

Type of the Paper (Article, Review, Communication, etc.)

# Sii-Mobility: an IOT/IOE Architecture to Enhance Smart City Services of Mobility and Transportation

Claudio Badii <sup>1</sup>, Angelo Difino <sup>1</sup> and Paolo Nesi <sup>1</sup>

<sup>1</sup> University of Florence, Department of Information Engineering,  
DISIT Lab, <https://www.disit.org>, <http://www.sii-mobility.org>, <name.surname>@unifi.it

**Abstract:** The new IoT/IOE (internet of things/everythings) paradigm and architecture permits to rethink about the way the Smart City infrastructures are designed and managed, on the other hand a number of problems have to be solved. In terms of mobility the cities that embrace the *sensing* era can take advantage of this disruptive technology to improve the quality of life of their citizen, also thanks the rationalization in the use of their resources. In Sii-Mobility, a national smart city project on mobility and transportation, a flexible platform has been designed and here, in this paper, is presented. It permits to setup heterogeneous and complex scenarios that integrate sensors/actuators as IoT/IOE in an overall scenario of Big Data, Machine Learning and Data Analytics. A detailed and complex case-study has been presented to validate the solution in the context of a system that dynamically reverse the traveling direction of a road segment, with all the safety conditions in place. This case study composes several building blocks of the IoT platform, which demonstrate that a flexible and dynamic set-up is possible, supporting off-grid, security, safety, cloud and mixed solutions.

**Keywords:** smart city, mobility, IOT Applications, safety critical.

## 1. Introduction

Since the rapidly spread of urbanization in the 1950s across the western world, mobility drove important aspects for growth and progress. At the beginning mobility gave the freedom to the people to move around with new means of transportation, independently from where they live or they work, but today due the strong linkage of traffic and communication system and the fact that the infrastructure is stressed to their limits, it is essential to think about alternative, dynamic routes and paths and a more efficient/optimized transportation system. Smart traffic planning, innovative public transport systems and personal multi-modal routing (for the citizen of city that have already embraced this new paradigm) are central for a complete interconnected infrastructure. To have less traffic jams and near-to-zero emissions with smart means of transportation have been demonstrated to produce relevant impact on the environment and quality of life in smart city, increasing positive virtuous habits from the population. Nowadays, the development of key enabling technologies as Internet of Things (IoT), internet of everything (IoE) are driving even more rapidly the growth of sustainable ecosystem. The switch to IoT/IOE paradigm is going to change (again) dramatically the way the citizens and city operators interact with its nearby infrastructure: how they move, how they get energy, how they make decision, how the city entities are managed and controlled [1].

The development of next generation IoT infrastructure is still at its early stage and main progress are expected to be made in the next years. According to Gartner [2] by 2020 up to 20.4 billion IoT device will be connected together. New tools to support the new paradigm are needed: smart management of the resources, better security for population, healthcare and engagement of the citizen in their everyday activities are some examples of the different scenario that can be unlocked with an embedded IoT ecosystem support [3].

This paper presents the work performed on defining a smart city architecture to enable the integration and implementation of a set of different and heterogeneous scenario for the IoT/IOE paradigm in the context of mobility and transport, where the accent on safety critical aspects is

relevant. Thus, a set of requirements and use cases are discussed and addressed, and thus, a complete experiment is detailed: a smart modality to manage a dynamic management of direction of a one-way road segment. This example employs the use of several modules of the architecture and puts in evidence how an off-grid solution can be implemented beside a complete cloud architecture, also respecting the safety critical constraints. The problem and the solution are explained to catch the value of having a complete platform from the sensing/actuating activities towards a programmable environment where the City Operators can implement their strategies [4], in a fully distributed manner.

## 2. Requirements and architecture

Most smart city solutions target several different aspects of the infrastructure to improve the sustainability of their services [5]. Several use case scenarios have been taken in consideration and some of them belong to very different application's domains, like healthcare management, smart mobility, citizen's security, efficient farming, etc. [6], [7]. In order to enable the City Operator, to introduce smart solutions, which are capable to help the citizens in their daily activities, a set of heterogenous and integrated tools are needed. These solutions have to be capable to support different scenarios in a smart and integrated way, to provide a mechanism to senses the context of the city, to drive the data in a singular or aggregated way, to provide at data scientists a set of tools for data analysis and a programming environment to computer scientist. A smart city platform should be capable to inform the citizens and City Operators, to collect their feedbacks about the city infrastructures and services, to provide an integrated info portal for ad-hoc and personal visual applications (Dashboards) and finally to provide to the platform's admins a way to monitor the functionalities of the solution.

In the context of mobility and transport, in wider view of mobility and transportation aspects of the city, the Intelligent Transport Systems (ITS) should be capable to manage semaphores, dynamic signages, speed control systems, smart parking, car sharing, etc. The revolution of IoT may also impact on mobility and transport applications, in distributed manner. An IoT platform for a smart city must support scenarios that focus on the way the citizen interacts with their infrastructures (in terms of streets, road, cycle path, public transport station, parking, etc.) [8], the way the City Operator of the mobility and transport unit of the city programs, configures the transportation infrastructure and how it is managed/monitored daily.

The main requirements we identified in the above described context are reported as follows. So that the solution has to be capable to:

- collect a big amount and heterogeneous data from the city infrastructure [9] in a continues way and support different ways for data injection (push, pull, on-demand, offline); support off-grid scenario and completely on-cloud architecture;
- assure an architecture that is complete and consistent due to the fact that some scenario can be safety-critical, and thus the communications among IoT elements, and with the Control Room (traffic congestion area, event mass moving, pedestrian cross paths), have to be safe; as well as their implementation has to be capable to overcome the cases of missing communication; in same case real-time or near-real-time safety mechanisms have to be provided [10];
- guarantee that the solution is distributed and scalable with respect to the points of controls, and that they may be capable to continue to work even if the connections with the Control Room, or each other are lost; the Control Rooms may be used to force situation and monitoring;
- guarantee that the data are managed properly, in respect of their ownership and the explicit signed consents they gave; protect for external intrusion and data tampering, man-in-the-middle attack and data sniffing; monitor suspect activities and enable inspection from law enforcement agencies; inform in case of a breach [11];
- store these data in a distributed storage and index them on the basis of different constraints imposed by the domain application [12]; support an efficient way for data retrieval and provide a complete system of backup and disaster recovery;

- provide a programmable environment to define algorithms for machine learning and data analysis, able to extract characteristics from the collected data (aggregation of data) [13] and elaborated them in order to implement predictive capabilities for guessing their future evolution: smart parking, predictions on traffic flow, traffic flow reconstruction;
- enable visual presentations of data via a set of dashboards composed by widget components that a City Operator can manage graphically; these widgets have to be manageable/configurable by a person that may not know deeply the technicisms behind the IoT infrastructure.

In the context of mobility and transport, the data exchanged and managed by IoT devices, toward the infrastructure in which the *local* computing capabilities, can be used via an IoT Edge (IoT device with enhanced capabilities of local computing). So that the features that those devices should be requested to locally address have been identified, and are reported as follows:

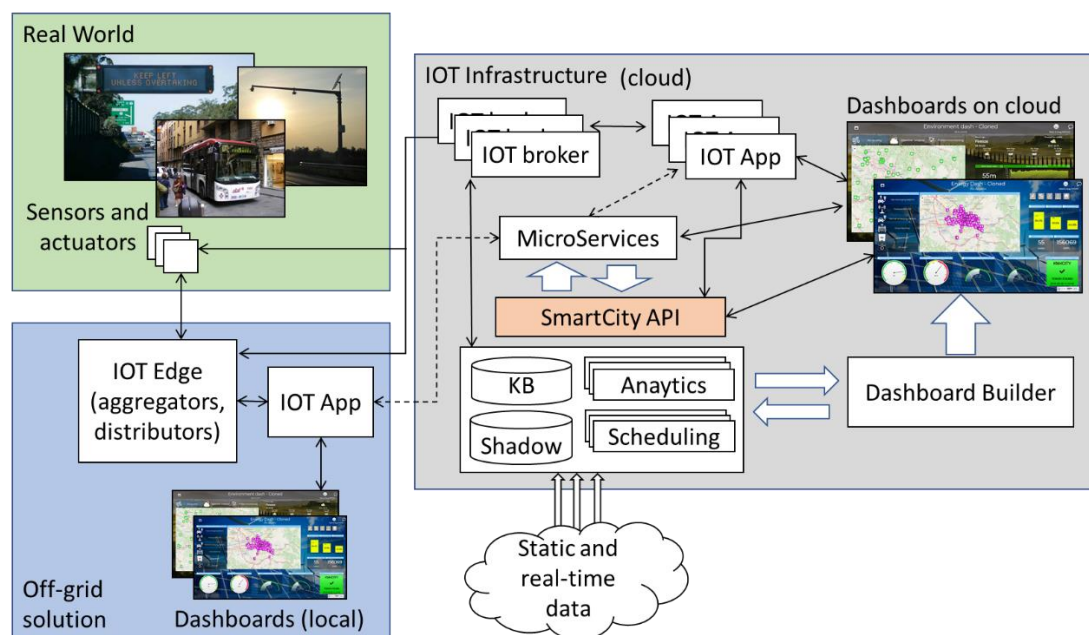
- dynamic signals on the road with respect to the conditions of the local sensors; display information to drivers and to people on the street: speed limits, presence of active/no-active smart speed meters, digital road display about the traffic condition and best solution for public transportation;
- collect information about traffic's flow sensors on the city road (how many cars in last time slot passed in a specific segment, mean speed of them, density of big truck, cars, motorbikes, etc.), estimate the traffic status in terms of density of equivalent vehicles, thus abstracting from vehicle kind;
- people flow moving in crucial areas based on cameras, laser, Wi-Fi, Bluetooth by counting sensors density of people. For example, placed in front of a bus/train station, in front of an important museum; counting the people flow in some strategic area of the city, in front of gates or around busy common path walk [14];
- implement change directions of traffic on the road with respect to the local conditions or on demand.

While other activities may need to be performed/delegated *remotely* on cloud in a mobility and transport central station and control room such as:

- predictions of: free car parking slots, delay of busses at bus stops, bikes to be shared from a given point, traffic flow in some points;
- display information to operators in Control Room to alarm in the case of early warning conditions or for current operations;
- traffic flow: catch and visualize the current global traffic situation, predict congestions, suggest alternative ways to reach a point of interest [15];
- video analysis: suggest alternative paths for tourist to avoid congestion areas (long queue) and to promote new and alternative tours;
- public transport: describe in a visual and integrated way the complete public transport infrastructure, enable queries on maps (when a bus will arrive at destination, how long does it take to commute to work) and support multi-modal routing (from here to there using different kind of public transportation means); integrate information of traffic flow to provide a better prediction;
- parking: track the availability of city parking areas, their weekly/daily use, suggest a free parking place in the area a citizen usually visit [16]; predict a parking place in a future assessment or parking area without any restriction due to street cleaning, public or private events;
- user behavior and engagement: analysis the citizen behavior in the use of the city infrastructure, highlight the behavior not sustainable and propose a switch of habits to promote virtuoso ones [17]; track if the suggestions are followed and rewards the "good" citizen;
- actuator: manage smart traffic board for dynamic setup or to inform the citizen about congestion when disruptive events occurs.

According to the above identified reasoning, the architecture designed and implemented resulted to be flexible enough to support the above requirements and use case scenarios, on IoT edge scenarios (off-grid) and cloud (with control rooms). In the context of this paper, we concentrate the attention to IoT Edge cases with some integration to cloud computing and control room. Figure 1 highlights the architecture on the basis of its major macroblocks.

Starting from the left upper part, the **Real World** and the environment are inter-connected to our infrastructure via a set of sensors and actuators. Thus, a multitude different of devices are supported and can be also classified on the basis of their energy consumption and distance in terms of network hops to our main backbone. A sensor can be implemented by using a simple *microchip* (i.e. esp8266 with microcontroller capabilities) that communicates a set of different data values via messages with some protocol and format. It can implement a button that can be connected to our backbone or to an aggregator device (IoT Edge) via a wireless network interface towards an Internet access point with standard communication capabilities (IEEE 802.x), or with patented technologies (such as LoRa, SigFox, OneM2M, ...). Most of this kind of sensors have no or low capabilities of embedding logic, and are just used to collect and transmit data to the platform or to perform some action on the basis of the data received. On similar capabilities, a *microcontroller board* (i.e. Arduino) can be used: it adds a slightly bit more dynamic setup, with aggregation capabilities between different type of sensors and actuators. These IoT devices are low energy consumption and usually works using strategies for sending data when they are significant or on-demand.



**Figure 1.** High level architecture of our IoT solution.

The data generated from the sensor or received by an actuator can be sent/received to/from an IoT Edge that act like an aggregator/distributor and that introduce the possibility to define some kind of logic (IoT Application) to enable **off-grid solutions**. IoT Edge are more capable devices are in substance *small single board computers* (i.e., Raspberry PI). They can be directly connected to a wireless network via a more powerful linkage, using little more energy, and can be installed where a remote powering is possible. They can act as sensors/actuators, as well as aggregators among of the lower level former IoT Devices or as router towards cloud IoT smart city backbone. *Handheld personal computer* (i.e., smartphones, tablets) can be regarded as IoT Edge devices and are also supported: they are similar device to the previous class, but with an embedded network connection to the Internet via 3G-5G capabilities, so they can act and communicate directly to the smart city infrastructure, even if the energy consumption has to be taken more in consideration; more generally any *personal computer*



and *Internet devices* can be integrated in the smart city IoT/IoE solution, whenever energy consumption and price is not an issue. Finally, a *virtual sensor and actuators* can be also created directly on cloud infrastructure for offering at the Operators and user to send messages and show results on some user interface. So that, a button on the user interface of an IoT Applications can be regarded as Virtual IoT Devices generating a message to an IoT broker. In IoT Edge raw data (or the elaborated ones) can be represented locally (in terms of closeness to the sensors/actuators) via local Dashboards.

In some cases, the logic (IoT Application) can use external services formalized as Microservices and exposed by the smart city platform. They are remotely exploited and, off course, when the Internet connectivity is lost, the application logic has to adapt the logic to work with the internal data and services according to its purpose. The Microservices provide nu smart city infrastructure are implemented using the SmartCity API [18] that work directly on top of Km4City Knowledge Base and a set of additional tools (for sake of readability, here we just presented two main macro-blocks: analytics and scheduling; for a more detailed description refer [19]).

In case an on-cloud solution it is chosen, the data can flow toward/from smart city **IoT infrastructure** by means of:

- a set of Context Brokers (Mosquito MQTT, Orion NSGI, RabbitMQ, etc.) which collect the data and allow other IoT application to make subscription to services;
- IoT Brokers towards a shadowing and indexing system to enable the collection of all changes performed on IoT Devices sensors/actuators and enable those data to be accessed by means of Smart City API for the IOT Applications, Data Analytics and Dashboards; This means that when the IoT devices goes off-line the system is capable to provide the last value and also the historical values that can also be used for publishing then as new data sets on Open Data Portals, automatically;
- IoT Edge devices on the field (Off-grid) which may have IoT applications inside also working with MicroServices and Dashboard Local or on Cloud via MicroServices.
- IoT Applications on cloud, which may be subscribed to IoT Brokers data, exploit MicroServices and thus services about IoT data Shadowing and Indexing, Data Analytics, Scheduling or processes, and Dashboards.

Several other management tools are available to the IoT Application programming logic in the backbone of the solution. In the context of this paper, the Dashboard Builder interfaces the IOT Application to the visual presentation of the data (Dashboards on a cloud). This module enables the City Operator to compose in a graphic dynamic and intuitive manner the visual presentations of data using a set of widgets that can be defined on top of raw or structured data. The collection of graphic widgets can be easily extended by developing specific PHP modules. The Dashboard Builder as the whole infrastructure and tools are Open Source.

The way the sensors/actuators are connected to the overall structure [20], which IoT Infrastructure's services are used by the IoT Application programming logics, and what is the best topology to connect the sensing to the elaboration side, strongly depends on the target scenario to implement. And, as well as, on the energy availability around the sensors, on the presence of an Internet connectivity (and its relative QoS) and on the required security (sensibility of the transmitted data).

The proposed IoT architecture of Sii-Mobility on cloud is the backbone of the solution. Completely virtualized on the Internet, it is the boilerplate where a City Operator interacts via a set of graphical tools to program algorithms for data aggregation, data analytics, and thus realizing solutions for computing predictions, anomaly detection, origin destination matrices, trajectories, etc. The data produced by this calculation can be also injected back in the solution knowledge base for further analysis or and exposed to the personal dashboards for graphical presentation.

Important to note, the data generated from the IoT devices are linked to the information about the user that own the device, so the platform is able, whenever in any tools of the ecosystem, to enforce and assure that a data is threaded properly and presented just when/where the owner of the

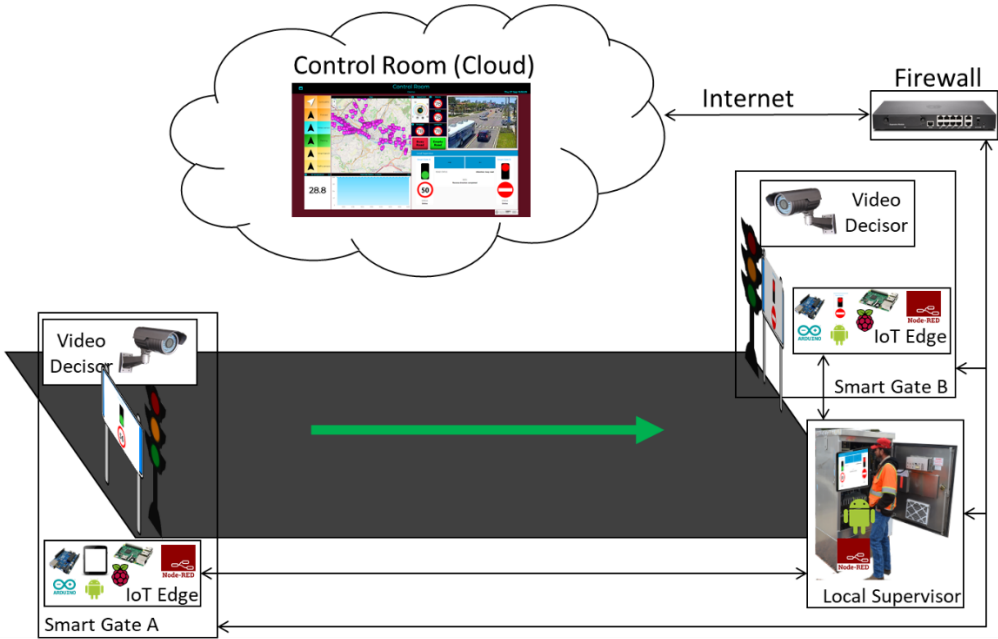
data gave his explicit consents. The data from a sensor are initially set private of the owner and thus he/she the only one that can visualize and utilize the data. Techniques of identification and authorization are implemented in any tools of the proposed solution exploiting existent service when it is possible (i.e. to provide a user single point of authentication we employ the use of the open-id connect protocol) or implementing new ad-hoc solutions where it was not available (i.e., to verify the device authorization to access an Orion broker (for example for a subscription) via a programmable firewall compliant with NGSI protocol). In the communication all data are encrypted to assure no other people can access the sensible information, and the connection protocol is also signed by a certification authority to prevent tempering (identity stealing). Thus, any transmission between the tool's component is protected using secure channels. Rule management for different kinds of rights and organizations in groups for classification of the users permit a flexible architecture to better organize the permissions in the overall managements of the user privacy. Finally, a delegation system is also available to permits the users to expose their data (providing specific right consent). The right management is enforced at level of IoT Devices and single IoT data (delegated device/datavalue), and can be delegated to other users, or to public (anonymously).

### 3. Experiment Key Study

In this experiment, a one-way variable road segment is taken into consideration: a road segment in which the direction of traffic flow can be dynamically inverted at any time according to the needs of the City, under the decision of the City Operator managing the section. As depicted in **Figure 2**, the solution consists of a number of elements to manage the traffic and their management by means of a set of interconnections. The main requirement, of course, is to be able to reverse the direction of travel without causing any accidents, putting in critical condition the drivers on the road.

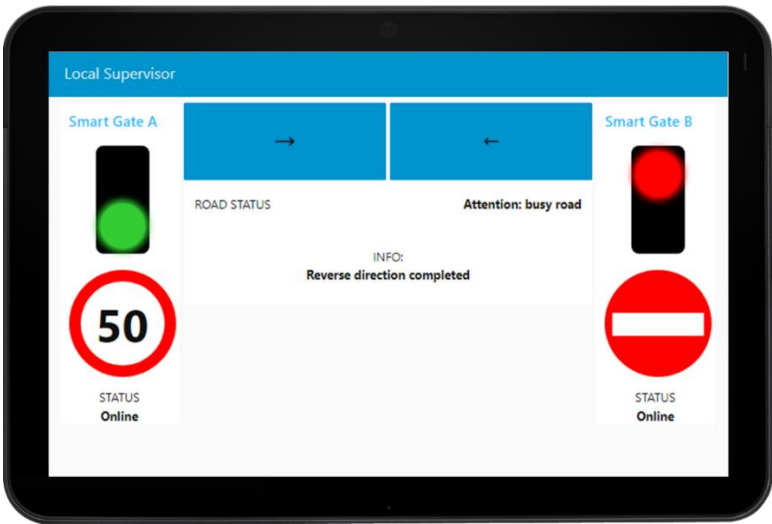
The system developed consists of two Smart Gate located on both sides of the segment to be managed, realized with a traffic light flanked by a display where information is shown to the drivers. The Smart Gates are controlled and connected to an IoT Edge each (they could be Android Device as well as Raspberry Pi or Arduino) that contain the IoT App logic to control the red-light, and the messages showed on the dynamic plate display (informing the that road cannot be accessed, or providing speed limit, etc.). The two IoT Edge Devices, in turn, are connected to a Local Supervisor IoT Edge device (also an Android Device) in the box of the Smart Gate system (usually on the side of the road segment) that can be used by the authorities to reverse the travel's direction of the segment under consideration. This Local Supervisor shows a simple and intuitive visual interface via a Dashboard (see **Figure 3**) and allows the reversal of the road with a simple press of a button in the interface.

The information coming from the Smart Gates through the Raspberry Pi and the commands sent from the Local Supervisor must have a high reliability and the City Operator must be sure that what he/she is seeing in the Local Supervisor is what the Smart Gates are showing to drivers at that precise moment. For this reason, the connection between the elements of the system are preferably performed by wired to guarantee higher reliability, but connections 5G would be a good replacement for the future.



**Figure 2.** Configuration of a one-way variable system on a road segment.

Depending on the geometry of the road segment, one or more cameras have to be installed, to be used by the Video Decisor subsystem to decide whether at a given time it is possible or not to complete the reversal operation; by controlling if the road is free of vehicles or whether the road is busy. In that case, the system has to wait until it become empty. The Smart Gates always have a complementary aspect: free access on one side and no access on the other. When switching, it is needed to check that the road is empty: the safety of people is at stake, so the software cannot be decided itself on it, but needs active human help.



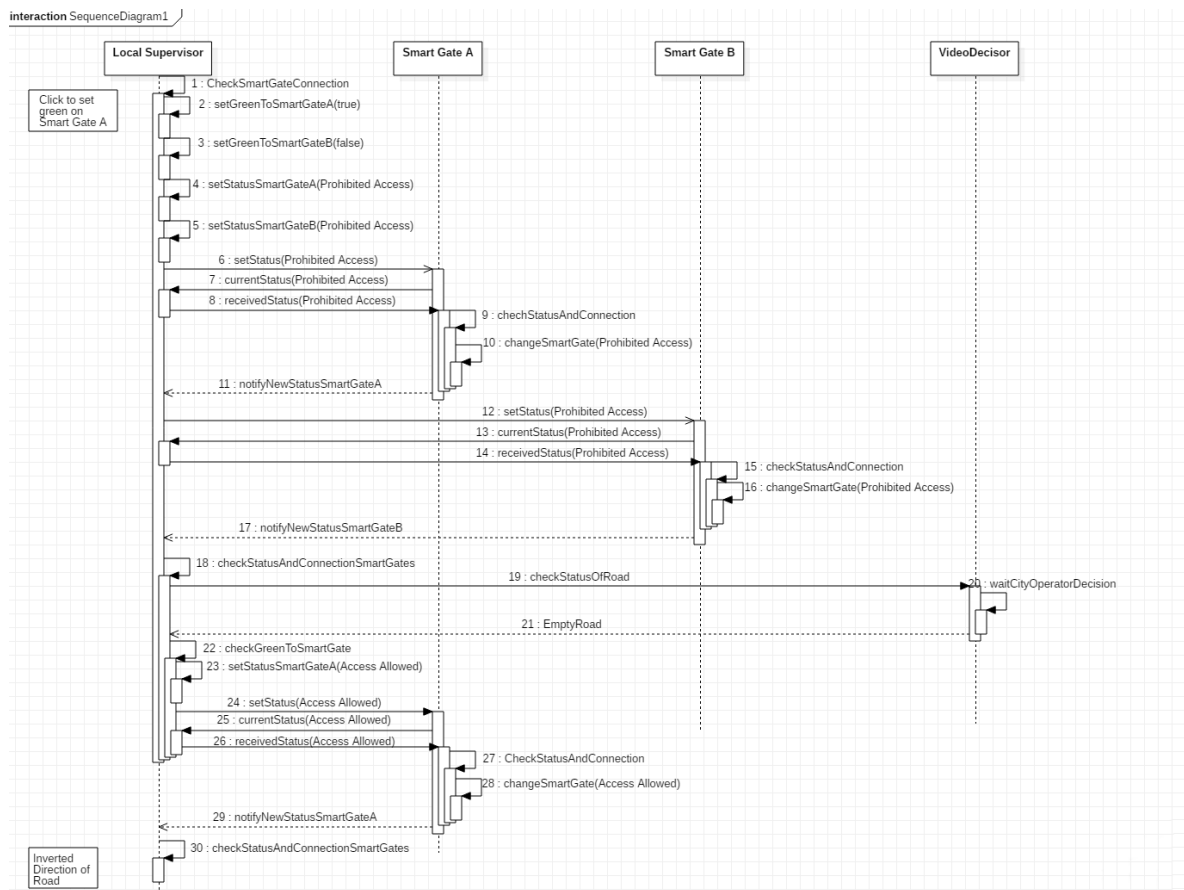
**Figure 3.** The interface (Dashboard) used to reverse the direction of travel on the road segment.

Clearly, the decision to reverse the one-way direction of a road can only be implemented safely if there are no vehicles in transit at that time. For this reason, a Video Decisor is used to detect the state of "Empty Road". This module is not directly connected to the integrated actuator but to the Control Room for safety reasons. In fact, it is not possible to rely on the judgement of an intelligent electronic device to acquire information that, if incorrect, has a direct impact on the safety of people. The information must necessarily be confirmed by a human operator on the Control Room.

For these reasons, the Video Decisor communicates directly to the Control Room, transmitting visual information based on which the operator can confirm and authorize the switching. The overall operational sequence of actions consists of:

- setting up of both actuators at the ends of the road in a state of "Prohibited Access";
- wait for confirmation of successful switching;
- wait for the "Empty Road" information from the Video Decisor and the Control Room;
- switch one of the two integrated actuators to the "Access Allowed" state.

These operations are described in detail in the sequence diagram in Figure 4.



**Figure 4.** Sequence diagram of operations that are performed in the phase of reversal of the road.

The first operation that starts the data flow is done by the operator who presses one of the two reverse gear buttons (at the top with the arrow shown in Figure 3). The system (1) checks if the connection with the Smart Gates is active and (2,3) records the target status that the Smart Gates will have according to the button selected by the operator.

It (4,5) saves locally the status "Prohibited Access" for both the Smart Gates and it (6, 12) sends the "Prohibited Access" signal to both Smart Gates and waits to receive from them an ACK message with the status they intend to change: in this case a "Prohibited Access" ACK must return (7,13). Once this ACK has been received, the system (8,14) responds with a second ACK containing the status "Prohibited Access".

If any ACK is lost, a message is shown on the screen indicating the possible inconsistency of the Smart Gates respect to the Local Supervisor and the Smart Gates show "Prohibited Access" until they receive another command and the "Smart Gate Change" protocol starts again.

If all the ACKs arrive at their destination, the Smart Gates (9-10, 15-16) turn to status "Prohibited Access" and they send a notification of the change made (11, 17, 18). The Video Decisor waits for the video to notify that the road is empty by the Control Room (19,20). When the "Empty Road" signal



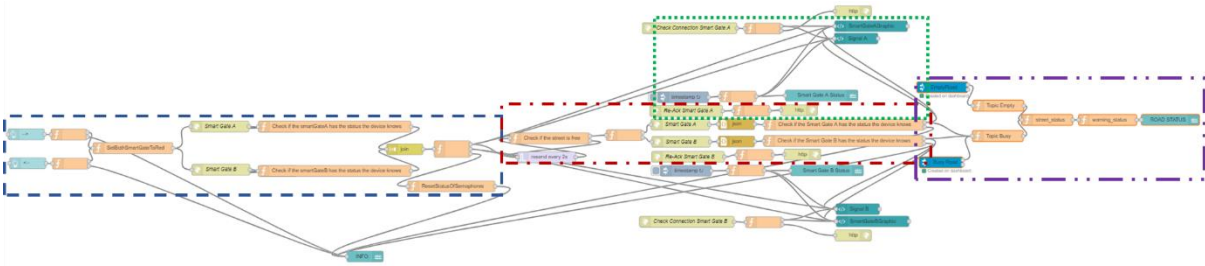
is notified (21), the Local Supervisor (22) reads the target status that the Smart Gates should have, in accordance with the operator's choice, and the "Allowed Access" signal is sent to the correct Smart Gate.

The same "Smart Gate Change" protocol as described above is started with this target Smart Gate (23-30). If the ACKs reach their destination, the Local Supervisor notifies that the reversal has been successful. If any ACK is lost, a message is displayed on the screen indicating the possible inconsistency of the Smart Gate respect to the Local Supervisor and the Smart Gate show "Prohibited Access" until they receive another command and the "Smart Gate Change" protocol starts again.

In addition, to increase the security of the reversal protocol, every two seconds the Smart Gate send their status to the Local Supervisor and if this does not arrive, the City Operator is informed that the Smart Gate that did not send the status is currently, and it is not online anymore, to the Local Supervisor.

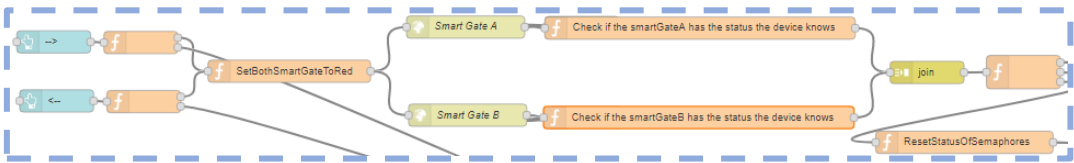
To avoid that the status is temporally misaligned with what was present at that time on the Smart Gate a timestamp is also added to the message. If the timestamp has a misalignment of more than 5 seconds, then the Smart Gate status may be inconsistent with what is present on the Local Supervisor and a message is shown to the City Operator.

The protocol described above has been implemented in the Smart Gates and in the Local Supervisor through the Node-Red platform via a Node-Red flow (Figure 5) which can be divided and explained in sub-flows (Figures 6,7,8) for greater clarity.



**Figure 5.** Complete flow of Node-RED that allows to realize the "Smart Gate Change" protocol

The first sub-flow segment (see Figure 6) allows the Local Supervisor to inhibit access from both sides of the road before allowing it on the chosen Smart Gate.

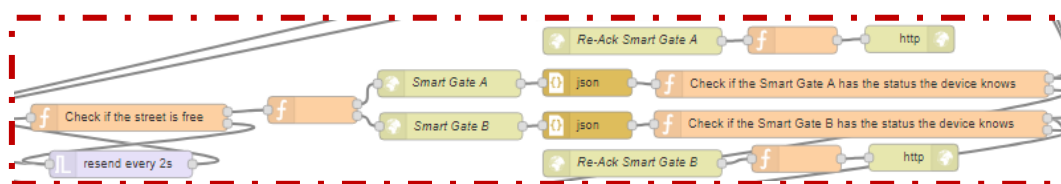


**Figure 6.** First sub-flow that allows the City Operator to set on both Smart Gate a "Prohibited Access" message. The entire flow is shown in Figure 5.

The flow starts with the two blue nodes on the left that correspond to the buttons (present in the interface of the Local Supervisor) that allow the City Operator to reverse the direction of travel. Depending on which of the two buttons is pressed, the first operation carried out by the following two orange nodes is to check that both Smart Gates have sent their signal within 5 seconds from when the button was pressed (1). If this is not the case, a message will be displayed indicating "Connection Problems" to the Smart Gates. If there are no problems the information, about which of the two Smart Gates will allow access after being inhibited by both Smart Gates, is saved (2,3). The following node sends the "Prohibited Access" signal to both Smart Gates, waiting in response for the ACKs containing the status that was sent (4-17). The last block (18) checks that both ACKs have returned, otherwise it signals connection problems between the Local Supervisor and the Smart Gates. If the

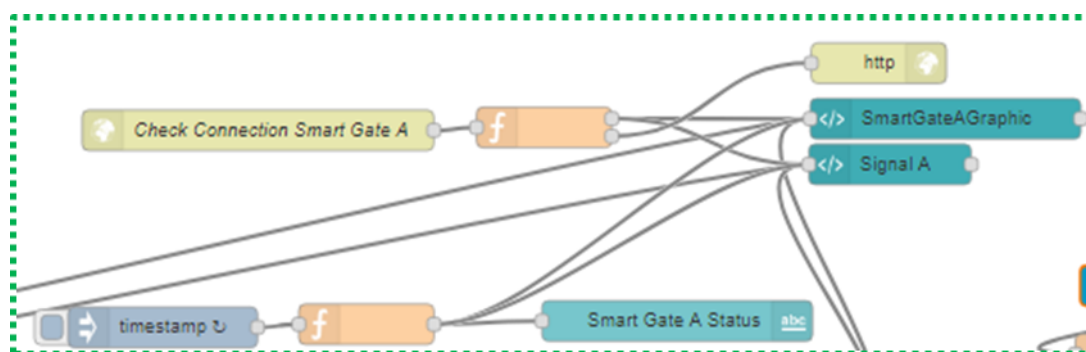
ACKs are returned but not correct, an inconsistency error is reported, and the Smart Gates remain in the "Prohibited Access" state waiting for a check or for a new signal to be sent that can be successful.

If the ACKs have been returned and are both correct then the flow continues, going to check if the state of the road is free (19). The check is carried out every 2 seconds and if it is successful and the road is empty (20,21), the status, that indicates which of the two Smart Gates should allow access, is recovered (22). A signal is sent to the right Smart Gate with the status of "Access Allowed" and is checked again that the ACK returns and that it is correct, if this happens the message that the reversal is successful is shown (23-30). If this is not the case, a message notifying the likely inconsistency between the Local Supervisor and the Smart Gates is displayed. This inconsistency should be cleared by periodically sending the statuses running the Smart Gates to the Local Supervisor every two seconds.



**Figure 7.** Second sub-flow that controls whether the road is marked as free and sends the "Access Allowed" message to the correct Smart Gate. The entire flow is shown in Figure 5.

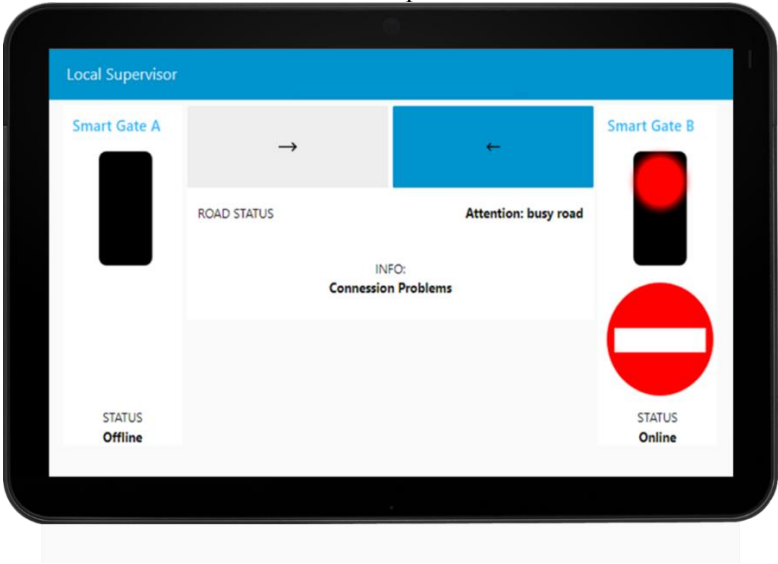
In this second sub-flow segment (see Figure 7), it is possible to notice the nodes that have a "Re-Ack" at the beginning of their name, through these nodes it is possible to receive the ACK and to resend the known status and the timestamp of the Local Supervisor to the Smart Gate (receivedStatus of the Sequence Diagram shown in Figure 4) that will answer with the last ACK (the one checked in the nodes whose name starts with "Check if", notifyNewStatusSmartGateA in the Sequence Diagram shown in Figure 4).



**Figure 8.** Third sub-flow that controls the consistency of status between the Smart Gates and the Local Supervisor. The entire flow is shown in Figure 5.

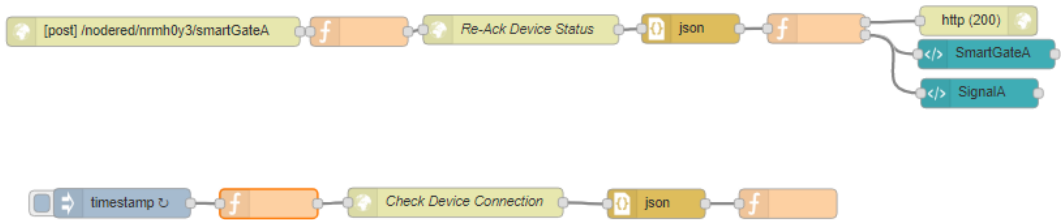
The third flow segment (see Figure 8) implements the part that maintains the consistency between Local Supervisor and Smart Gates (here is shown just a synthesis of the flow for Smart Gate A, but the flow for Smart Gate B is identical). Every two seconds the Smart Gate A notifies its status to the Local Supervisor through a direct call that arrives on the node "Check Connection Smart Gate A". The data that arrives is the status of the Smart Gate and the timestamp of when the message was created: the status is sent to the dashboard to maintain or recreate consistency and the timestamp is saved. Periodically, the lower part of the flow checks that the timestamp sent by the traffic light and the one saved is no older than a predefined time. If this time is exceeded, the Local Supervisor interface (Figure 9) will notify the City Operator that no longer a connection with the Smart Gates exists and it will remove the signals related to the latter because we cannot know if they are any more consistent.

The last flow segment (see Figure 10) is used on the Smart Gates for the logic complementary to the one written above is as follows. From the first node at the top the calls are made by the Local Supervisor when the status of the Smart Gate must be changed. When a status change message arrives, the new status is saved and is sent to the Local Supervisor to make sure that the message has arrived correctly. If the return ACK contains the same new status as the one already sent in the first message, then the Smart Gate changes its status. The bottom flow checks every two seconds if the connection with the Local Supervisor is active. If not, the status is set to "Prohibited Access" until City Operator intervention or resolution of the connection problem.

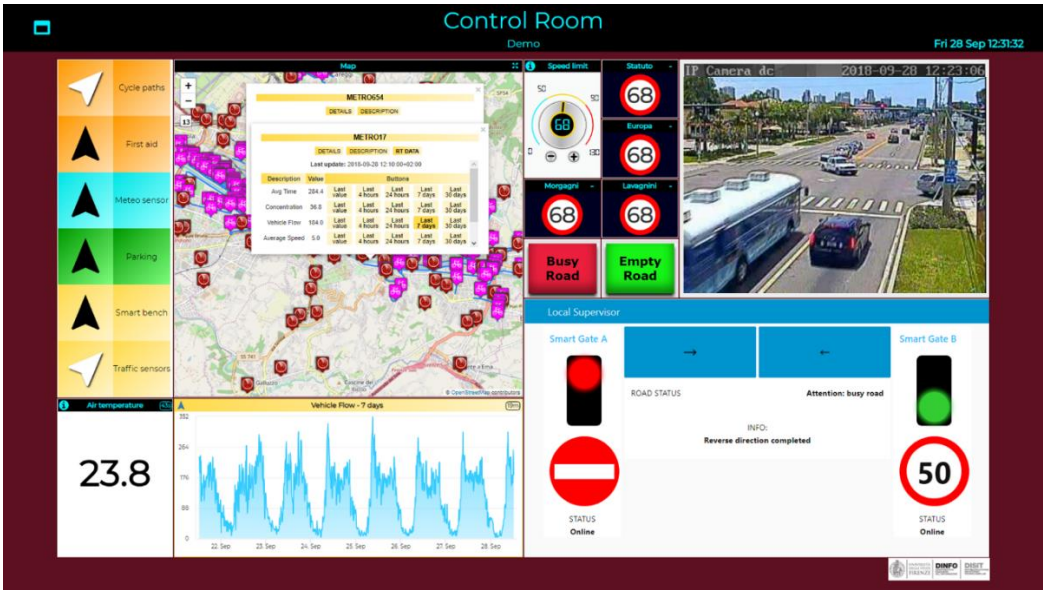


**Figure 9.** This figure shows the Local Supervisor interface in case of inconsistency and following a connection problem with the Smart Gate A.

The system described above allows to control the travel’s direction of the road being in front of the Local Supervisor and does not allow any remote control. Using the tools developed and made available by the Sii-Mobility architecture, it is possible to create a simple and intuitive dashboard as shown in Figure 11. Through this dashboard it is possible to remotely control the reversal of the considered road, having much more information at our disposal than if in front of the Local Supervisor. It is possible to have the real-time images available from the Video Decisors (note the typo) to make sure that the road is free during the reversal, it is possible to modify the signals shown in the Smart Gates and it is possible to interact directly with the Local Supervisor to launch the command that starts the reversal protocol described above.

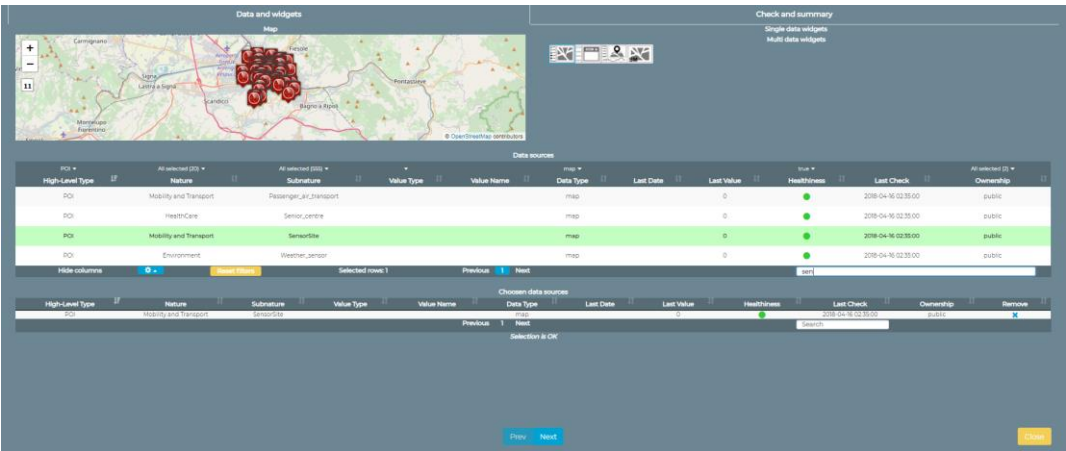


**Figure 10.** Flow that realizes the logic of the Smart Gate. The upper part is the one that receives the status and changes it if the ACK exchange is successful. The upper part is the one that checks if the connection is always active between the Local Supervisor and the Smart Gate.



**Figure 11.** The Control Room shows a large amount of information and actions that the City Operator can do through this Dashboard. It allows to have static and real-time information (including historical data) of the POIs that are in the part of the map visible on the left. It allows to send data to the logic developed via Node-red through direct calls (Local Supervisor view), through web sockets (buttons "Empty Road" and "Busy Road") or through IoT brokers (Dimmers and road signs).

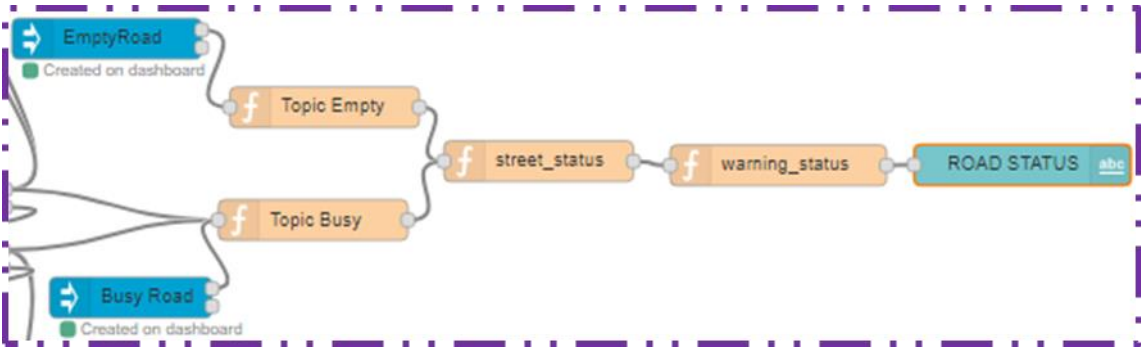
On the left of the image in Figure 11 we can see how it is possible to retrieve information from all the static and real-time data saved by the Sii-Mobility ecosystem. A map can be inserted with points of interest that can be added or removed dynamically by pressing the buttons on the left of the map. For example, in the case of the study presented here, the operator in the Control Room can show the position and the real-time values of the traffic sensors in the entire area affected by the change in gear of the road with the system installed. Moreover, for each single sensor it is possible to obtain, in addition to the real-time value of the sensor, also the historical data up to one month before the request. Once the choice has been selected, the trend of that sensor in the selected period is shown in the bottom panel. The use of this Widget is done through a Dashboard Wizard (Figure 12), that utilizes the services provided by a Dashboard Builder module and allows the user who is creating the Dashboard to decide which are the services he wants to see inside the map and on where the area should be centered.



**Figure 12.** Creation of a widget (the map with the selectors of the various POIs) through the Wizard. In the example, traffic sensors are selected as POIs.

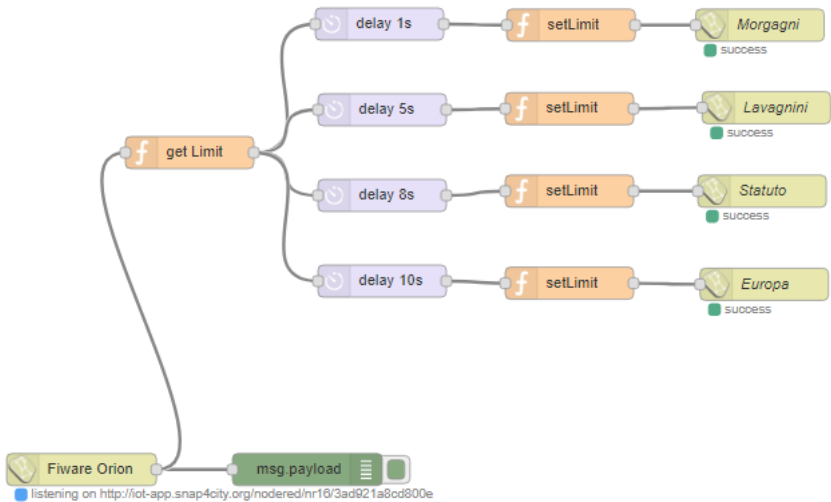
In the top right corner of Figure 11 it shows the signal received by the camera part of the Video Decisor subsystem and based on which it is possible for the operator to decide if the road is free or not. Next to the video there are two buttons that allow the operator to send the signal to the Local Supervisor whose interface is shown in the lower part of the Control Rooms. The connection with the interface is sent to an Android device located in the box on the side of the road segment and the operator in the Control Room can act as if he was in front of the Local Supervisor. In this way, if the connection is lost, it is not possible to send signals to the Smart Gates as they cannot access the interface.

The "Empty Road" and "Busy Road" buttons are part of the Video Decisor subsystem. The importance of these buttons is to make a dashboard in the Control Room interactive: it allows the City Operator to send commands that can be read and, basing on the value that these commands have sent, to execute the linked logic. These nodes were added to the dashboard directly by creating the flow on node-red. In fact, using the blue nodes that can be seen on the left of figure 13, the user can decide in which of his dashboards add buttons that, if pressed, will send a message in the flow of node-red outgoing from the respective blue nodes. When the flow is executed the button appears on the selected dashboard and sends data to the node inside the flow.



**Figure 13.** Forth sub-flow that allows the interaction of the dashboard with Node-red. Through this flow, data can be sent from the buttons in the Control Room to the logic that resides in the Local Supervisor. The entire flow is shown in Figure 5.

In the development of the subsystem of the Video Decisor the flow, that follows the reception of data from the buttons, controls which of the two buttons has been pressed and saves the value of free/busy road. It will be used to allow the change of state of a chosen Smart Gate.





**Figure 14.** Flow that allows City Operator to read from an entity connected to the Dimmer and write these values to four other entities connected to the road signs in the Control Room. These entities reside within an IoT Broker.

Moreover, in addition to the impulsive buttons, other actuators can be inserted in the dashboard for example a Dimmer: an example is shown in Figure 11 that present a button that is between the 4 road signs and the map. This Dimmer is connected to an entity on a Context Broker and when it is turned around by the users, the value of the connected entity changes. The detailed flow in Figure 14 allows to read the value of the entity associated with the dimmer in Event-Driven mode and write the 4 entities associated with road signs in the dashboard that will change their value based on that set by the Dimmer.

#### 4. Discussion

The way the user moves around the city dramatically changed in the last twenty years and the increase of integration between the city infrastructure with the sensors/actuators that are able to act-on/catch its status is enabling nowadays a wide range of new application to be built on top of new IoT architectures and platforms. A set of requirements, these architectures have to be satisfied in terms of smart mobility, has been identified and a brief description of the logical separation between local computation (off-grid solutions) and cloud elaboration has been presented. The major building block of our IoT solution has been explained and a brief description of the main functionalities of the tools have been highlighted.

A complete and real problem has been used to validate our IoT solution and its modularity: a road segment in which the direction of travel can be reversed at any time according to the needs of the City Operator that manage that section. We described all the problems related to its automation and how it's possible to employ our IoT building blocks to enable a secure workflow, integrating a local computation with a cloud solution. Some snaps of the logic of the IoT application has been presented, to highlight the simplicity of the programming environment supported.

**Funding:** This research was funded by MIUR with the Smart City national founding project Sii-Mobility SCN\_00112.

**Acknowledgments:** The authors would like to thank the MIUR, the University of Florence and companies involved for co-founding Sii-Mobility project. Km4City is an open technology of research of DISIT Lab.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results

#### References

1. Su, K., Li, J., & Fu, H. (2011). Smart city and the applications. In *Electronics, Communications and Control (ICECC), 2011 International Conference on* (pp. 1028-1031). IEEE.
2. Gartner.com. Available online: <https://www.gartner.com/newsroom/id/3598917> (accessed on 01 October 2018).
3. Jin, J., Gubbi, J., Marusic, S., & Palaniswami, M. (2014). An information framework for creating a smart city through internet of things. *IEEE Internet of Things journal*, 1(2), 112-121. IEEE.
4. Schaffers, H., Komninos, N., Pallot, M., Trousse, B., Nilsson, M., & Oliveira, A. (2011). Smart cities and the future internet: Towards cooperation frameworks for open innovation. In *The future internet assembly* (pp. 431-446). Springer.
5. Bačić, Ž., Jogun, T., & Majić, I. (2018). Integrated Sensor Systems for Smart Cities. In *Tehnički vjesnik*, 25(1), 277-284.
6. Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). Internet of things for smart cities. In *Internet of Things journal*, 1(1), 22-32. IEEE.

7. Kamilaris, A., Gao, F., Prenafeta-Boldú, F. X., & Ali, M. I. (2016). Agri-IoT: A semantic framework for Internet of Things-enabled smart farming applications. In *Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on* (pp. 442-447). IEEE.
8. Gebresselassie, M., & Sanchez, T. (2018). "Smart" Tools for Socially Sustainable Transport: A Review of Mobility Apps. In *The Future of Urban Transportation and Mobility Systems*, 2(2), 45. MDPI.
9. Fazio, M., Paone, M., Puliafito, A., & Villari, M. (2012). Heterogeneous sensors become homogeneous things in smart cities. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on* (pp. 775-780). IEEE.
10. Rathore, M. M., Ahmad, A., Paul, A., & Rho, S. (2016). Urban planning and building smart cities based on the internet of things using big data analytics. In *Computer Networks*, 101, 63-80. Elsevier.
11. Elmaghraby, A. S., & Losavio, M. M. (2014). Cyber security challenges in Smart Cities: Safety, security and privacy. In *Journal of advanced research*, 5(4), 491-497. Elsevier.
12. Sowe, S. K., Kimata, T., Dong, M., & Zettsu, K. (2014). Managing heterogeneous sensor data on a big data platform: IoT services for data-intensive science. In *Computer Software and Applications Conference Workshops (COMPSACW), 2014 IEEE 38th International* (pp. 295-300). IEEE.
13. Poncela, J., Vlacheas, P., Giaffreda, R., De, S., Vecchio, M., Nechifor, S., ... & Demestichas, P. (2014). Smart cities via data aggregation. In *Wireless personal communications*, 76(2), 149-168. Springer.
14. Pau, G., Campisi, T., Canale, A., Severino, A., Collotta, M., & Tesoriere, G. (2018). Smart Pedestrian Crossing Management at Traffic Light Junctions through a Fuzzy-Based Approach. In *Future Internet*, 10(2), 15. MDPI.
15. Bellini, P., Bilotta, S., Nesi, P., Paolucci, M. & Soderi, M. (2018). WiP: Traffic Flow Reconstruction from Scattered Data. In *Smart Computing (SMARTCOMP), 2018 IEEE International Conference* (pp. 264-266). IEEE.
16. Badii, C., Nesi, P. & Paoli, I. (2018). Predicting available parking slots on critical and regular services by exploiting a range of open data. In *IEEE Access* 6 (2018). IEEE.
17. Badii, C., Bellini, P., Cenni, D., Difino, A., Paolucci, M. & Nesi, P. (2017). User Engagement Engine for Smart City Strategies. In *Smart Computing (SMARTCOMP), 2017 IEEE International Conference* (pp. 1-7). IEEE.
18. Badii, C., Bellini, P., Cenni, D., Martelli, G., Paolucci, M. & Nesi, P. (2016). Km4City Smart City API: an integrated support for mobility services. In *Smart Computing (SMARTCOMP), 2016 IEEE International Conference* (pp. 1-8). IEEE.
19. Bellini, P., Nesi, P., Paolucci, M., & Zaza, I. (2018). Smart City architecture for data ingestion and analytics: processes and solutions. In *2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService)* (pp. 137-144). IEEE.
20. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing* (pp. 13-16). ACM.