A Web Crowdsourcing Platform for Territorial Control in Smart Cities

Andrea Pazienza[1 [0000−0002−6827−2585]], Domenico Lofù[1,2 [0000−0001−6413−9886]], Giampaolo Flace[1 [0000−0003−3648−9656]], Marco Salzedo[1 [0000−0002−8261−6831]], Pietro Noviello[1 [0000−0003−3881−9188]], Eugenio Di Sciascio[2 [0000−0002−5484−9945]], and Felice Vitulano[1 [0000−0002−6059−8177]]

[1] Innovation Lab, Exprivia S.p.A. – Via A. Olivetti 11, Molfetta (I-70056), Italy
{andrea.pazienza, domenico.lofu, giampaolo.flace, marco.salzedo, pietro.noviello, felice.vitulano}@exprivia.com
[2] Polytechnic University of Bari – Via E. Orabona 4, Bari (I-70125), Italy
{domenico.lofu, eugenio.disciascio}@poliba.it

Abstract. Nowadays citizens engage with smart city ecosystems in several ways using smartphones, mobile devices, connected cars, and drones. Pairing devices and data with a city's infrastructure and services can improve sustainability and achieve an improvement in awareness and territorial control. Communities can improve energy distribution and decrease traffic congestion with the help of IoT technologies. To support and streamline such a process, in this paper we introduce a Web crowdsourcing platform as a Common Operational Picture dashboard to interoperate with smart devices, collect urban data from them, and monitor the city in real-time. Its application to the Metropolitan City of Bari is presented and discussed.

Keywords: Web Crowdsourcing · COP · Smart City.

1 Introduction

Crowdsourcing is based on the concept of involving the participation of a large number of individuals to solve certain problems together. Web crowdsourcing provides a new model for solving problems by gathering data also from interactive objects. The Internet of Things (IoT) offers services in almost all daily fields through advanced connectivity of devices, systems, and services, particularly smart applications. Hence, the idea of exploiting the Web crowdsourcing concept to smart city projects may help in fostering the activity of territorial control and monitoring by collecting data from citizens, including feedback and evaluation about urban services, city functioning, quality of life, as well as improvement suggestions. In particular, taking into consideration also autonomous devices, in particular Unmanned Aerial Vehicles (UAVs) such as drones, and self-driving vehicles, their IoT characteristics can be exploited to collect data from a limited urban area while achieving territorial control and obtaining an improved urban situational awareness.

In this context, a Common Operational Picture (COP) is defined as a shared representation of widespread and general knowledge concerning operation. Situational Awareness (SA) is at the basis of decision processes to maintain and understand what is happening in a certain situation and leverage this information to avoid or mitigate risks. SA can thus be considered as a purposeful act of an entity of a comprehensive system. This act directs interest to the relative combination of circumstances and entities that have aims and goals in common contexts at a certain moment.

In this work, we present a smart city crowdsourcing Web platform capable of monitoring the territory in near real-time, involving the creation of COP-based dashboards, able to collect data from a number of heterogeneous sources. Such data can be processed and analyzed to show geo-located information on a dedicated interactive map. In particular, we are interested in the geo-tracking of moving objects, and the real-time consultation of the data received and their trend over time. Furthermore, from a SA perspective, this solution may help to understand the perception of events, especially in critical ones. For example, exceeding the maximum speed of a vehicle inside the city or their presence in forbidden areas, will have to activate appropriate alarms or signaling mechanisms that will be managed by city authorities.

The remainder of the paper is structured as follows. Section 2 provides an overview of related work and technologies which were investigated as background knowledge. The next section presents our proposal, i.e. a Web crowdsourcing platform, including its functional architecture and the employed technological stack. Section 4 describes a possible scenario of territorial control by monitoring a smart city by means of our Web crowdsourcing platform. Finally, Section 5 concludes the paper, with an outline of future work.

2 Related Work

One of the steps towards SA is to make a transition into Smart Cities. Here, IoT play an important role to gather relevant information from the city, citizens, and the corresponding communication networks that transfer the information in real-time. In this respect, the works in [10, 7] proposed solution for a smart city mobility monitoring platform based on crowdsourcing mobile solutions to gather and provide open traffic data, supporting an ecosystem for collaborative development of new Intelligent Transportation Systems applications.

Other works, such as in [6, 8, 3], investigated the role of IoT in smart city projects, analysing the evolution of research in the field of the smart city and related technologies, including crowdsourcing.

Since the popularity of the crowdsourcing for performing various tasks online increased significantly in the past few years, several frameworks and Web-based solutions have been provided, witnessing the need for tools to develop and run subjective quality assessment experiments [4, 9]. Authors in [5] proposed a framework to enable a worldwide crowdsourcing approach to the generation of OBD-II data, similarly to OpenStreetMap for cartography.

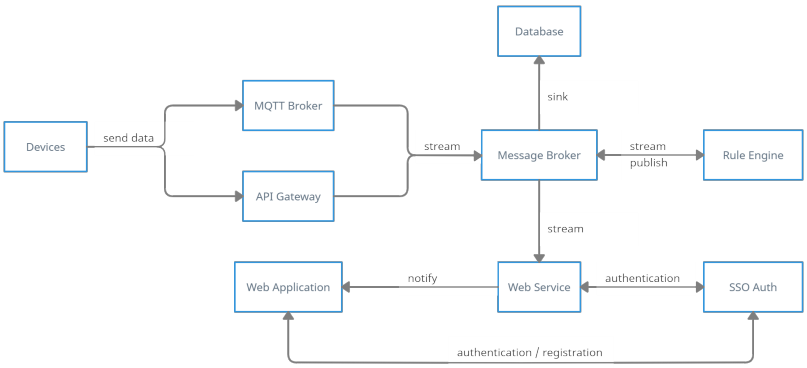

Fig. 1. Architectural diagram of the Web crowdsourcing platform.

In this landscape, COP-based solutions aiming at achieving situational awareness scenarios have been proposed, devoted to maintaining an understanding of what's going on around people at every moment and using that information to mitigate risks [1, 2].

3 Crowdsourcing Data from Smart Autonomous Devices

This section presents the functional architecture of the Web crowdsourcing platfom, paying attention to the flow of data, ranging from the collection in the field to the persistence and use of them on the system. In Figure 1 the architectural diagram is shown.

The intelligent devices for data collection in the field are enabled to transmit data, after authentication, using the protocols in two ways, namely MQTT (Message Queuing Telemetry Transport) and REST (REpresentational State Transfer). The devices that support the transmission of data via MQTT, convey them into an MQTT broker on specific topics for each type of device. If the devices do not support the MQTT communication protocol, the transmission of data is allowed by making requests to specific REST API endpoints. In this case, a massive transmission of data is also allowed. The internal data management of the system is mediated by a *Message Broker (MB)*, which acts as a communication bus common to all system components. In the MB the data are merged into two streams based on the type of information, called *tracking* and *events* respectively. The tracking messages correspond to those sent by smart devices, generally made up of information such as: instant time of detection, values detected by sensors and the current geographical position. The *API Gateway* component, once received, publishes them on the topic *tracking* of the MB. Instead, the messages arriving on the MQTT broker are consumed and inserted in the MB in the same way. Notifications and alerts are generated by the *Rule Engine* component, which has the task of applying user-defined rules and controls

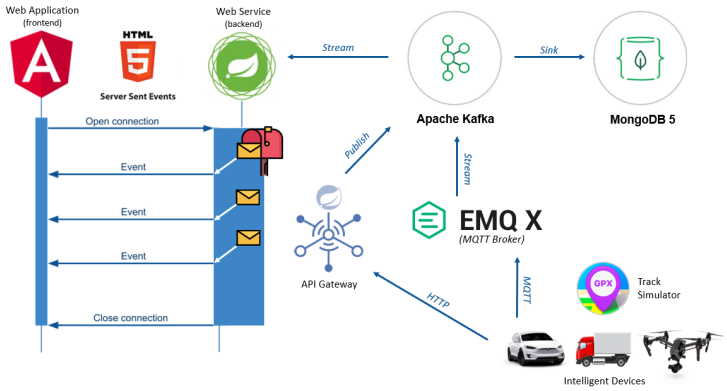

Fig. 2. Data Flow of the Crowdsourcing Architecture

to the data it receives in streaming from the MB topic *tracking*. Examples of rules may concern the exceeding of threshold values, access to forbidden areas, exit from geofence areas, and so on. Any data in transit on the topics of the MB is persistent on a time-series database within tables or collections. The choice of the type of database is dictated by the need for efficient reading queries over time intervals. A *Single Sign-On (SSO)* solution is used for managing user roles and data, system authentication, and for the use of services such as password registration and recovery. The *Web Service* component has back-end functionality and exposes all the services necessary for the Web application that the end-user will use. The main services concern the fetch of context information for initialization purposes by querying the database, and the presentation of the data in near real-time from the MB. A *Web Application* allows the end-user to access all the tools of the COP such as the geo-localized display of information and events in order to be able to reconstruct in any moment the SA of the monitored territory.

3.1 Technological Stack

As shown in Figure 2, moving objects send data to the platform via MQTT or HTTP protocol. In the first case, the messages are published on specific topics present in the MQTT broker, implemented with *EMQ-X*[3]. Subsequently, the MB, represented by *Apache Kafka*[4], uses a special connector to consume them by collecting them in more generic topics (e.g., *tracks*). In the second case, instead, the intelligent devices send data by making HTTP requests to specific REST API implemented in the *API Gateway*. The latter, once the data has been received, in sequence or in batches, publishes them directly on specific Kafka topics. At this point, once the messages arrive on the Kafka topics, they are persisted

[3] EMQ-X: https://www.emqx.io/
[4] Apache Kafka: https://kafka.apache.org/

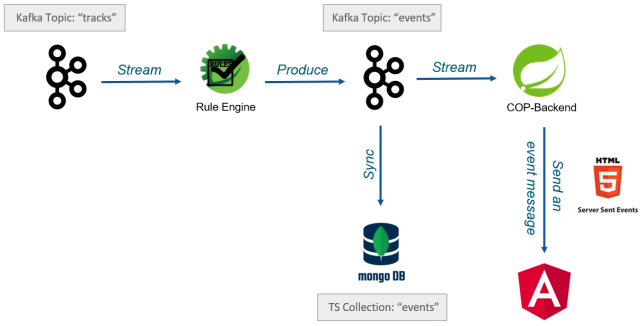

Fig. 3. Rule Engine diagram for the management of alarm and notification events.

on the *MongoDB 5* database[5], storing their content in the form of documents within specific timeseries collections. At the same time, the *Web Service* assumes the role of consumer of the messages present in the Kafka topics, in order to serialize them and transmit them in near real-time through notifications to the *Web Application*, using the *Server Sent Event* (SSE) protocol.

The generation of notifications (e.g., alarms), as shown in Figure 3, is taken over by the *Rule Engine*, which consumes the tracking messages present on Kafka in order to subject them to certain rules hand-coded, such as simple conditions on threshold values or violation of constraints on geofence areas. In the latter case, a GIS library is used which is part of the Mapbox SDK and is able to verify whether or not a geographical point (i.e., the actual position of the device) belongs to a geographical area delimited by a polygon. In particular, keeping track in memory of the last known positions for each monitored device, it is possible to trace the input and output of devices within these areas. If a rule is triggered, an event message is generated which can be of the type alarm or information, and finally published on the Kafka topic *events*. At this point, through a Kafka connector of type *MongoDB SinkConnector*, once the messages in the topic are received, they are synchronized on the database in the time series collection *events*. Finally, the Web Service component subscribes to the topic *events* consuming the messages and then sending them to the front-end via the SSE protocol. More precisely, the *SseEmitter* component of *Spring*[6] is used to create an SSE channel dedicated to events, accessible after authentication. Finally, the *Web Application* implemented with *Angular 12*[7], subscribes to this channel. User authentication is managed through SSO using *Keycloak*[8], which is also entrusted with saving user data, and registering and managing accounts for them. The interaction between the Web Application user and the SSO is mediated by a back-end service called *COP-Auth* which acts as middle-ware.

[5] MongoDB: https://www.mongodb.com/
[6] Spring: https://spring.io/
[7] Angular: https://angular.io/
[8] Keycloak: https://www.keycloak.org/

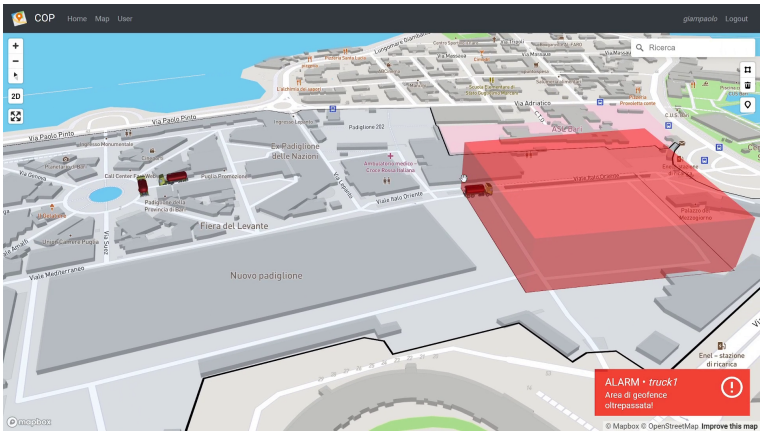

Fig. 4. Dynamic display of 3D objects and notifications.

This component exposes REST APIs that allow the front-end to register and authenticate the user at a higher level. After the user successfully logs into the system, she receives an auth-token used for invoking the secure back-end resources provided by the micro-services. The latter, before allowing any request to the resources, interact with SSO and evaluate the validity of the authentication token provided in input. To render the map of the territory and display on it the objects dynamically monitored on the Web application, *MapboxGLJS*[9] is used, a Javascript library that exploits the *WebGL 2* hardware acceleration. In particular, the *Deck.GL* library[10] is used to render the layers in 3D.

4 The Case of Metropolitan Area of Bari

The proposed solution has been applied in the context of the Italian national project called *Casa delle Tecnologie Emergenti*, in the Metropolitan Area of Bari, to allow near real-time monitoring of the exhibition districts, nameley a limited area representing a small-scale model of a smart city in which the tracking data of self-driving vehicles, traditional vehicles and drones, related to telemetry and environmental information, allow the monitoring staff to visually identify any dangers in a promptly and interactive fashion, through notifications or alarms generated autonomously by the system. In particular, the alarms are generated through the rule engine of the Web crowdsourcing platform through the analysis of time windows and the streaming of the captured incoming data.

In the presence of events inside the "Fiera del Levante", a large complex that has been selected as an experimental site, a large influx of people is expected, whose security and privacy must be guaranteed. Thanks to our Web crowdsourcing platform it is indeed possible to define geofence areas for which drones

[9] MapboxGLJS: https://www.mapbox.com/mapbox-gljs
[10] Deck.GL: https://deck.gl/

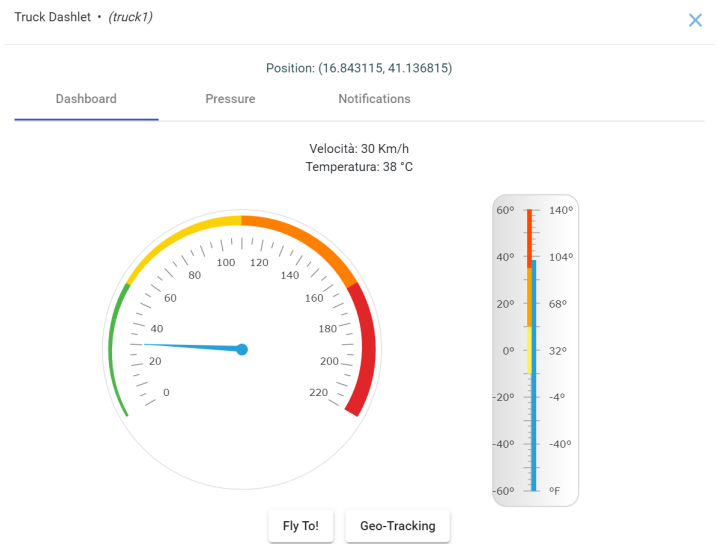

Fig. 5. Telemetry Dashlet.

cannot fly and areas forbidden to the circulation of both self-driving and traditional vehicles. Furthermore, since there are moving vehicles, it is also necessary to monitor the speed of each vehicle in order to efficiently manage traffic control. The Web crowdsourcing platform allows the monitoring of moving objects, such as self-driving vans and drones, and also the management of the definition and control of access to unauthorized areas by each of them.

Therefore, as shown in Figure 4, the monitoring operator can interact with the devices tracked by the system, having access not only to information related to their position, but also to a whole series of telemetry data that can be viewed in real time in dashlets containing animated widgets, graphs or tables. It is worth noting that in the early stages of the project, we simulated classes of smart objects in order to generate realistic telemetry values. In particular, once authenticated, the operator chooses the *Map* item in the menu, then a map implemented with *Mapbox* is rendered. Here, the boundaries of the monitored place are delimited by a polygon, outlined in black. Trucks and drones are rendered on the map using the *Deck.GL* 3D library, automatically resized according to the zoom set by the operator.

Interestingly, by clicking on any device, a dashlet appears where, as can be seen in Figure 5, it is possible to view the speed and temperature detected in real time of the vehicle through dynamic widgets, as well as graphs relating to the trend of the detections. Finally, the *notifications* tab shows the history of all the alarm and information events recorded in a tabular and paginated form. Events are notified in near real-time at the bottom right, with a event type colored box showing the type of event, the identifier of the concerned vehicle, and information content.

5 Conclusion

In this work, a Web crowdsourcing platform has been presented as a COP solution aimed at geolocalizing and monitoring drones in flight, self-driving electric vehicles, traditional vehicles, pedestrians, and IoT devices installed in urban areas, as well as crowdsourced information, gathered for territorial control in the framework of the monitoring of Bari city.

In future works, such a solution can take advantage of the recent introduction of the 5G network which allows intelligent devices to transmit data extremely fast, reducing the problem of latency.

Acknowledgements This work was funded by the Italian Ministry of Economic Development (MISE) as part of the Emerging Technologies Support Program - FSC 2014/20 - Asse I.

References

1. Ardito, C., Di Noia, T., Fasciano, C., Lofù, D., Macchiarulo, N., Mallardi, G., Pazienza, A., Vitulano, F.: Towards a situation awareness for ehealth in ageing society. In: Proceedings of AIxAS 2020. pp. 40–55 (2020)
2. Ardito, C., Di Noia, T., Fasciano, C., Lofù, D., Macchiarulo, N., Mallardi, G., Pazienza, A., Vitulano, F.: Management at the edge of situation awareness during patient telemonitoring. In: Proc. of 19th International Conference of the Italian Association for Artificial Intelligence (AIxIA 2020). vol. 12414, pp. 372–387 (2021)
3. Boccadoro, P., Daniele, V., Di Gennaro, P., Lofù, D., Tedeschi, P.: Water quality prediction on a sigfox-compliant iot device: The road ahead of waters. Ad Hoc Networks **126**, 102749 (2022)
4. Hoßfeld, T., Hirth, M., Korshunov, P., Hanhart, P., Gardlo, B., Keimel, C., Timmerer, C.: Survey of web-based crowdsourcing frameworks for subjective quality assessment. In: 2014 IEEE 16th International Workshop on Multimedia Signal Processing (MMSP). pp. 1–6. IEEE (2014)
5. Loseto, G., Gramegna, F., Pinto, A., Ruta, M., Scioscia, F.: A mobile and web platform for crowdsourcing obd-ii vehicle data. OJIOT **7**(1), 43–58 (2021)
6. Pazienza, A., Macchiarulo, N., Vitulano, F., Fiorentini, A., Cammisa, M., Rigutini, L., Di Iorio, E., Globo, A., Trevisi, A.: A novel integrated industrial approach with cobots in the age of industry 4.0 through conversational interaction and computer vision. In: Proc. of the Italian Conf. on Comp. Ling. (CLiC-it 2019) (2019)
7. Pazienza, A., Polimeno, G., Vitulano, F., Maruccia, Y.: Towards a digital future: an innovative semantic IoT integrated platform for industry 4.0, healthcare, and territorial control. In: 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC). pp. 587–592. IEEE (2019)
8. Shahrour, I., Xie, X.: Role of internet of things (iot) and crowdsourcing in smart city projects. Smart Cities **4**(4), 1276–1292 (2021)
9. Staletić, N., Labus, A., Bogdanović, Z., Despotović-Zrakić, M., Radenković, B.: Citizens' readiness to crowdsource smart city services: A developing country perspective. Cities **107**, 102883 (2020)
10. Suciu, G., Butca, C., Dobre, C., Popescu, C.: Smart city mobility simulation and monitoring platform. In: 2017 21st International Conference on Control Systems and Computer Science (CSCS) (2017)