Smart Devices for Automated Emergency Calls

Mihai Buf[1], Barbara Guerra[2], and Marco Manso[2](✉)

[1] Orange Romania, Bucharest, Romania
[2] Rinicom, Lancaster, UK
marco@rinicom.com

Abstract. The Internet-of-Things (IoT) promises to transform our society into smart environments, incorporating smart objects that cooperate to fulfil specific goals. Amongst its many applications, emergencies can also benefit from IoT principles and use of automation for a better emergency response and reducing the number of fatalities. Smart devices can be used to detect emergency events (e.g., fire, presence of hazardous gases) and automatically trigger alerts to emergency services. However, emergency services today mostly rely on circuit-switch networks and audio-based calls. Therefore, in this paper, we describe our concept to apply the IoT paradigm to the concept of automated calls, in which audio calls are generated from preformatted messages and a text-to-speech engine. Supported by an implemented prototype, our approach brings the benefits of automated calls without requiring significant investments to the infrastructure and systems of emergency services.

Keywords: Internet of Things · Text-to-speech · Emergency services

1 Introduction

The Internet-of-Things (IoT) promises to transform our society into smart environments, environmentally aware and well informed about its wellbeing, interests and security. IoT refers to the extension of the Internet paradigm to the world of objects and places, each able to communicate their own data and access aggregated information from other objects and places. Building blocks of the future IoT, smart objects cooperate to fulfil specific goals [1] finding applications across many societal domains - including agriculture, energy, environment, health and security - providing "live" capabilities such as monitoring, processing and actuation. Their presence is becoming ubiquitous and global: Gartner forecasted that 8.4 billion connected things will be in use worldwide by 2017 and will reach 20.4 billion by 2020 [2].

The emergency services sector is not immune to the penetration of IoT concepts and applications.

Emergency Services (ES) deal with urgent life threatening situations that require a swift response. While they mainly rely on voice calls (via the 112 number for most of Europe), initiatives like eCall, part of the eSafety initiative of the European Commission [3], represents a most notable and successful effort in bringing automated calls and data exchange with ES into reality. The eCall is applied to the specific event of a car accident: if the car's sensors detect a crash, the car automatically calls the nearest

G. Fortino et al. (Eds.): InterIoT 2017/SaSeIoT 2017, LNICST 242, pp. 27–37, 2018.
https://doi.org/10.1007/978-3-319-93797-7_4

emergency centre. Even if no passenger is able to speak, e.g. due to injuries, a 'Minimum Set of Data' is sent, which includes the exact location of the crash site[1], the time of incident, the accurate position of the crashed vehicle and the direction of travel. The eCall concept was designed having in mind circuit switched emergency calls, specifically working in 2G and 3G networks using an in-band modem [4]. Despite, it is expected that eCall cuts emergency services response time by up to 50% in the countryside and 40% in urban areas. It is estimated that eCall can reduce the number of fatalities by at least 4% and the number of severe injuries by 6% [5].

Further applications can be deployed to the benefit of citizens. As proposed in [6], these include eHealth, Smart Building applications and the next generation eCall that fully exploit IoT concepts, IP and packet switched networks.

Despite novel approaches for ES, such as those taken by the EU funded action NEXES (Next Generation Emergency Systems) [7], which embrace IP technologies and data exchange, ES worldwide mainly rely on circuit switched voice calls.

This paper presents a concept supported by a prototype implementation that applies smart devices and IoT concepts to the ES context. It brings the benefits of automated features, sensor data and data processing while supporting legacy voice emergency calls thus able to operate with any ES deployed nowadays.

The remainder of this paper is structured as follows. It starts by presenting a fictional but representative use-case of a smart plant that had a chemical leak incident causing harm to humans. It then presents the overall concept to generate automated calls (from sensors to ES), followed by describing the system high-level architecture, including interfaces, protocols and components. The system workflow is then described, presenting the sequencing between the relevant steps, followed by the structure of the emergency messages. This paper finalises with the conclusions.

2 Emergency Use Case: Smart Chemical Plant

This section describes the use-case of a chemical plant that uses smart sensors and automated alerts to generate automatic emergency calls in a situation where human intervention is not possible. It is based on a use-case described in [8].

In the early hours of the morning, a tank at a small chemical plant develops a breach and hazardous materials begin leaking out. The security guard investigates the incident but fails to follow the appropriate security precautions and is quickly taken ill upon inhalation of noxious fumes. The guard attempts to call for assistance using his mobile device. Before being able to initiate the call, the guard loses consciousness.

Because the NEXES System has been adopted at the chemical plant, an automated emergency call via the NEXES telematics function is triggered following the detection of unusual and dangerous levels of chemicals in the air by installed sensors. Text-to-speech (TTS) is generated to convert a pre-defined message - describing the emergency type and, if relevant, incorporating sensor data - into audio in a form that can be understood by call takers operating in legacy ES. Upon receiving this automatic call,

[1] https://ec.europa.eu/digital-single-market/en/ecall-time-saved-lives-saved.

the emergency call taker develops awareness of the emergency situation and dispatches relevant personnel and resources, including specialist containment and decontamination crews and personal protective equipment.

This *futuristic* scenario highlights the potentially vital role of telematics in specific emergency scenarios. The use of telematics and automated calling means that in situations in which citizens are incapacitated, emergency services can still be timely contacted and properly informed about the incident, thus enhance the emergency response in terms of response time, the safety of the citizen and, in this situation, the safety of first responders as well.

3 Automated Emergency Calls for Legacy ES

The concept described in this paper consist in deploying IoT systems to perform sensing and incident detection together with components capable to trigger specific actions - specifically initiate an emergency call - and convert digital messages into audio in a form that can be received and understood by any ES nowadays.

The system concept overview is presented in Fig. 1. It comprises one or more connected sensors (i.e., smart devices) responsible for measuring specific physical quantities of interest and detecting key events (e.g., heart rate monitoring, seismic activity, gas concentration and smoke detection). The sensors communicate with a Gateway/Hub that collects sensor data and can further run data fusion and processing capabilities that may contribute to increase detection reliability (e.g., both fire and smoke sensor detect a relevant event). The Hub also has the capability to generate event messages into audio (using its TTS engine) and, for our specific use-case, initiate emergency calls[2] using the public network.

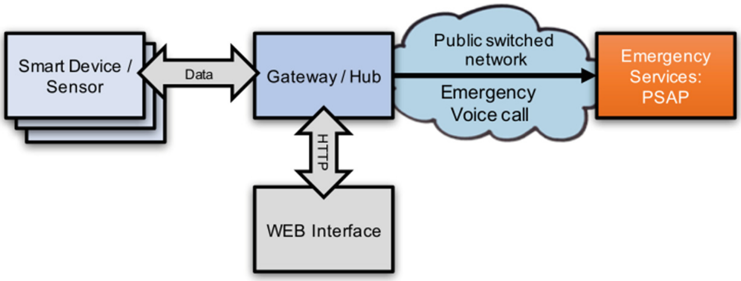

Fig. 1. System concept overview

Additionally, a web interface is devised allowing the user to perform management functions such as system configuration, reporting, alert recipient configuration, etc.

[2] Other approaches could instead opt for calling the owner phone or a service operator.

The chosen configuration allocates eventual "heavy" computational tasks, such as data processing and TTS, to the Hub. The sensors are often resource constrained devices[3] thus their requirements are "lightweight".

4 System Architecture, Interfaces and Protocols

Our proposed setup for automated emergency calls considers smart devices in a cluster configuration connected to a Hub (within a LAN or via a router). The Hub manages the sensor clusters and performs functions such as sensor data processing and initiation of emergency calls. Note that in our setting, sensors are installed in fixed locations. Next, we present the high level architecture of the system (focusing on the Hub components) and describe its main interfaces. The high level architecture is depicted in Fig. 2.

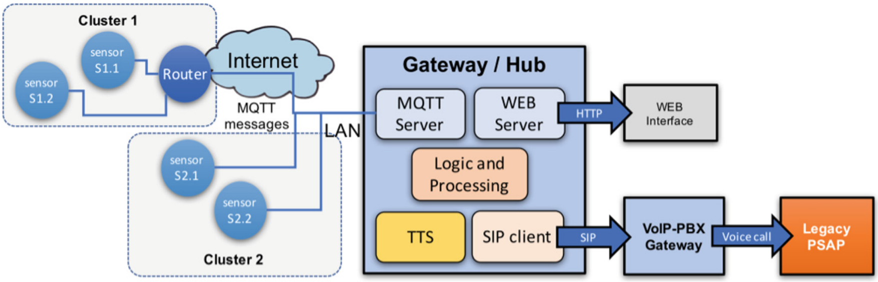

Fig. 2. High level architecture

4.1 Interfaces and Protocols

All-Over-IP.
The Internet Protocol (IP) is chosen as common protocol between all devices and components within the system. IP has become a dominant technology worldwide successfully linking billions of devices over the Internet and within private networks. IP is used to interconnect smart devices, the Hub, web-clients and initiation of calls (using SIP).

Sensor Data Exchange Middleware: MQTT.
The Message Queue Telemetry Transport (MQTT) is a message broker middleware based on the publish-subscribe paradigm standardised as ISO/IEC PRF 20922 (ISO/IEC 20922:2016). It has become very popular in the IoT community given its lightweight Client Server publish-subscribe messaging transport protocol, making it fit for resource-constrained devices.

[3] The resource constraints can be related to low processing power, low memory and low power, as well as network constraints.

Smart devices share information using MQTT, by publishing information to specific topics. The Hub receives data from sensors by subscribing to those topics.

Management Interface: HTTP.
The HUB system management is performed over the Hypertext Transfer Protocol (HTTP) supported by most web browsers.

Emergency Calls: SIP, TTS and PBX Gateway.
Our concept involves the automated establishment of an emergency call to emergency services (specifically, a Public Answering Safety Point (PSAP)) in case of an emergency. For this purpose, the Hub supports the SIP protocol, which is widely used for IP telephony all over the world. Additionally, to "reach" the legacy public switched telephone network (PSTN), a VoIP-PBX gateway is added to convert SIP signalling and VoIP to a form supported by PSTN. To generate audio, the Hub uses its TTS engine.

4.2 Hub System Architecture

The Hub architecture is designed taking in consideration the following main requirements:

- The Hub shall connect multiple sensors and smart devices via MQTT protocol;
- The Hub shall be able to process sensors' data and reliably determine the occurrence of an emergency situation;
- The Hub shall be able to establish SIP calls;
- The Hub shall generate TTS to generate audio to be streamed as part of a SIP call (the audio message should contain caller information and type of incident, including "live" sensor data if relevant);
- The Hub shall store information about caller - such as name, address, location - and make it available for the PSAP in case of emergency.

The Hub main components allowing to realise the above requirements are described next.

MQTT Server.
The MQTT Server manages publishers, subscribers and data exchange. It is responsible for message management, based on the publish/subscribe paradigm, disseminating messages to subscribers when these are published.

In our configuration, each sensor (i.e., publisher) has a specific topic to which it publishes messages (i.e., sensor data or alert). Via MQTT, the Hub allows authorised users to subscribe to sensor messages to e.g., monitor the area and be notified in case of emergencies. Internally, the "Logic and Processing", described later, also subscribes to sensor messages.

SIP Client.
The Hub includes a SIP client in order to be able to initiate SIP calls in case emergencies are detected. The SIP Client performs all required steps for setup (e.g., registration) and manage calls (e.g., initiation, establishment and termination).

TTS.
The TTS engine allows to convert text messages to audio. It requires significant processing and storage capabilities (typically above 1 GHz processor and several Mbytes of storage for audio voice files) typically not present in sensor devices.

Web-Server.
The Web-server allows the remote configuration and management of the Hub (and the system). It allows the user or administrator perform functions like system configuration (e.g., security, setup users, setup sensors), generate reports, configure recipients of alerts, "live" monitor the system, etc. It also allows to setup emergency events (e.g., fire), workflows and associated messages (see Sect. 5).

Logic and Processing.
The "Logic and Processing" is a central element of the Hub. It receives sensor data (via MQTT) and processes it to determine the occurrence of an emergency event and, if so, triggers related actions such the automatic initiation of an emergency call. In order to ensure high levels of reliability (even a small number of false positive or false negative detections are not acceptable in ES) techniques such as Complex Event Processing (CEP), which combine and correlate data from multiple sources to identify events, might be explored, as proposed in [10].

5 Workflow

The workflow related with our implementation of automated calls based on smart devices is depicted in Fig. 3 and described next.

The process starts with the **initialisation** procedure where the system setup and configuration is performed, including establishment of MQTT topics (to where sensors publish messages), SIP client registration, and the fulfilment of required authentication steps. After initialisation, the system enters a continuous loop. In it, each smart device **measures** their parameters of interest (for example, temperature, gas concentration or smoke detection) and publish the **value** using **MQTT** according to the defined criteria (e.g., raw measurement, measurement when above a certain threshold value or event) and to the appropriate topic.

The Hub **reads the sensors' values**, by receiving MQTT messages published in topics it subscribed.

The Hub then processes received messages to determine if there is an emergency situation (when out of "**value safe side**"), which can be based on direct sensor data (e.g., a smoke sensor produces a message with value "true" for fire indication), a threshold value for sensor data (e.g., the sensor temperature is above 60 °C) or the result from correlating data between multiple sensors (e.g., temperature sensor report values above 60 °C and smoke sensor indicates presence of fire). It is critical that this step is highly reliable otherwise, in case of false positives, false emergencies are reported (resulting in unnecessary deployment of resources) or, in case of true negatives, emergencies are missed (resulting in loss of material and possibly lives).

In case an emergency is detected, an emergency call is automatically initiated. In addition, a **local alarm** can be triggered as well to pre-defined recipients.

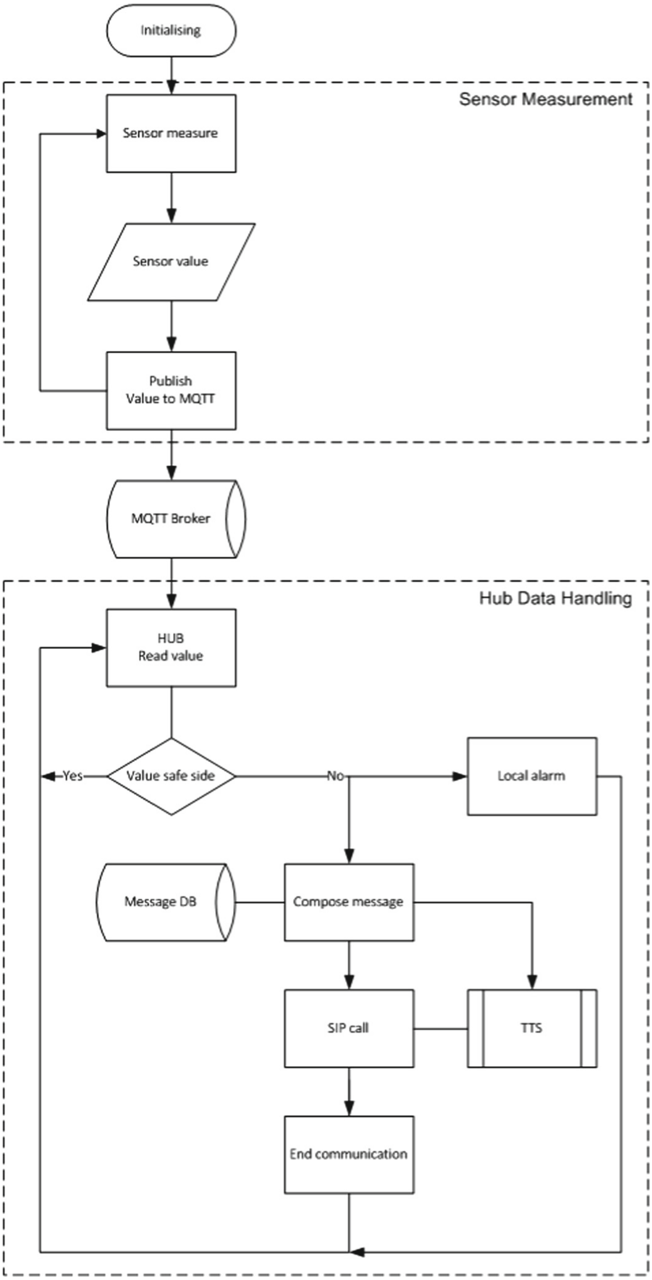

Fig. 3. Automated call workflow

The following steps are taken when **establishing an emergency call**:

- Based on the emergency type and using a database of predefined messages (**message DB**), **a text message is composed** containing pertinent information about the emergency, which includes type of incident, date&time and sensor information (where relevant), but also caller information (name, address and location);
- Initiate a **SIP outgoing call** to the emergency services (which will use the VoIP/PBX Gateway to reach the PSTN and subsequently the PSAP);
- Once the call is initiated, activate the **TTS engine** to convert the text message to audio (and thus automatically report the emergency in a human understandable form). To ensure the message is received and understood by the PSAP (human) operator, it can be replayed a number of times (in our case, we set to 3 times);
- Terminate the call.

6 Messages

During an emergency call, audio is generated from a text message that is built based on the type of event. Given the context of our application, the generated message needs to be simple, easy to understand and contain all relevant information for emergency services to react. As such, the text message should contain:

- The indication that it is an automatic message;
- Information about the type of emergency;
- Information about the subscriber/user (subscriber name);
- Location information (civic address);
- Optionally, associated sensor information when useful.

Example of pre-formatted messages, for situations of fire detection and abnormal temperature detection, are presented next. Note that dynamic fields inside brackets are filled based on configuration values or sensor data.

Fire emergency predefined message:
sipspeakfire: linphonecsh generic "speak default This is an automated emergency call from {{subscriber name}} home, address is {{civic address}}. Fire detected inside home. Please help."
High abnormal temperature predefined message:
sipspeakfire: linphonecsh generic "speak default Environment alarm at {{subscriber name}} home, address is {{civic address}}. The temperature is {{tdata}} degree. The humidity is {{hdata}}%. Please help."

7 Prototype Results

We implemented a simple prototype consisting of several deployed sensors (e.g., temperature, humidity, smoke, fire) and one Hub. For management, configuration and remote access the “Home Assistant” automation platform[4] was used.

For test purposes, a simulated PSAP supporting SIP was used (thus no VoIP-PBX Gateway was deployed). All data was exchanged within a local network.

Figure 4 presents collected sensor data in graphical form pertaining to smoke, temperature and humidity sensors. On top, it is shown the moment an event is triggered caused by the temperature sensor being above a predefined threshold (in our example, set to 35 °C, bar changing from blue to orange). The event initiated a SIP call with the simulated PSAP. The conveyed message was a “High abnormal temperature”, containing temperature data, in audio form.

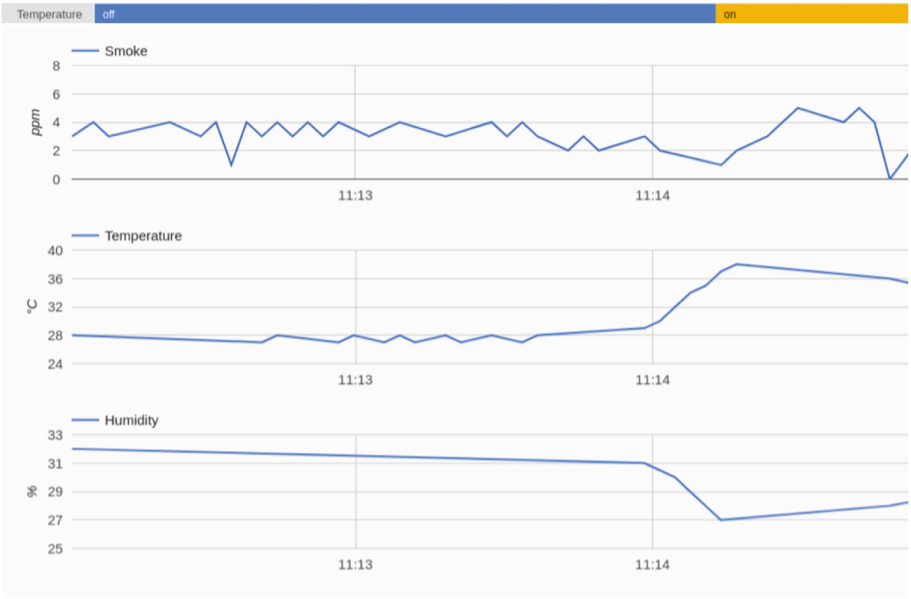

Fig. 4. Example of sensor data visualisation and event triggering (Color figure online)

8 Conclusion

The IoT paradigm and increasing presence of smart devices across all sectors of society continues to find new applications including in emergencies. The eCall initiative brought the concept of automated calls in situations of serious car accidents, aiming to improve emergency response time and reduce the number of fatalities. It is reasonable

[4] https://home-assistant.io.

to expect that other types of emergencies may benefit from this concept as well. In this paper, we presented the application the IoT paradigm to the concept of automated calls, specifically addressing smart buildings, where information provided by smart devices can be used to detect emergency events, such as fire and chemical gas contamination. Leveraged by widely used technologies and protocols in IoT (e.g., IP, MQTT), our application considers the legacy nature of ES and PSAP specifically, which mainly operates with audio-based calls. As such, we proposed a (simple) concept in which audio calls are generated from preformatted messages and TTS engine. In this way, IoT and the concept of automated calls can be implemented without requiring significant changes (and investments) to the ES infrastructure and systems.

Future work will include the application of CEP for improved reliability and security recommendations to protect the system from potential misuse and/or malicious exploitation. Furthermore, considering the forecasted evolution for next generation ES to be implemented over the coming decades, as envisaged and explored in the NEXES Action, we will explore in future work the use of IP and data exchange to data-enabled PSAPs, following concepts such as the next generation eCall [9] and smart environments [6].

Acknowledgment. This paper has been prepared as part of the NEXES Research and Innovation Action, which has received funding from the EU's Horizon 2020 research and innovation programme under Grant Agreement No. 653337.

References

1. Fortino, G., Guerrieri, A., Lacopo, M., Lucia, M., Russo, W.: An agent-based middleware for cooperating smart objects. In: Corchado, Juan M., Bajo, J., Kozlak, J., Pawlewski, P., Molina, Jose M., Julian, V., Silveira, R.A., Unland, R., Giroux, S. (eds.) PAAMS 2013. CCIS, vol. 365, pp. 387–398. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38061-7_36
2. Gartner, Inc.: Forecast: Internet of Things — Endpoints and Associated Services, Worldwide (2016)
3. https://ec.europa.eu/digital-single-market/ecall-time-saved-lives-saved
4. 3GPP TR 26.967 V8.0.1: eCall Data Transfer; In-band modem solution
5. European Commission: Road safety statistics: what is behind the figures? La Valette, 28 March 2017. http://europa.eu/rapid/press-release_MEMO-17-675_en.htm
6. Manso, M., Guerra, B., Carjan, C., Sdongos, E., Bolovinou, A., Amditis, A., Donaldson, D.: The application of telematics and smart devices in emergencies. In: Gravina, R., Palau, Carlos E., Manso, M., Liotta, A., Fortino, G. (eds.) Integration, Interconnection, and Interoperability of IoT Systems. IT, pp. 169–197. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-61300-0_9
7. NEXES. http://nexes.eu
8. Manso, M., Amditis, A., Carjan, C., Donaldson, D., Guerra, B., Jigman, A., Sdongos, E.: The application of telematics and smart devices in emergencies. In: Proceedings of 2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI), Berlin, Germany, 4–8 April 2016

9. Gellens, R., Tschofenig, H.: Next-Generation Pan-European eCall, draft-ietf-ecrit-ecall-07.txt, The Internet Engineering Task Force, 19 February 2016. https://tools.ietf.org/html/draft-ietf-ecrit-ecall-07
10. Chen, C.Y., et al.: Complex event processing for the Internet of Things and its applications. In: 2014 IEEE International Conference on Automation Science and Engineering (CASE), pp. 18–22, August 2014