Infrastructure-Less Prioritised Communication Platform for First Responders

1st Philip Wright
Engineering Ingegneria Informatica S.p.A.
Rome, Italy
philip.wright@eng.it

2nd Ilmija Asani
Engineering Ingegneria Informatica S.p.A.
Trento, Italy
ilmija.asani@eng.it

3rd Nelson Pimenta
Instituto de engenharia de sistemas e computadores inovacao
Lisbon, Portugal
nelson.pimenta@inov.pt

4th Paulo Chaves
Instituto de engenharia de sistemas e computadores inovacao
Lisbon, Portugal
paulo.chaves@inov.pt

5th William Oliff
University of Greenwich
London, United Kingdom
william.oliff@greenwich.ac.uk

6th Georgia Sakellari
University of Greenwich
London, United Kingdom
g.sakellari@greenwich.ac.uk

***Abstract*—Infrastructure-less environments present unique challenges in emergency environments, where resources are scarce and information needed to be shared in an efficient manner. With the use of IoT technologies in emergency response situations the amount of information generated can be overwhelming for both the First Responders (FRs) and the infrastructure-less network that might connect them. Therefore, architectures need to be implemented that can prioritise and regulate this information before displaying it to the FRs or sending it over in an efficient manner. This paper introduces such an architecture and the core tools that comprise it. This work, is part of the RESCUER project[1], an EU-funded project aiming at developing a FR centered technology toolkit that will empower the next generation of FRs by enhancing their operational capacity and safety, specifically in adverse conditions, both environmental and infrastructure-wise.**

***Index Terms*—Information Prioritisation, Data Regulation and Orchestration in IoT environments, Ad Hoc Networks, First Responder**

I. Introduction

First Responders (FRs) play a critical role in maintaining public safety, as they are the front-line rescuers in emergencies and major crises. During these incidents, FRs rely on communication technology such as radios, cell phones, and computer-aided dispatch to gather information and coordinate appropriate incident response. During their missions the main problems they face regarding the technology they carry with them is connectivity, reception and bandwidth of the devices and being overwhelmed by a sensory overload and not being able to have a clear situational awareness [1].

The authors demonstrate and provide a way to design and to implement a real infrastructure, part of the RESCUER toolkit, that satisfies the requirements coming from real scenarios of FRs. Moreover, this architecture can be adapted in a different context than the RESCUER project and it could be applied in different scenarios (see Section III, where examples are illustrated in more detail).

The RESCUER toolkit consists of a number of different tools, or modules, that aim to enhance the capabilities of a FR by enhancing their senses, their ability to detect threats (e.g, dangerous gases), find victims, communicate their finding and provide situational awareness.

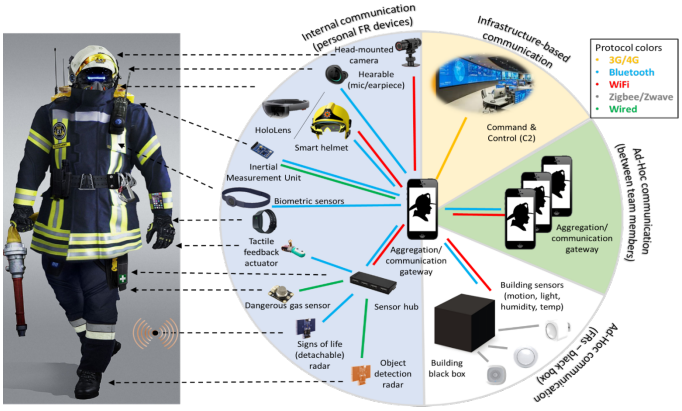

Fig. 1. RESCUER components

Figure 1 shows the components of the RESCUER architecture, their connections and inter-relations. There are two types of tools in this architecture; first the software tools that utilise the hardware (i.e., IoT devices such as helmet-mounted cameras, in-ear hearables, inertial measurement units (IMU), biometric sensors, dangerous gas detectors, and radars) which process IoT data and augment the FR's senses (e.g., tools for detecting gas, victims, enhancing vision in smoked environment, estimating self-position, etc.). We call those tools *capability* tools. The second type of RESCUER tools are the software tools that enable the capability tools to share their outputs and communicate their findings; we call those tools *enabling* tools.

[1]This project has received funding from the European Union's Horizon 2020 Research & Innovation Programme under Grant Agreement No. 101021836

All the data produced by the *capability* tools (augmented sense information) is collected to a central device, which runs a publish/subscribe Message Broker and serves as an aggregation communication gateway which enables the sharing of information between the different tools in a harmonised way. It also enables the information to be easily accessible from the *enabling* tools such as the prioritisation tool and the orchestrator that we will see later on. Both the prioritisation and the orchestrator tools are enabling technologies for sharing the most important information. The prioritisation tool assigns a priority to the information, i.e., messages, according to their importance, while the orchestrator takes into account this priority, alongside other metrics (e.g., network conditions, energy, cognitive load of the FR, etc.) and regulates whether a message should be shared between FRs so as to not overwhelm the network or whether it should be published to the broker in order to be shown to the FR. Information can be visualised to the FR either on a smart helmet or a state-of-the-art Augmented Reality (AR) device such as the HoloLens.

In this paper we present how our proposed technologies can support the FRs in their missions, starting from the RESCUER project use case and generalising it to be adapted outside the project in a any infrastructure-less situation.

This paper is divided into the following sections: Section I presented the problem, Section II details firstly the way information is prioritised, then how it is regulated and finally the communications platform used to exchange information between FRs. Section III presents different scenario used to describe possible applications; and finally Section IV summarises our conclusions.

II. System Architecture

Within this section, we firstly detail the overall architecture of our communications platform and describe the information flow of our design, and then describe each individual *enabling* tool. We are describing this in the context of the RESCUER project, but this can be easily extended to any infrastructure-less communication platform that uses IoT sensor devices to enhance FR capabilities. The problem statement of this paper relates to how once we receive the processed IoT information we prioritise it, in order to then regulate the amount of information to be shared through an infrastructure-less Ad Hoc network.

A. Overview

Let us consider that each tool has a software (SW) component and, optionally, a hardware (HW) component, usually a sensor or, more in general, an IoT device. Due to our focus on infrastructure-less solutions, all HW and SW components of all tools are located, or hosted, on an individual rescuer's person. The different devices, including sensors and processing devices hosting the SW components, connect to each other via a Personal Area Network (PAN). Inter-tool communication is achieved via a publish/subscribe message broker, to which all SW components have access via the PAN network.

Fig. 2 demonstrates a simplified version of the architecture focusing on the flow of information. HW devices communicate exclusively and bidirectionally with their corresponding SW components: they send raw data to the SW components for processing and are controlled by them. The SW component processes the raw data from the HW component and outputs processed information, which it publishes to the message broker. From there, information is accessible to the SW components of other modules that may require it, and may be forwarded to a visualisation tool, or sent to other team members or to a visualisation interface through the Communications Infrastructure tool.

The amount of information published to the broker is regulated by the Data Sharing Orchestrator (DSO) component which decides which message should be published and become available to other tools or to be visualised by the FR or be shared between FRs via the communication component, based on different aspects such as the priority of the message or the quality of the network.

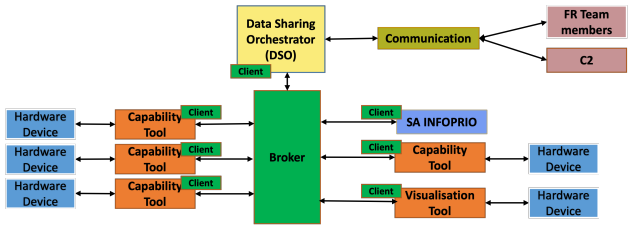

Fig. 2. Simple data flow diagram in each FR

This architecture can work with any subset of the *capability* tools, provided that the *enabling* tools (DSO, Information Prioritisation and Communication Infrastructure) are in place. This can be very useful, in different use-case scenarios and real operational conditions involving different capability and enabling tools (see section III for further details). Likewise, changes or upgrades in individual modules will not affect the whole system, as long as the message protocol formats and structures for exchanging information are observed.

B. Situation-Aware Information Prioritisation

Firstly, we describe a tool that prioritises the information received by any *capability* tool that process IoT sense information. Situation-Aware INFOrmation PRIOritisation (SA INFOPRIO) in disaster scenarios has the goal of providing emergency management teams with a clear perception of the scene, highlighting the relations among the actors involved (FRs, victims, etc.) and relations with environmental factors (risks, hazards, points-of-interest) in respect to time and space.

1) Relevant Literature: Part of SA INFOPRIO is developed as a Complex Event Processing (CEP) framework. CEP's aim is to detect relevant or critical situations (complex events) in real time. It is used to analyse and correlate huge amounts of data in the form of events. It allows for pattern recognition and triggering actions based on a combination of multiple events of different types, coming from different data sources.

CEP processes in real-time incoming events based on an existing pattern. CEP systems execute data processing, removing any irrelevant data at the beginning. As soon as the incoming events are compared to all the stored patterns, the result/response is sent out straight away, giving the process real-time capabilities [2].

In CEP application development, at least three different CEP application development paradigms, exist:

- Data Manipulation Language (DML) extensions to handle time windows, etc., and are embodied into a conventional application development stack
- Visual event-oriented programming language. Their development is at the core of the visual tool, which creates executable programs without requiring any code generation step
- Rules engine. This paradigm consists of a set of rules, which are then executed by a suitable engine. This is the approach followed for SA INFOPRIO

Currently there are several well-known tools, both open-source and commercial ones, of CEP technology. In general, these tools can be classified into: Event Processing Platforms (EPPs), Distributed Stream Computing Platforms (DSCPs) and CEP Libraries (CEPLs).

CEP can analyse a stream of events coming from different data sources in real-time and to come up with meaningful and actionable insight. CEP work is based on a collection of data from various sources and then processes them to give comprehension.

2) SA INFOPRIO Tool Description: The role of the SA INFOPRIO tool is to prioritise the messages generated by the enabling tools in order to the DSO service to regulate the amount of information exchanged between different entities.

This tool is able to process any data produced by each different data source of the ecosystem, with the aim to build and maintain a clear representation of the operational environment in which the mission is carried out and can be used to support FRs in emergency management operations. For this, as a first step the tool might involve consolidation with FRs to identify their requirements before progressing to an automated way of assigning priorities to information.

SA INFOPRIO helps in preventing the overloading of FRs with such a huge amounts of information, especially with multiple data with with similar and redundant information, or with the data not relevant to the FR's current role, position, or to the mission, time, place, and context they are operating in. This type of criteria will be used as rules, for relevant information to be recognised, processed, and prioritised, while not relevant (or already propagated) information will be ignored at particular processing stages. This will allow to provide FRs (e.g. through a visualisation interface application) with only crucial information, that is in line with their present cognitive capabilities, and that could be delivered to them on their devices via the DSO service.

Thus, this module is able to detect, to process and further to propagate, only relevant and useful information which will help to increment the awareness about current situation. In the situation awareness and emergency response domains such relevant information is called an event, and the logic of dealing with it is called an event processing.

SA INFOPRIO is part of an Event-Driven Architecture (EDA) designed for the detection, analysis, and response to events temporarily ordered and obtained from multiple sources.

More specifically:

1) Detection: receiving events which will be analysed using CEP technology. Event capture is usually applied on an already existing information system that provides events as output
2) Analysis: processing and correlating the information in the form of events according to the previously defined patterns to detect critical or relevant situations in real-time
3) Response: notifying the system, software or device in question, when detecting a particular situation of interest. In addition, the engine can only notify that a given pattern or rule has been detected, and maybe update the instance. Alternatively, a new event can be generated

A CEP application can be conceptualised through an Event Processing Network (EPN), see Figure 3 for further details. In the EPN the main nodes, beside the producers and consumers of events, are the event processing agents that can validate, enrich, aggregate and fuse events information in real-time.

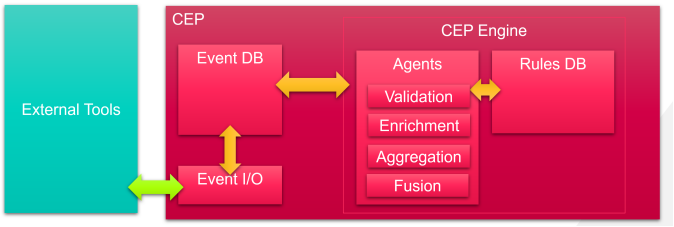

Fig. 3. EPN design

SA INFOPRIO has four types of rules, each one is interpreted by a specialized agent able to carry out a procedure on the message. Following the description of the four agents:

- Any new data received from a capability tool (via the Message Broker) is handled as a *new Event* and is immediately consumed by the *Validation Agent* that ingests and validates it. The validation process allows to identify inconsistencies in the received event and tries to solve them when it is possible. If internal validation rules are satisfied for the received event (i.e., the event contains all mandatory properties, consistent dates, valid category, location is known or resolvable), the event has passed validation stage (process)
- The *Enrichment Agent* will process each validated event trying to enrich it with pertinent information, e.g., adjusting its priority based on value provided in the enrichment rule, setting location from the location of same Device Source, i.e., from the location of same FR (if already

received), or adding any other useful data by producing an enriched event. The Enrichment Agent will be driven by a set of enrichment rules that provide the set of information about the types of enrichment to apply and the characteristic of the events to which these enrichment operations must be applied.

- The *Aggregation Agent* then processes each enriched event with the aim to identify a set of already processed events belonging to the same category and that are timely-spatially correlated with the current enriched one, and that can be aggregated (the condition to be satisfied to proceed with the aggregation, it will provide through the aggregation rules) in a single event. The aggregation of multiple events makes easier the following processing phases by reducing the complexity and redundancy of several events that include the same informative content.
- The *Fusion Agent* continuously tries to identify patterns of aggregated events that are related to specific threats. These events are fused together by producing a new valid (complex) event classified according to the event taxonomy. The event patterns are provided by a set of "Fusion Rules" that provide the patterns to match along with the characteristics of the event that will be produced by the fusion process when the pattern will match. Therefore, through the fusion process new events are generated by the CEP when specific patterns of event are identified.

The priority level of a message is currently set manually based on the preference of the user, however, we plan to use Artificial Intelligence (AI) techniques and technologies to add AI based capabilities to this CEP Solution, allowing it, for example, to apply classification models that can be continuously adapted to the incoming data.

C. Data Sharing and Orchestration

Secondly, in this section we describe how to regulate and decide which of the prioritised information received from the SA INFOPRIO, described before, should be shared between different FRs in order not to overwhelm the network connecting them, or which information should be shown to the FRs themselves as to not mentally overwhelm them if we have relevant information. The amount of information and messages exchanged between tools depends on the number of *capability tools* an FR has on them. This means that the information generated could at times be overwhelming for both the FR or the communication network that connects the different components and the FRs between them. Of course, not all information will have the same importance at a specific time and operation, thus a mechanism is needed to first prioritise it (see Section II-B) and then to regulate it. The role of the DSO is twofold; first to regulate the information provided to an FR by considering how important the information is, and second to regulate the information exchanged between FRs by considering the quality of the communication network and the battery levels of the involved CGWs.

1) Relevant Literature: Data orchestration is a process that allows to coordinate and automate the data flow of a system to make meaningful data available to their intended receivers in the fastest and most efficient way.

Data orchestration in IoT is a relatively new area of research, but has been explored before in other contexts, such as automated services management, service-oriented architectures, virtualisation, security, resource management and task scheduling. For example, in [3], an orchestrator-based architecture was proposed to enhance security in Software Defined Networking, by analysing network traffic and turning on and off the applications for detecting attacks on the system.

In the IoT area, the authors in [4] present a smart patient health monitoring system, based on an optimised scheduling mechanism using IoT-tasks orchestration architecture to monitor vital signs data of remote patients in the home, as well as in the hospital, for providing reliable health services to remote patients by minimising information lost during context switching of sensors and tasks failure frequency.

In [5], the authors examine the orchestration of multiple real-time IoT workflows in a heterogeneous fog computing environment by proposing a partial computations and error propagation model and a dynamic scheduling heuristic, while [6], considers a dynamic decision support system that considers the execution of the IoT workflows in different processing layers (locally, edge cloud). The work conducted in [7] showcases an optimal orchestration mechanism is proposed to automate the processes of mission-critical IoT applications by introducing a multi-level optimised orchestration mechanism at task-level. Results show that the operation plan is flexible and with scaling up the problem size, the orchestration is still graceful and within the requirements of mission-critical applications.

Moreover, in [8] a lightweight containerised orchestration framework is proposed that deploys a distributed middleware layer to support clustering of MQTT brokers to collectively disseminate messages between large number of MQTT clients. The experimental evaluations confirm the viability of the implemented system in terms of message throughput, latency, resiliency, and lightweight-ness.

2) DSO Tool Description: Here we describe two separate components, the Message Broker and the Data Sharing Orchestrator (DSO). The Message Broker allows sharing information between the different tools, while the DSO module allows regulating this information according to its priority and subject to network and device performance and energy constraints. Together, they are the primary enabler for information exchange, placed at the centre of all capability tools in each FR, allowing them to share between them the augmented sense information they generate, which can be ultimately presented to each FR through the visualisation tools. Moreover, it allows FRs to exchange this information between them via the Communication network. In this paper we will use the term DSO Service to describe the combination of those two entities (i.e., message broker and DSO).

The aim of the DSO Service is twofold. To allow the sharing of the augmented sense data generated by the capability tools (e.g., modules such as augmented sensing, victim detection, etc.) in a harmonised way through the Message Broker, and secondly and more importantly, through the DSO module, the regulation of how much information should be actually shared based on the priority of the information and other aspects that might be available by the other capability tools (e.g., such as the tools measuring the biosignals and cognitive capabilities of each FR, the network and energy capabilities of the hosting devices, etc).

As previously shown in Fig. 2, the DSO Service sits in the middle of the data flow architecture. The Message Broker is a publish/subscribe MQTT service [9] that enables the sharing of each capability software tool's output with other modules in a lightweight, fast to deploy and harmonised way. The DSO module regulates the information of the subscribing services and enables sharing with other entities by running a real-time, lightweight multi-criteria decision support mechanism based on different parameters: the information priority level, the current cognitive load balance of the FR, the battery level, and the network availability (Fig. 4).

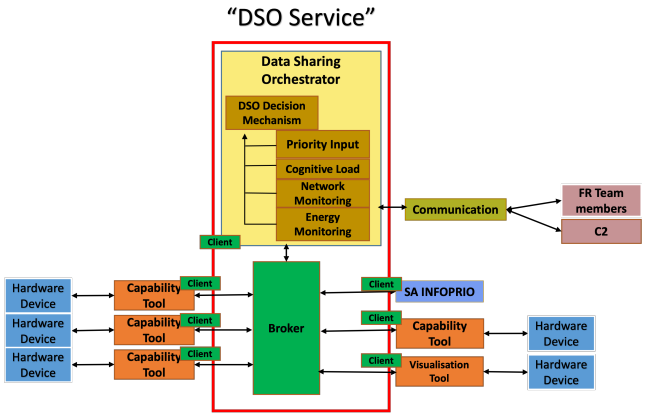

Fig. 4. DSO High Level Architectural Design

Within an FR, the MQTT Message Broker enables the capability tools belonging to the same PAN to exchange messages. Each tool will publish their outcome to a corresponding unique *fromtool* topic. These messages will be first received by the SA INFOPRIO II-B, which will assign a priority to each message and publish on it's own **fromtool** topic, in which the DSO module is subscribed to. An *internal* DSO decision mechanism will receive these as inputs and will decide in near real-time which information should be released to its corresponding unique *fromdso* topic which other tools, such as the visualisation tools, are subscribed to.

To enable tools and more specifically the DSO, hosted on the PAN of each FR, to exchange messages and data over the Communication network, some form of network bridge is required, as devices and related tools on the FRs' PAN are unable to directly reach the network. This will need to be located on the CGW, to allow communication and message exchange between different FRs.

For messages that need to be shared between FRs, the DSO Service will trigger an *external* DSO decision mechanism, which takes into consideration data gathered from internal monitoring functions (e.g., energy and network monitoring), rather than the priority as this will be evaluated in the receiving FR, and decides which information should be transmitted through the Communication network to other FRs in a way that tries to preserve the battery life of the network as long as possible and not contribute to the congestion of the network if its performance is not good.

D. Communication Infrastructure

Finally, in this section we describe the communication infrastructure that can be used after the DSO decides which information should be shared between FRs, as to not overwhelm the network. After natural or man-made disasters, the public communication infrastructure often becomes overwhelmed or experiences partial or complete failure. Ensuring a reliable and resilient communication link between Search and Rescue teams is crucial for establishing a shared operational picture among all involved actors, and improving informed decision-making processes [10].

1) Relevant Literature: Within the existing literature, there are numerous studies that highlight the significant influence of communication systems and strategies on the overall effectiveness and success of emergency operations [11].

Wireless Ad Hoc Networks (WANETs) have emerged as an attractive technology for emergency communication [12]. These networks are quickly deployed without the need for pre-existing communication infrastructures, and exhibit important attributes such as high resilience, self-healing, and self-organization, making them particularly well-suited for communication in temporary or rapidly evolving emergency situations. A diverse range of Ad Hoc based networks has been explored to respond to emergency communication needs, including Wireless Mesh Networks (WMN), Mobile Ad Hoc Networks (MANET), Vehicular Ad Hoc Networks (VANET), Fly Ad Hoc Networks (FANET), and Sea Ad Hoc Networks (SANET) [11].

In [13], a resilient easily-deployable network solution is presented to address the problem of extending existing communication networks. The proposed solution employs nodes for relaying communications in a mesh network, installed on Unmanned Aerial Vehicles (UAV)s, mounted on tripods on the ground, or even on Unmanned Ground Vehicles (UGV)s. Each node is simultaneously a Wi-Fi Access Point that enables connection to the mesh network and provides local Wi-Fi network access for the ground elements.

A LoRa/IEEE 802.11s mesh-based networking architecture for UAV swarms is proposed in [14]. The proposed solution facilitates seamless data transmission between drones and ground control center by exploiting the different strengths of

the two communication protocols. Different data rates and operational ranges can be achieved, depending on environmental conditions and the specific scenario being addressed.

2) Communication Tool Description: Our communication tool provides a self-organised intra-FR communication network, that allows FRs to collaborate efficiently and securely without using any pre-existing communication infrastructure. This network is provided by a customized communication module, the Communication Gateway (CGW), hosted by each team member, compliant with FR's Personal Protective Equipment (PPE).

The communication between tools hosted locally by each team member is ensured through a PAN, when the different modules connect to the Wi-Fi Access Point (AP). When connected to this PAN, local tools can exchange messages and share information using a service provided by the DSO (see sub-section II-C). Through this service, the CGW also shares status information such as the list of connected nodes, network metrics, battery status, or internal temperature.

For intra-team communication, there is a need to establish communications in an Ad Hoc mode, where devices (nodes) interact directly instead of using a central wireless router or access point. The Local Area Network (LAN) standard for wireless devices IEEE 802.11s, an IEEE 802.11 amendment for mesh networking, meets project requirements that mandate the use of standard communications protocols to provide the needed infrastructure-less network. While transmission power limitations imposed by regulatory requirements limit the range of these networks, the IEEE 802.11s standard enables multi-hop communication, introduces wireless frame forwarding and routing capabilities at the MAC layer, which allows for increased coverage (see Fig. 5), and brings new interworking and security [15].

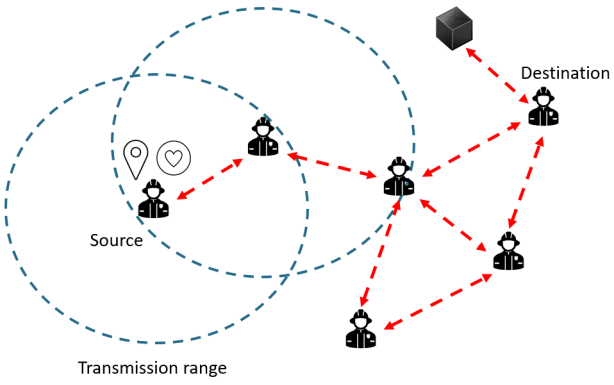

Fig. 5. Typical Ad Hoc (mesh) communication network architecture

The CGW uses the IEEE 802.11s standard to create a self-organized, infrastructure-less communication network supported in a mesh network topology. This WANET is transparent to FRs, as they move during a search and rescue operation, they are dynamically added or removed from the network without any user intervention. The CGW implements also a needed network bridge between PAN and WANET networks, which allows the tools hosted in different FR PANs to exchange messages using the DSO service, and information can then be delivered to the user visualisation modules if needed.

Moreover, The CGW can also work as an external module, acting as a WANET relay to increase the network coverage at a disaster site.

When a reliable infrastructure is available, the CGW can also provide access to a remote control center, supported through a 3G/4G modem.

III. Demonstration Scenarios

Initial prototypes of the *enabling* tools described in this paper and the interactions between them and with other *capability* tools have already been tested in three different scenarios, which will be also used to properly evaluate them in the second phase of the project.

The first use case, takes place in partially collapsed buildings to emulate an earthquake scenario. Urban search and rescue teams with the different Iot-based *capability* tools on them execute specific search and rescue scenarios where, e.g., multiple *capability* tools are used to detect victims behind walls, in smoke, dark conditions, detect dangerous gas leaks, estimate the location of FRs and victims, etc. In those kind of conditions, where an infrastructure does not exist and operations are usually hampered by the adversity of the conditions, all this information needs to be communicated to the FR themselves or between them. Our *enabling* tools are needed to have a communication infrastructure that can be quickly and effectively setup and and enable to send only crucial information that preserves the life of the network and does not overwhelm the FR, especially since e.g., aftershocks of the earthquake could potentially worsen the conditions.

The second use case is an tunnel, emulating a vehicle crash scenario. The *capability* tools in this scenario are helping to detect victims and objects in smoke and dark conditions, to detect dangerous gases, to monitor the vital signals of FRs wearing oxygen masks, estimate the indoor location of FRs and victims, etc. Again, the need of our *enabling* tools is crucial to allow the highest priority information to be communicated inside and outside the tunnel is a seamless and effective way.

Finally, the third and last use case, is a mountain rescue scenario, where a group of hikers got lost in the mountain and need to be rescued with some more injured than others. Again our *enabling* tools need to support the exchange of critical information, such as detecting victims in snow conditions, sharing their outdoor location in an efficient, infrastructure-less and timely way.

In all three scenarios described the presence of our *enabling* tools previously is crucial for the success of the mission, since:

- Communication Infrastructure provides a stable communication among the FRs without the need of an external

infrastructure, to overcome the lack of connectivity in the area of an emergency.

- SA INFOPRIO assigns priorities to the different data coming from various capability tools operating within the FRs and based on the current conditions and the situation of each FR
- DSO allows only the relevant data to reach the FRs on the field, and keeps them focused on the important information.

IV. Conclusion & Future Work

In this paper we have described a communication platform that allows prioritised information to be shared in an efficient and infrastructure-less way.

We have presented an architecture that combines our three *enabling* tools: SA INFOPRIO, DSO and Communication Infrastructure, which can be used in realistic rescue scenarios and can enable various capability tools to utilise IoT technologies and empower the next generation of First Responders to perform their job in a more efficient and safe way.

Our next steps will be to work on techniques and algorithms that optimise our described tools and conduct experiments to validate their efficiency under our three realistic pilot scenarios.

References

[1] Morrison, Kerrianne & Dawkins, Shanee & Choong, Yee-Yin & Theofanos, Mary & Greene, Kristen & Furman, Susanne. (2021). Current Problems, Future Needs: Voices of First Responders About Communication Technology.

[2] Complex Event Processing. https://datainsights.de/complex-event-processing/

[3] Zaalouk, A., Khondoker, R., Marx, R. and Bayarou, K., 2014, May. OrchSec: An orchestrator-based architecture for enhancing network-security using network monitoring and SDN control functions. In 2014 IEEE Network Operations and Management Symposium (NOMS) (pp. 1-9). IEEE.

[4] Iqbal, N., Imran, Ahmad, S., Ahmad, R. and Kim, D.H., 2021. A scheduling mechanism based on optimization using IoT-tasks orchestration for efficient patient health monitoring. Sensors, 21(16), p.5430.

[5] Stavrinides, G.L. and Karatza, H.D., 2021. Orchestrating real-time IoT workflows in a fog computing environment utilizing partial computations with end-to-end error propagation. Cluster Computing, 24(4), pp.3629-3650.

[6] Jaddoa, A., Sakellari, G., Panaousis, E., Loukas, G. and Sarigiannidis, P.G., 2020. Dynamic decision support for resource offloading in heterogeneous Internet of Things environments. Simulation Modelling Practice and Theory, 101, p.102019.

[7] Ahmad, S., Khudoyberdiev, A. and Kim, D., 2019. Towards the task-level optimal orchestration mechanism in multi-device multi-task architecture for mission-critical IoT applications. IEEE Access, 7, pp.140922-140935.

[8] Thean, Z.Y., Yap, V.V. and Teh, P.C., 2019, November. Container-based MQTT broker cluster for edge computing. In 2019 4th International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE) (pp. 1-6). IEEE.

[9] Message Queue Telemetry Transport (MQTT), link https://mqtt.org

[10] Andreassen, N.; Borch, O.J.; Sydnes, A.K. Information sharing and emergency response coordination. Saf. Sci. 2020, 130, 104895

[11] Wang, Q.; Li, W.; Yu, Z.; Abbasi, Q.; Imran, M.; Ansari, S.; Sambo, Y.; Wu, L.; Li, Q.; Zhu, T. An Overview of Emergency Communication Networks. Remote Sens. 2023, 15, 1595.

[12] D. Reina, M. Askalani, S. Toral, F. Barrero, E. Asimakopoulou, and N. Bessis, "A survey on multihop Ad-Hoc networks for disaster response scenarios," International Journal of Distributed Sensor Networks,SAGE Publications Sage UK.

[13] Rocha da Silva, T., Fernandes, L., Gonçalves, J., Chaves, P., Bexiga, V. (2022). Relay Communication Solutions for First Responders. In: Martins, A.L., Ferreira, J.C., Kocian, A. (eds) Intelligent Transport Systems. INTSYS 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 426. Springer, Cham.

[14] Davoli, L.; Pagliari, E.; Ferrari, G. Hybrid LoRa-IEEE 802.11s Opportunistic Mesh Networking for Flexible UAV Swarming. Drones 2021, 5, 26.

[15] Hiertz, G., et al.: IEEE 802.11s: the WLAN mesh standard. IEEE Wirel. Commun. 17(1),104–111 (2010). (E-ISSN 1558-0687).