

Research Article

Next Generation Emergency Services Based on the Pan-European Mobile Emergency Application (PEMEA) Protocol: Leveraging Mobile Positioning and Context Information

Urban Sedlar ¹, James Winterbottom,² Bostjan Tavcar,³ Janez Sterle ⁴,
Jaka Cijan ¹, and Mojca Volk ¹

¹University of Ljubljana, Slovenia

²Deveryware S.A., France

³Administration of the Republic of Slovenia for Civil Protection and Disaster Relief, Slovenia

⁴Internet Institute, Ltd., Slovenia

Correspondence should be addressed to Urban Sedlar; urban.sedlar@fe.uni-lj.si

Received 30 October 2018; Revised 4 February 2019; Accepted 5 March 2019; Published 24 March 2019

Guest Editor: Maurizio Casoni

Copyright © 2019 Urban Sedlar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we analyze requirements of next generation 112 emergency services in the era of ubiquitous mobile devices and sensors and present the design, implementation, and piloting results of our testbed, which was developed within the H2020 project NEXES. The system leverages a multihop location-aware PEMEA routing network that finds the geographically closest responsible public service answering point (PSAP) and supports cross-border application roaming. Our reference mobile implementation utilizes multiple device and network-based positioning technologies, which, combined, both outperform traditional cell-tower based positioning and provide a means for detecting fraudulent calls. The system is extensible and can establish a variety of communication channels after the initial emergency session is set up; we demonstrate this with an interoperable WebRTC-based video call. The obtained results demonstrate the viability and flexibility of PEMEA-based over-the-top emergency services, show high user acceptance when comparing them with existing solutions, and thus pave the road for further rollout of such systems.

1. Introduction

In recent years, ubiquitous sensing and positioning have enhanced several applications for general population. For example, web search engines commonly tailor results to user's location; nearby events and stores are promoted on social networks and through advertising platforms. Commercial applications even extend to indoor positioning based on advanced Wi-Fi network models and Bluetooth beacons [1].

However, emergency services throughout the world are still largely relying on network-based mobile phone tracking, which either utilizes cell identification, cell coverage, triangulation, or trilateration. Such approaches yield accuracy in

the range of a couple of hundred meters at best, which is much worse than the previous landline number-to-location mapping. For improved emergency services a better location is needed, which has to be provided either by a denser deployment of base stations [2] or by other sensors or mechanisms, implemented in the handset itself [3–5].

The latter option is already viable today; off-the-shelf mobile handsets commonly use a Global Navigation Satellite System (GNSS) receiver [6] and can look up the location of nearby Wi-Fi access points [7] and cell IDs in large crowdsourced databases provided by companies such as Apple, Google, or Combain. General-purpose emergency call solutions that leverage such handset-based positioning today are either SMS- or HTTP-based or encompass proprietary

applications that are tied to a specific PSAP (e.g., a national mobile application tied to a national PSAP).

In this paper, we describe how we deployed and tested an end-to-end system for data-based emergency calls based on a new roaming-capable standard for Pan-European Mobile Emergency Application (PEMEA) [8]. The described testbed comprises a mobile application with a backend server, PEMEA network nodes, a PSAP system, and multiple over-the-top and legacy communication services, ranging from a PSTN and SIP calls to a WebRTC video call.

In Section 2 we examine the related work in this area. Section 3 outlines the requirements that we elicited from various system stakeholders. Section 4 describes system design and Section 5 the implementation details. In Section 6, we evaluate the system from technical and user perspectives. In Section 7, we provide concluding remarks.

2. Related Work

Multiple initiatives and projects have been trying to address the problem of contacting emergency services and providing better information faster [9]. The NG112 Long Term Definition [10] document by the European Emergency Number Association (EENA) focuses primarily on bringing together today's heterogeneous telecommunications systems, such as legacy telephony networks, telco-managed IP-based voice services (such as Voice over LTE), and a dedicated Emergency Services Internet Protocol Network (ESInet). Most of these systems are telco voice centric and will be interconnected using gateways and a number of specialized functional elements for call routing, media transcoding, location information exchange, and similar. As a vision for the future, integration with over-the-top (OTT) systems is also foreseen, such as web-based solutions, but these are currently nonexistent in practice and only recently have standards begun to emerge. Various research projects are extending this space in the directions of accessibility for the elderly and people with disabilities, social media integration, haptics, automated calling, and similar [11–14].

Rudimentary data exchange, which is possible in such systems, already provides a range of possible benefits over voice calls, starting with more efficient and less error-prone data collection when every second counts [3, 15, 16], automated triggering as it happens in the vehicular domain with eCall [17–19], to the possibility of automated triage on the PSAP side in case of large-scale events [20].

The actual mechanics of data delivery are usually solved on a national level, with integrations between operators and PSAPs, either by event-based message delivery or by the use of location and information platforms with a central database (e.g., the Polish PLI CBD).

A notable and more recent standard in the mobile domain is the Advanced Mobile Location (AML) (ETSI Technical Report (TR) EMTTEL-00035), which is considered as an enhancement of legacy emergency systems. Today, AML is supported in majority of mobile devices on the market (Android and iOS based), but only a handful of countries

currently support it. AML typically uses an invisible-to-the-user text or binary SMS [21] to send the emergency-related data when an emergency number is dialed. It can be configured to post the data to a REST API as well.

Other innovative solutions in this space include using text-to-speech to relay preset information and location information to the dispatcher automatically using voice, as well as using proprietary information repositories to store such information. Another possibility is the E.164 Number to URI Mapping (ENUM) standard that maps phone number identities to records in the Domain Name System (DNS); this is already implemented in some countries and could be used for the discovery of a Location Information Server (LIS), storing the user's location. In our opinion, these solutions are a poor fit in terms of both required flexibility and regulations.

In addition, none of these solutions has adequately addressed the aspects of international roaming. This is relevant not only due to the increasing number of travelers, but also because it ultimately enables mobile application interchangeability. A small PSAP can thus leverage a higher-quality app developed by another region or country. A proposal addressing this problem is the recently published PEMEA standard [8, 22, 23], which is also an important first step towards the interconnection of the telco emergency infrastructure systems with the OTT world. In this paper, we try to demonstrate the viability of PEMEA in practical scenarios. Further, we aim to demonstrate that it can be extended with support for various communication technologies; we focus specifically on web-based OTT technologies that can provide an emergency services ecosystem that is secure and reliable and at the same time sufficiently open for new developers and services.

3. Requirements Engineering

We defined our design guidelines and constraints through the process of requirements engineering, with the help of a number of real world emergency systems stakeholders, including 112 PSAP operators, first responders, third-party solution developers, and equipment vendors. During the process, we have organized multiple workshops and tabletop exercises for collection of requirements, as well as feedback, both before and during the system implementation. Besides the usual requirements of reliable operation, user-friendliness, providing adequate feedback to the caller, support for GNSS positioning, and support for video communication to cater especially to the deaf and hard of hearing people, we also identified multiple less obvious requirements and features:

- (i) Positioning using multiple technologies (both device based: GNSS, Wi-Fi, and mobile network based)
- (ii) Medical data transfer with adequate data privacy: on-demand transfer of data only to the ambulance service dispatchers, bypassing intermediaries
- (iii) Ability to capture as much contextual information as possible (battery level, network type, data downlink, and uplink speeds) to guide further communication decisions and constraints

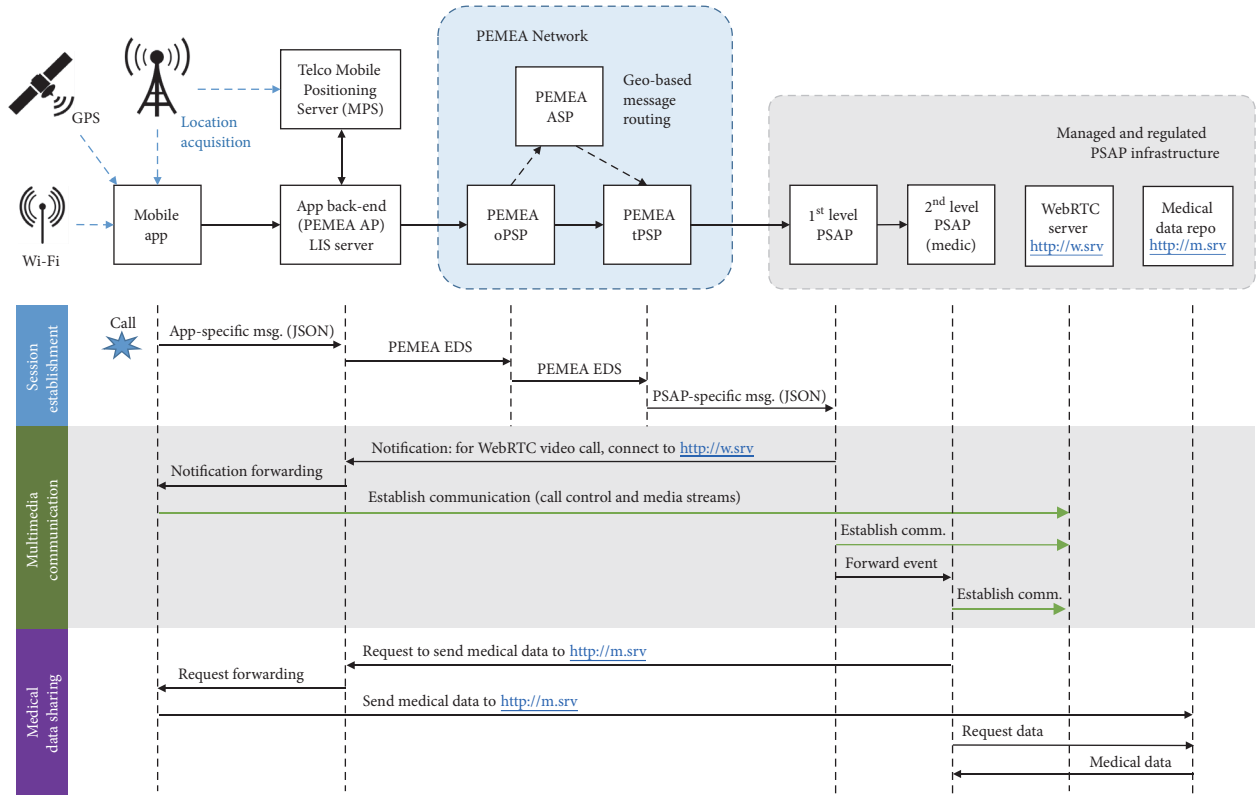


FIGURE 1: Pilot system architecture and main data flows. Green arrows denote media (audio/video) streams.

- (iv) International roaming, to enable an app connected to its own backend to be used abroad
- (v) Effortless extensibility with PSAP’s own content and web-based services, such as web-based surveys
- (vi) Chats, media (images and uploaded videos) should be stored on the premises of the destination PSAP
- (vii) All audio and video communication should be recorded and stored on the premises of the destination PSAP.

Based on these requirements, the system design was iteratively refined until the requirements were met and most of the existing and desired procedures could be executed from beginning to end.

4. System Design

The final end-to-end architecture of the system is presented in Figure 1, together with three main groups of message flows. In the following sections, we describe each subset of the architecture.

4.1. PEMEAs Network. The core of the architecture is shown in a blue block (Figure 1). PEMEAs architecture consists of basic routing elements, PEMEAs Service Provider (PSP) components. Each PSP can be either an entry node for emergency messages (termed originating PSP or oPSP), or an exit node that talks to a PSAP (in this case it is called a

terminating PSP or tPSP). An identical node can operate at a higher level and forward messages between PSPs; then it becomes an Aggregating Service Provider (ASP). A message with a predefined format is passed between the PSPs and ASPs, called the *EmergencyDataSend* (EDS) message. EDS is defined with an XML schema in the PEMEAs specification [22] and includes geographic coordinates of the sender, as well as some routing-related information (time to live, list of hops, etc.). All information passed between PEMEAs nodes is communicated through SSL REST APIs that are mutually authenticated using X.509 certificates.

To secure the system, PEMEAs expects all entry and exit nodes to trust their sources and destinations, respectively, by using a Fully Qualified Domain Name (FQDN) whitelist.

4.2. Data Source and Destination. The sources of emergency messages in the EDS format are called Application Providers (AP). In most cases, a source is expected to be a mobile app backend (relaying data of all of its users), rather than an individual mobile app. Mutual authentication and whitelisting mechanisms make it easy to block a rogue backend that is sending out a large amount of fraudulent calls. To store the locations, a Location Information Server (LIS) can be used, which was in our case collocated with the AP.

On the other side of the network, an exit node is expected to be a PSAP. There are no data format requirements here, and integration is expected to be done on a case-by-case basis with existing proprietary PSAP software. Additionally,

PSAPs typically operate in different hierarchies, according to national specifics. The received data can be forwarded between PSAPs, making it possible to share the same set of data and communication channels.

4.3. Multimedia Communication. The PEMEA system in itself serves only to establish a basic session. No multimedia or additional communication is sent through the PEMEA nodes. Rather, the source AP provides a set of callback Uniform Resource Identifiers (URIs) that can be used to establish multimedia streams, trigger calls, and exchange ancillary data. Figure 1 shows how a media session establishment by means of a callback URI takes place; the PSAP notifies the AP directly that a Web Real-Time Communication (WebRTC) session is taking place in a WebRTC web application on a provided Uniform Resource Locator (URL) address. Once both the mobile application and the PSAP console open the same web view, a communication session is established using web-based technologies. This makes it easy to deploy not only real-time multimedia communication, but also text chat, or any kind of web-based collaboration. Other actors (such as 2nd level PSAPs, medics, and firemen) can also participate in the discussion by opening a unique URL of the chat room with a temporary session token. Since any actual communication service is hosted on a managed and regulated infrastructure of the PSAP, it also allows session recording and largely solves the problems of data management, data retention, and auditing.

4.4. Sensitive Data Exchange. Exchange of any kind of data after the delivery of the initial PEMEA message happens directly between the PSAP and the AP. However, to prevent even the AP from having access to the sensitive user-related data (photos, media, medical information, and similar), a more secure mechanism was implemented, as evident from Figure 1, as follows: a request for sensitive data is sent to the AP and then relayed to the mobile app. This request includes an endpoint under the control of the PSAP, where the data can be safely deposited. In the next step, the app asks the user for permission to deposit private data, and, if permission is given, the data is sent directly from the mobile app to the PSAP-controlled endpoint.

5. Pilot Implementation

The following sections present details about our pilot implementation that was undertaken with the end goal of being able to evaluate the system in a large demonstration event with real-world stakeholders.

5.1. Mobile Application. The Android mobile application features a large SOS/112 (112 is the standard European emergency number, equivalent to 911 in the US. More information at <http://www.eena.org/>) button to start a call. In addition, the app includes a detailed profile form where the citizen can beforehand provide all relevant contact or medical information (see Figure 3). The data model for this was inspired by Apple's Medical ID on iOS, a feature that also hints how

the future emergency services apps could look when deeply integrated into the mobile OS.

5.2. Mobile App Backend. The app backend (AP) serves as a link to the PEMEA network and as a relay to reach the mobile app from the outside world when it is behind a NAT/Firewall. In addition, we integrated it with the telco infrastructure to obtain another location data point from the mobile network itself. This data can either validate or invalidate the location reported by the phone. For each emergency session, the backend server generates multiple tokens and sets up multiple proxy URLs, such that all communication sent to the proxy URLs is relayed back to the mobile app. Finally, the backend assembles the PEMEA *EmergencyDataSend* message, which it then forwards to the *originating* PEMEA Service Provider node (oPSP).

5.3. PEMEA PSP with GIS-Based Message Routing. PEMEA PSP element was developed from scratch according to the specification [22] and tested during interoperability tests with the NEXES [14] project partners. The PSP element receives the incoming EDS message, parses out the location, and forwards the message, to either a known endpoint, another PSP, or an ASP element (default gateway) when no better route is found. The PSP was set up to serve 13 regions in Slovenia (Figure 4) based on their *shapefiles* and a PostGIS query was used to determine if a location falls within any of the regions.

5.4. PSAP. To receive the calls and initiate further communication, we developed a simplified PSAP console with a Hyper-Text Markup language (HTML) based frontend. The use of web technologies made it easier to upgrade the established emergency sessions to other web-based channels, such as HTML information display, WebSockets-based chat, and WebRTC video communication.

5.5. Multimedia: WebRTC Server. Multimedia communication was implemented through WebRTC, an HTML5-based real-time communication stack that is available in most modern web browsers. We used WebRTC for its ease of creating multiparty video chatrooms and the requirement of simplicity to join such calls on the side of the first responders; in case of WebRTC, only a modern Web browser is needed.

However, WebRTC is by design peer-to-peer, which means that media cannot be easily recorded, even when required by regulations. Thus, we used a slightly modified architecture, with a server placed in lieu of a peer; the server then acts as a central point of communication, providing audio/video mixing and recording. A mobile app screen with an active video chat room based on the open-source Kurento WebRTC server is shown in Figure 2(d).

5.6. Other Web-Based Services. By using a flexible web view, many more modes of communication can be implemented using the standard popular web development stacks. A static HTML-based notification is shown in Figure 2(a); in non-critical situations, interactive apps such as web-based surveys can help with crowdsourced data collection (Figure 2(b)).

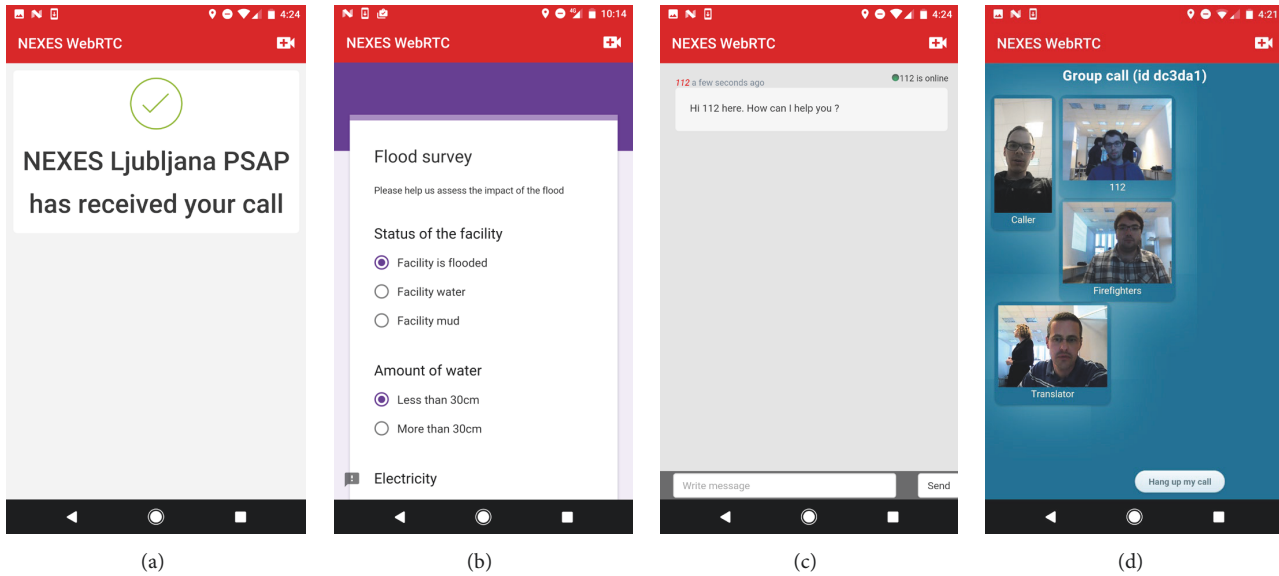


FIGURE 2: Full-featured Chromium browser inside the mobile application, for hosting different extensions. From left to right: (a) web-based notification; (b) a survey; (c) WebSocket-based chat application; (d) WebRTC-based multiparty video.

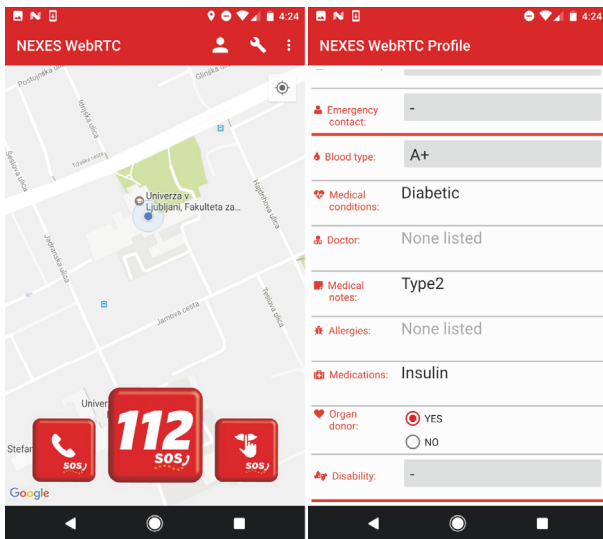


FIGURE 3: Citizen Application for Android: main screen with location display and emergency call button (left); user information screen (right).

Similarly, we developed a web-based chat, leveraging WebSockets for communication. The mobile side of the app can be seen in Figure 2(c). But the high pace of emerging web technologies already makes it possible to develop and deploy Web Graphics Library (WebGL) based visualizations and Web Virtual Reality (WebVR) or Web Augmented Reality (WebXR) scenes that could in the future help users by displaying virtual reality worlds or overlays.

6. Evaluation

Evaluation was performed at our public demonstration event with participants and observers from all major stakeholder

groups (citizens, including special needs minorities, as well as national police, fire brigade, ambulance service, municipality disaster relief team, equipment vendors, and telcos).

The example depicted in Figure 5 shows a deaf person who requires a sign language interpreter for communication. Once the session is established, the 112 operator is able to see the profile of the “caller” and their disability. The operator can thus decide to establish a video call and then patches a sign language interpreter into the multiparty video call to help with communication. Once the information is received, the caller can remain online as the 2nd-level fire department PSAP joins in on the same video call. All communication is relayed through the central WebRTC server located with the PSAP, which also performs recording. Such setup allows easy participation of any required stakeholder, in most cases without the need for special applications; all that is required is a modern web browser and the knowledge of the temporary session token identifying the video chat room.

In addition, four more scenarios were successfully demonstrated and role-played:

- (i) A mudslide event: uploaded pictures and video clips help emergency services identify scope of the event. The 112 PSAP and a 2nd-level fire department PSAP are involved.
- (ii) A lost person event: The 112 PSAP and a 2nd-level ambulance service PSAP are involved and find the citizen through the periodic location updates.
- (iii) A traffic accident of a tourist: the call is relayed through tourist’s home country and the PEMEA network, arriving to the geographically closest PSAP. The 112 PSAP and a 2nd-level police PSAP are involved.
- (iv) A hiking accident: a diabetic citizen cannot explain what is going on, but a preauthorized medical data

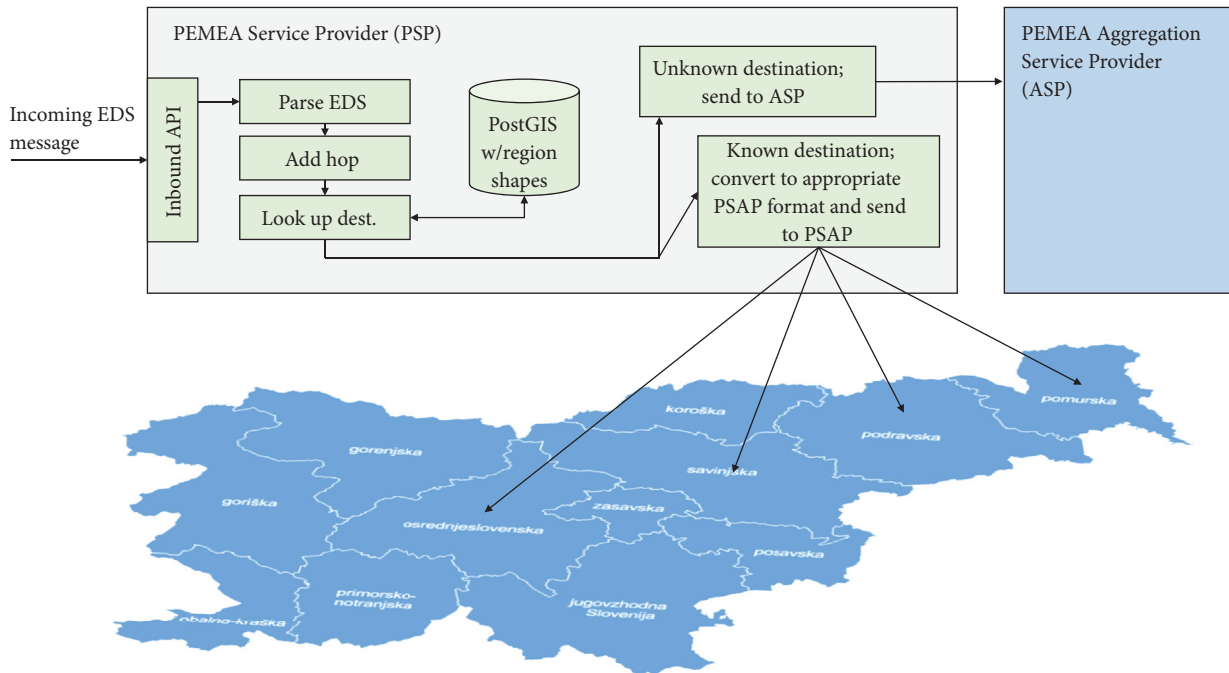


FIGURE 4: PSP architecture with PostGIS shapes for regions in Slovenia. Messages with served location are transformed to PSAP-specific format and sent to appropriate PSAP.

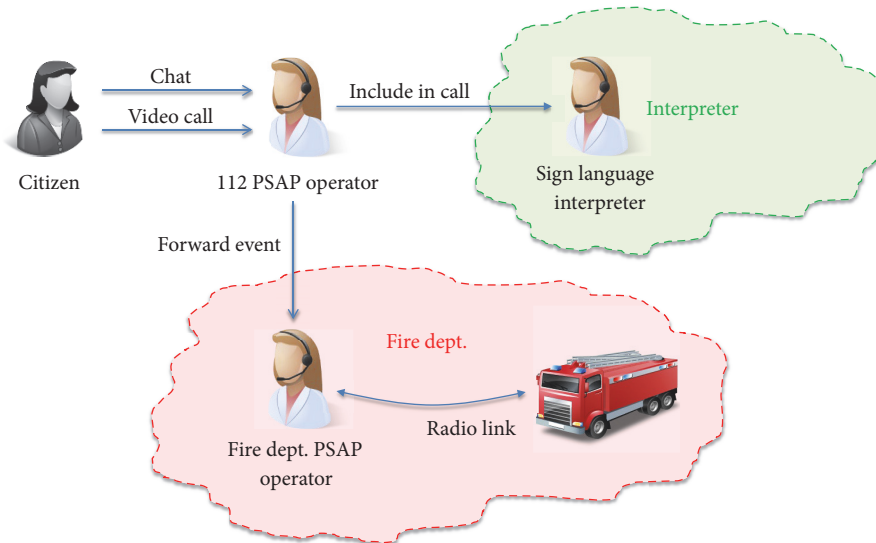


FIGURE 5: Example scenario of a deaf person calling 112 services; a sign language interpreter is added to the video call to help with communication.

request helps the 112 PSAP to determine this is a medical emergency.

6.1. *User Acceptance and Feedback.* After the demonstration event, the participants were able to experiment with the solution and try out the roles on the side of the citizen (mobile app) as well as on the side of the PSAP and

emergency response organizations (ERO). Of over 60 participants, 25 were willing to provide written feedback by filling out one of two types of surveys (from the perspective of either PSAP/ERO, the citizen, or a deaf citizen), rating the improvement of the current situation on a 5-point Likert scale. All three cohorts agreed that the presented features improved upon the existing systems; a detailed breakdown of the questions and the responses is presented in Figure 7.

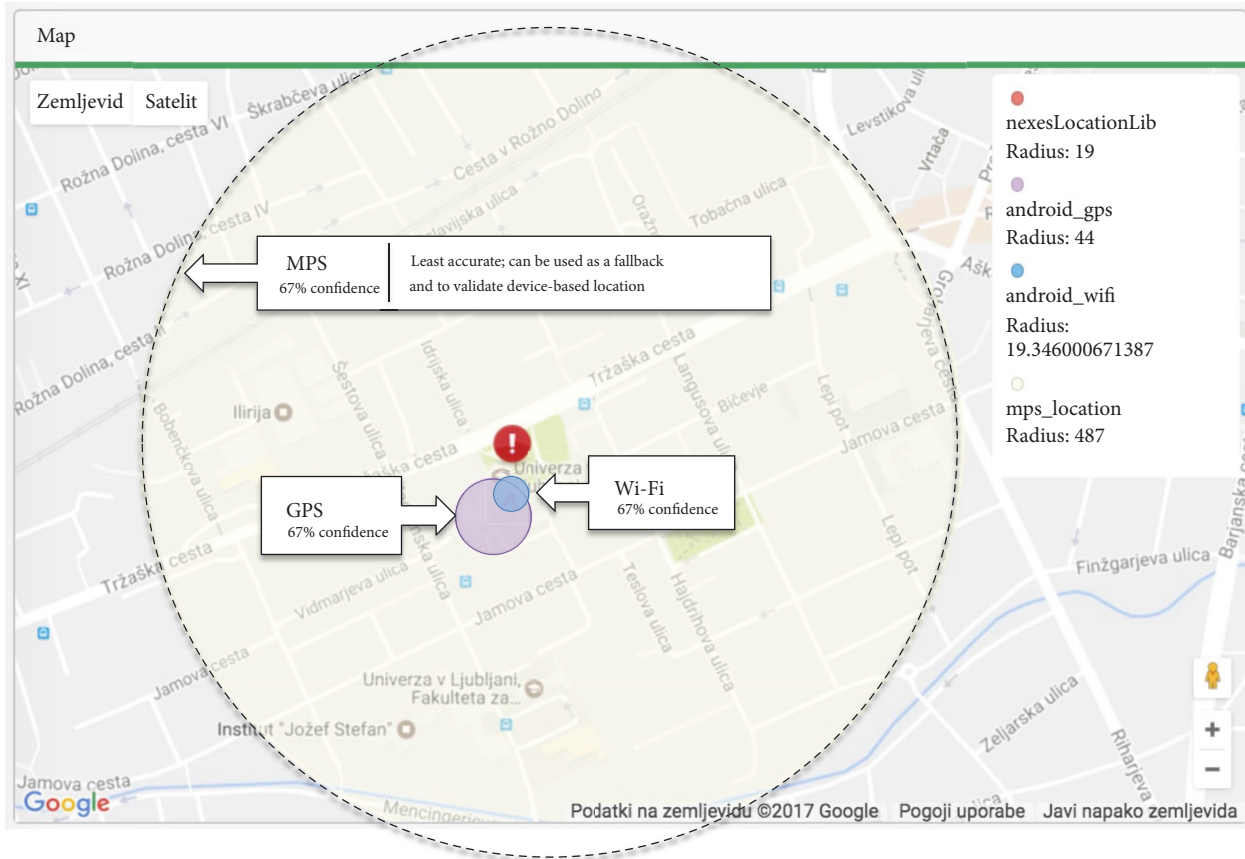


FIGURE 6: PSAP dashboard screenshot, demonstrating all three obtained locations: GPS, Wi-Fi, and telco network-based MPS.

In addition, during the subsequent discussions, the professional cohort expressed that the location feature was most important, followed by media sharing, and, lastly, due to a small percentage of international incidents, roaming. The citizens cohort ranked the importance from better location (most important), chat, international roaming, to video (least important). The deaf cohort ranked video and text chat as the most important features.

6.2. Positioning Accuracy. The following differences between client-based positioning technologies (GPS and Wi-Fi) and an existing telco Mobile Positioning System (MPS) were determined, based on 400 test calls in different locations around our campus (see also Figure 6) (see Table 1).

Best GPS improvement over the actual MPS result was determined as 442x smaller radius (outdoors), yielding 195000x smaller search area; worst improvement of GPS over MPS was only a 2x smaller radius (4x smaller search area). Additionally, best improvement of Wi-Fi over MPS was 148x (22000x smaller search area), but Wi-Fi also performed worse than MPS in 24% of our test cases, which proves it is not very reliable but can still provide valuable insight in majority of the cases.

6.3. International Roaming Capabilities. One of the goals of the pilot was to demonstrate international roaming of

an emergency data exchange (emergency session establishment), which was achieved by leveraging the PEMEA testbed routing capabilities.

For the purpose of the pilot, every emergency session establishment was performed using the PEMEA *Emergency-DataSend* message; in case of national scenarios they were performed through only one, national, PSP element that served as both oPSP and tPSP at the same time. In these scenarios, the main task of the PSP was resolution of local PSAPs based on Slovenian region boundaries.

However, in the international roaming scenario, a foreign citizen used their own national (Spanish) 112 mobile application. This application was by design only able to talk with its own backend in Spain, which had no mapping of Slovenian regions. Spanish backend thus sent data to its only known PSP in Spain, and over a number of hops the PEMEA network elements routed the message to the appropriate PSAP in Slovenia. This is shown in Figure 8.

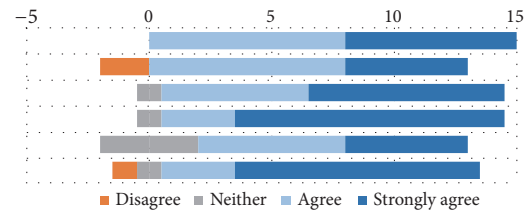
6.4. System Overhead. This section presents the application power consumption, CPU utilization, and overall system delays of the pilot solution. Three consumer mobile devices were used for analysis of the system overhead: HTC 10 running Android 6, Sony Xperia XZ running Android 8, and Google Pixel 3 running Android 9.

TABLE 1

	GPS reported accuracy (radius)	Wi-Fi reported accuracy (radius)	MPS reported accuracy (radius)
Min	1 m (outdoors)	3.6 m	440 m
Max	245 m (indoors)	1700 m	592 m
Mean	38.9 m (all situations)	452 m (all situations)	480 m
Median	15 m (all situations)	37.5 m (all situations)	487 m

Does the demonstrated solution improve current PSAP/Emergency response organization (ERO) capabilities?

PSAP/ERO feedback (N=15)	Strongly disagree	Disagree	Neither	Agree	Strongly agree
Overall	0	0	0	8	7
Text chat only	0	2	0	8	5
Media transfer only	0	0	1	6	8
Video call only	0	0	1	3	11
International call routing	0	0	4	6	5
Improved location	0	1	1	3	10



Does the demonstrated solution improve current capabilities of emergency calling?

Citizen feedback (N=6)	Strongly disagree	Disagree	Neither	Agree	Strongly agree
Overall	0	0	0	2	4
Text chat only	0	0	1	3	2
Media transfer only	0	0	0	3	3
Video call only	0	0	0	3	3
International call routing	0	0	1	2	3
Improved location	0	0	1	2	3

Deaf association citizen feedback (N=4)	Strongly disagree	Disagree	Neither	Agree	Strongly agree
Overall	0	0	0	2	2
Text chat only	0	0	1	1	2
Media transfer only	0	0	1	1	2
Video call only	0	0	1	1	2
International call routing	0	0	2	2	0
Improved location	0	0	1	2	1

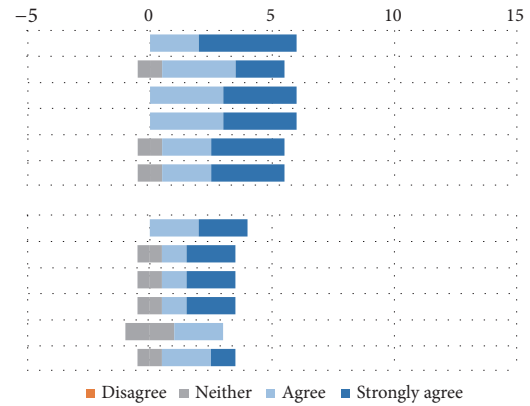


FIGURE 7: PSAP/emergency response organization and citizen feedback regarding overall improvements of the proposed system compared to existing solutions.

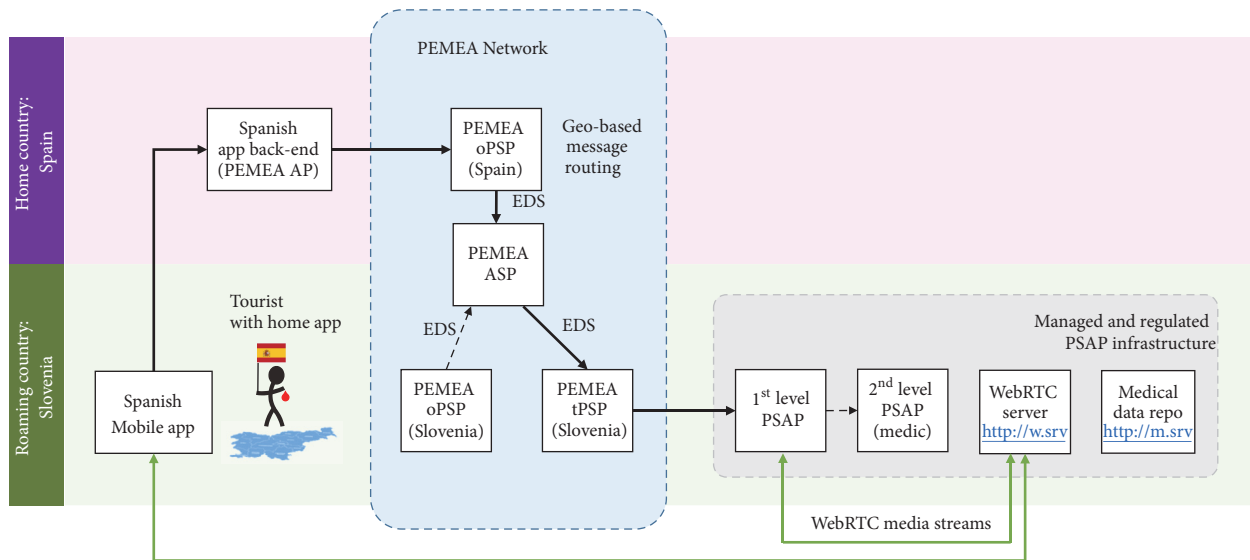


FIGURE 8: Roaming scenario architecture; thick black arrows indicate the path of the EDS message; green arrows indicate WebRTC media streams.

6.4.1. Mobile App CPU Usage and Power Consumption. To estimate CPU usage, Android Debug Bridge was used, with the command `adb shell dumpsys cpuinfo`. For power consumption estimation, we used Digibites AccuBattery, which logs the battery discharge rate reported by the hardware. All measurements were done with the app in foreground and screen brightness set to 80%.

The CPU usage of the citizen mobile app is low in idle state, that is, less than 1%. Similarly, the estimated power usage is negligible, even though the app activates GNSS reception immediately at start-up, to be able to acquire a satellite fix by the time user initiates a call. GNSS does contribute to the power consumption, but since that is the most important information to relay to the PSAP, it is enabled regardless of the battery level.

The majority of the battery drain comes from the CPU usage if a video call is established. CPU usage during WebRTC video call ranges from 80% on newer Pixel 3 to over 120% (more than 1 core fully utilized) on both older models. The estimated total current draw ranges between 850 mA and 1150 mA across the 3 hardware models, which is in line with the observed discharge rate of between 30% and 42% of total battery capacity per hour during a 1-hour WebRTC call.

When the user establishes an emergency session, we also profile the connectivity using Internet Control Message Protocol (ICMP) ping and perform a short downlink and uplink bandwidth test by transferring a small file. The main purpose of this test is to be able to determine if video call is a viable alternative for reaching the user later. Due to the battery drain of video calls, we omit both bandwidth profiling and video call establishment if battery level is below 40%.

6.4.2. System Delays. The delays in the system come from the following parts: (i) initial request transmission, which depends on mobile network latency and bandwidth, (ii) application server (AP) processing time (with added overhead of MPS resolution delay), (iii) PSP node processing (message parsing, validation, and routing) at each hop (multiplied by the number of hops), and, finally, (iv) the PSAP server processing time. Using only one PSP hop between the AP and the PSAP, we measured the average end-to-end message delay in urban environments (4G network) at 1.87 seconds, which includes the network profiling tests. With network profiling disabled, the average end-to-end message delay on a 4G network was 1.36 s from the mobile app to the PSAP dashboard. In contrast, with 3G network connectivity, the average end-to-end message delay was 2.2 s, and, with only 2G connectivity, the average delay increased to 4.90 s without network profiling, which was automatically disabled in both latter cases.

Another significant source of delays was connected to the WebRTC video call establishment, which was enabled on 4G networks only. This took on average 5.8s, which includes pushing the WebRTC app URL to the device, downloading a client-side WebRTC HTML5 app to the phone, assembling a Session Description Protocol (SDP) message, transferring it, and finally establishing a media stream.

7. Conclusions

In this article we presented the design of a next generation 112 emergency system based on the PEMEA protocol. The solution has been designed and implemented from scratch by the partners of the H2020 project NEXES and was extensively tested in a pilot setting at the University of Ljubljana, with a demonstration event with attendance of all major system stakeholders (first responders, PSAP operators, and solution vendors). The viability of the developed system was demonstrated in five complex scenarios, and extensibility of the PEMEA system was confirmed using multiple modes of communication and information sharing. Due to the simplicity of the underlying PEMEA protocol, multiple additional integrations have followed ours in a short time span, and the extensibility of the protocol could easily be leveraged by third-party communication apps to deliver a robust emergency calling experience. The next steps include further development and productization of the components to fit into the nascent PEMEA ecosystem supported by EENA, as well as tackling the usability challenges of the end users where steps need to be taken to raise awareness about emergency apps and make them more user-friendly and generally useful. Finally, we also want to address how the emergency systems like this can leverage telco-grade 5G network features, such as prioritized slices, QoS, direct mode communication, and fine-grained network positioning.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The research and development work was in part supported by the European Commission (Grant Agreement no. 653337), the Republic of Slovenia, and the European Regional Development Fund (project 5GSafety).

References

- [1] Z. Chen, H. Zou, H. Jiang, Q. Zhu, Y. C. Soh, and L. Xie, "Fusion of WiFi, smartphone sensors and landmarks using the kalman filter for indoor localization," *Sensors*, vol. 15, no. 1, pp. 715–732, 2015.
- [2] M. Koivisto, A. Hakkarainen, M. Costa, P. Kela, K. Leppänen, and M. Valkama, "High-Efficiency Device Positioning and Location-Aware Communications in Dense 5G Networks," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 188–195, 2017.
- [3] M. Šterk and M. Praprotnik, "Improving emergency response logistics through advanced GIS," *Open Geospatial Data, Software and Standards*, vol. 2, no. 1, 2017.

- [4] M. Ortiz, M. De Sousa, and V. Renaudin, "A new PDR navigation device for challenging urban environments," *Journal of Sensors*, vol. 2017, Article ID 4080479, 2017.
- [5] Z. Deng, X. Liu, Z. Qu, C. Hou, and W. Si, "Robust heading estimation for indoor pedestrian navigation using unconstrained smartphones," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 5607036, 11 pages, 2018.
- [6] A. Stern and A. Kos, "Positioning performance assessment of geodetic, automotive, and smartphone gnss receivers in standardized road scenarios," *IEEE Access*, vol. 6, pp. 41410–41428, 2018.
- [7] L. Vidmar, M. Stular, A. Kos, and M. Pogacnik, "An automatic Wi-Fi-based approach for extraction of user places and their context," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 154958, 15 pages, 2015.
- [8] "ETSI TS 103 478 V1.1.1 (2018-03): Emergency Communications (EMTEL); Pan-European Mobile Emergency Application," Article ID 103400, https://www.etsi.org/deliver/etsi_ts/103400_103499/103478/01.01.01_60/ts_103478v010101p.pdf.
- [9] F. Liberal, J. O. Fajardo, C. Lumbreras, and W. Kampichler, "European NG112 Crossroads: Toward a new emergency communications framework," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 132–138, 2017.
- [10] EENA, "Term Definition Document," 2012, <https://web.archive.org/web/20180115155320>.
- [11] EMYNOS, "nExt generation eMergencY commuNicatiOnS H2020 project," 2018, <https://www.emynos.eu/>.
- [12] Y. Rebahi, K. T. Chiu, N. Tcholtchev, S. Hohberg, E. Pallis, and E. Markakis, "Towards a next generation 112 testbed: The EMYNOS ESInet," *International Journal of Critical Infrastructure Protection*, vol. 22, pp. 39–50, 2018.
- [13] E. K. Markakis, A. Lykourgiotis, I. Politis, A. Dagiuklas, Y. Rebahi, and E. Pallis, "EMYNOS: Next generation emergency communication," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 139–145, 2017.
- [14] NEXES, "NEXt generation Emergency Services H2020 project," 2018, <https://nexes.eu/>.
- [15] D. Corral-De-Witt, E. V. Carrera, J. A. Matamoros-Vargas, S. Munoz-Romero, J. L. Rojo-Álvarez, and K. Tepe, "From E-911 to NG-911: Overview and Challenges in Ecuador," *IEEE Access*, vol. 6, pp. 42578–42591, 2018.
- [16] I. Osebor, S. Misra, N. Omoregbe, A. Adewumi, and L. Fernandez-Sanz, "Experimental simulation-based performance evaluation of an sms-based emergency geolocation notification system," *Journal of Healthcare Engineering*, vol. 2017, Article ID 7695045, 12 pages, 2017.
- [17] R. Öörni and A. Goulart, "In-Vehicle Emergency Call Services: ECall and beyond," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 159–165, 2017.
- [18] R. Öörni and T. O. Korhonen, "ECall minimum set of data transmission - Results from a field test in Finland," *IET Intelligent Transport Systems*, vol. 8, no. 8, pp. 639–647, 2014.
- [19] M. Cabo, F. Fernandes, T. Pereira, B. Fonseca, and H. Paredes, "Universal access to eCall system," *Procedia Computer Science*, vol. 27, pp. 104–112, 2014.
- [20] A. H. Tapia, N. A. Giacobe, P. J. Soule, and N. J. LaLone, "Texting for large-scale disasters: developing practical technical innovations for emergency management at public universities," *International Journal of Public Administration in the Digital Age (IJPADA)*, vol. 3, no. 3, pp. 73–85, 2016.
- [21] "Advanced Mobile Location (AML) Specifications & Requirements," 2016, http://www.eena.org/download.asp?item_id=165.
- [22] "Pan-European Mobile Emergency Application (PEMEA) Protocol and Procedures Specification," 2018, http://www.eena.org/download.asp?item_id=175.
- [23] J. Winterbottom, B. Casse, C. Lumbreras, P. Sanders, I. Gomez, and P. Kolios, "Pan-European Mobile Emergency Application (PEMEA) Requirements and Functional Architecture," 2018, <https://eena.org/wp-content/uploads/2018/11/Pan-European-Mobile-Emergency-Application-PEMEA-Requirements-and-Functional-Architecture.pdf>.



Hindawi

Submit your manuscripts at
www.hindawi.com

