




# ENSURE - Integration of Volunteers in Disaster Management

Frank Fuchs-Kittowski<sup>1,2</sup> , Michael Jendreck<sup>1</sup>, Ulrich Meissen<sup>1,2</sup>, Michel Rösler<sup>1</sup>,  
Eridy Lukau<sup>1</sup>, Stefan Pfennigschmidt<sup>1</sup>, and Markus Hardt<sup>1</sup>

<sup>1</sup> Fraunhofer FOKUS, CC ESPRI, Berlin, Germany  
{Frank.Fuchs-Kittowski, Michael.Jendreck,  
Ulrich.Meissen, michel.matthias.roesler, eridy.lukau,  
stefan.pfennigschmidt, markus.hardt}@fokus.fraunhofer.de

<sup>2</sup> HTW Berlin, Berlin, Germany  
{Frank.Fuchs-Kittowski, Ulrich.Meissen}@htw-berlin.de

**Abstract.** Volunteers can be a valuable support for disaster management. Disaster management must be able to coordinate volunteers in order to benefit from their support. Interactive, collaborative, and mobile technologies have the potential to overcome the challenges involved in integrating volunteers into crisis and disaster management. This paper presents the ENSURE system, designed to effectively integrate volunteers for an improved approach to disaster management. The system supports the aid forces in recruiting, managing, activating, and coordinating volunteers in the event of a large-scale emergency. To achieve this, ENSURE provides necessary functions such as volunteer registration, volunteer profiles, sending alerts, and volunteer activation (via the mobile app). The system uses a subscription-based approach in which the volunteers agree to take part in an emergency operation by responding to an alert. The system architecture provides technical insights into how to implement crowdtasking systems. It contains seven logical components to provide the necessary features. It was designed to ensure scalability, performance, and availability. The results of a first comprehensive evaluation of ENSURE, which was performed as a large-scale exercise directed by the disaster relief forces in Berlin, were, without exception, positive in all three evaluation areas: efficiency & security, clarity & usability, reliability & availability.

**Keywords:** ENSURE · Volunteers · Crowd tasking · Disaster management

## 1 Introduction

In recent years, more and more citizens are willing to actively help during crises and disasters [1], such as during the flood in 2013 in Saxony-Anhalt [2]. The engagement of so-called “unbound volunteers” can be a valuable contribution while coping with a disaster [3]. Nevertheless, this kind of support can be problematic if it is not coordinated by emergency personnel. The lack of efficient coordination can result in the support being ineffective, with overcrowded or understaffed operation sites. This can lead to

frustration among the volunteers [4] and can hinder the actions of the emergency personnel or may even result in unintentional damages [5].

Modern (i.e., interactive, collaborative, and mobile) technologies can help to effectively involve spontaneous unbound volunteers in disaster management in case of an emergency [6]. Web 2.0 has created concepts of participation, such as crowdsourcing [7], which facilitate the engagement of volunteers and which have successfully been used in crisis management [8, 9]. In addition, the widespread use of mobile devices among the population offers a high potential to change the means of communicating with citizens in the event of a disaster, and to simplify the participation of the citizens as active volunteers [10]. With the help of mobile applications, crisis management can provide current on-site information via mobile devices in real time as well as organize and coordinate the activities of the volunteers at specific locations.

This paper presents the ENSURE system, designed to effectively integrate volunteers for an improved approach to crisis management. The system enables the registration, coordination, and notification of volunteers. The paper is structured as follows: Sect. 2 provides a brief introduction into crowdsourcing and crowdtasking in the context of disaster management and the conceptual architecture of crowdtasking systems is presented. In Sect. 3, the ENSURE system is considered from a user's perspective by discussing the functions it provides. Details on the technical system architecture are presented in Sect. 4. A brief overview about implementation and evaluation of the system is provided in Sect. 5. Related work is discussed in Sect. 6. A summary of the paper is given in Sect. 7.

## 2 Mobile Crowd Sourcing and Tasking in Disaster Management

### 2.1 Existing Approaches and Concepts

There is a variety of approaches to mobile IT systems for the integration of volunteers into crisis management. The majority of these systems are primarily designed for users to collect or assess information on their mobile devices, e.g., CrisisTrackers, Ushahidi, GeoChat, Mobile4D, Cross, Diadem, CrowdHelp, RE-ACTA. This type of involvement, in which simple, digital tasks are performed on-site by volunteers, can be described as "Mobile Crowdsourcing" [11] (or "Mobile Crowdsensing", if data are captured using the built-in sensors of the mobile devices). A sub-form of mobile crowdsourcing is "Mobile Crowdtasking", in which volunteers undertake special physical tasks (such as filling sandbags, providing first aid to injured people, protecting cultural assets, securing dangerous places, etc.) and report on them if appropriate.

Two forms of mobile crowdtasking can be distinguished based on the task distribution scheme [11], with one approach being independent, autonomous task selection and the other coordinated task assignment. In the first case, the volunteers choose their own tasks from a pool of globally available tasks. In the second case, qualified volunteers are efficiently given appropriate tasks with the aim of fulfilling the objectives of the application as best as possible. In a narrower sense of the term, only the subcategory in which the volunteers are not addressed as a group, but rather assigned individual tasks, is referred to as crowdtasking [16].

The potential of crowdtasking systems remains largely untapped with only a few examples of such systems currently in use to assign real, physical tasks to volunteers (e.g., filling sandbags). Scientific approaches and projects for such integration of unbound volunteers are Hands2Help [12], AHA [13], and KOKOS [14]. In addition to these scientific approaches, there are already some projects based on practical experience, such as ZUKS [15], and Team Österreich [16], aimed at the coordinated involvement of volunteers. Other helpful systems such as Mobile Retter, instantHelp, FirstAED, or Pulsepoint (which notifies a registered user of an accident in the area according to the user’s skills) are also aimed at involving and coordinating volunteers, but they are mainly used for *ad hoc* lifesaving, i.e., they are specifically designed for first aid, and not common tasks in crisis management.

All the approaches to mobile crowdtasking mentioned so far have the following points in common: they provide methods and tools to recruit a greater number of volunteers, mobilize and activate them when needed, and coordinate their activities. To achieve this, a specific control system is required in order to distribute the tasks to suitable volunteers. A mobile app is also necessary for the volunteer to receive the tasks, coordinate with others, get involved, and capture and send data or reports to the control system. Additionally, the data captured and the reports sent (e.g. about task fulfilment) by the volunteers have to be analysed and evaluated by the control centre to gain a better awareness of the situation and to support campaign and crisis management.

### 2.2 Conceptual Architecture for Mobile Crowdtasking Systems

Based on these application examples as well as proposals for a general architecture for crowdsourcing applications [11], volunteer management systems [16], and a general process of crowdtasking [17–19], we identify the following components of a general architecture for crowdtasking systems (see Fig. 1).

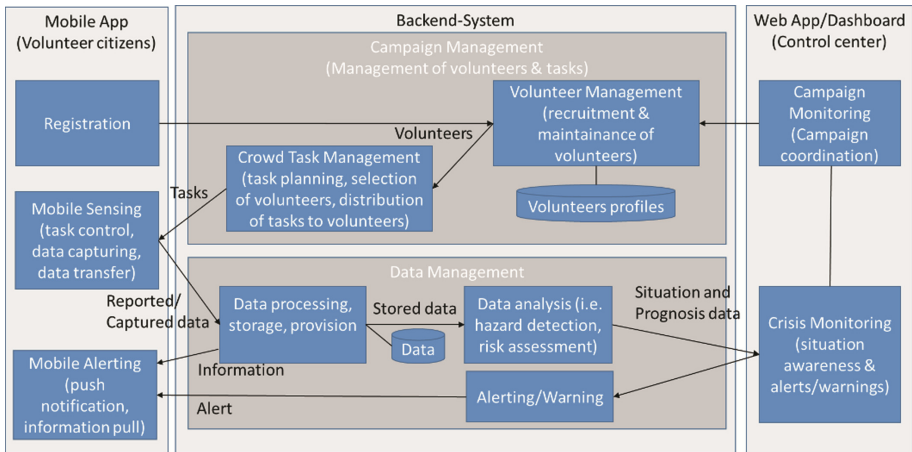


Fig. 1. Conceptual architecture of crowdtasking systems

Typical *roles* within this architecture are “campaign manager”, “crisis manager” (both within the control centre), and “volunteer (citizen)”.

- **Campaign manager (control centre):** initiates and monitors the crowdtasking campaign, including the definition of the campaign as well as recruitment, control, and coordination of well-suited volunteers.
- **Volunteer (citizen):** contributes to the crowdtasking campaign by (voluntarily) accepting and fulfilling tasks using his/her own mobile device. Often volunteers are members of the general public (“unbound volunteers”), but they can also be in a formal relation to the organizer (“formal voluntary engagement”, see [16]).
- **Crisis manager (control centre):** accesses and processes the data captured or reports sent by the volunteers for better situational awareness and crisis management.

The conceptual architecture is divided into three independent *runtime systems*:

- Backend system (server) with two main components: (1) **Campaign Management** provides subcomponents for **Crowd** or **Volunteer Management** (campaign planning, volunteer management, recruiting and maintenance of volunteers), and **Crowd Task Management** (task planning, selection of volunteers, distribution of tasks to volunteers). (2) **Data Management** contains components for data processing (pre-processing, storage, processing, and provisioning), data analysis (i.e. hazard detection, risk assessment), and notification/warning.
- Mobile App (mobile client) for the volunteers with the components for **Registration** (motivation of volunteers, creation and maintenance of a volunteer profile), **Mobile Notification** (push notification, pull information), and **Mobile Sensing** (task control, data capturing, and data transfer).
- Web App (web client) for the control centre providing dashboard features for **Campaign Monitoring** (campaign coordination, monitoring of task fulfilment), and **Crisis Monitoring** (situational awareness, decision support).

### 3 ENSURE System Concept

In this section, the concept of the ENSURE system is presented. The most important functions of the system are discussed from the users’ perspective. The aim of the system is to support disaster relief forces in recruitment, administration, activation, and coordination of volunteers in urban spaces in case of disasters. From the user perspective, the ENSURE system distributes help requests or alerts in the event of a hazard or emergency (Figs. 2, 3, and 4). These requests or alerts are delivered to the volunteers via a mobile app (Fig. 5). Volunteers agree to receive requests through a subscription-based system. Therefore, the system provides certain functionalities such as

- Registration of volunteers,
- Profiling of volunteers,
- Notification of volunteers (see Figs. 2, 3, and 4),
- Activation of volunteers (see Fig. 5)

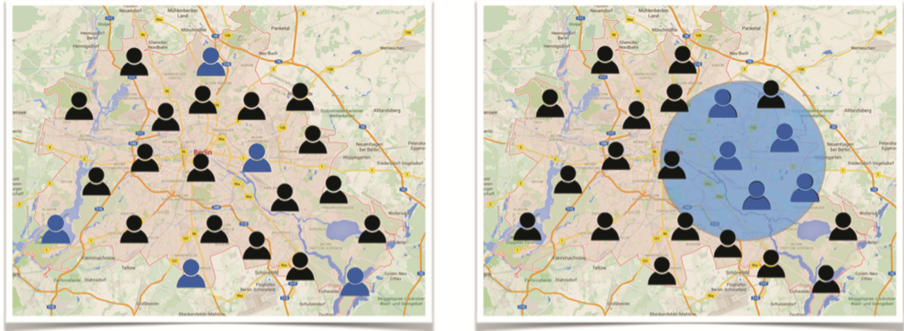


Fig. 2. Alerting of volunteers: coordination system (left: topic-based alert, right: regional alert)

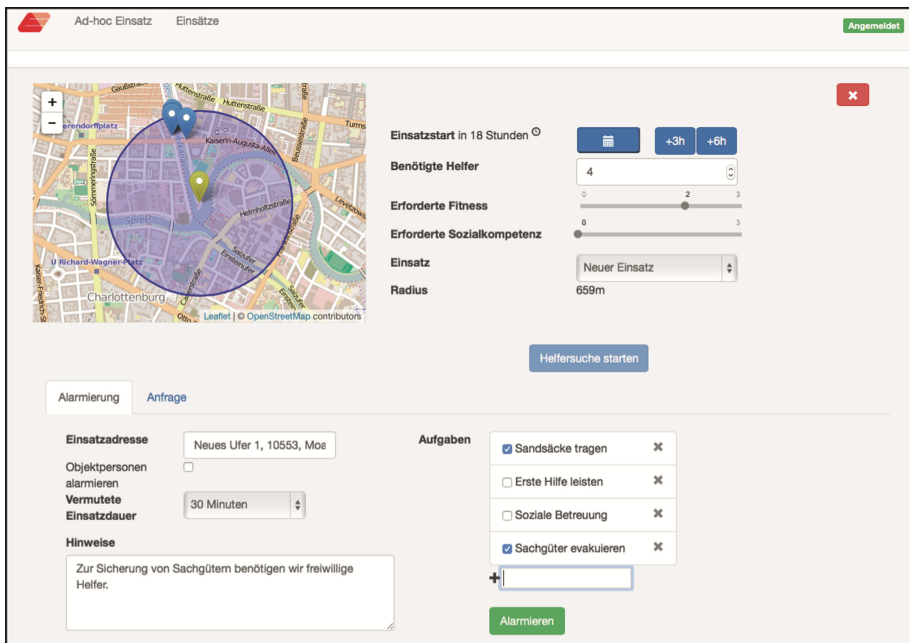
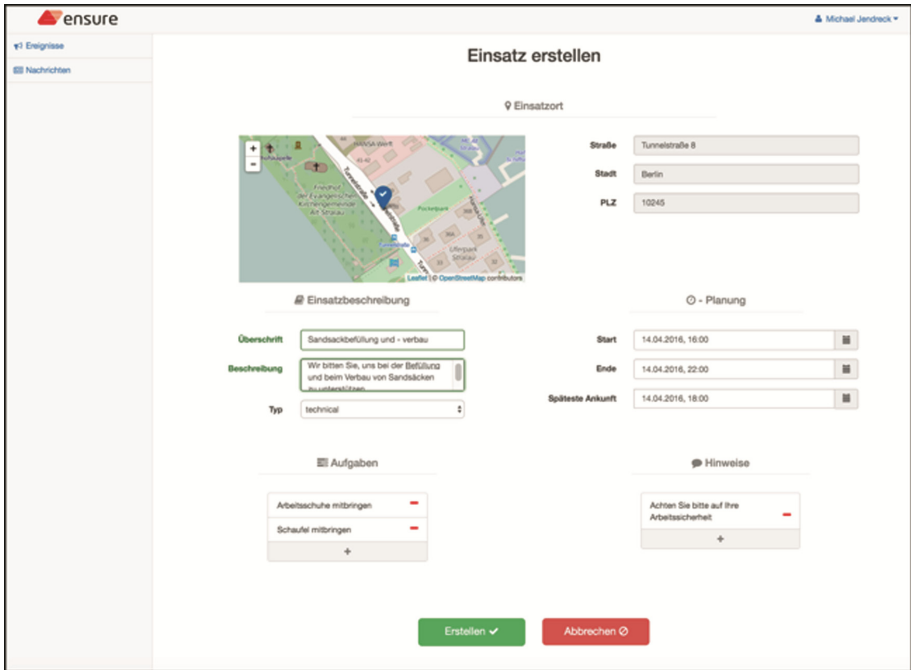


Fig. 3. Alerting of volunteers: coordination system (regional alert)



**Fig. 4.** Alerting of volunteers: coordination system (topic-based alert)

by means of a mobile app for the volunteers (see Fig. 5) and a coordination system (control centre) for the disaster relief forces (see Figs. 2, 3, and 4).

### 3.1 Volunteer Registration

Volunteers must first register via the mobile app in order to receive alerts. When the app is run for the first time, the user is presented with a project guide to encourage him/her to “join in”. If the user decides to take part, this is technically equivalent to registering and is taken as implicit consent to create a user profile.

### 3.2 Volunteer Profiles

One of the basic principles of data protection is data minimization. The ENSURE system implements this as much as possible. Only the following information is stored (anonymously) in the base profile of a user:

- *Current location:* It is necessary to know the approximate current location of the volunteer in order to send regional alerts. With regard to data protection and informational self-determination, a volunteer must actively agree, after registration, to receive regional alerts and accept that their location is therefore always recorded by the system.

- *Fitness level and social skills:* In order to effectively activate volunteers in the event of a hazard or emergency, it should be possible to filter volunteers based on some properties. Information (subjective assessments) regarding physical fitness and social skills are stored in the base profile of the volunteer. This information is based on answers supplied by the volunteer to questions posed during the app setup.

In addition to the base profile, the system allows volunteers to add so-called “profile extensions”. These profile extensions can provide information such as third-party verified qualifications (e.g., first aid), driving license class, technical expertise, etc.

### 3.3 Alerts

The help requests (or alerts) sent by the system to the volunteers are created by the app control centre by means of a web-based control system.

Filtering and alerting of volunteers is a multistep process and proceeds as follows:

- *Alert type:* It must first be established whether to send a regional alert or a topic-based alert.
- *Filtering:* Volunteers are chosen based on the location and time of the emergency, the number of volunteers needed, and any special skills that may be required.
- *Alert details:* Additional information such as specific tasks, estimated duration of the emergency operation, instructions, etc., can be added to the alert.

Volunteers can be alerted in one of two ways:

- The control system can trigger a *regional alert* to all volunteers in a given area (Fig. 2, right and Fig. 3). With this type of alert, volunteers receive a request if they are in the immediate vicinity of the emergency based on their current location.
- Alternatively, volunteers can receive *topic-based alerts* by subscribing to a certain topic, e.g., “Flooding 2013 – Dresden” (Fig. 2, left and Fig. 4). The location of the volunteer does not play a role in this type of alert, and so does not need to be known. The volunteer can freely select which type of alerts to receive in the mobile app.

In addition to regional and topic-based alerts, alerts can be further classified as early warning alerts and *ad hoc* alerts:

- *Early warning alerts:* If the warning period is sufficiently long, the potential volunteers can be requested to confirm their willingness to participate in advance. This procedure can be used for both regional and topic-based alerts and is particularly beneficial for improving planning. A questionnaire can also be sent to the volunteers in order to collect information about special skills that may be required during the emergency operation. The results of the questionnaire are stored as profile extensions in the system and can be used as filtering criteria.
- *Ad hoc alerts:* These alerts do not have a warning period or a planning phase. The idea is that volunteers in the immediate vicinity of an emergency (e.g., a medical emergency) arrive as soon as possible and provide first aid. In order to not lose time creating the alert, only the location and the emergency operation code must be given. If the emergency is assigned an address, this is set automatically as the target location.

Extra details can be entered in a free-text field in the creation form for the alert. As the control system is web-based, all fields of the alert form can be filled with URL parameters. Specialized procedures can pass pre-existing information to the control system via a URL link.

Other functions that are available on the control system include:

- Sending updates to all users,
- Sending additional information about an emergency operation to all users taking part in that operation,
- A detailed view of ongoing emergency operations (including volunteer feedback).

### 3.4 Volunteer Activation

For activation (Fig. 5), the selected volunteers receive a message via push notification on their smartphones. The users are shown all the information about the emergency operation in the app and can decide whether to accept or reject the request. They can also respond to a questionnaire regarding special skills for the emergency operation if one was sent with the alert.



Fig. 5. Volunteer activation via the mobile app



### 4 System Architecture

In this section, the architecture of the ENSURE system is presented (see Fig. 6).

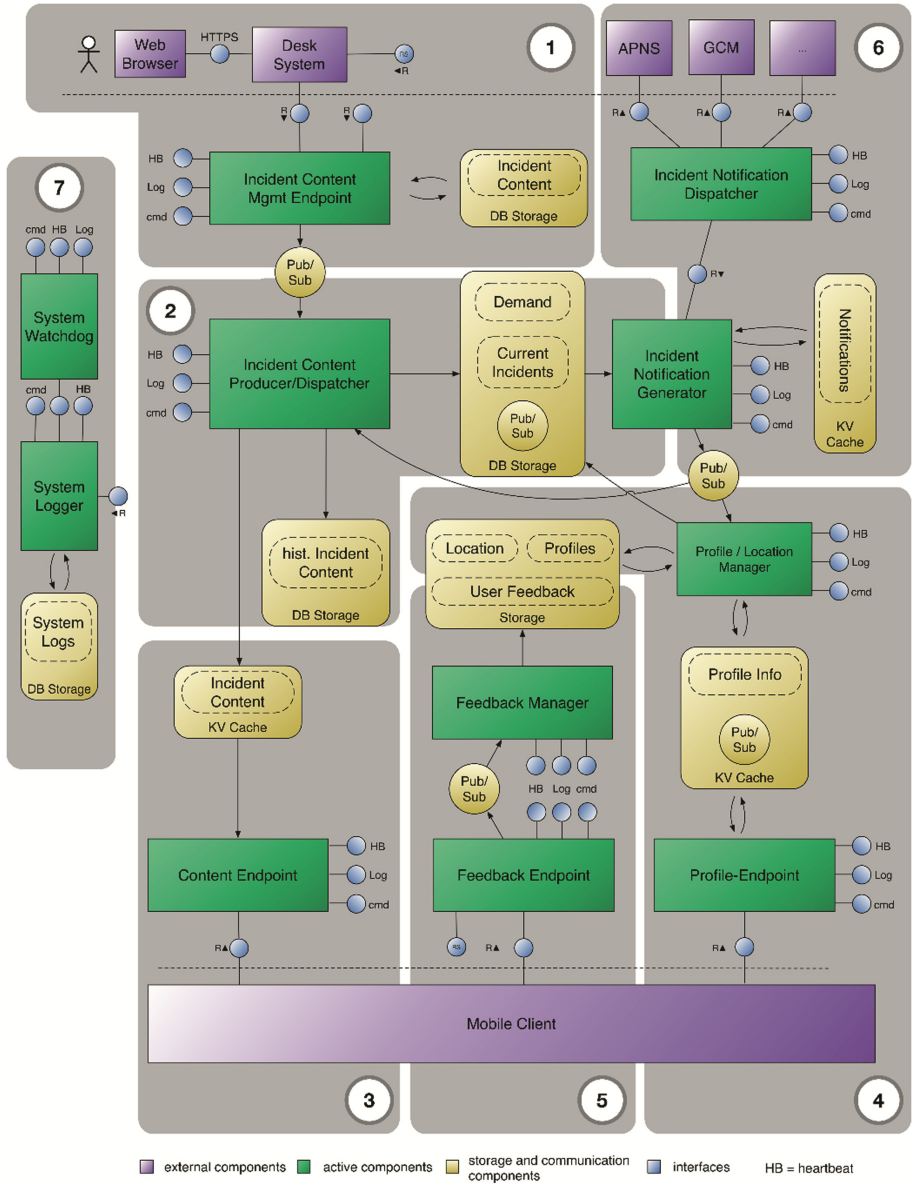


Fig. 6. ENSURE – system architecture (mobile app, web app, backend)

## 4.1 Components

The components of this system can be divided in seven logical parts (subsystems).

- **Accepting and managing events (1).** The *Incident Content Management Endpoint* is responsible for managing incoming events (alerts or requests from the control system). It accepts event messages (raw events) via different external interfaces and saves them in the *Incident Content Storage*. New or updated events are forwarded to the *Incident Content Producer/Dispatcher* and thus to the distribution system.
- **Distribution system (2).** The *Incident Content Producer/Dispatcher* has two main tasks: it prepares the raw data of the incoming events and is responsible for the distribution system.
  - The processing is performed according to how the information will be further distributed. The system stores the data required for sending the notifications in the *Demand/Incident Storage* and informs the *Incident Notification Generator* about the new or updated event. It also generates the data required by the end users for the textual and graphical processing on their devices. These data are made available for retrieval via the *Incident Content Cache*.
  - The *Incident Content Cache* serves as a buffer for content to be delivered (e.g., events such as alerts or inquiries, additional information). The *Demand/Incident Storage* maps subscriptions from system users and event messages in a high-performance runtime model that facilitates efficient matching between both types of information.
- **Content retrieval (3).** The *Content Endpoint* provides the interface through which content may be queried. This includes queries of processed events and relevant additional information. The data are taken from the *Incident Content Cache*. The *Content Endpoint* is designed for maximum scalability and employs dynamic load distribution.
- **Management of personal and device information (4).** The *Profile Endpoint* presents the interface through which the user can query, add, or modify profile information. Similar to the *Content Endpoint*, the *Profile Endpoint* uses a dynamic load distribution in order to enable, among other things, multiple profile updates within a short period of time. The *Profile Endpoint* retrieves answers to queries from the *Profile Info Cache*, which holds frequently requested data as well as current profile information in a high-performance key-value cache. The *Profile/Location Manager* is responsible for managing all profile-related data in the system. User inquiries are forwarded to the manager if they cannot be answered directly from the *Profile Info Cache*. All data are persistently stored in the *Location/Profile/Feedback Storage*. Changes to subscriptions are also sent to the *Demand/Incident Storage* and forwarded to the *Incident Notification Generator* to check if a notification is required.
- **Notification service (6).** The *Incident Notification Generator* is responsible for matching subscriptions (current position) and events.
  - The service has access to the *Demand/Incident Storage*, a special runtime database in which the subscription and event data have already been combined. The main task is to retrieve the mailing addresses of the relevant volunteers for each of the event messages and send these data to the *Incident Notification Dispatcher*. In addition,

this component has a store in which the notification statuses of the individual devices/users are held.

- The *Incident Notification Generator* also informs other system components of which devices are expected to send queries in the near future before triggering the notification delivery. This makes it possible to prepare the system so that requested information is more readily available.
- The *Incident Notification Dispatcher* is responsible for sending the notifications over different channels. It receives the event messages generated by the *Incident Notification Generator* along with the corresponding mailing addresses and sends the notifications.
- **User feedback (5).** The *Feedback Endpoint* allows all user feedback (task accepted/rejected, feedback after completion of a task, as well as feedback regarding the basic profiling) to be received and forwarded to the *Feedback Storage*. Furthermore, the content management system can access all feedback information via this endpoint and prepare it for the dispatchers or use it as filter parameters (profile extensions) for the volunteer search.
- **System monitoring (7).** The *System Watchdog* is an instance of a checking system that checks to see if all active components are working properly within a given timeframe.
  - The so-called *Heartbeat* is a complex object that is regularly sent by each active component to the *System Watchdog*. This object contains, among other things, information on the current status, errors, processed objects, as well as environment variables (such as required memory). With these data, the *System Watchdog* can react in the case of faulty behaviour and take control using the *Command Interface*.
  - The *System Logger* receives log entries from all active components, prepares this information, and stores it in the System Logs Storage. Furthermore, it offers a request interface for retrieving log entries from both internal and external components, e.g., for support.

## 4.2 Performance, Scalability, and Data Security

The system architecture was designed to ensure scalability, load balancing, performance, and availability. In order to realise the performance and scalability of the system, the following design decisions were consistently applied within the architecture and later in the implementation:

- *Key-value caches* are used to quickly respond to queries. These are designed for the purpose of load distribution using multiple distributable instances. The number of instances can be dynamically adjusted depending on the expected or actual load. Additionally, key-value storage instances provide advantages over relational storage instances in terms of scalability.
- All data are persistently stored in databases. This means that all cache instances have persistent data storage. If an (unexpected) restart occurs, the information in the caches can be restored.
- Communication between the individual components in the system backend is asynchronous.

## 5 Implementation and Evaluation

The ENSURE system has been developed in an iterative process. In each iteration, new functionality is added to the system and evaluated. For the evaluation of the system, several alerting exercises as well as two large-scale field tests directed by the disaster relief forces in Berlin (Berlin fire brigade and red cross) are carried out. The results of evaluations are used for improvement of the existing system, which will be evaluated in the next iteration.

The **implementation** of the system is based on the following technologies:

- The **mobile client**, used by the volunteers, was implemented as a native mobile app for the mobile platforms iOS and Android.
- The **control system**, used by the control centre, was implemented as a single-page web application using the client-side JavaScript framework AngularJS.
- The **backend**, providing the functionality for the two clients and ensuring satisfaction of non-functional requirements like performance, robustness, and scalability, was implemented using server-side JavaScript technology included in the MEAN stack (MongoDB, Express.js, AngularJS, and Node.js) as well as Redis as a key value store.

A comprehensive **evaluation** of the system was performed as a large-scale exercise directed by the disaster relief forces in Berlin. In total, 24 volunteers and 120 aid workers took part in this exercise. Volunteers had to fulfil 14 different tasks before the arrival of the aid forces as well as simultaneously with the work of the disaster relief forces.

The evaluation criteria concentrated on the three established requirements on software applications in disaster management [6]: efficiency & security, clarity & usability, reliability & availability. The evaluation results were, without exception, positive in all three evaluation areas. In particular, the app was found to be very usable (SUS score: 90 points), highly stable, and had short response time as well as a good success rate when storing a personal profile. First aid tasks were especially well performed (observation) and the volunteers followed instructions very accurately (clearing vehicle access routes, etc.) (See [20] for details.)

Currently, another large-scale field-test of ENSURE is taking place in Berlin under the name “Support the Berlin fire brigade”. The ENSURE app is available in app stores (iOS and Android) for the field test so that anyone can participate as a volunteer in the exercise.

## 6 Related Work

Overall, the ENSURE approach described above has a number of similarities with existing concepts and systems (see Sect. 2.1), but it also shows significant differences.

The aim of the ZUKS project (Zivile Unterstützung im Katastrophenschutz) is to acquire helpers and to organise and deploy them. Similar to the ENSURE project, notification, organisation, and coordination is carried out using a control system. Activating previously registered helpers and sending task details is done analogously to a smartphone app. However, there is a considerable divergence in the integration of both

systems. ENSURE provides the technical platform to the emergency and rescue services for free use; in contrast, the ZUKS project integrates the separate organisations into the existing system, and then the system manages the helpers. Furthermore, ENSURE is defined by anonymous registration and the continued organisation and coordination of helpers on the ground (including food and accommodation).

Similar to ENSURE, the Hands2Help project offers an app-based coordination and notification system that is designed to support emergency and rescue services in coordinating volunteers. Control centres can define a request for help by filling out a form. Volunteers can offer help and indicate their abilities through the app. If the system detects a match between an offer and a request, the relevant users are automatically contacted. This continues until the appropriate number of suitable volunteers is found. In contrast to ENSURE, no control system is provided for manual notification of volunteers by a control centre since the matching between offers and requests as well as the notification is automatically taken care of by system algorithms. The fact that every user can publish a request for help also differentiates both projects. In addition, Hands2Help requires compulsory data on the availability of the users with regard to time and location. ENSURE, in addition to notifying volunteers based on their location, offers topic-based notifications that do not require the location data of the volunteers.

The AHA project (Automatisiertes Helferangebot bei Großschadensereignissen) and the KOKOS project (Kooperation mit freiwilligen Helfern in komplexen Schadenslagen) both run parallel to the basic idea of ENSURE and also have the goal of involving the population in dealing with damage. However, there are differences in the concrete implementation of this objective. The KOKOS project focusses on integrating the public into crisis management. In addition to the volunteers themselves, the AHA project especially considers the technical equipment of the volunteers to be a useful resource, and it requests and registers availability.

In the Team Österreich project, volunteers can be registered with the Red Cross Austria as helpers and can be notified via various channels (SMS, email, etc.) in case of an emergency. Suitable volunteers are identified for a task, taking into account their place of residence as well as distance to the task location, so that enough volunteers with the right skill set are automatically notified. ENSURE relies on a content management system, which allows control centres to notify a certain number of volunteers and if necessary, request more help. The ENSURE app also offers users the possibility to report a delay in arriving at the task site so that the “non-appearance” of helpers can be taken into account by the control centre and included in the planning and coordination.

## 7 Conclusions

Disaster management is currently changing. On the one hand, citizens are very willing to actively help during crises and catastrophes and support the local emergency services. On the other hand, citizens are also increasingly seen as major players and a key resource in disaster relief. Nevertheless, the integration of volunteers into disaster management is still in its infancy.

There are a variety of approaches to implementing mobile IT systems for integrating citizens into crisis management. However, the potential of crowdtasking systems remains largely untapped with only a few examples of such systems in existence and a low level of interest and consideration in the research community.

This paper presents the ENSURE system. The goal of ENSURE is to effectively integrate volunteers for an improved approach to disaster management. The system enables the registration, coordination, and alerting of spontaneous volunteers. The functionality and architecture of the system are described in detail. ENSURE provides a concrete example that mobile crowdtasking applications can contribute to the effective involvement of volunteers in disaster relief.

A general, conceptual architecture for mobile crowdtasking systems was also proposed in this paper. The characterisation of mobile crowdtasking systems helped to understand the individual components and their attributes as well as the existing relationships and interdependency among the components, as determined by the system as a whole. The generic system architecture allows specific applications to be designed, classified, and evaluated, e.g., the ENSURE system.

The system architecture of ENSURE is not only an example of the use of the conceptual architecture. It is also an example of a system architecture that ensures scalability, load balancing, performance, and availability. It demonstrates the feasibility of developing robust crowdtasking applications. Identifying variables and features that affect the scalability and performance of a system is helpful when planning its design or making improvements to it once it is in operation.

The great potential, but also the associated difficulties, are only slowly coming to light. Questions regarding the validation and quality of the services provided by the volunteers as well as the handling of data protection issues arising from the use of personal data are currently being discussed. In addition, volunteer motivation and incentive systems for volunteer participation are important issues. In the end, evaluations of emergency operations must show how much the operations could be improved by the participation of the volunteers.

The systems that are known so far for the successful integration of the population into disaster relief show the starting points for the necessary change: citizens should not only be seen as recipients, but as an important resource in disaster management. Even though there may already be some examples of citizen integration in crisis management, these are always isolated and individual cases. So far, there is no unified, integrated concept covering all phases of disaster management, in which the different parties – professional forces, volunteer forces, and citizens – are well coordinated.

## References

1. Ohder, C., Röpcke, J.: Hilfebedarf, Hilfeerwartung und Hilfsbereitschaft bei einem Stromausfall. Ergebnisse einer Bürgerbefragung in Berlin. *Crisis Prev.* **2**, 33–35 (2014)
2. Geißler, S., Sticher, B.: Hilfeverhalten in Katastrophen und die Folgen für das Katastrophenmanagement - am Beispiel des Hochwassers 2013 in Magdeburg. *Poliz. Wiss.* **4**, 53–70 (2014)
3. Reuter, C., Heger, O., Pipek, V.: Social media for supporting emergent groups in crisis management. In: *Proceedings of CSCW 2012 Workshop on Collaboration and Crisis Informatics, International Reports on Socio-Informatics (IRSI)*, vol. 9, no. 2, pp. 84–92 (2012)
4. Kircher, F.: Ungebundene Helfer im Katastrophenschutz. In: *BrandSchutz*, vol. 68, no. 8, pp. 593–597. Kohlhammer, Stuttgart (2014)
5. Schorr, C., Bierger, A., Weber, T., Max, M., Schulze, M.: Die Rolle der ungebundenen HelferInnen bei der Bewältigung von Schadensereignissen. Dt. Rotes Kreuz, Berlin (2014)
6. Mauthner, J., Engelbach, W., Engel, K.: Informationstechnologien für das Freiwilligenmanagement in Katastrophenschutz und Krisenmanagement. In: *Engagiert im Katastrophenschutz - Impulse für ein zukunftsfähiges Freiwilligenmanagement*, pp. 165–178. Wochenschau Verlag, Schwalbach (2015)
7. Howe, J.: The rise of crowdsourcing. *Wired* **14** (2006). [www.wired.com/wired/archive/14.06/crowds.html](http://www.wired.com/wired/archive/14.06/crowds.html)
8. Kaufhold, M.A., Reuter, C.: Vernetzte Selbsthilfe in Sozialen Medien am Beispiel des Hochwassers 2013. *i-com* **13**(1), 20–28 (2014)
9. Schimak, G., Havlik, D., Pielorz, J.: Crowdsourcing in crisis and disaster management – challenges and considerations. In: Denzer, R., Argent, R.M., Schimak, G., Hřebíček, J. (eds.) *Environmental Software Systems*, vol. 448, pp. 56–70. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-15994-2\\_5](https://doi.org/10.1007/978-3-319-15994-2_5)
10. Reuter, C., Ludwig, T., Pipek, V.: Ad hoc participation in situation assessment - supporting mobile collaboration in emergencies. *ACM Trans. Comput.-Hum. Interact.* **21**(5), article 26 (2014)
11. Fuchs-Kittowski, F., Faust, D.: Architecture of mobile crowdsourcing systems. In: Baloian, N., Burstein, F., Ogata, H., Santoro, F., Zurita, G. (eds.) *CRIWG 2014. LNCS*, vol. 8658, pp. 121–136. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-10166-8\\_12](https://doi.org/10.1007/978-3-319-10166-8_12)
12. Hofmann, M., Betke, H., Sackmann, S.: Hands2Help – Ein App-basiertes Konzept zur Koordination freiwilliger Helfer. *i-com* **13**(1), 36–45 (2014)
13. Detjen, H., Geisler, S., Bumiller, G.: Nutzeranforderungen eines Systems zur automatischen Helferbereitstellung. In: Weibecker, M., Burmeister, A., Schmidt (eds.): *Mensch und Computer 2015*, pp. 11–18. Oldenbourg, Stuttgart (2015)
14. KOKOS: Project-Website, 31 March 2017. <http://kokos-projekt.de/>
15. ZUKS e.V.: ZUKS – Zivile Unterstützung im Katastrophenschutz – Konzept im Juni 2015, 31 March 2017. [https://www.zuks.org/static/files/Konzept\\_1.2.pdf](https://www.zuks.org/static/files/Konzept_1.2.pdf)
16. Neubauer, G., Nowak, A., Jager, B., Kloyber, C., Flachberger, C., Foitik, G., Schimak, G.: Crowdtasking – a new concept for volunteer management in disaster relief. In: Hřebíček, J., Schimak, G., Kubásek, M., Rizzoli, Andrea E. (eds.) *ISESS 2013. IAICT*, vol. 413, pp. 345–356. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-41151-9\\_33](https://doi.org/10.1007/978-3-642-41151-9_33)
17. Auferbauer, D., Ganhör, R., Tellioğlu, H.: Moving towards crowd tasking for disaster mitigation. In: *ISCRAM 2015 - Proceedings of the 12th International Conference on Information Systems for Crisis Response and Management* (2015)

18. Middelhoff, M., Widera, A., van den Berg, R., Hellingrath, B., Auferbauer, D., Pielorz, J., Havlik, D.: Crowdsourcing and crowdtasking in crisis management: lessons learned from a field experiment simulating a flooding in city of the Hague. In: ICTDM2016- International Conference on Information and Communication Technologies for Disaster Management (2016)
19. Auferbauer, D., Ganhör, R., Pielorz, J., Tellioğlu, H.: Crowdtasking: field study on a crowdsourcing solution for practitioners in crisis management. In: ISCRAM 2016 - Proceedings of the 13th International Conference on Information Systems for Crisis Response and Management (2016)
20. Jendreck, M., Meissen, U., Rösler, M., Lukau, E., Fuchs-Kittowski, F.: ENSURE - increasing resilience by integration of volunteers in disaster management. In: Environmental Informatics – Stability, Continuity, Innovation: Current trends and future perspectives based on 30 years of history, Shaker, Aachen, pp. 131–140 (2016)