse button

numeric keyboard

switch button

dimmer

gauge chart

single content

speedometer

time trend

geolocator

Bar content

Column content

web content

S4CKPIData

get my kpdata

get my kpdata values

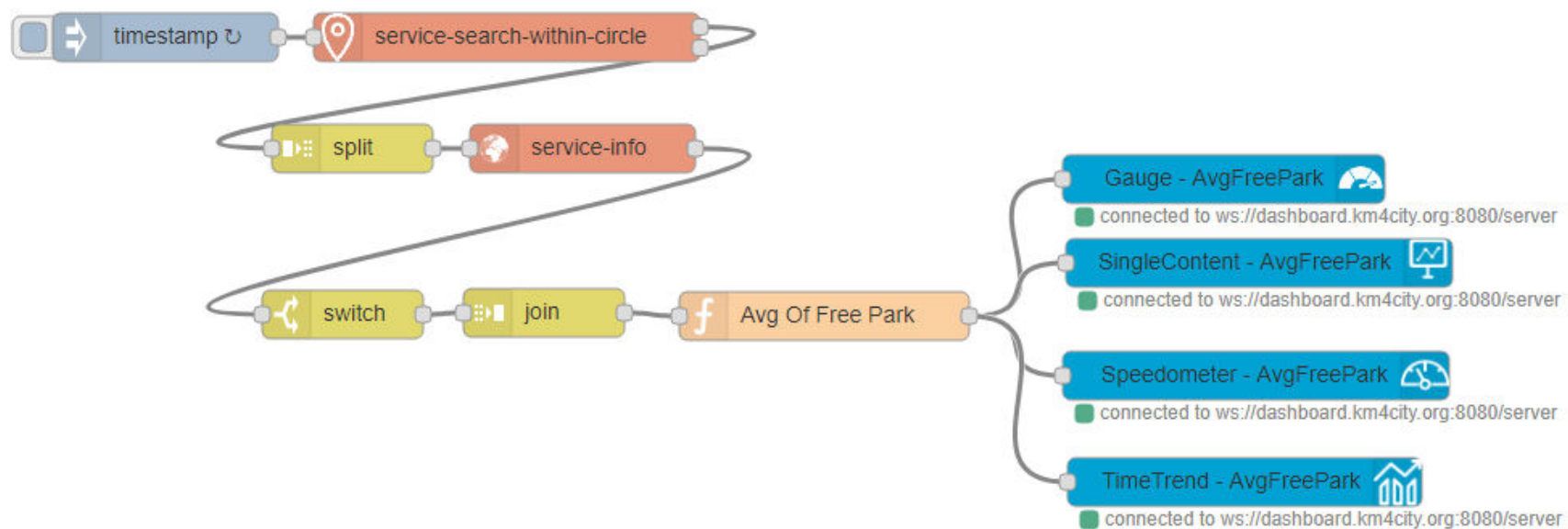
get public kpdata values

get delegated kpdata values

save my kpdata values



# One Possible Solution





# Nodes configuration 1/2

**inject**

**Payload**

**Topic**

**Repeat**

every

☒ Inject once at start?

**service info**

**Name**

**ServiceUri**

**Language**

**split**

**Array**

Split using

**gauge chart**

**single content**

**speedometer**

**time trend**

**Dashboard**

**Name**

**Widget**

**Name**

**Avg Of Free Park**

**Name**

**Function**

```

1 var sum = 0;
2 for (var i = 0; i < msg.payload.length; i++){
3   sum = sum + parseInt(msg.payload[i].realtime.results.bindings[0].freeParkingLots.value);
4 }
5 msg.payload = parseInt(sum/msg.payload.length);
6 return msg;

```



# Nodes configuration 2/2

service search within circle

Max Results

100

Language

French

Latitude

43.775246

Longitude

11.250564

Max Distance

6.534

Categories

car\_park

car park

Transfer Service And Renting

Car Park

switch

Name

Name

Property

msg.payload.realtime.results

is not null

→ 1

join

Mode

manual

Combine each

msg.payload

to create

an Array

Send the message:

After a number of message parts

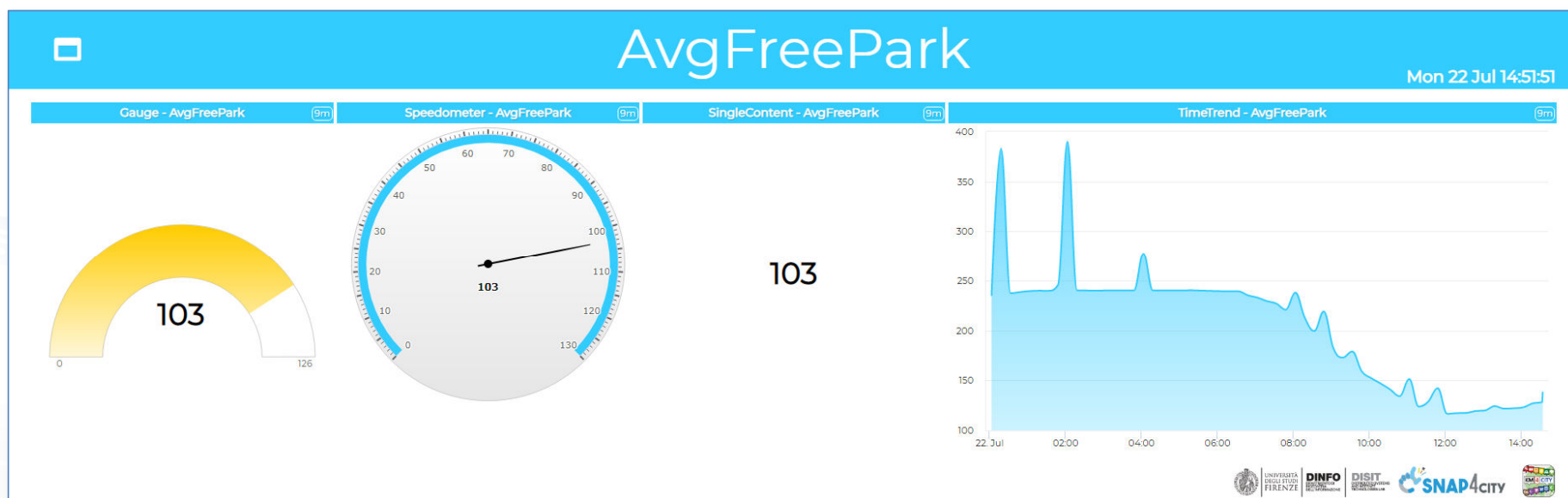
count

After a timeout following the first message

3



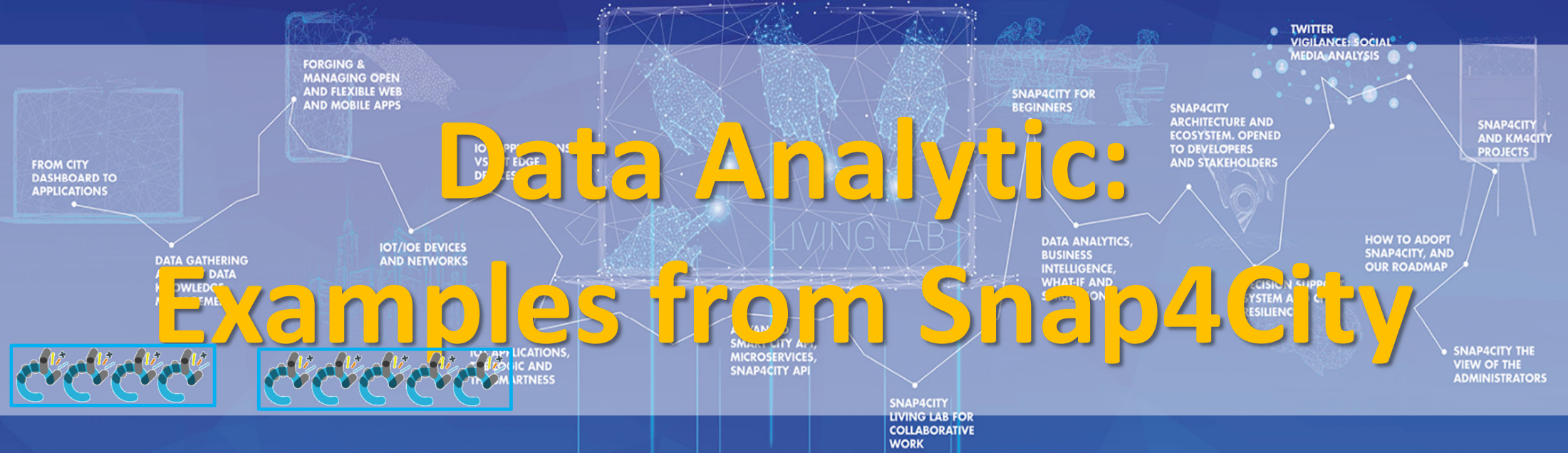
# Resulting Dashboard



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

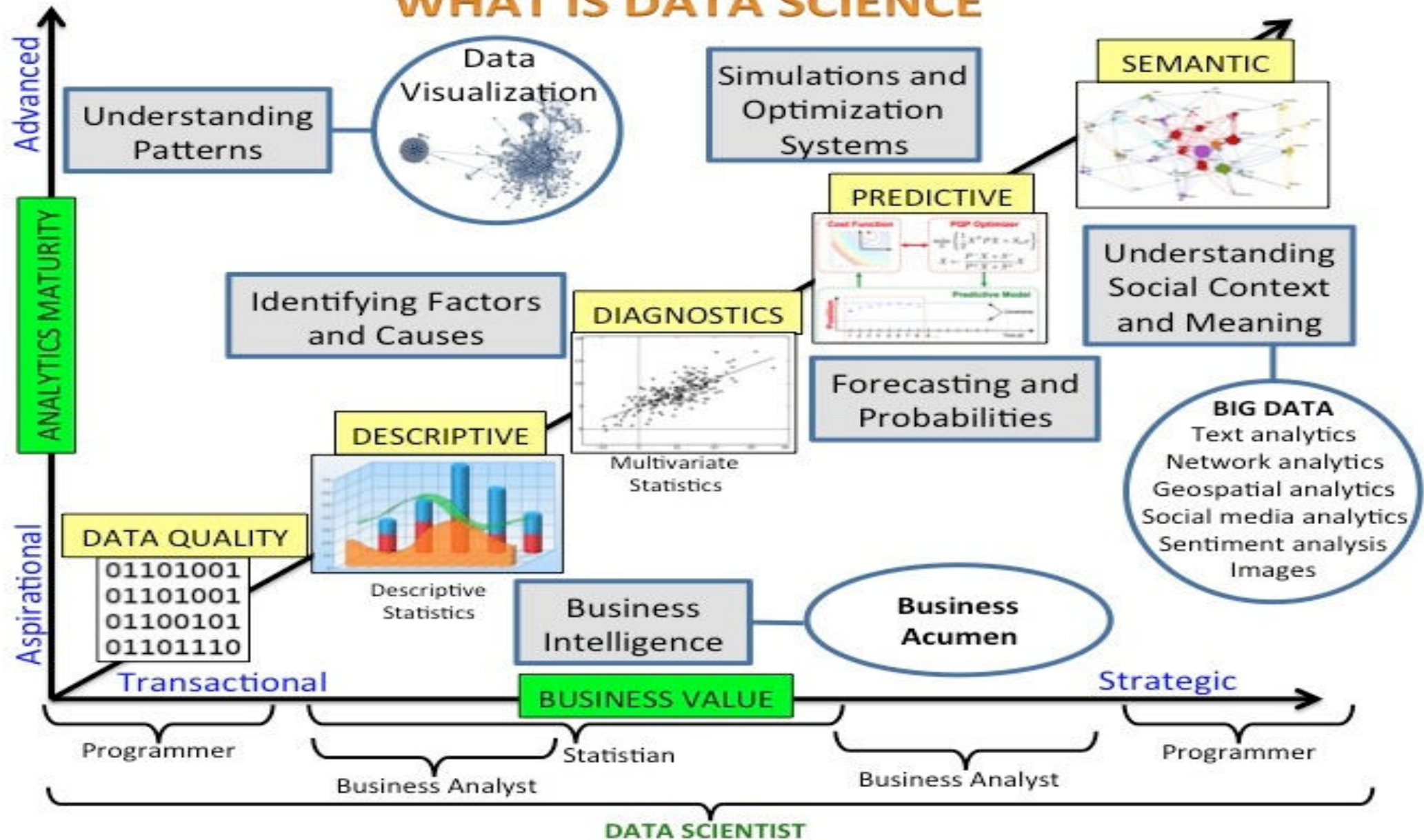


## Data Analytic: Examples from Snap4City





# WHAT IS DATA SCIENCE







UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB



**SNAP4CITY**



# 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](#)





UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB

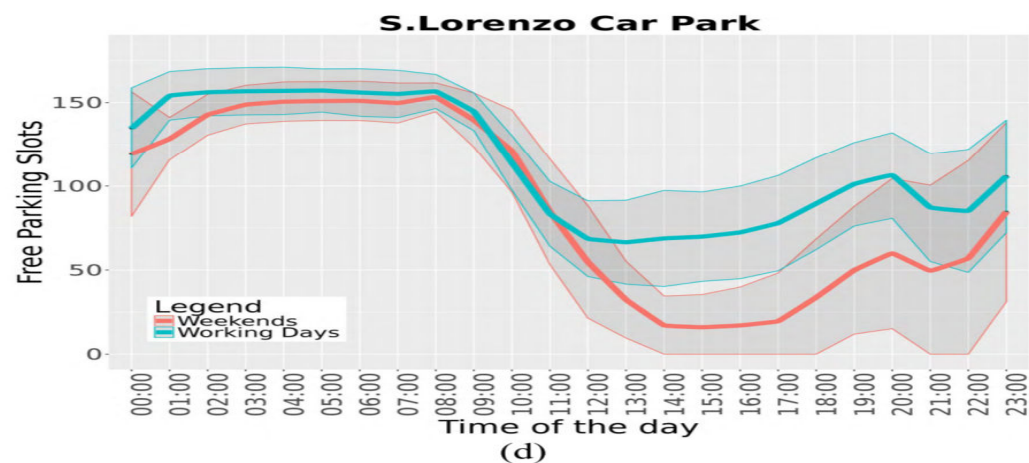
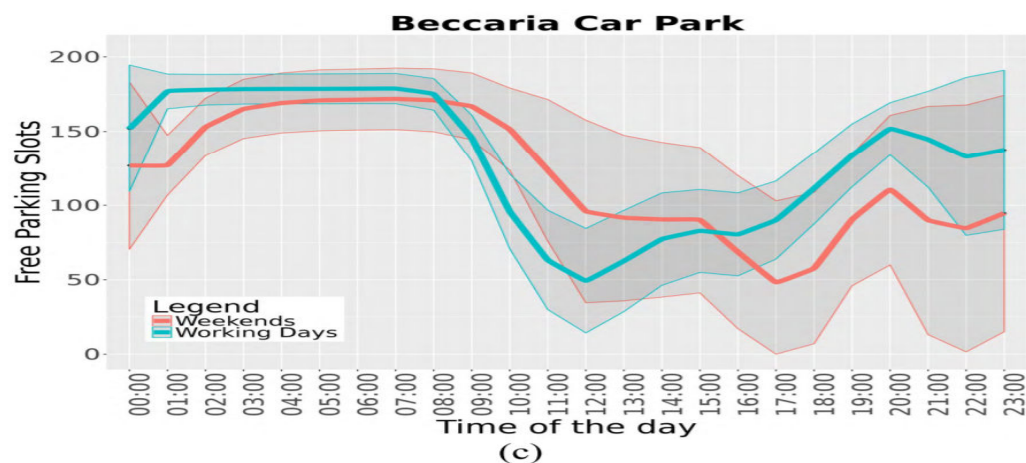
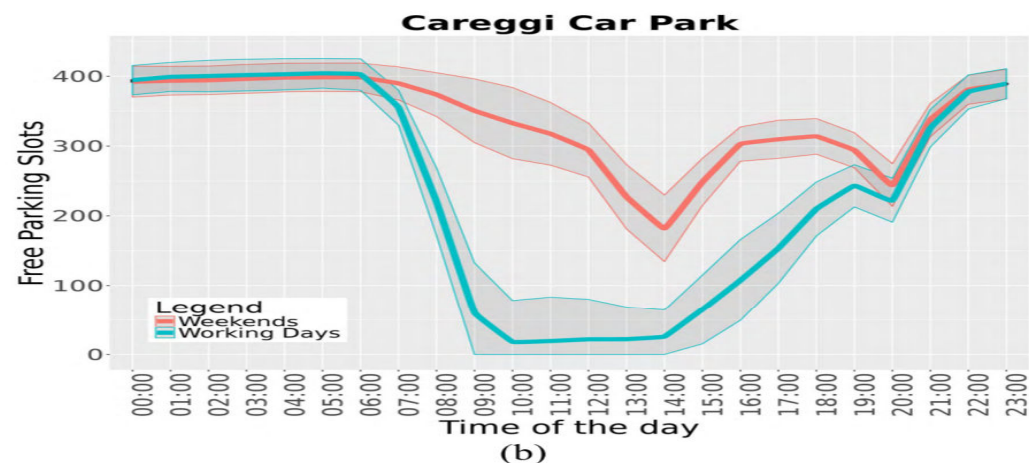
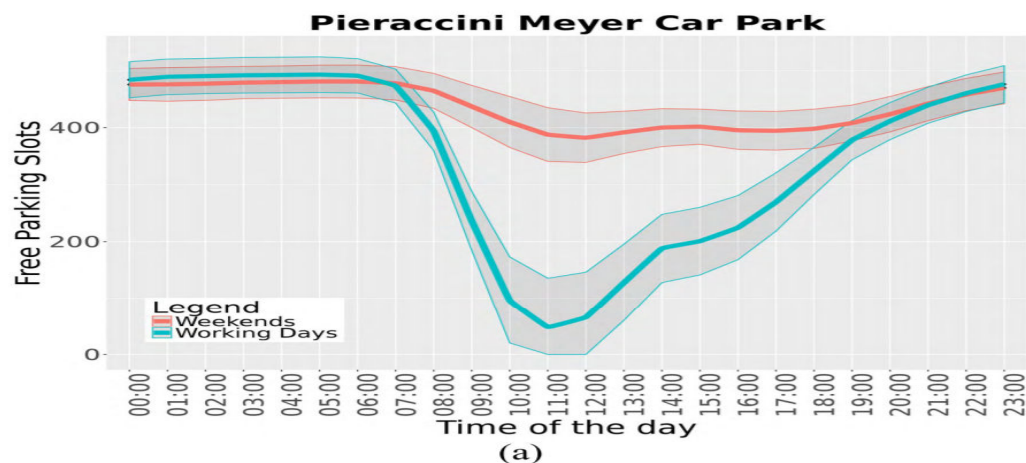


# *Smart Parking: predictions*





# Free Parking space trends



## 12 parking areas in Florence

Snap4City (C), 23 July 2019

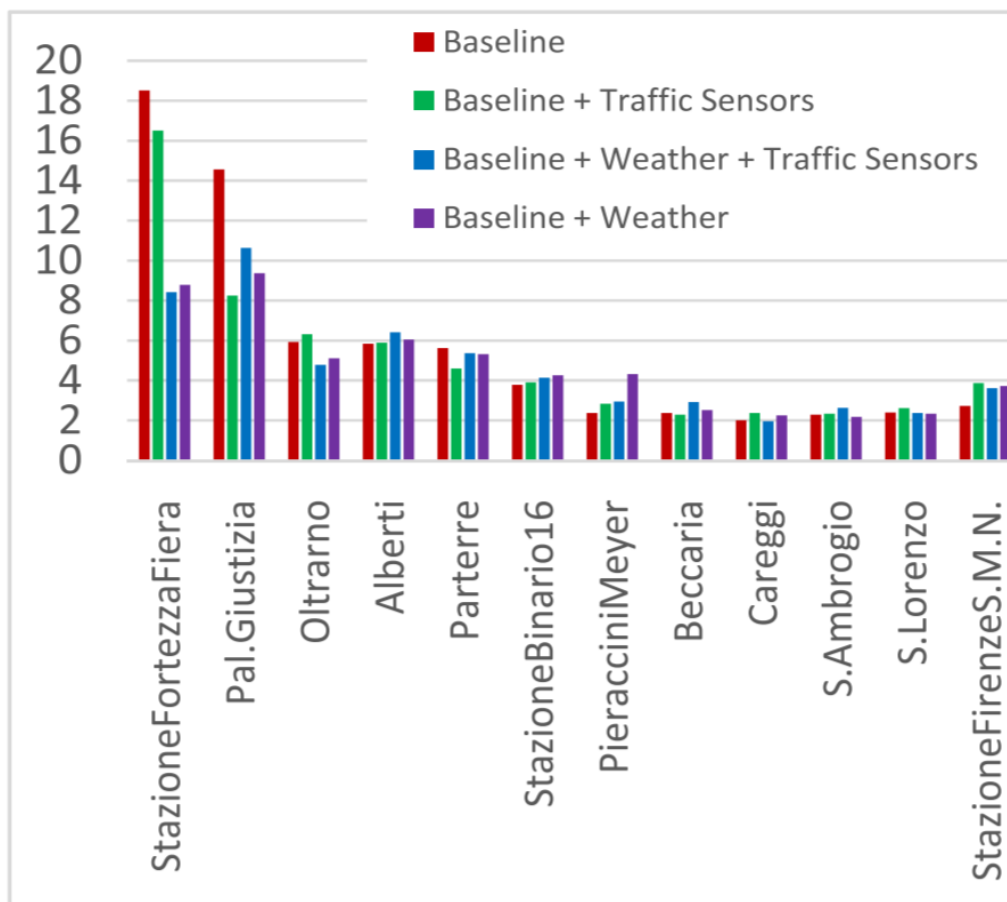


# Free Parking PREDICTIONS



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
<b>Careggi car park</b>			
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
<b>Pieraccini Meyer car park</b>			
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
<b>S. Lorenzo car park</b>			
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
<b>Beccaria car park</b>			
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





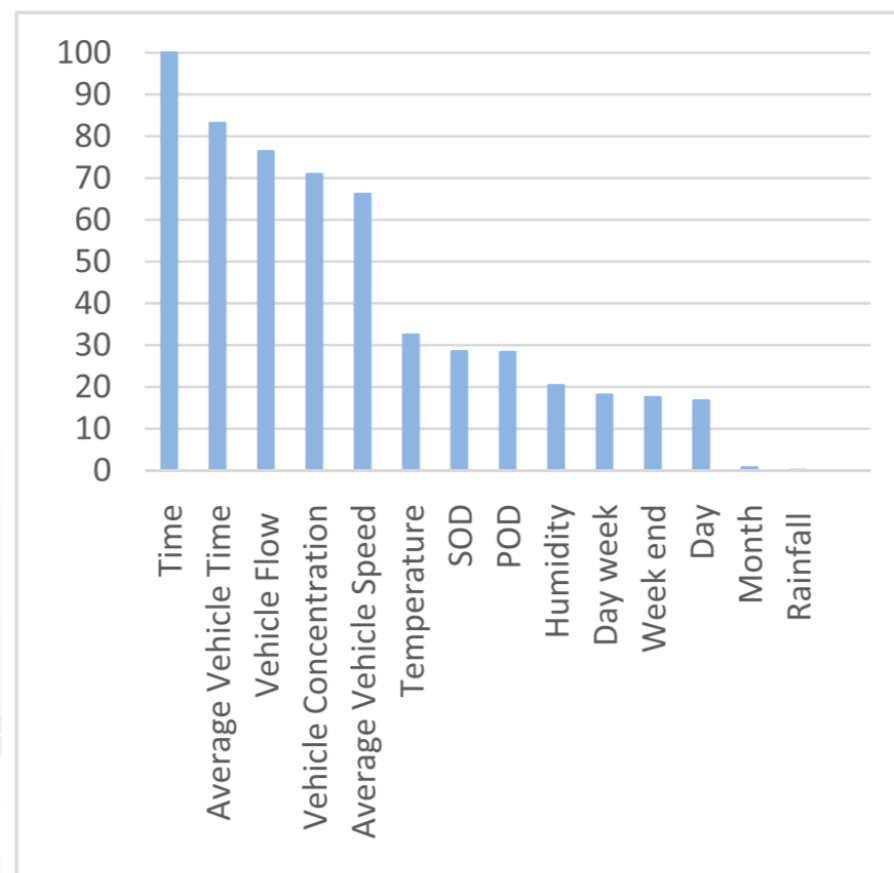
# Free Parking PREDICTIONS



## Performances

## Relevance of Variable

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





# 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%



Snap4City (C), 23 July 2019

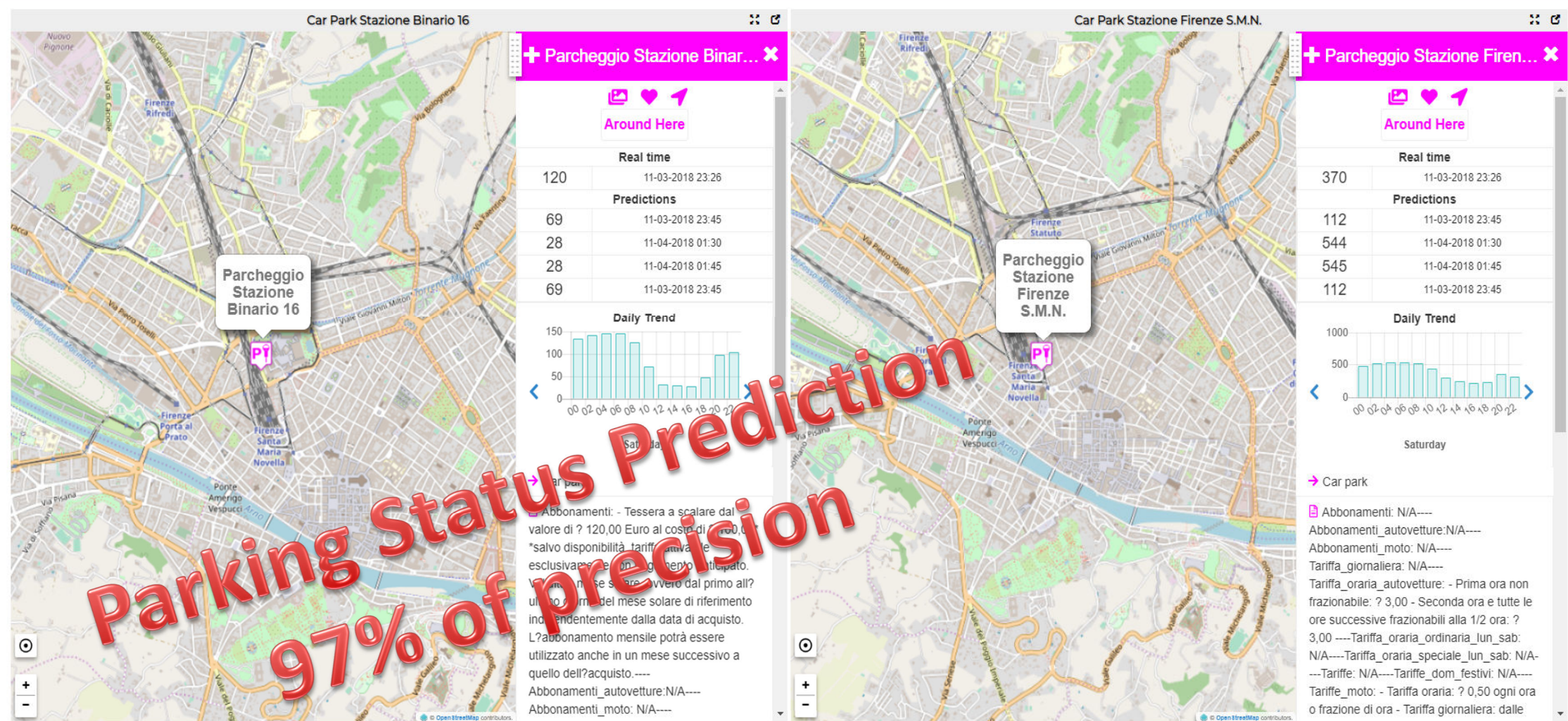






# Monitoring Station for Parking

Sat 3 Nov 23:39:55



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



# Predictions on Parking

- 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/>

**IEEE**  
Access®





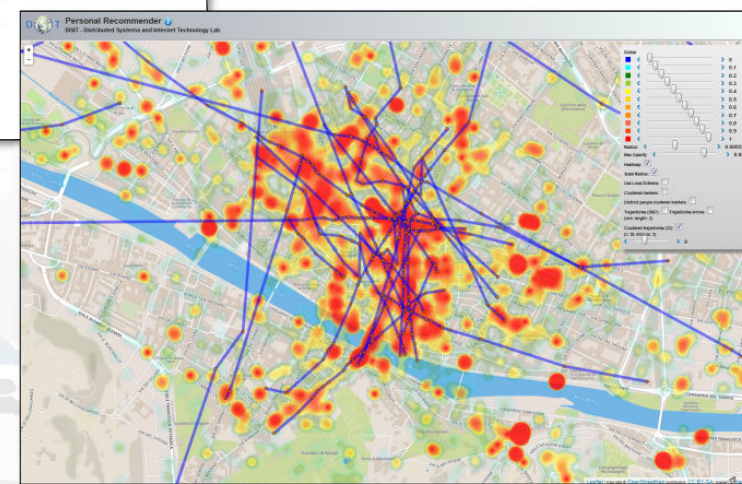
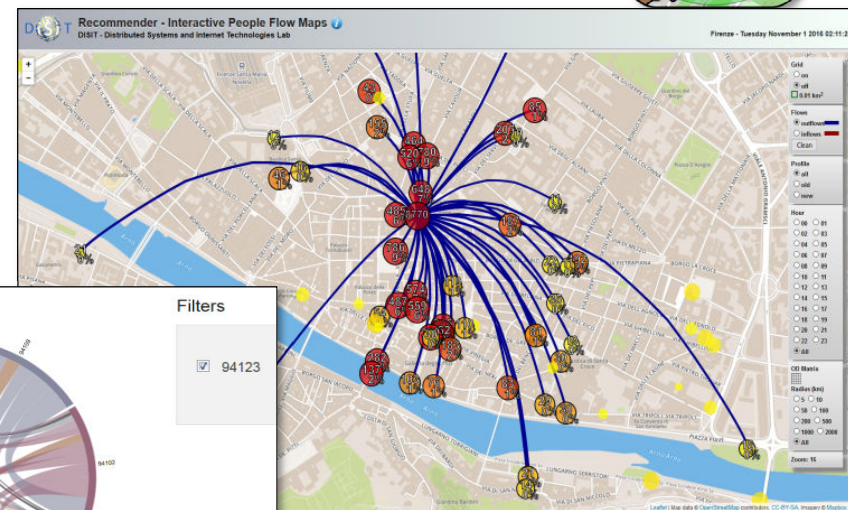
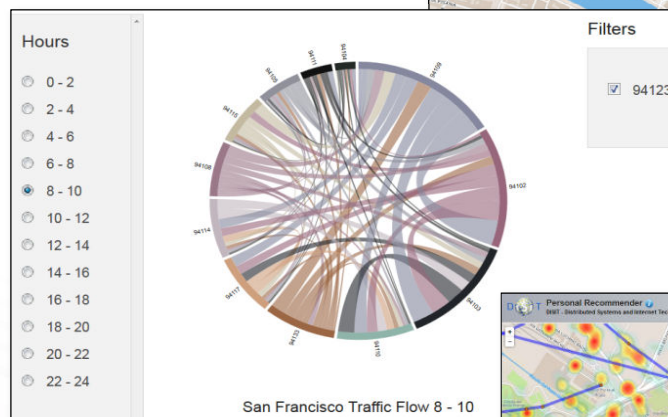
# *User Behaviour Analysis via Wi-Fi, OD Matrices, Trajectories*





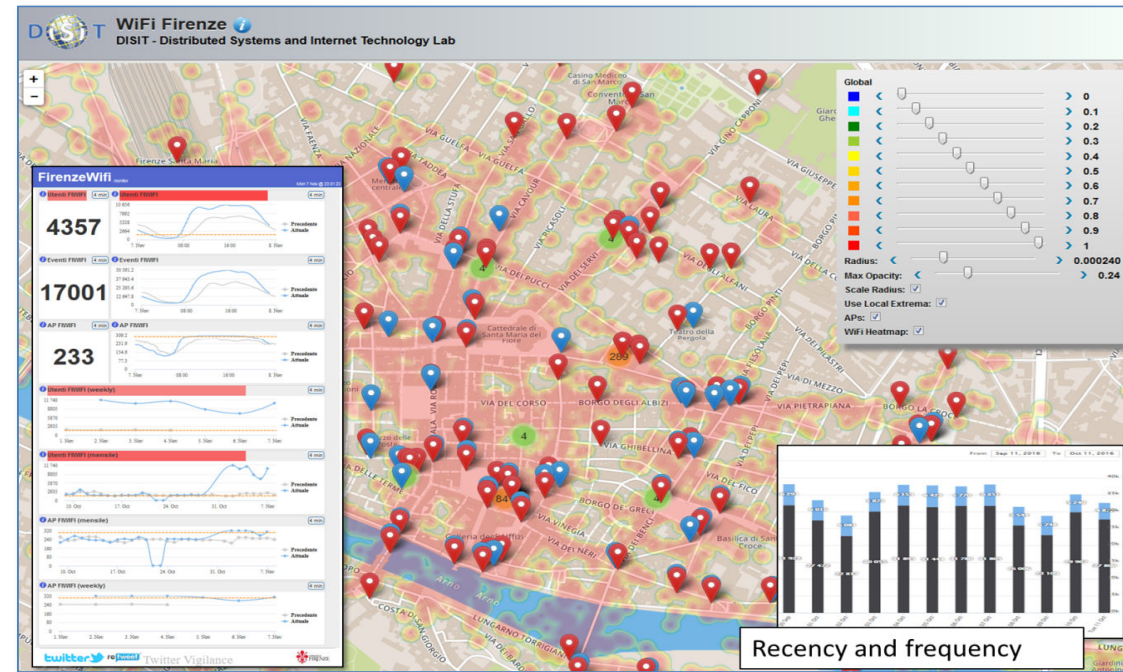
# User Behaviour Analysis

- **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**

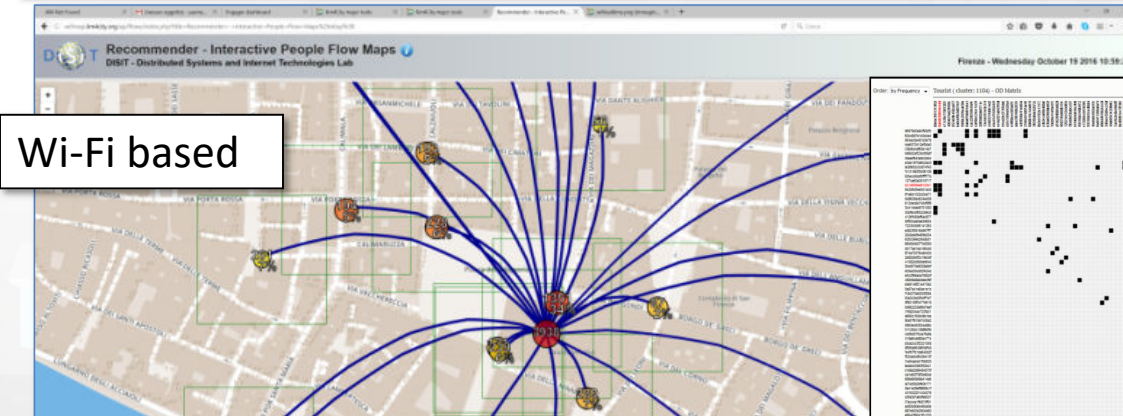




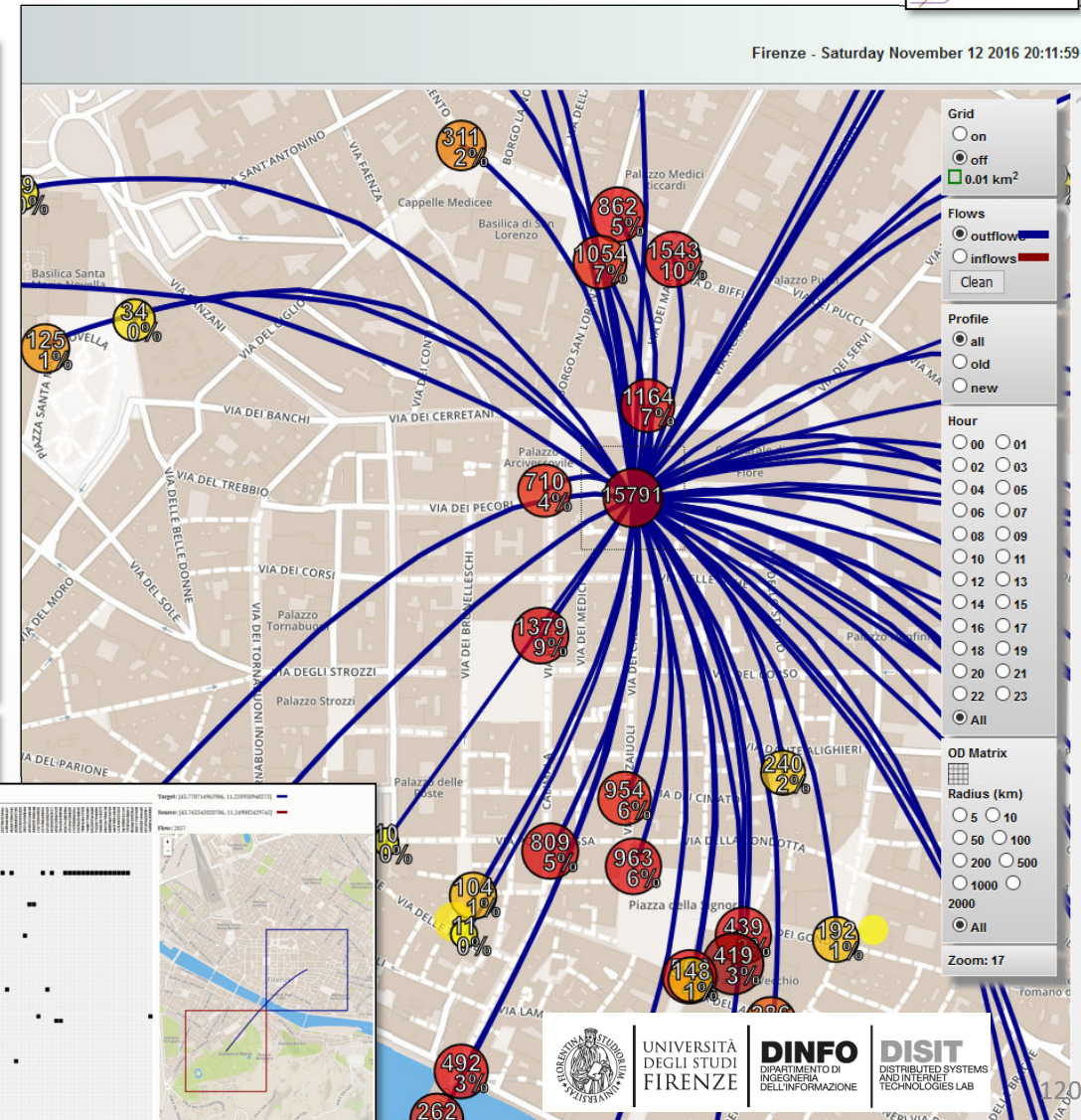
# Origin Destination Matrix Estimation



Recency and frequency

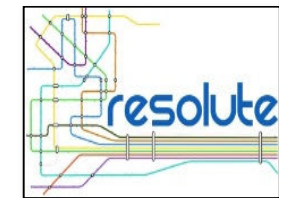


Wi-Fi based



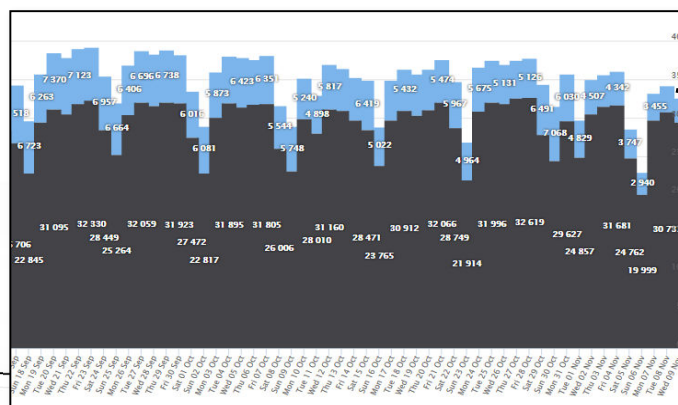
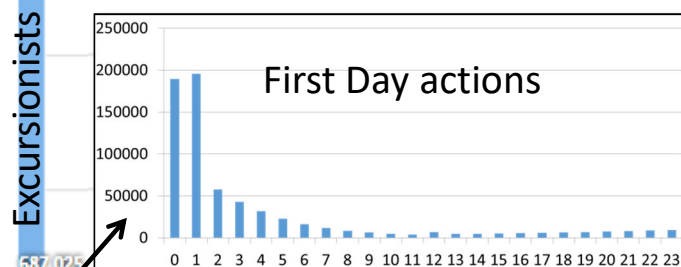


# User Behaviour Analysis

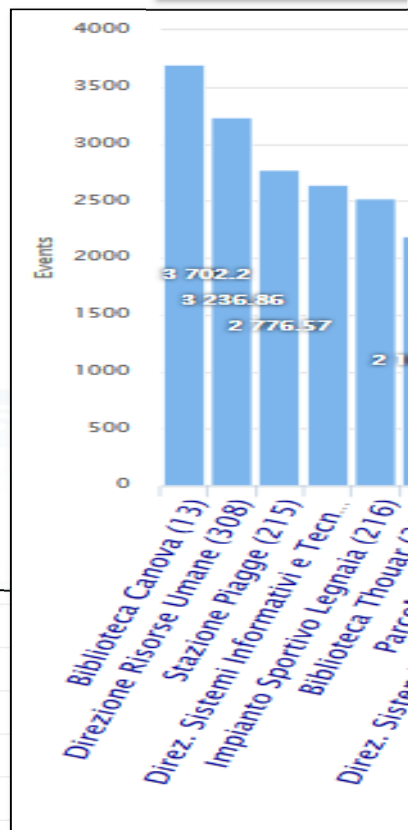


Distinct APs: 343  
Distinct APs (last 24 hours): 311  
Distinct Users (last 180 days): 1102098  
Distinct Excursionists (last 180 days, < 24 h): 687025

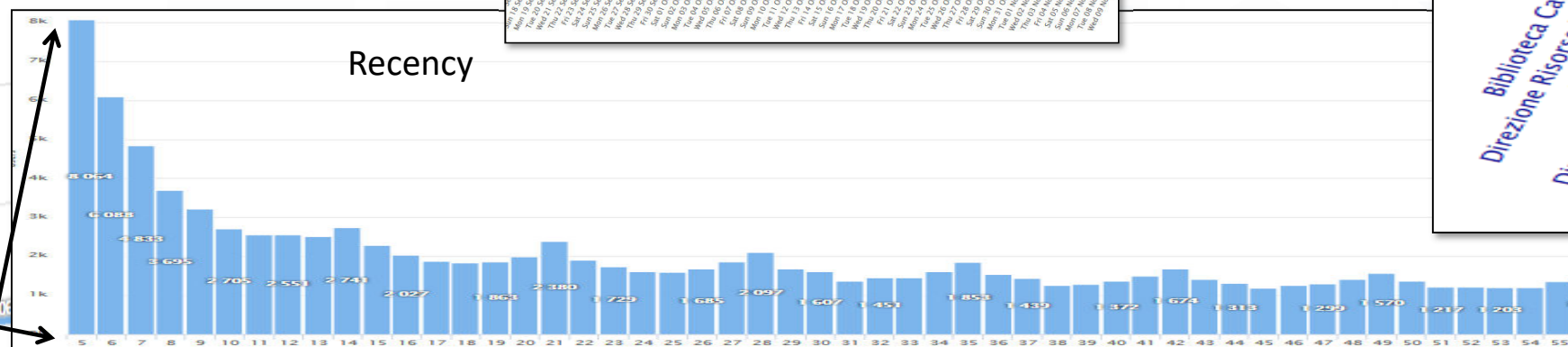
Where



New City Users  
VS  
Returning



Recency





# Characterizing City Areas

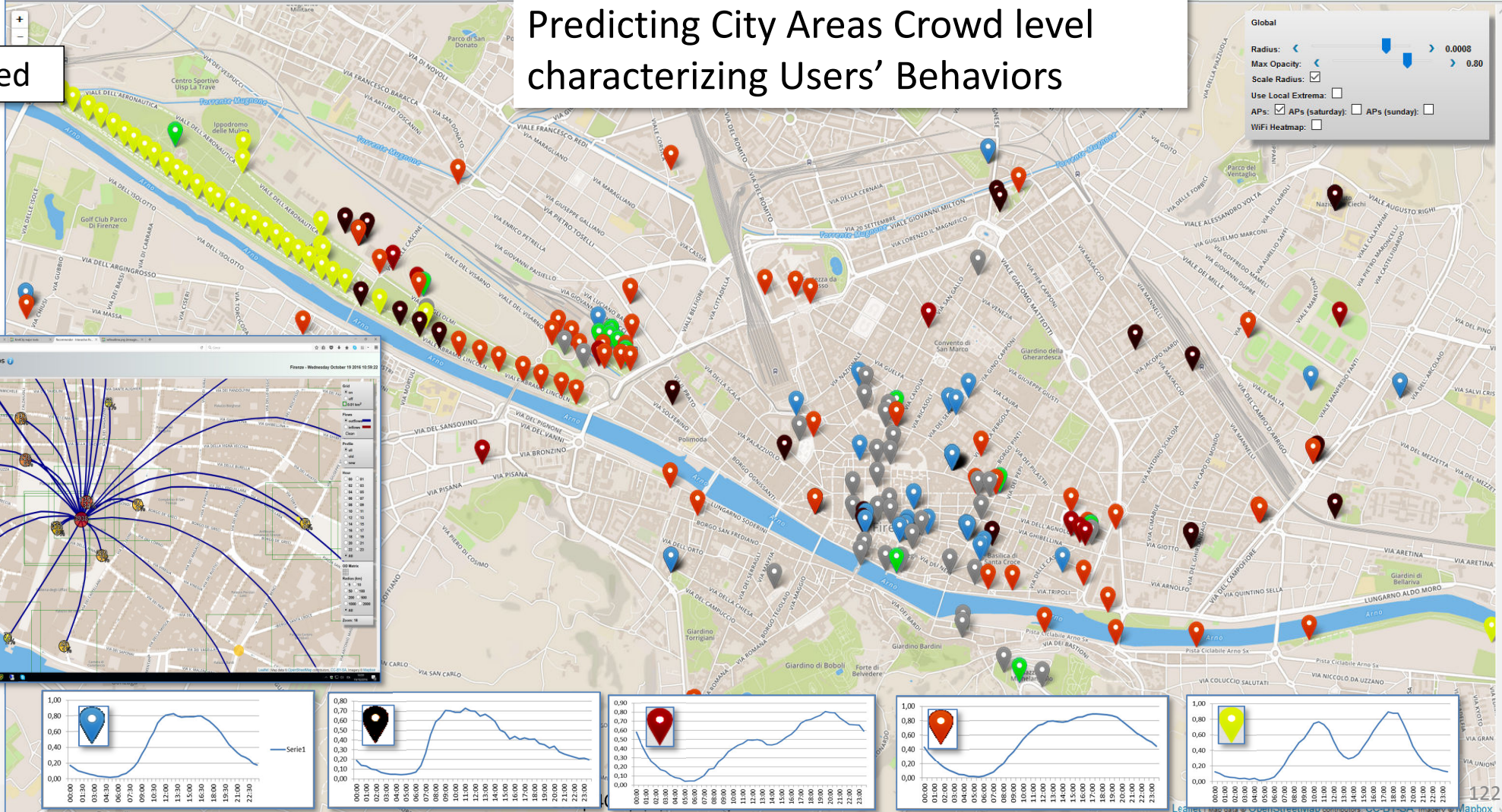


DISIT Firenze Wi-Fi: Access Points Clusters Coverage Map  
DISIT - Distributed Systems and Internet Technologies Lab

Firenze - Saturday November 12 2016 19:16:33

Wi-Fi based

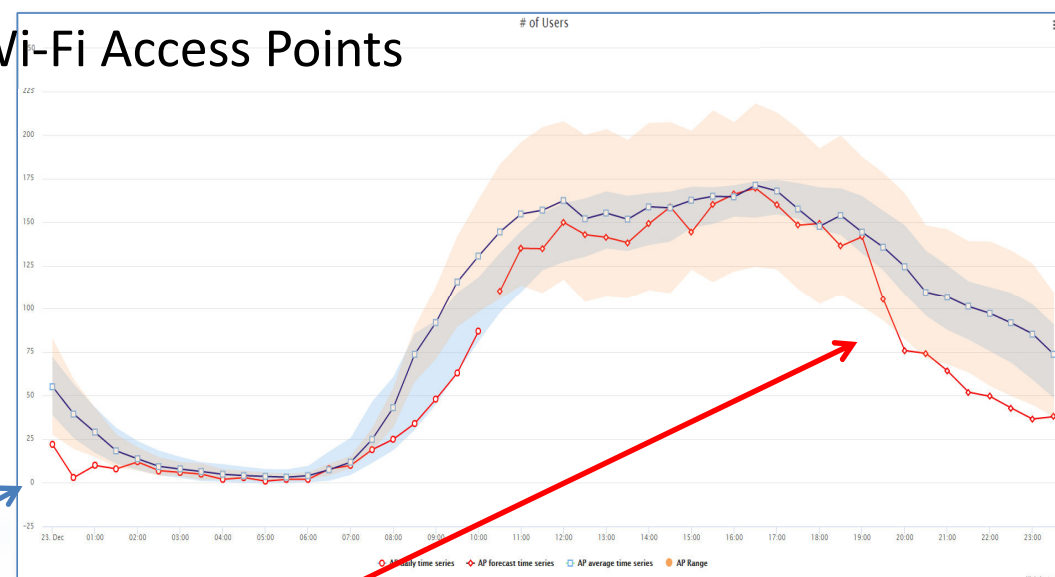
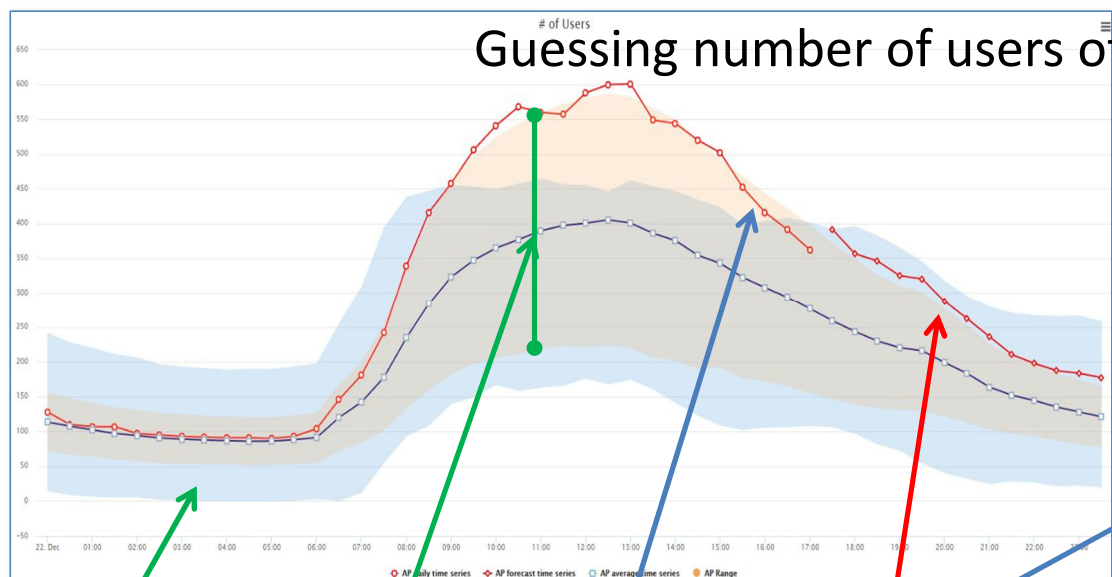
Predicting City Areas Crowd level  
characterizing Users' Behaviors





# Prediction and Identification of Anomalies

# of Users  
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>





# *Traffic Flow Reconstruction from Traffic Sensors Data*

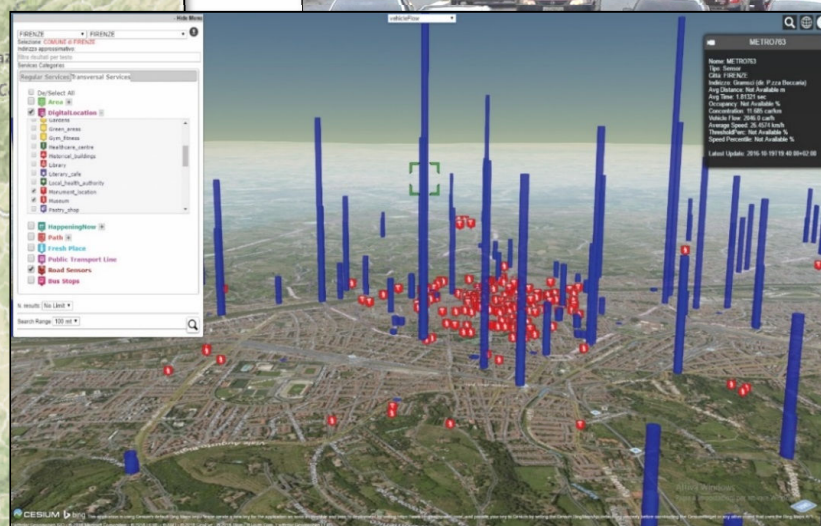
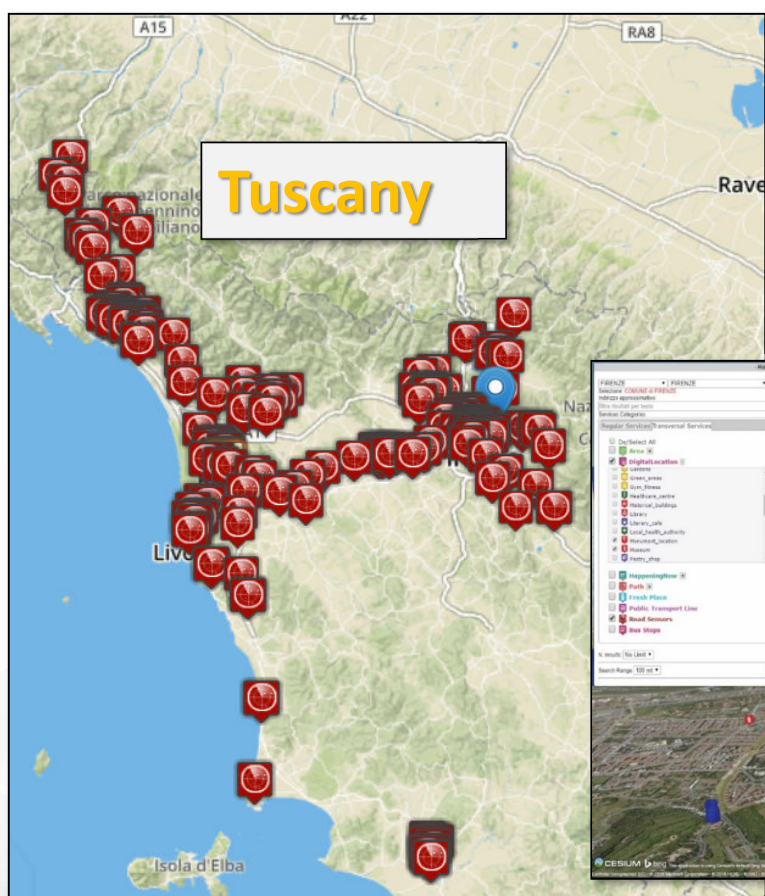




# Traffic Flow Tools

## Spire and Virtual Spires (cameras), Bluetooth, ...

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



Snap4City (C), 23 July 2019



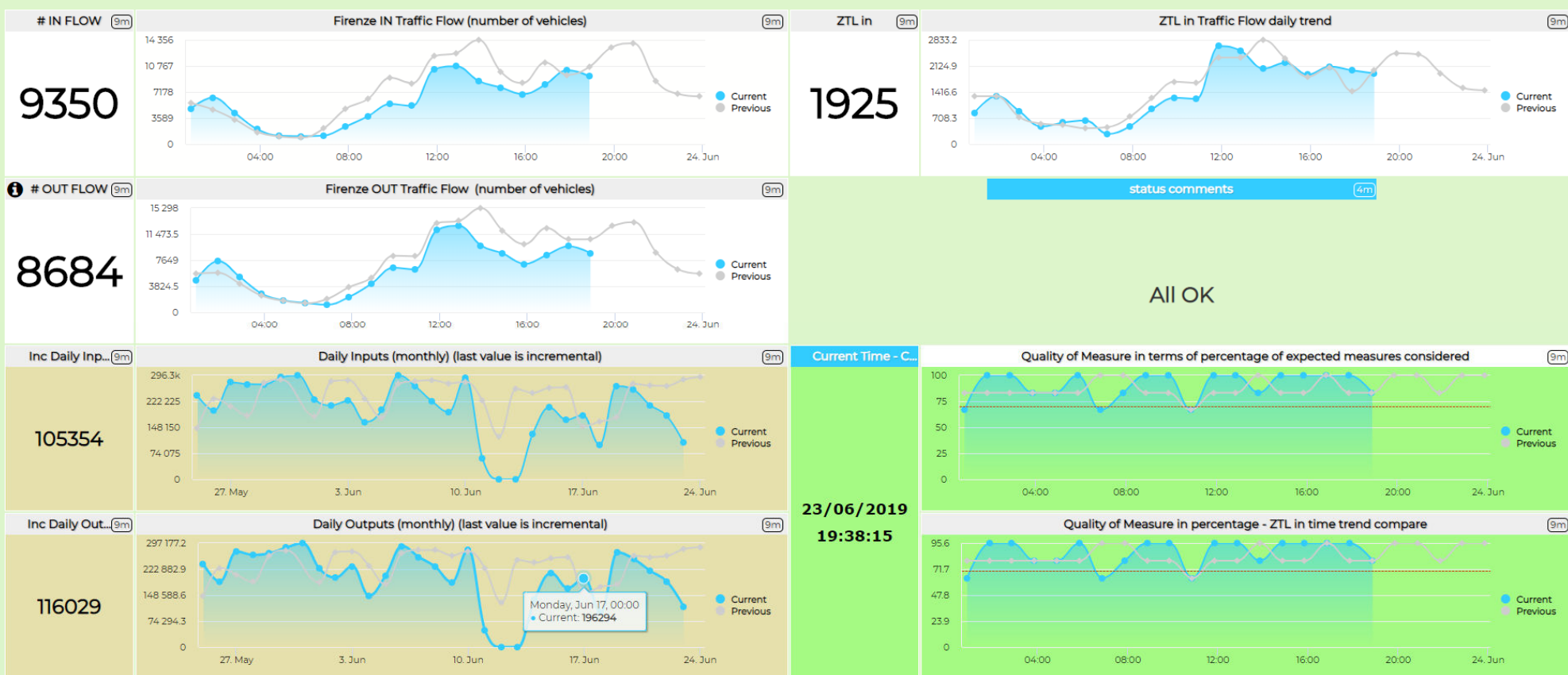
# Traffic Flow data



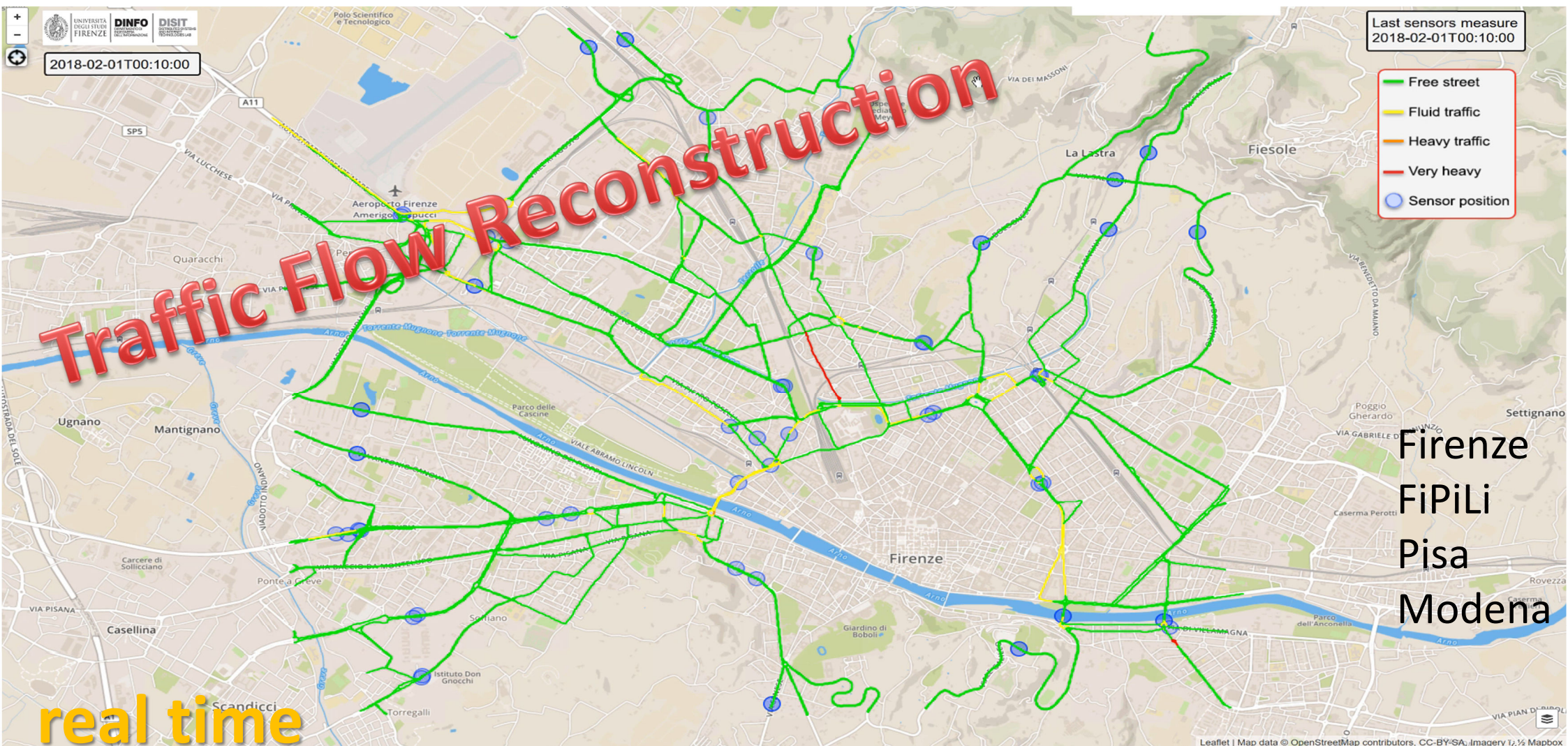


# Traffic Flow Monitoring - Firenze - Cloned

Sun 23 Jun 19:38:15







<http://firenzetraffic.km4city.org>

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

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

Snap4City (C), 23 July 2019

129





# Traffic Flow Reconstruction for the cities

Sun 23 Jun 19:40:52



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

Snap4City (C), 23 July 2019



## 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



# *Modal & Multimodal Routing for Navigation and Travel Planning*





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
- DISIT Lab portal

## Service Map

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**

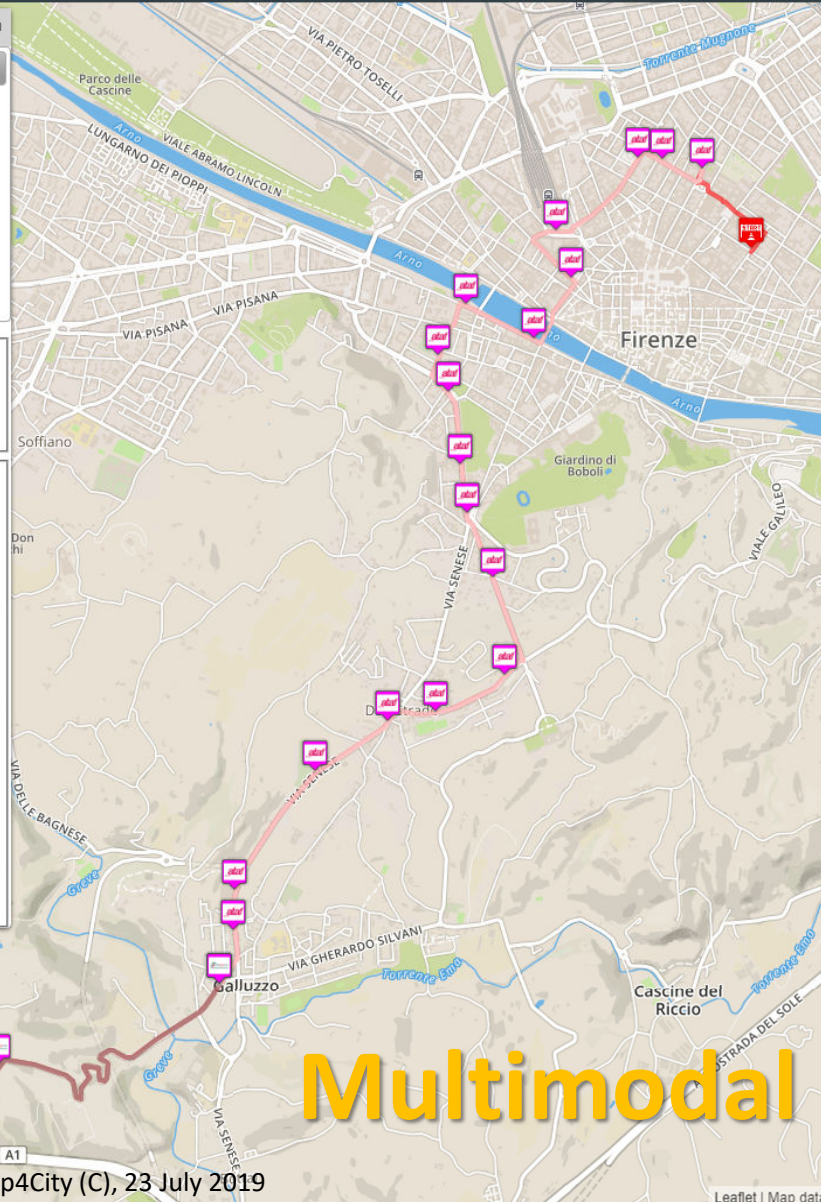
Actual Selection:  
Coord: 43.7130,10.9272  
Address: [VIA DI PRATOVECCHIO, 58, EMPOLI](#)

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 : Arazzeri - 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 Sanzia Fr.157 - Via Sanzia Fr. Coop Sn 1002m (08:08:00)
12. Via Raffaello Sanzia 45m (08:10:00)
13. nd 33m (08:10:35)



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...

Icons: Search, Refresh, Home, Print





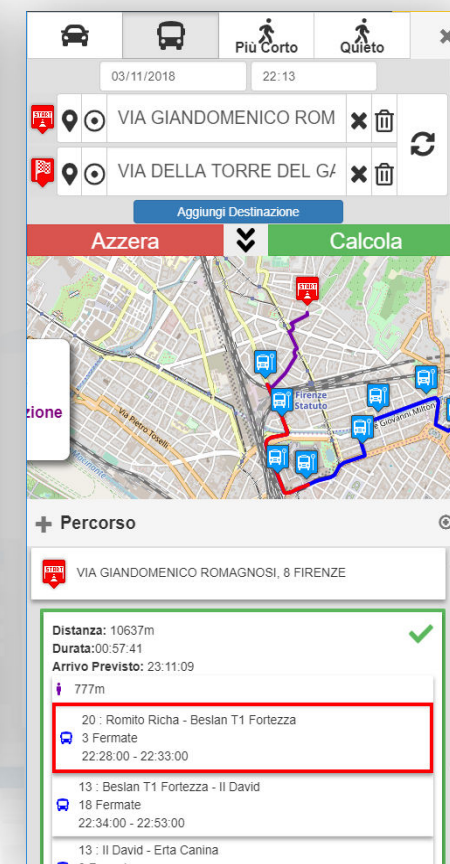
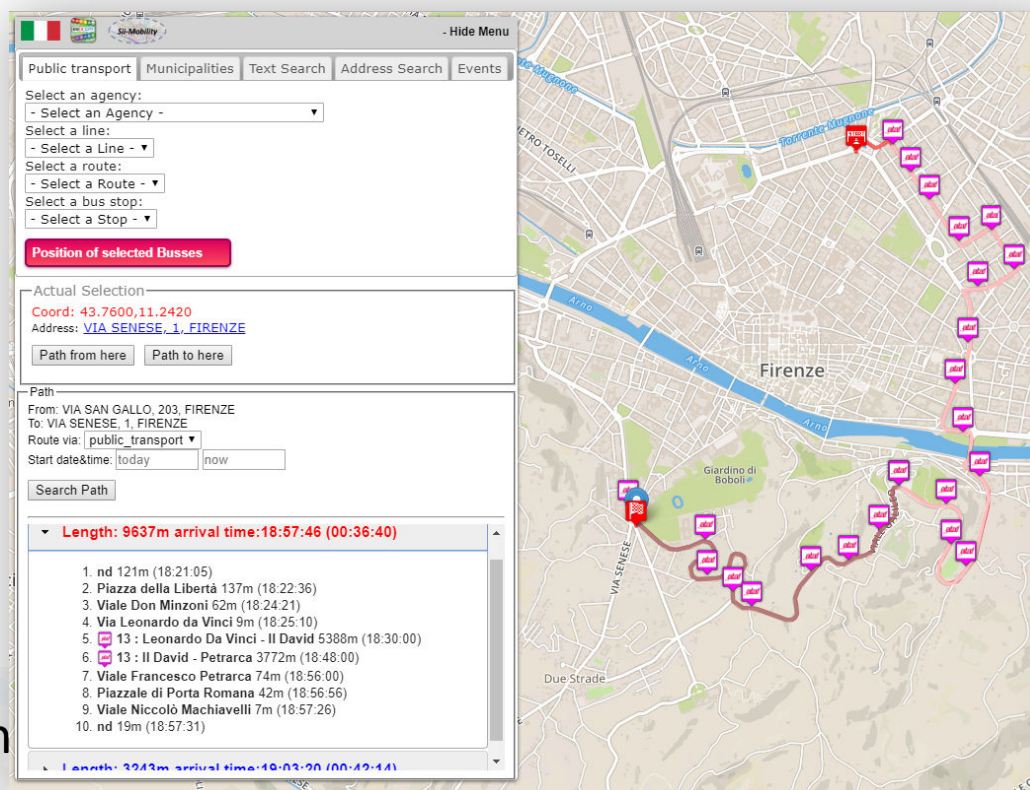
# Routing and Multimodal Routing

## 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

Micro Applications: Travel Planner, routing - Helsinki

05/05/2019 12:03

Kolmas linja, 17 Helsinki

Tyynenmerenkatu, 9 Helsinki

Reset Add Destination Calculate

Distance: 4124m  
Time Length: 00:27:41  
ETA: 12:30:55  
241m

6: Kallion virastotalo - Rautatieasema  
4 Stops  
12:08:00 - 12:16:00

7: Rautatieasema - Huutokonttori  
5 Stops  
12:17:00 - 12:26:00

Shortest Quiet

05/05/2019 12:03

Kolmas linja, 17 Helsinki

Tyynenmerenkatu, 9 Helsinki

Reset Add Destination Calculate

Distance: 3885m  
Time Length: 00:49:32  
ETA: 12:52:46

Shortest Quiet

05/05/2019 12:03

Kolmas linja, 17 Helsinki

Tyynenmerenkatu, 9 Helsinki

Reset Add Destination Calculate

Distance: 3908m  
Time Length: 00:49:48  
ETA: 12:53:02



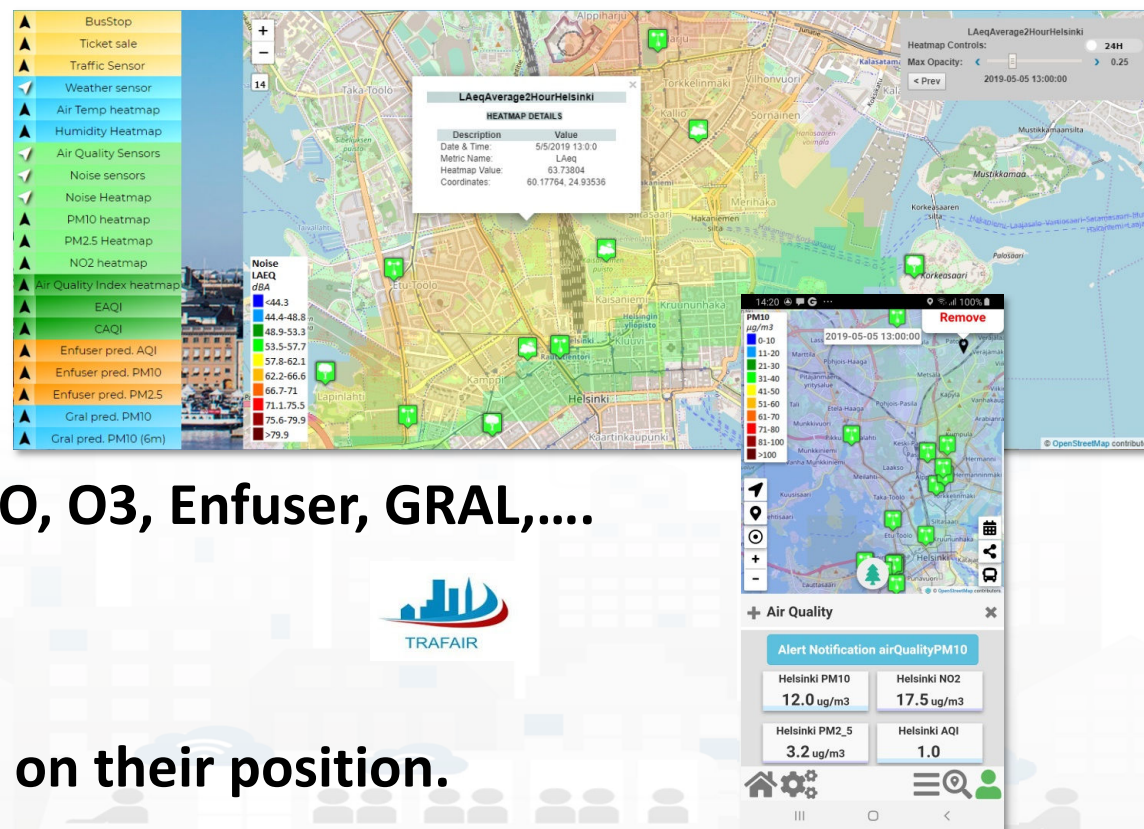
# *Environmental Data: Predictions, Early Warning*





# Data Analytics: Heatmaps

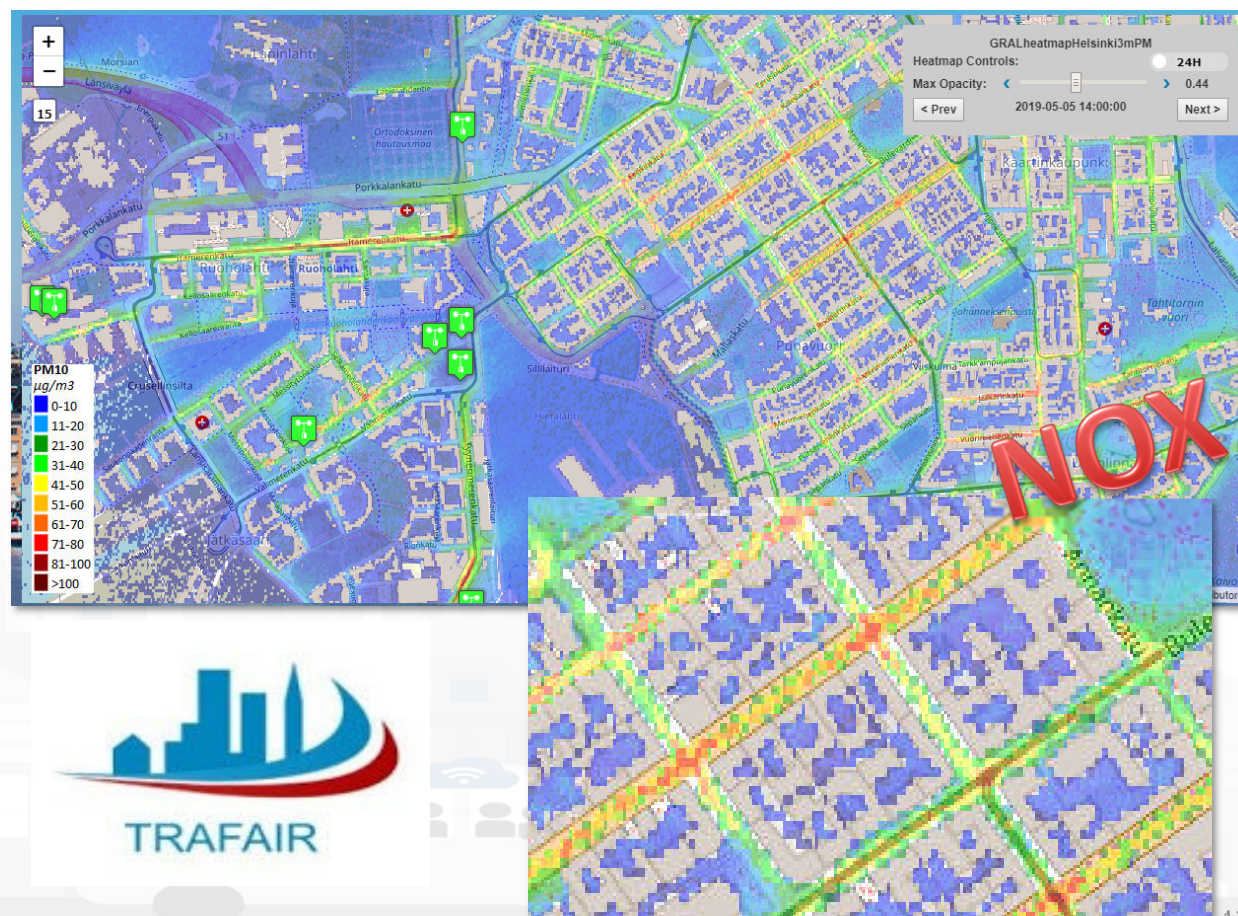
- Over the Gaussian Heatmaps
- Calibrated heatmaps on the basis of Interpolated data for:
  - From 200x200 to 4x4 mt
  - PM10, PM2.5, SO2, NO2, Noise, NO, O3, Enfuser, GRAL,....
  - Any programmed Color map
  - Animations over H24
  - Picking values in any place, values on their position.
  - On Web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





# 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

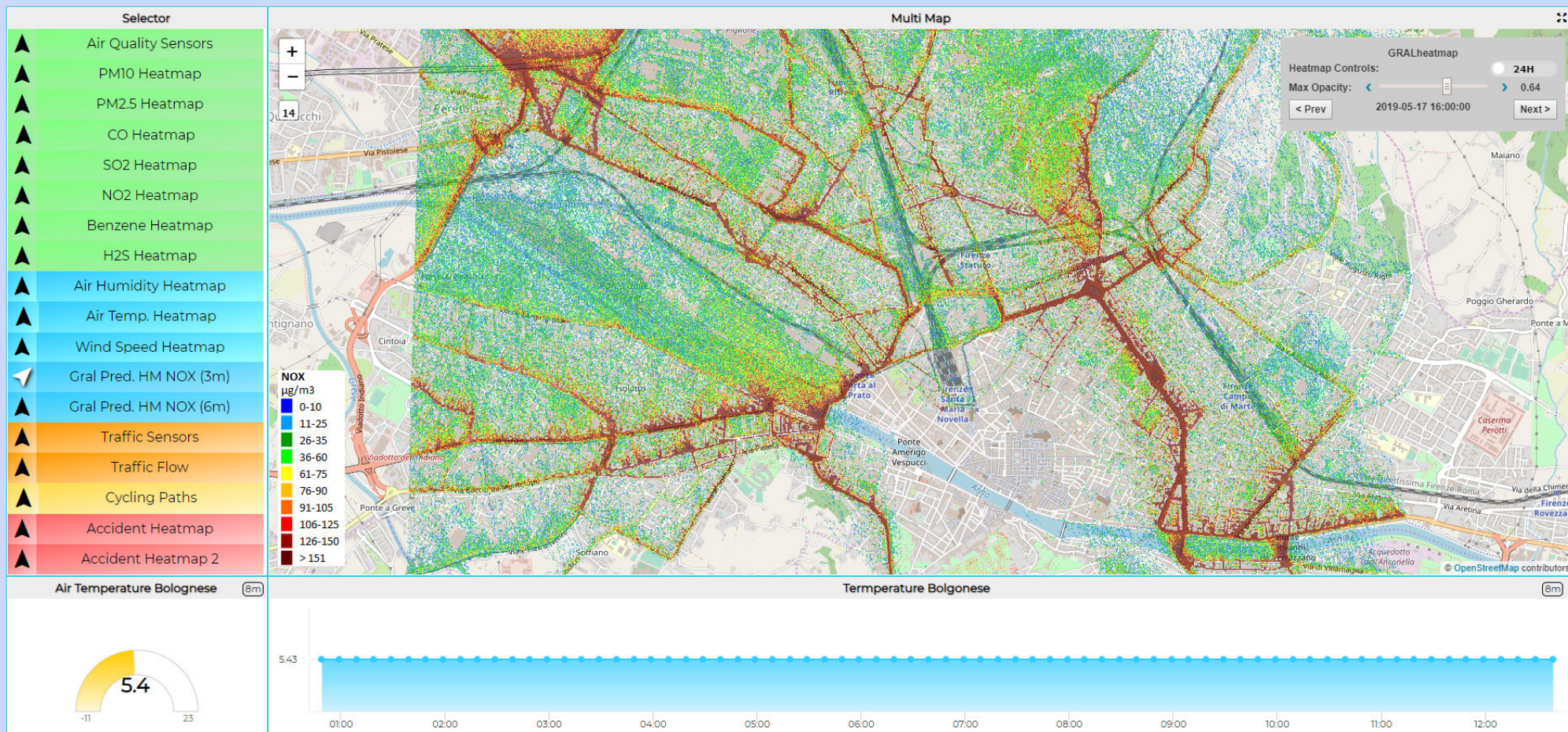




## Heatmap Firenze - trafair

different data

Fri 17 May 12:49:34



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



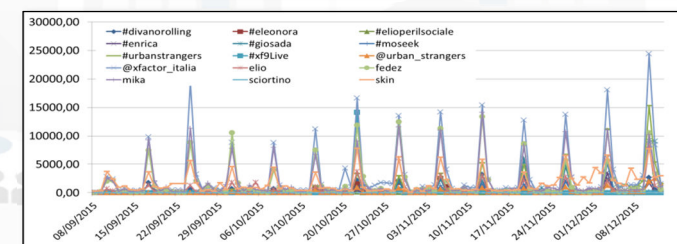
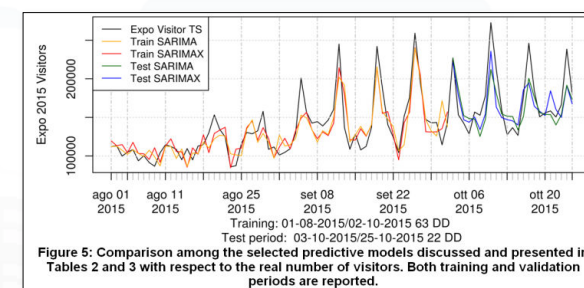
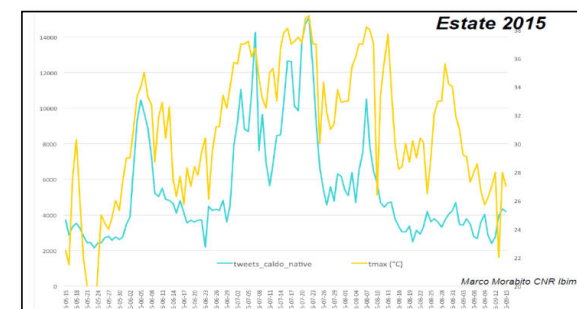
# *Social Media Analysis: Early Warning, Predictions*





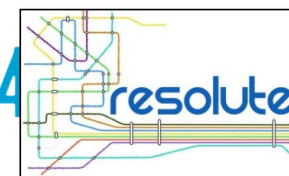
## Prediction/Assessment

- 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



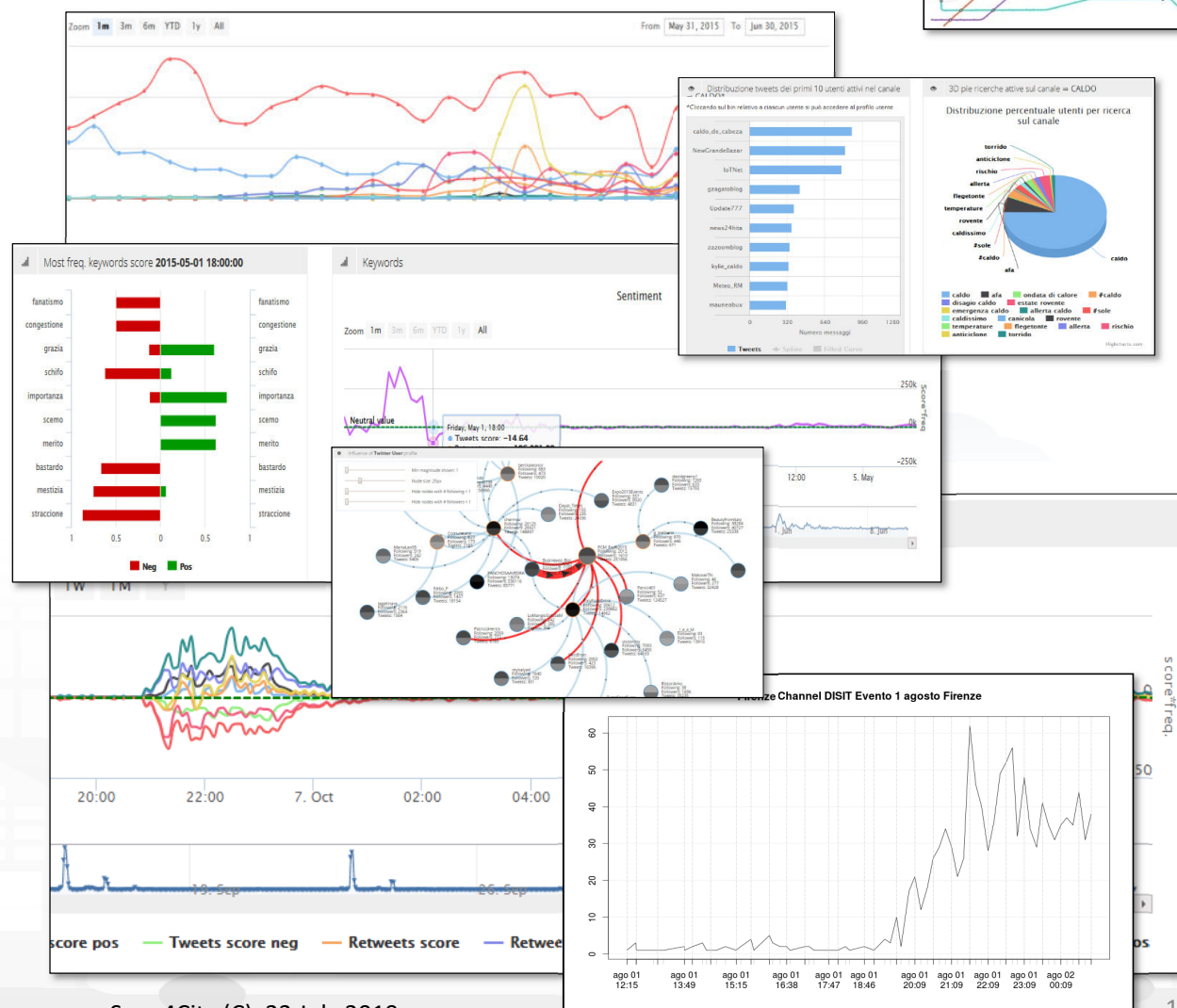


# Twitter Vigilance SNAP4



- <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**



Snap4City (C), 23 July 2019



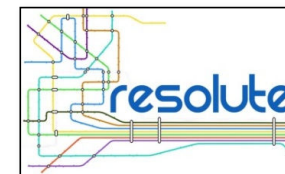


UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

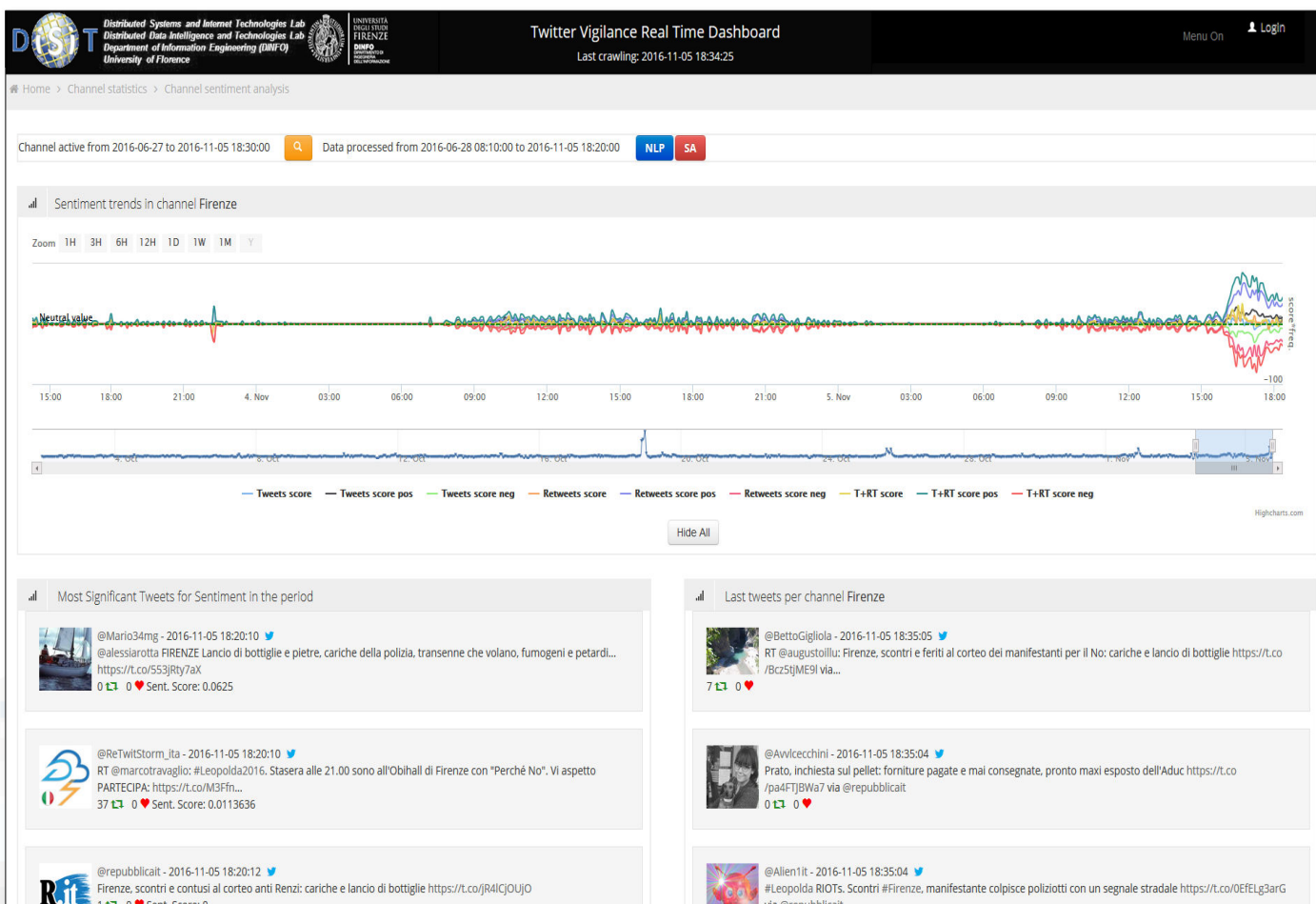
**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB

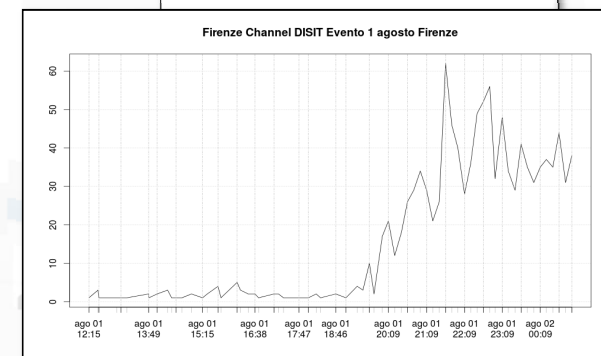
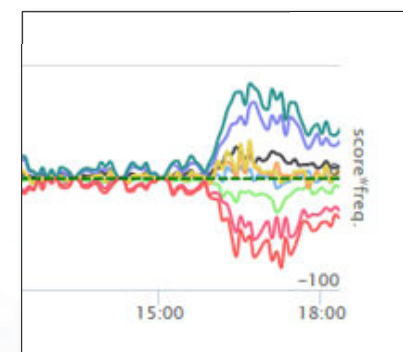
# Twitter Vigilance



## Twitter Vigilance RT: sentiment analysis



## Real time Early Warning



Snap4City (C), 23 July 2019



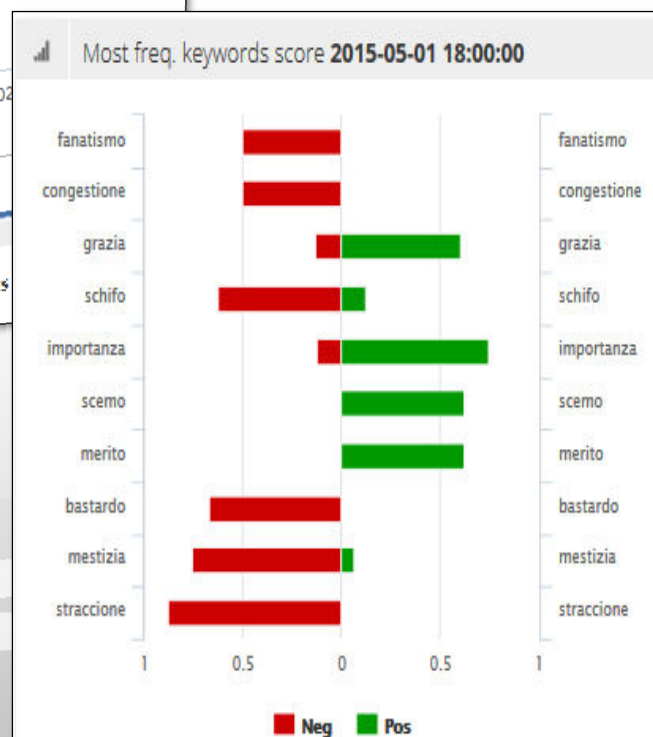
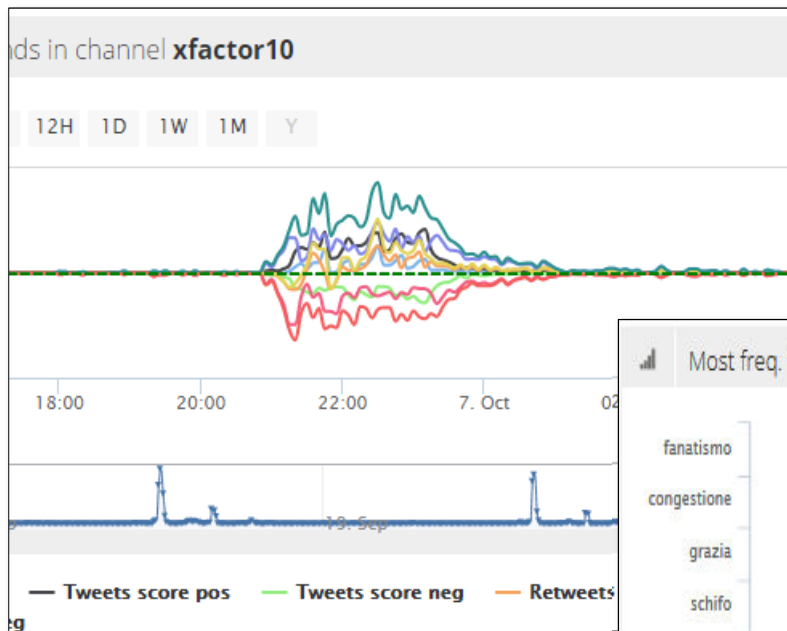


UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB  
<http://www.disit.org>

# Sentiment Analysis



**Twitter Vigilance**





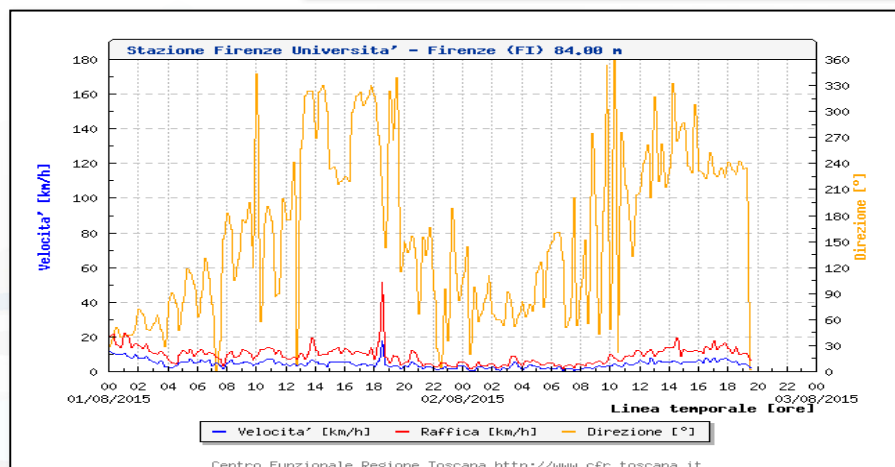
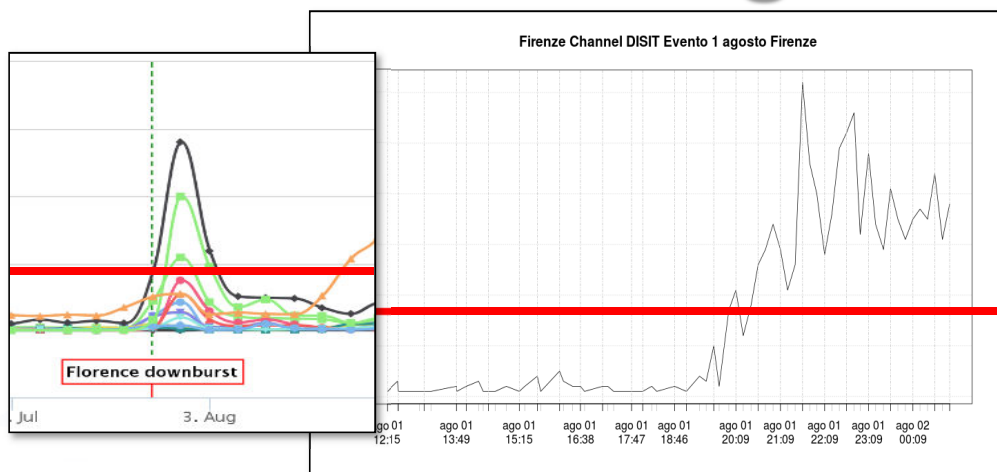
UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

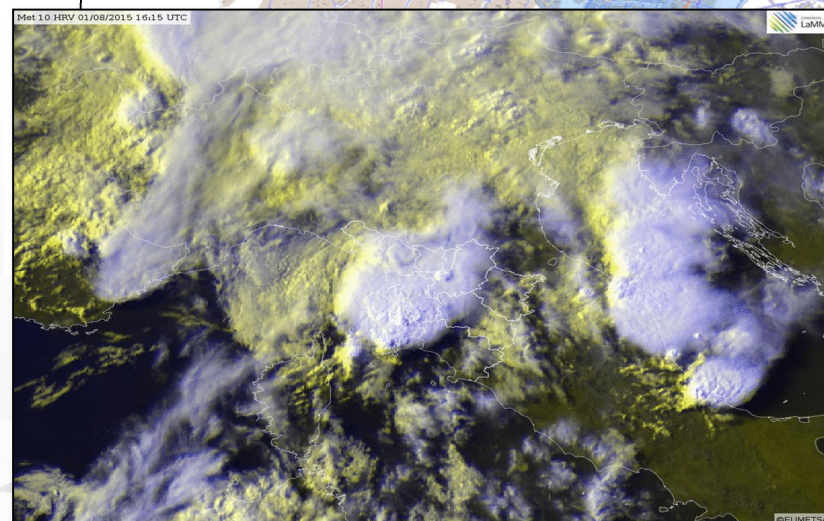
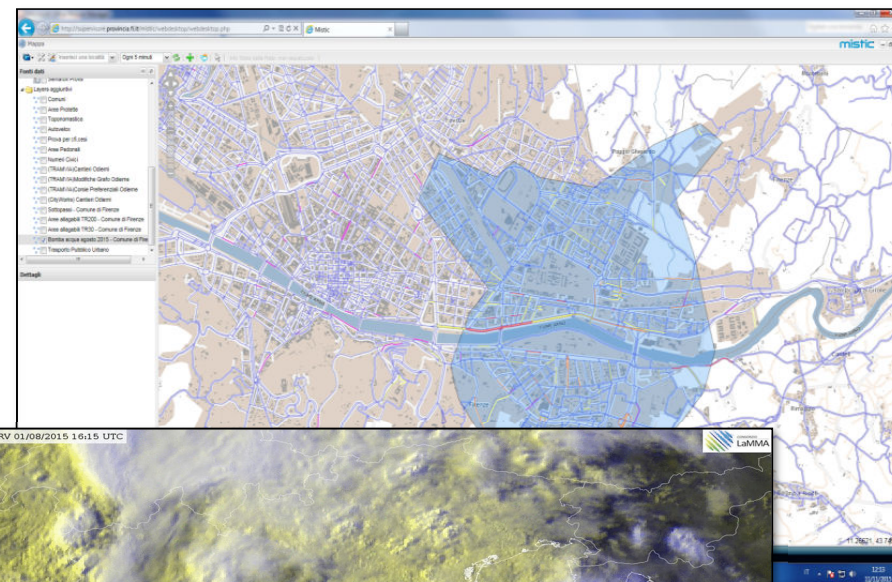
**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB

<http://www.disit.org>

# Twitter Vigilance and Water Bomb



**Twitter Vigilance**



Snap4City (C), 23 July 2019



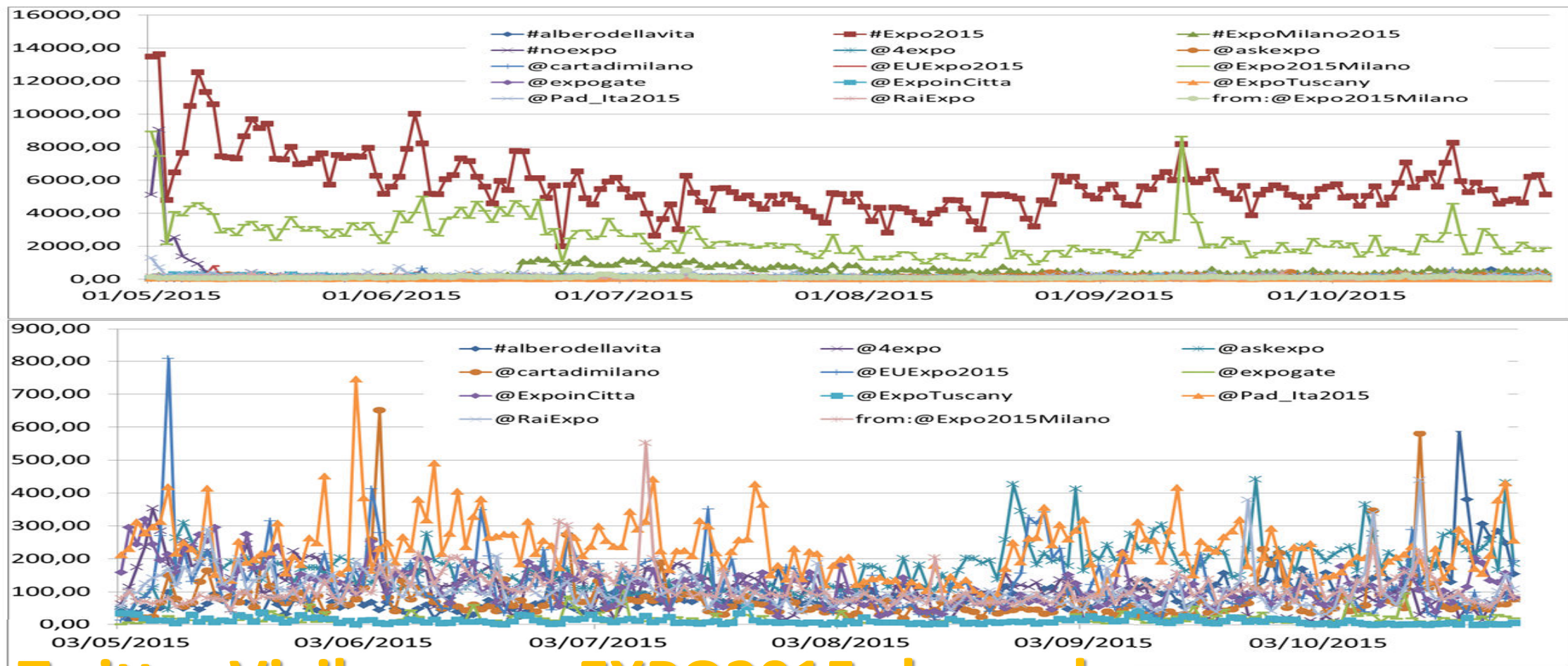


UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB

# Predicting EXPO2015



## Twitter Vigilance on EXPO2015 channel

**Twitter Vigilance**

Snap4City (C), 23 July 2019





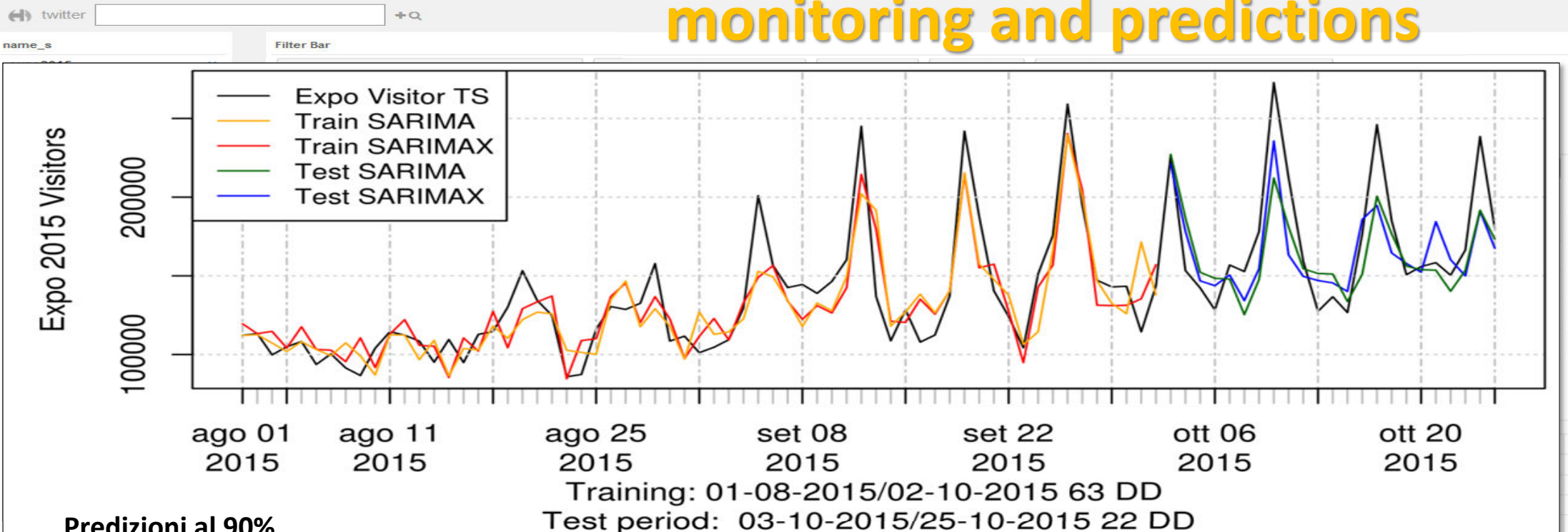
UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**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

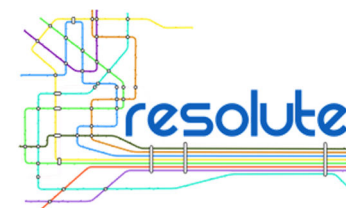
## Case Study B2

Predicting volume of visitors for tuning the services

**Twitter Vigilance**

dozens of cars burned down during #noexpo protest in #milan <http://t.co/ltacp8mpkq> <http://t.co/llsgtqtpjt>  
rt @aut\_omnia: black bloc used smoke bombs to blind cops, then changed clothes, dropped gear and slipped into crowd. genius. #noexpo <http://t.co/2972qxocq>  
rt @maurubiani: #noexpo black bloc #noexpo3 grazie, viana per @illmanifesto <http://t.co/ni8elmfn0>





# 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>



# Data Analytic: Enforcing and Exploiting

FROM CITY  
DASHBOARD TO  
APPLICATIONS

DATA GATHERING  
AND C  
KNOWLEDGE  
MANA

FORGING &  
MANAGING OPEN  
AND FLEXIBLE WEB  
AND MOBILE APPS

IOT/IOE DEVICES  
AND NETWORKS

IOT APPLICATIONS,  
THE LOGIC AND  
THE SMARTNESS

ADVANCING  
SMART CITY API,  
MICROSERVICES,  
SNAP4CITY API

SNAP4CITY  
LIVING LAB FOR  
COLLABORATIVE  
WORK

SNAP4CITY FOR  
BEGINNERS

DATA ANALYTICS,  
BUSINESS  
INTELLIGENCE,  
WHAT-IF AND  
PULSION

SNAP4CITY  
ARCHITECTURE AND  
ECOSYSTEM. OPENED  
TO DEVELOPERS  
AND STAKEHOLDERS

TWITTER  
VIGILANCE, SOCIAL  
MEDIA ANALYSIS

HOW TO ADOPT  
SNAP4CITY, AND  
OUR ROADMAP

SNAP4CITY  
AND KM4CITY  
PROJECTS

SNAP4CITY THE  
VIEW OF THE  
ADMINISTRATORS





UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB

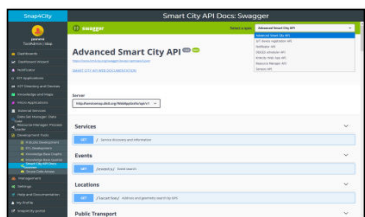


**SNAP4CITY**

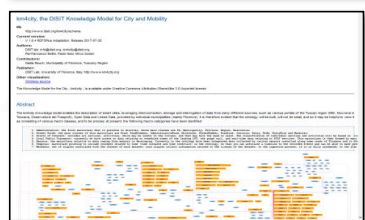
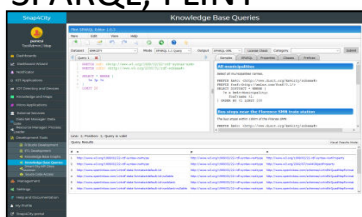


# Data Analytics Dev. in R Studio and/or Tensor Flow

Swagger



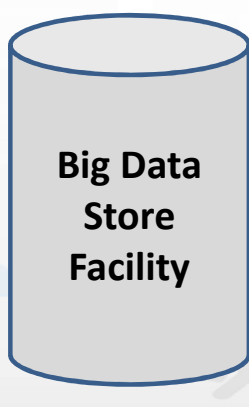
SPARQL, FLINT



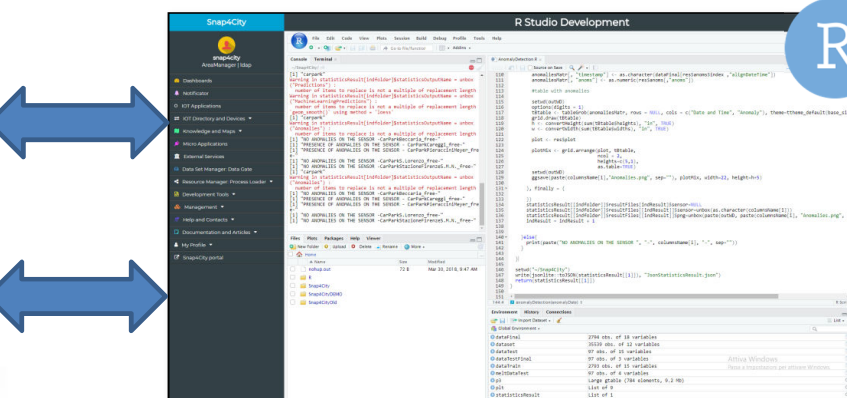
Ontology Schema



LOG.disit.org



Smart City API from Knowledge Base and other tools



R Studio®



Creating  
MicroServices



Using them into  
IOT Applications

Saving /  
Sharing  
reusing



Resource Manager



Snap4City (C), 23 July 2019



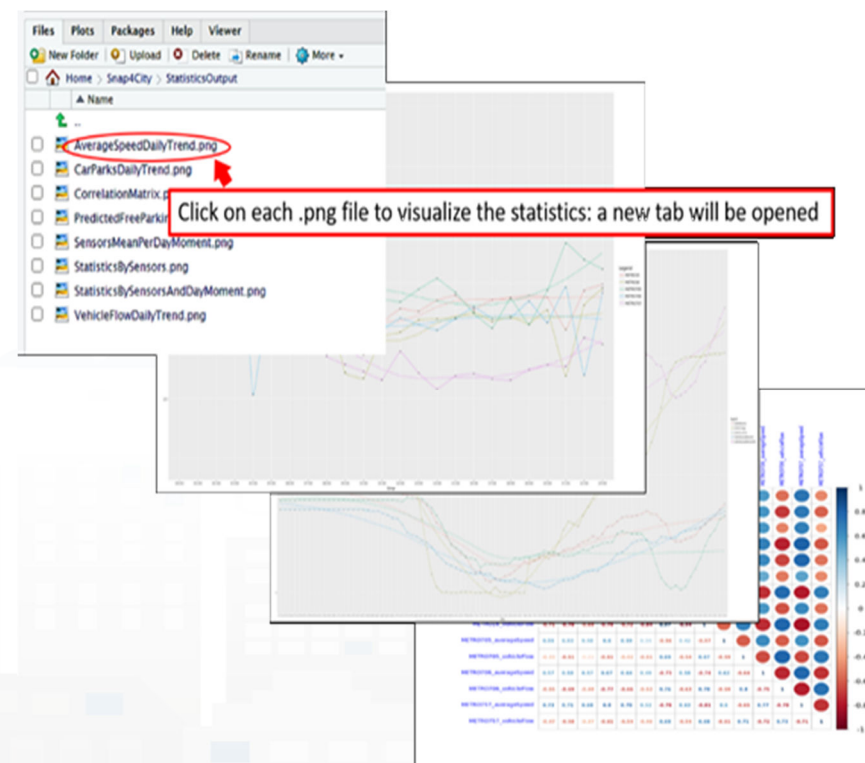
# Developer in R Studio + Tensor Flow

R Studio Development

```

110 anomaliesMat[, "timestamp"] <- as.character(dataFinal$res$anoms$Index, "alignDateTime")
111 anomaliesMat[, "anoms"] <- as.numeric(res$anoms[, "anoms"])
112
113 #table with anomalies
114
115 setwd(outDir)
116 options(digits = 1)
117 ttable <- table2rob(anomaliesMat, rows = NULL, cols = c("Date and Time", "Anomaly"), theme=theme_default(base_size=12))
118 grid.draw(ttable)
119 h <- convertHeight(sum(ttable$heights), "in", TRUE)
120 w <- convertWidth(sum(ttable$widths), "in", TRUE)
121
122 plot <- res$plot
123
124 plotMix <- grid.arrange(plot, ttable,
125                          ncol = 2,
126                          heights=c(5,1),
127                          as.table=TRUE)
128
129 setwd(outDir)
130 ggsave(paste(columnsName[i], "Anomalies.png", sep=""), plotMix, width=22, height=h+5)
131
132 }, finally = {
133
134   statisticsResult[[indFolder]]$resultFiles[indResult]$sensor=NULL
135   statisticsResult[[indFolder]]$resultFiles[indResult]$sensor=unbox(as.character(columnsName[i]))
136   statisticsResult[[indFolder]]$resultFiles[indResult]$png=unbox(paste(outDir, paste(columnsName[i], "Anomalies.png", sep=""), indResult = indResult + 1)
137
138
139 }
140
141 }
142
143
144
145
146 setwd("~/Snap4City")
147 write(jsonlite::toJSON(statisticsResult[[1]]), "jsonStatisticsResult.json")
148 return(statisticsResult[[1]])
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

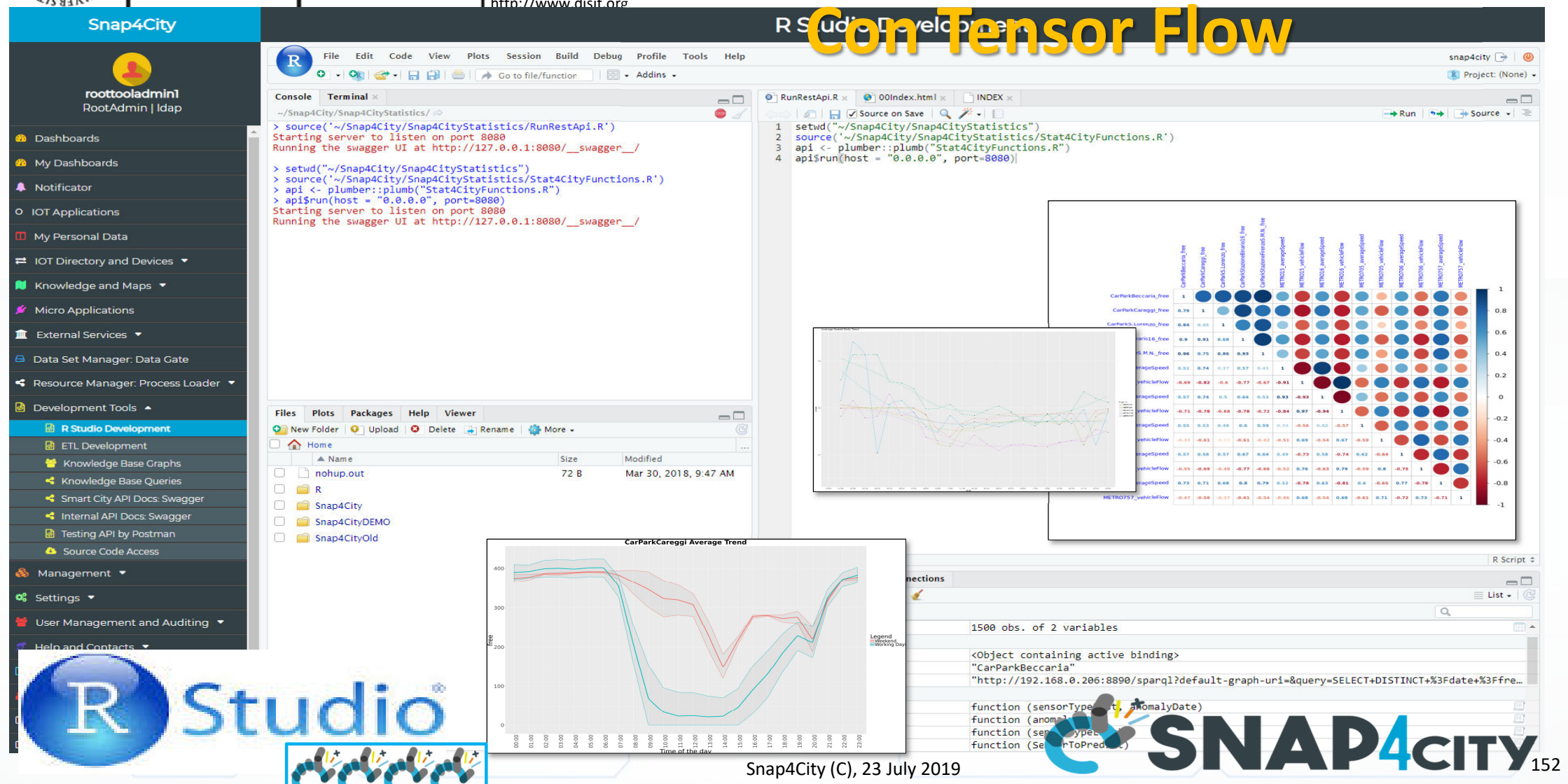




# Data Analytics in R Studio

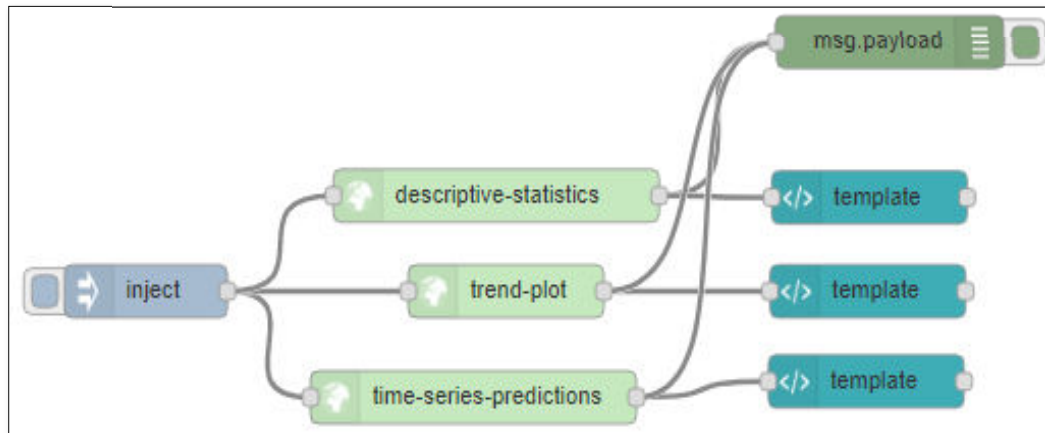
## R Studio Development

### Con Tensor Flow

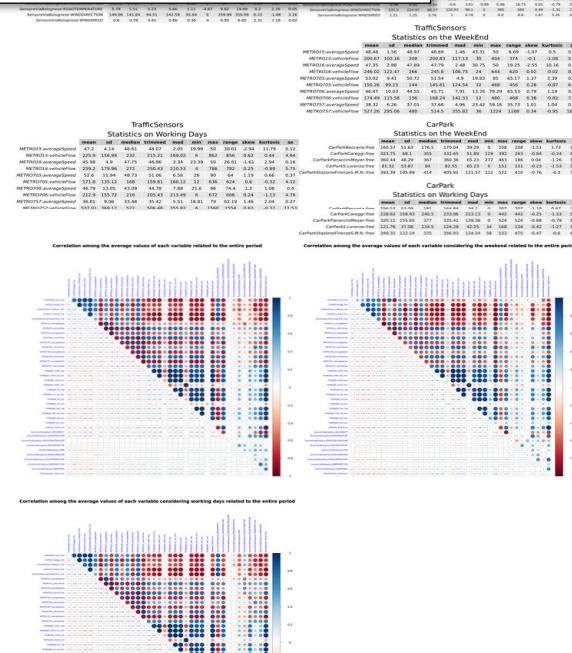




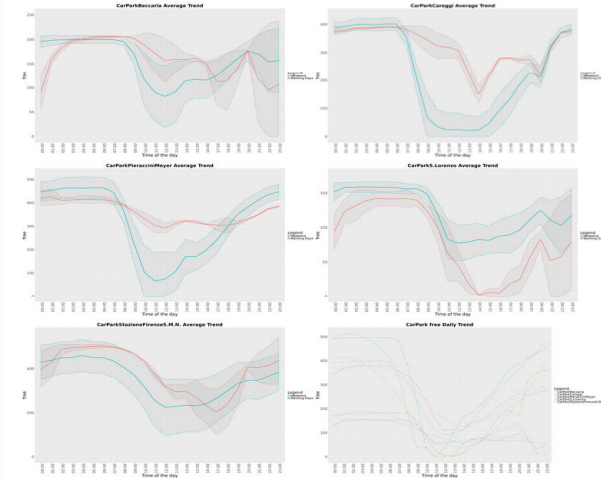
# From R studio data analytics to MicroService



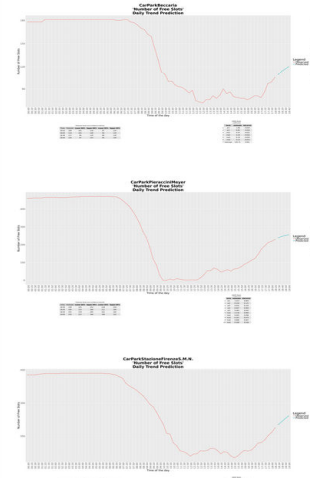
R Studio algorithms are automatically transformed into MicroServices for your IOT Applications



Trend Plot



Time Series Predictions







# Developing in R Studio and/or Tensor Flow

**SNAP4CITY**

AreaManager | Idap

- Dashboards
- Notifier
- IOT Applications
- IOT Directory and Devices
- Knowledge and Maps
- Micro Applications
- External Services
- Data Set Manager: Data Gate
- Resource Manager: Process Loader
- Development Tools
- Management
- Help and Contacts
- Documentation and Articles
- My Profile
- SNAP4CITY portal

**R Studio Development**

```

110 anomaliesMat[, "timestamp"] <- as.character(dataFinal[resAnoms[, "alignDateTime"])
111 anomaliesMat[, "anoms"] <- as.numeric(resAnoms[, "anoms"])
112
113 #table with anomalies
114
115 setwd(outDir)
116 options(digits = 1)
117 ttable <- tableRob(anomaliesMat, rows = NULL, cols = c("Date and Time", "Anomaly"), theme=theme_default(base_size=
118 grid.draw(ttable)
119 h <- convertHeight(sum(ttable$heights), "in", TRUE)
120 w <- convertWidth(sum(ttable$widths), "in", TRUE)
121
122 plot <- resPlot
123
124 plotMtx <- grid.arrange(plot, ttable,
125 hcol = 2,
126 heights=c(5,1),
127 as.table=TRUE)
128
129 setwd(outDir)
130 ggsave(paste(columnName[i], "Anomalies.png", sep=""), plotMtx, width=22, height=h*5)
131
132 }, finally = {
133
134 statisticsResult[[indFolder]]$resultFiles[indResult]$sensor=NULL
135 statisticsResult[[indFolder]]$resultFiles[indResult]$sensor=unbox(as.character(columnName[i]))
136 statisticsResult[[indFolder]]$resultFiles[indResult]$png=unbox(paste(outDir, paste(columnName[i], "Anomalies.png", sep=""),
137 indResult = indResult + 1
138
139
140 }
141
142 else{
143 print(paste("NO ANOMALIES ON THE SENSOR ", "-", columnName[i], "-", sep=""))
144
145
146 }
147
148 setwd("/SNAP4CITY")
149 write(jsonlite::toJSON(statisticsResult[[1]]), "jsonStatisticsResult.json")
150 return(statisticsResult[[1]])
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Environment | History | Connections

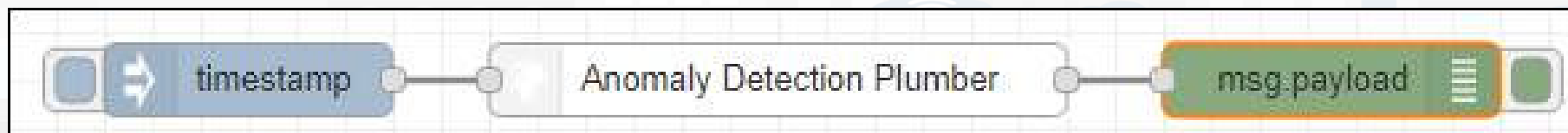
Object	Class	Size
dataFinal	data.frame	2794 obs. of 18 variables
dataset	data.frame	35539 obs. of 12 variables
dataTest	data.frame	97 obs. of 15 variables
dataTestFinal	data.frame	97 obs. of 3 variables
dataTrain	data.frame	2795 obs. of 15 variables
metaDataTest	data.frame	97 obs. of 4 variables
p3	list	large ttable (784 elements, 9.2 Mb)
plt	list	list of 9
statisticsResult	list	list of 1

Files | Plots | Packages | Help | Viewer

Home > Snap4City > StatisticsOutput

- AverageSpeedDailyTrend.png
- CarParksDailyTrend.png
- CorrelationMatrix.png
- PredictedFreeParks.png
- SensorsMeanPerDayMoment.png
- StatisticsBySensors.png
- StatisticsBySensorsAndDayMoment.png
- VehicleFlowDailyTrend.png

Click on each .png file to visualize the statistics: a new tab will be opened







UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

**DINFO**  
DIPARTIMENTO DI  
INGEGNERIA  
DELL'INFORMAZIONE

**DISIT**  
DISTRIBUTED SYSTEMS  
AND INTERNET  
TECHNOLOGIES LAB

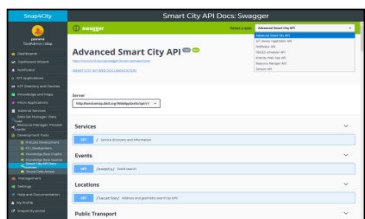


**SNAP4CITY**

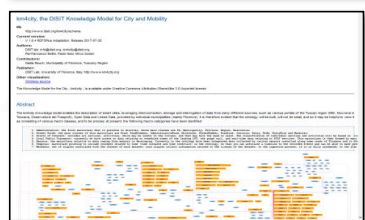
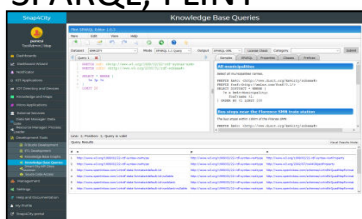


# Data Analytics Dev. in Java, Python, ..

Swagger



SPARQL, FLINT



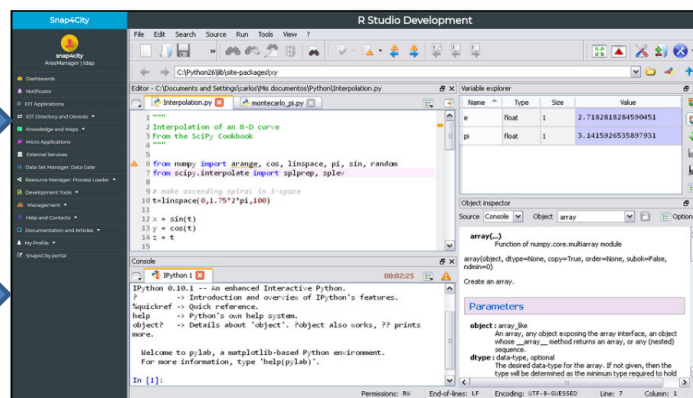
Ontology Schema



LOG.disit.org



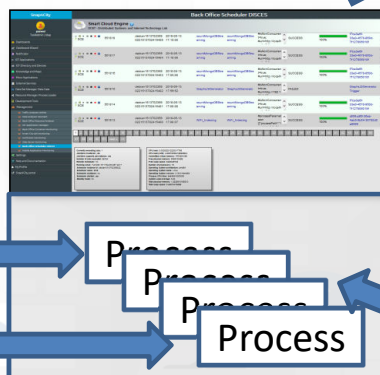
Smart City API from Knowledge Base and other tools



Coding  
Testing



DISCES scheduler



Distributed Back Office

Saving /  
Sharing  
reusing

Monitoring



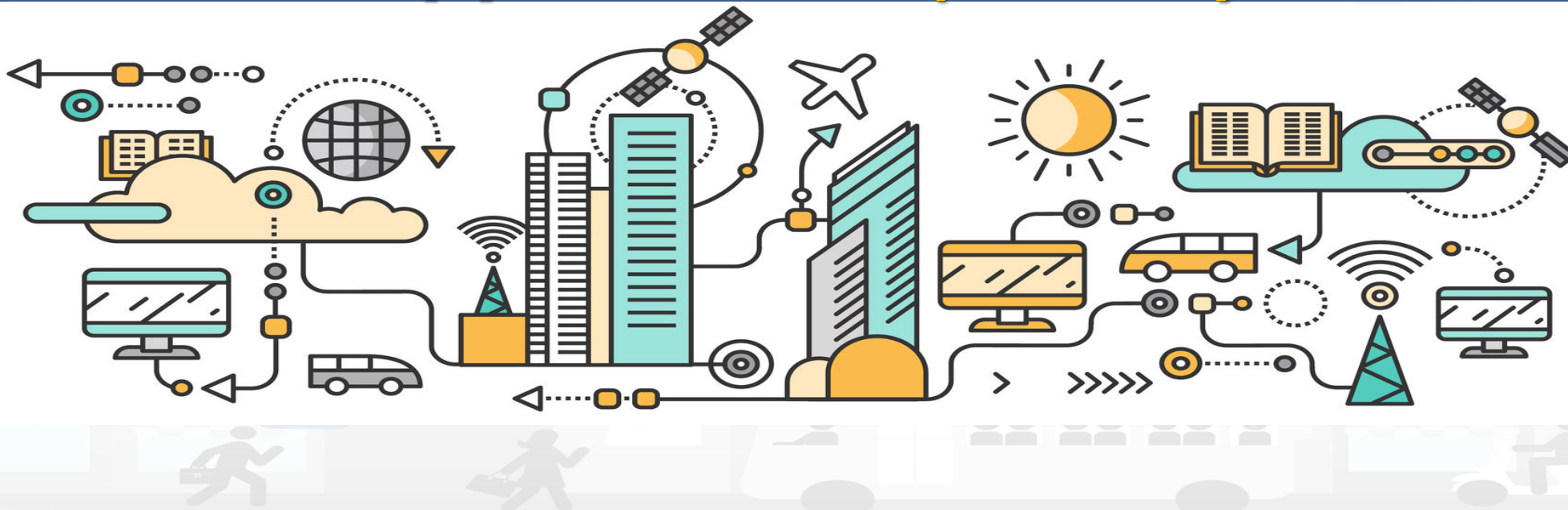
Resource Manager



Snap4City (C), 23 July 2019



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







# Real Time Data Analytics using Studio®

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"
```

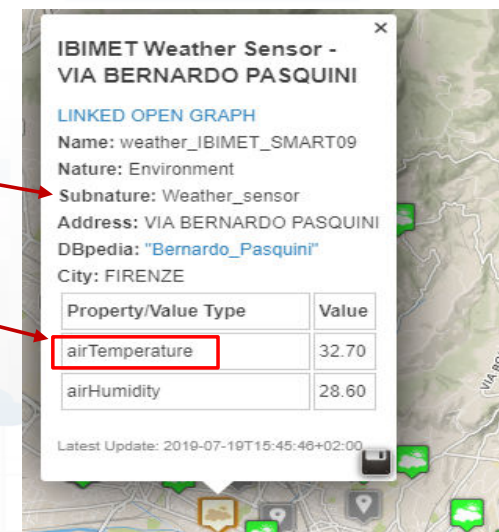
```
sensorCategory = "Weather_sensor"
```

```
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](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=  
42.50247797334869;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=42.67897316354954;9.954032295814045;44.00523270268637;12.063407295814045&categories=Weather\\_sensor&maxResults=0&maxDists=0.1&format=json](https://servicemap.disit.org/WebAppGrafo/api/v1/?selection=42.67897316354954;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/?serviceUri="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",toDateDateTime,sep=" "),
    "clustered"= 0, "latitude"=lat, "longitude"=long,
    "value"= meanObs, "date"= paste(toDateDateTime, "Z", sep=""), "org"="DISIT")

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





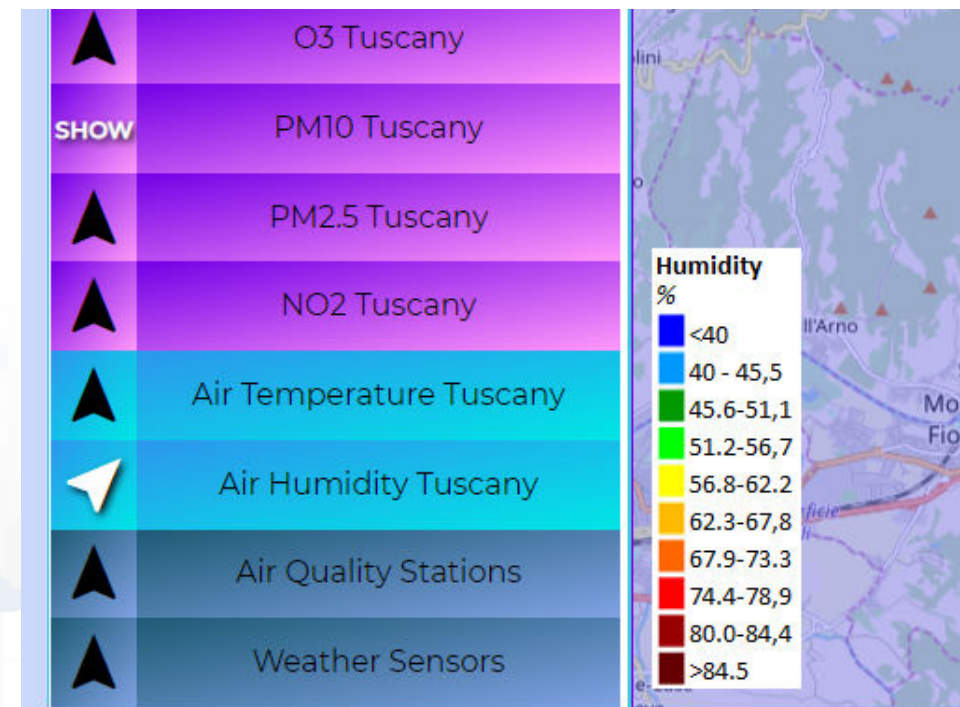
# Real Time Data Analytics using R Studio

## 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









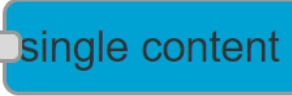

```
[  
{  
  "mapName": "airTemperatureTuscanyTest",  
  "metricName": "airTemperature",  
  "description": " Air Temperature heatmap ... ",  
  "clustered": 0,  
  "latitude": 43.1,  
  "longitude": 11.1,  
  "value": 25.5,  
  "date": "2019-07-23T10:00:00Z"  
  "org": "DISIT"  
}, { [...]} ]
```



# IOT App for Real Time Data Analytics

## How to create a Data Analytics IOT Application

### What we need:

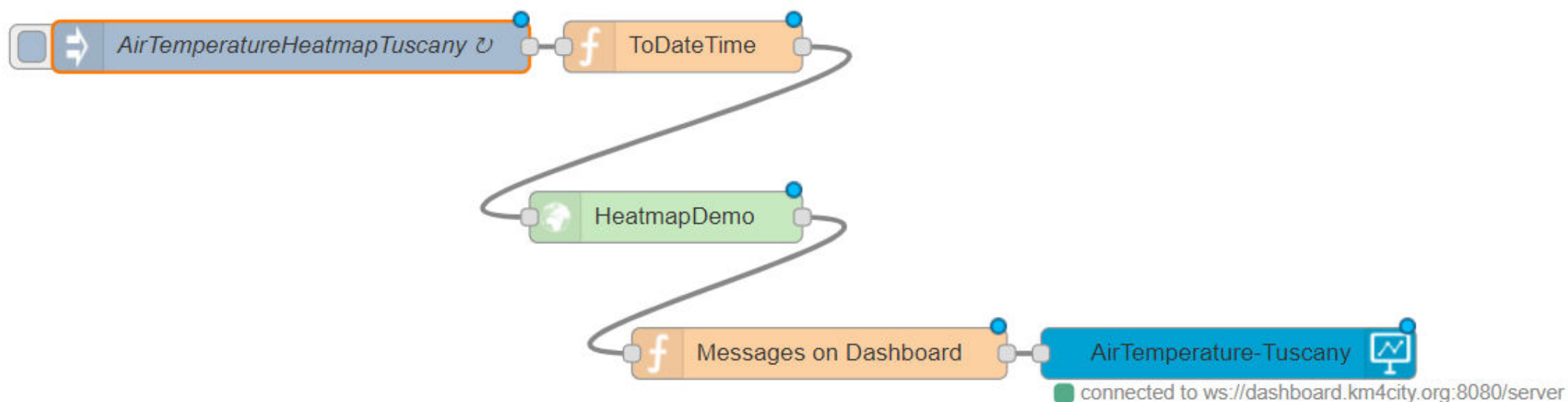
-  inject  To insert the R function parameter
-  plumber data analytic  To upload the R script and create a plumber instance
-  function  To visualize strings/numbers/html on a dashboard
-  single content  To execute JavaScript code on output messages





# 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:

**Edit inject node**

Delete Cancel Done

node properties

**Payload**

**Topic**

**Repeat**

every

☒ Inject once at start?

**Name**

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

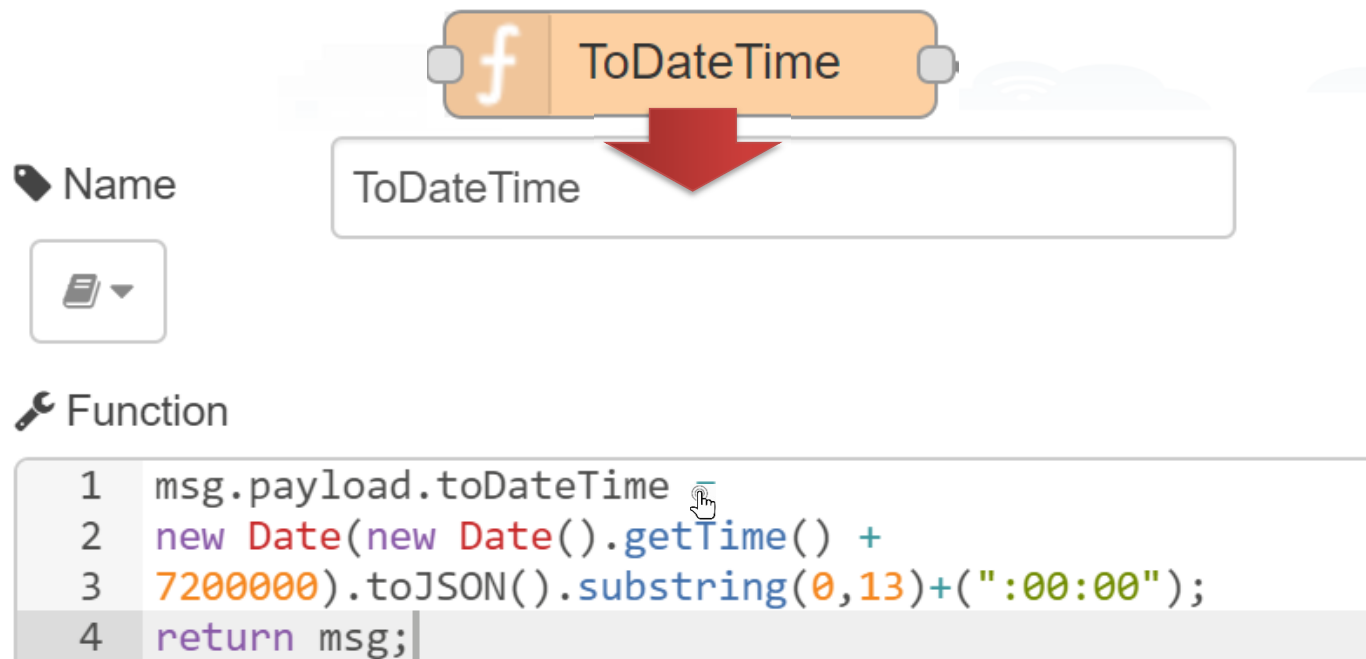
```
{  "varName": "airTemperature",
  "heatmapName":
  "airTemperatureTuscanyTest",
  "fromDateTime": "2-hour",
  "sensorCategory": "Weather_sensor"
}
```



# 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):



The screenshot shows the Node-RED interface with a Function node named "ToDateTime" selected. The node is represented by an orange box with a white 'f' icon and the text "ToDateTime". A red arrow points from the node to a configuration panel below it. The configuration panel has a "Name" field containing "ToDateTime" and a "Function" field containing the following JavaScript code:

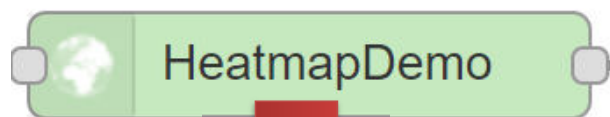
```
1 msg.payload.toDateime
2 new Date(new Date().getTime() +
3 7200000).toISOString().substring(0,13)+(":00:00");
4 return msg;
```



# IOT App for Real Time Data Analytics

## Nodes Configuration – Plumber Data Analytic Node

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



Edit plumber-data-analytic node

Delete Cancel Done

node properties

Name HeatmapDemo

Relative Uri /TuscanyHeatmap

Script R Upload TuscanyHeatmap (3).R

Create Plumber Data Analytic

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



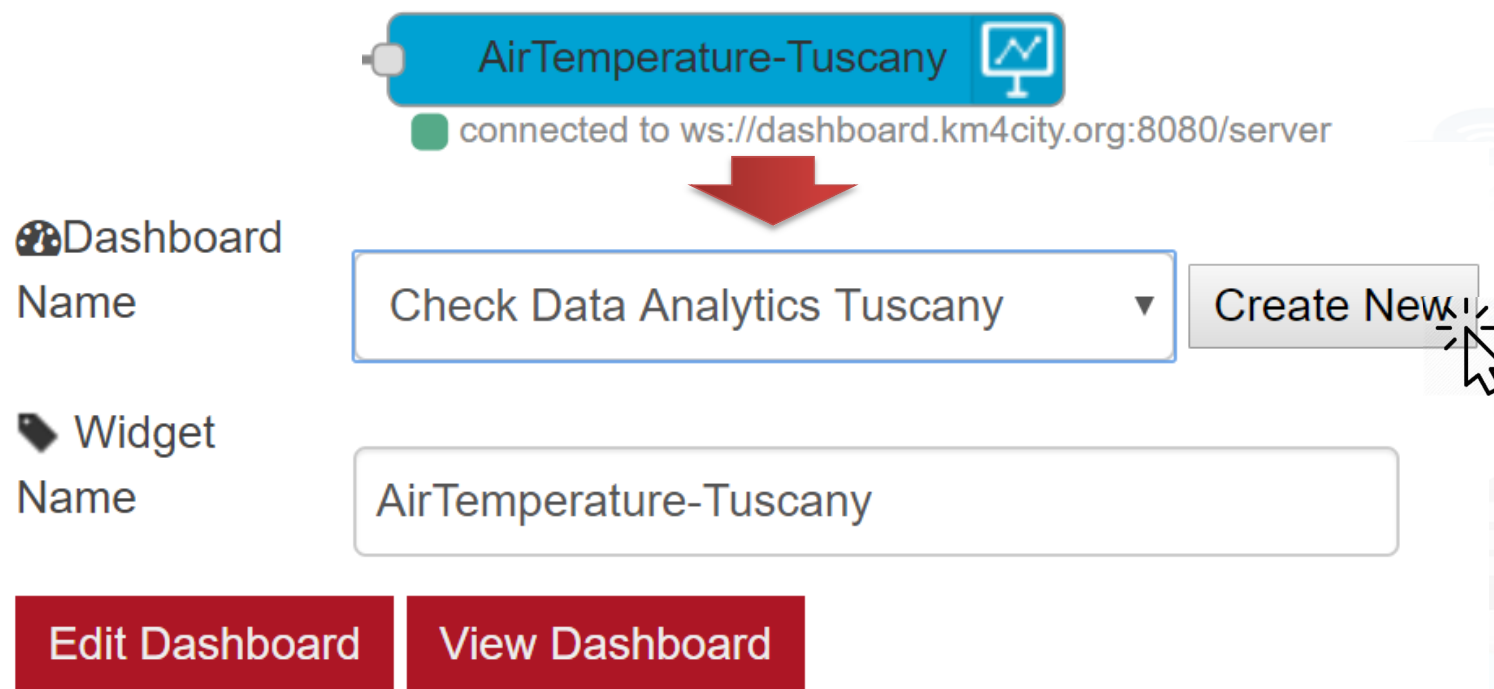
```
1 msg.payload=msg.payload.message+" "+msg.payload.dateTime;
2 if(msg.payload.indexOf("Completed")!= -1){
3     msg.payload ="<span style='color:green;'>"+
i 4     msg.payload + "</span>"
5 } else if (msg.payload.indexOf("No Availabe Data") != -1){
6     msg.payload ="<span style='color:orange;'>"+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



**AirTemperature-Tuscany** 

connected to ws://dashboard.km4city.org:8080/server

**Dashboard**

Name  Create New

**Widget**

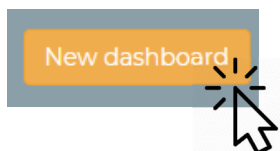
Name

**Edit Dashboard** **View Dashboard**

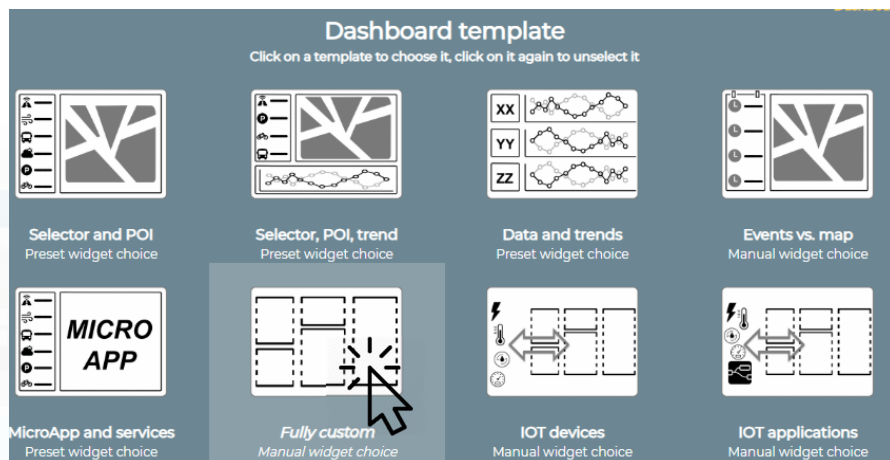


# Wizarded Heatmap Visualization

1. Create a New Dashboard from Dashboard (Public) 



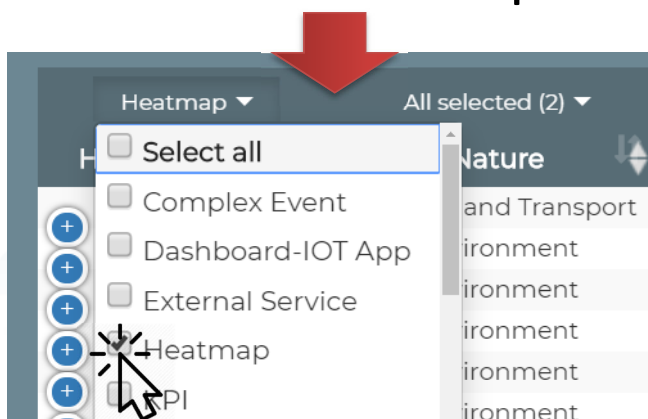
2. Insert a Dashboard Title and select a Dashboard Template



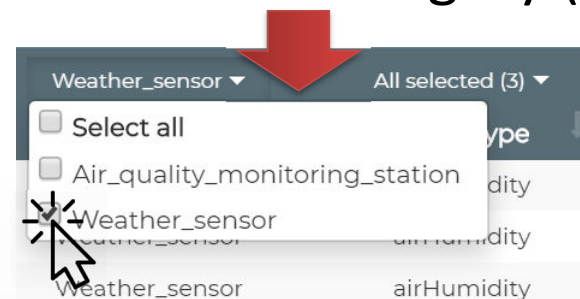


# Wizarded Heatmap Visualization

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)

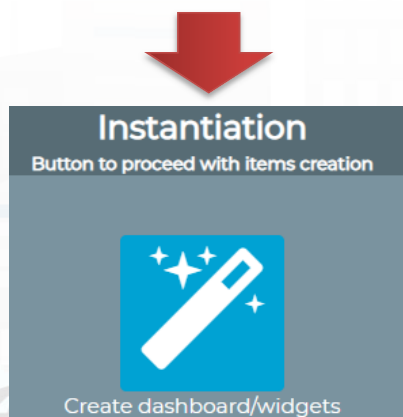
All selected (3) ▾		heatmap ▾	
Value Type	Value Name	Data Type	Last Date
airHumidity	AirHumidityAverage24HourFlorence	heatmap	2019-04-08 13:27:52
airHumidity	AirHumidityAverage2HourFlorence	heatmap	2019-07-22 13:00:00
airHumidity	airHumidityTuscanyTest	heatmap	2019-07-22 12:00:00



# 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



**Data and widgets**

Check and sum

Single data widgets  
Multi data widgets

Map Controls:  
FilterMap GPSUser GPSOrg

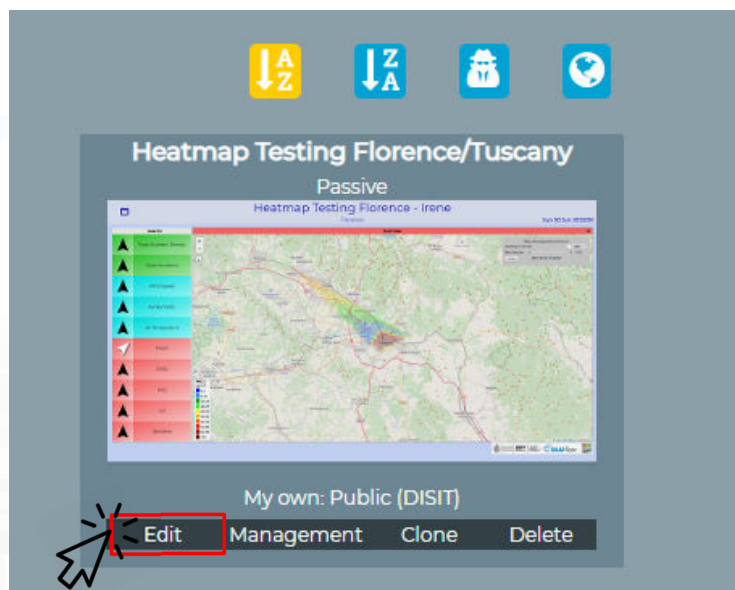
Data sources

Value Type	Value Name	Data Type
airHumidity	AirHumidityAverage24HourFlorence	heatmap
airHumidity	AirHumidityAverage2HourFlorence	heatmap
airHumidity	airHumidityTuscanyTest	heatmap

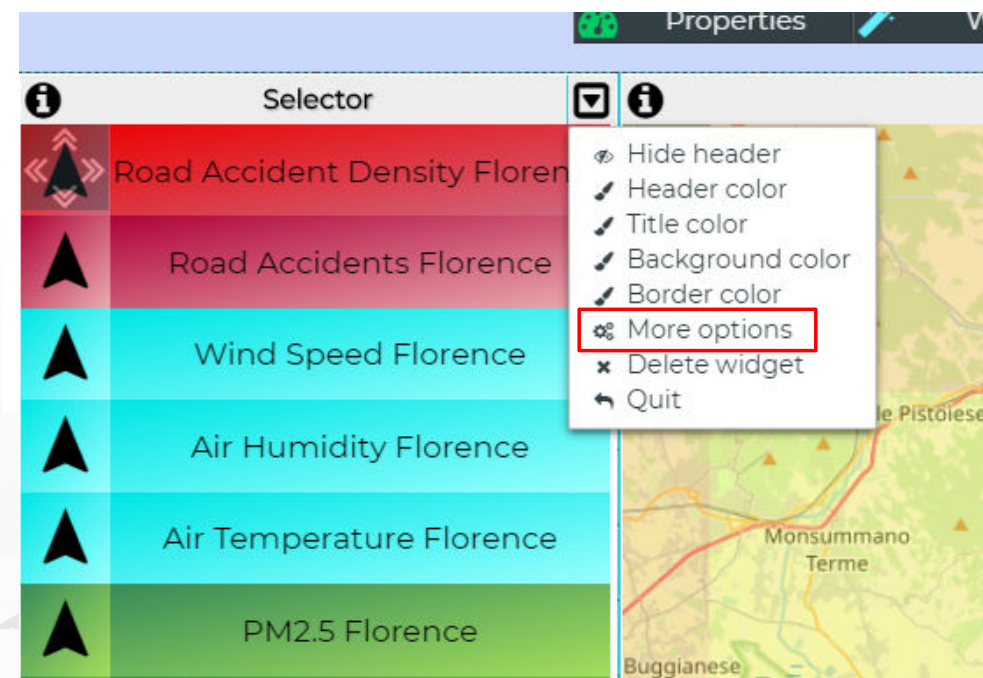


# Manually Heatmap Visualization

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

Specific widget properties

Map widgets Multi Map

Active rows font color rgba(0,0,0,1)

Default	Symbol mode	Symbol choice	Symbol preview	Description	Query	Color1	Color2	Data widgets
<input type="checkbox"/> No	<input type="checkbox"/> Auto		▲	Road Accident...	https://he...	rgba(23, 0, 0, 1)	rgba(20, 0, 0, 1)	Nothing se ▾
<input type="checkbox"/> No	<input type="checkbox"/> Auto		▲	Road Accident...	https://wm...	rgba(17, 0, 0, 1)	rgba(23, 0, 0, 1)	Nothing se ▾
<input type="checkbox"/> No	<input type="checkbox"/> Auto		▲	Wind Speed Fl...	https://wm...	rgba(0, 255, 0, 1)	rgba(15, 0, 0, 1)	Nothing se ▾

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

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

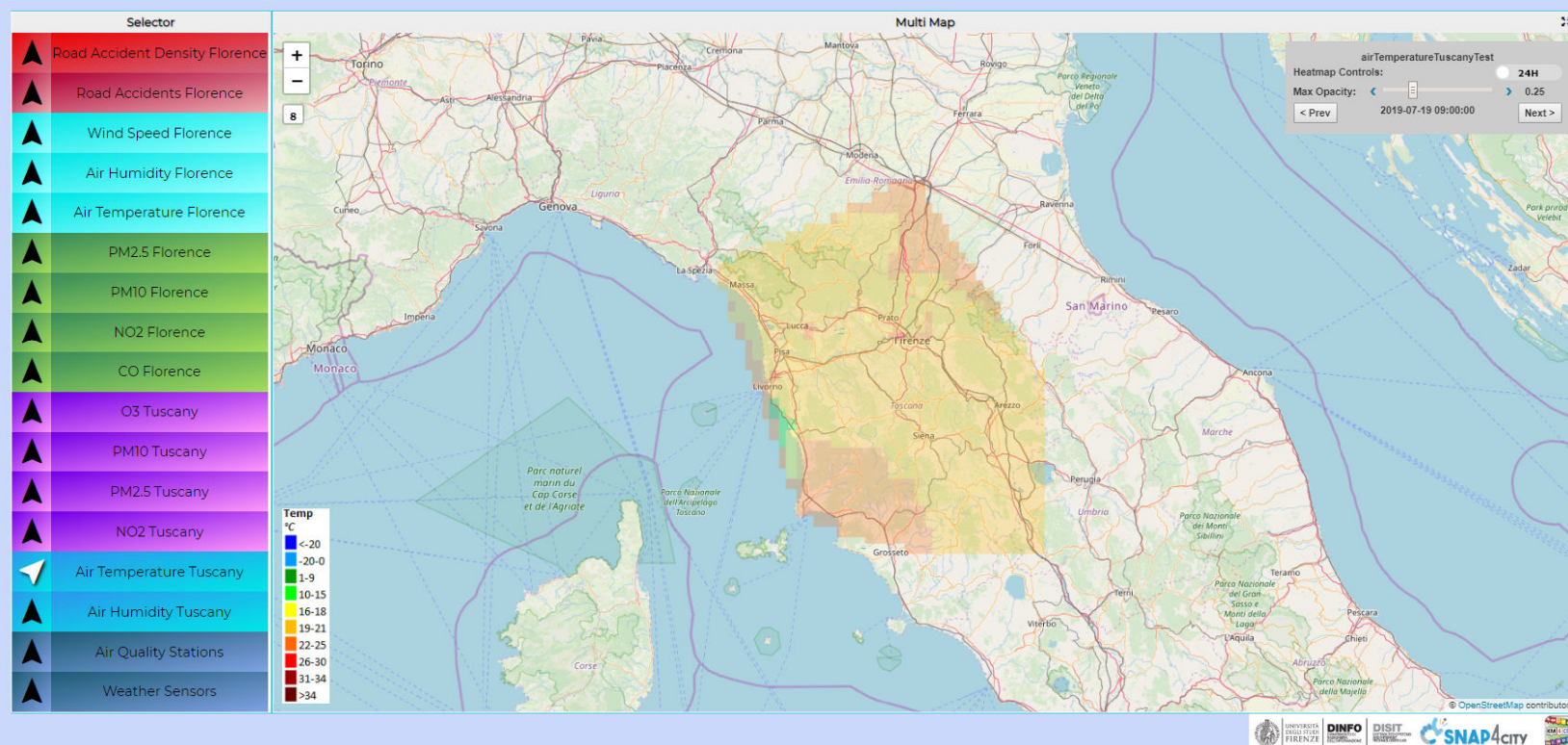


# Heatmap Visualization

## Heatmap Testing Florence/Tuscany

Irene

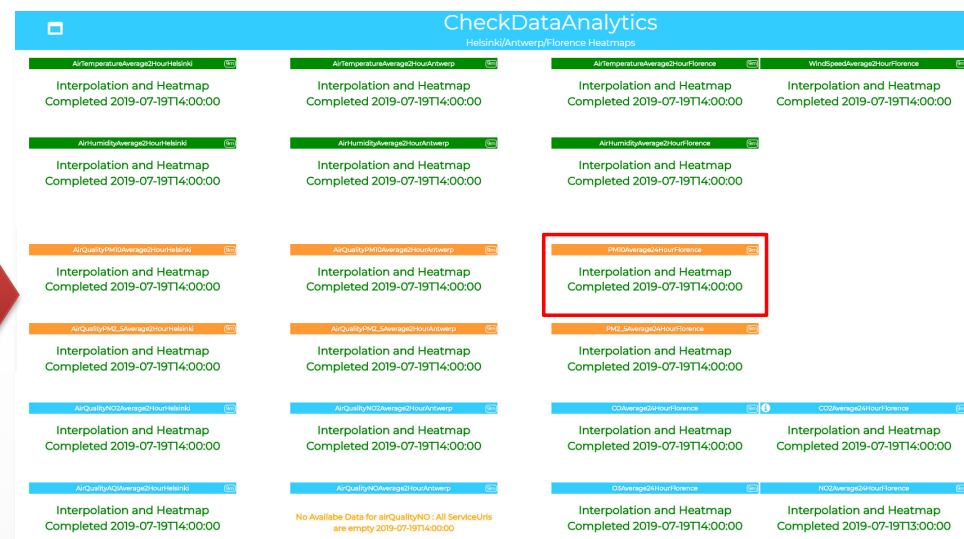
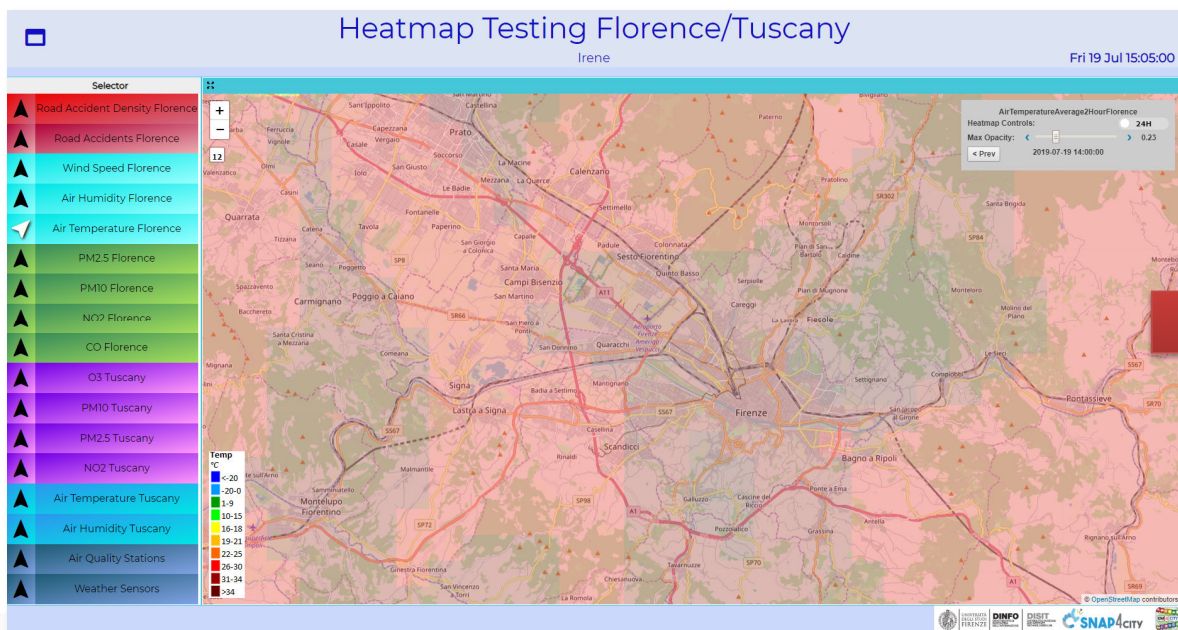
Fri 19 Jul 12:04:21



<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

FROM CITY  
DASHBOARD TO  
APPLICATIONS

DATA GATHERING  
AND CITY DATA  
KNOWLEDGE  
MANAGEMENT

FORGING &  
MANAGING OPEN  
AND FLEXIBLE WEB  
AND MOBILE APPS

IOT/IOE DEVICES  
AND NETWORKS

IOT APPLICATIONS,  
THE LOGIC AND  
THE SMARTNESS

ADVANCED  
SMART CITY API,  
MICROSERVICES,  
SNAP4CITY API

SNAP4CITY  
LIVING LAB FOR  
COLLABORATIVE  
WORK

SNAP4CITY FOR  
BEGINNERS

DATA ANALYTICS,  
BUSINESS  
INTELLIGENCE,  
WHAT-IF AND  
PREDICTION

SNAP4CITY  
ARCHITECTURE AND  
ECOSYSTEM. OPEN  
EVEN FOR 5  
AND 10 YEARS

TWITTER  
VIGILANCE, SOCIAL  
MEDIA ANALYSIS

HOW TO ADOPT  
SNAP4CITY, AND  
OUR ROADMAP

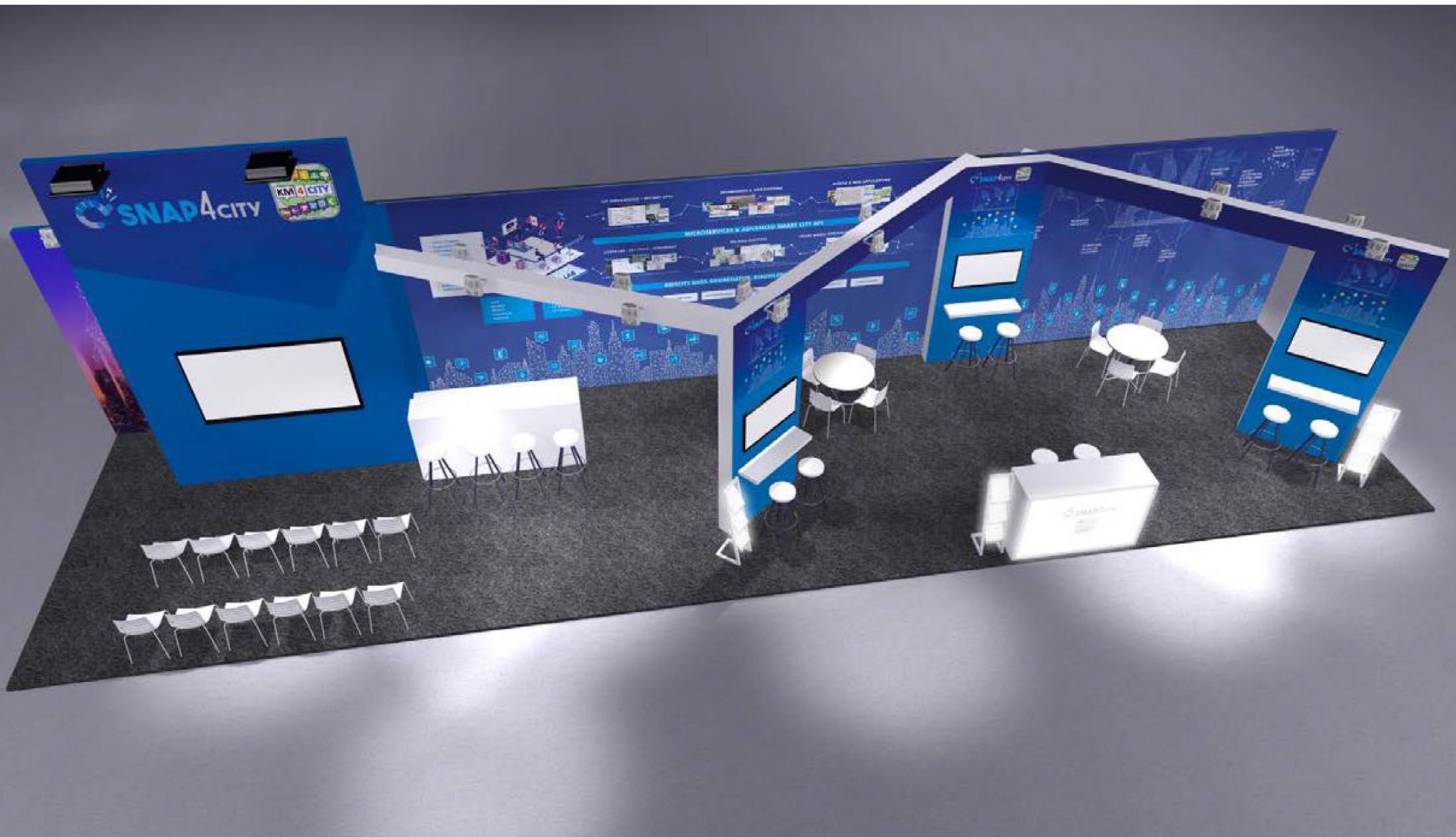
SNAP4CITY  
AND KM4CITY  
PROJECTS

SNAP4CITY THE  
VIEW OF THE  
ADMINISTRATORS











## Acknowledgements

FROM CITY  
DASHBOARD TO  
APPLICATIONS

DATA GATHERING  
AND CITY DATA  
KNOWLEDGE  
MANAGEMENT

FORGING &  
MANAGING OPEN  
AND FLEXIBLE WEB  
AND MOBILE APPS

IOT APPLICATIONS  
VS IOT EDGE  
DEVICES

IOT APPLICATIONS,  
THE LOGIC AND  
THE SMARTNESS

ADVANCED  
SMART CITY API,  
MICROSERVICES,  
SNAP4CITY API

SNAP4CITY  
LIVING LAB FOR  
COLLABORATIVE  
WORK

SNAP4CITY FOR  
BEGINNERS

WHAT-IF AND  
SIMULATION

SNAP4CITY  
ARCHITECTURE AND  
ECOSYSTEM. OPENED  
TO DEVELOPERS  
AND STAKEHOLDERS

DECISION SUPPORT  
SYSTEM AND CITY  
RESILIENCE

TWITTER  
VIGILANCE SOCIAL  
MEDIA ANALYSIS

HOW TO ADOPT  
SNAP4CITY, AND  
OUR ROADMAP

SNAP4CITY  
AND KM4CITY  
PROJECTS

SNAP4CITY THE  
VIEW OF THE  
ADMINISTRATORS



# 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.



INEA CEF-TELECOM Project  
funded by European Union



Horizon 2020  
European Union Funding  
for Research & Innovation



Horizon 2020  
European Union Funding  
for Research & Innovation



MINISTERO DELL'ISTRUZIONE DELL'UNIVERSITA' E DELLA RICERCA







*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](http://www.snap4city.org)

Email: [snap4city@disit.org](mailto: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