Internship Report

# The Everbridge Public Warning Chatbot Initiative

**Koen Vogel**
8 July 2022

**Internal Supervisor:** DR. A.J. van Hessen
**Everbridge Supervisor:** M. Bot

Master
Interaction Technology

University of Twente

# Contents

# 1 Assignment

In short, designing a user-centric platform extension of the One2Many public warning dissemination system, which improves the disaster resilience and reach of the public. This is done by providing more detailed and reliable information and the ability to ask questions to authorities using automated Machine Learning methods.

### 1.0.1 Introduction

One2Many's public warning systems like NL-Alert are effective solutions for alerting citizens to local emergencies, and allow their users to be quickly informed. However, there is limited space for information in broadcasted messages, which means more detailed information and instructions are often searched for elsewhere, like news websites. There are also people who do not fully understand (non-Dutch speakers for example) the message contents. The goal of this assignment is to design a solution to improve the disaster resilience and reach of the public, by providing more detailed and reliable information, and offer more answers to users' questions.

### 1.0.2 Project

A platform is proposed as a trustworthy solution to offer more in-depth information and interaction, which builds upon the existing public warning system (in the form of an attached link or something like). The user should be able to quickly find additional information on the current emergency, as well as concrete instructions that might not fit in the short broadcast message (e.g. in case of fire, close windows and doors and disable air conditioning). Finally, users should be able to ask questions to authorities, which could be handled by an ML-powered chat-bot to save human resources. The platform should be easily accessible and intuitive to use, especially in stressful situations like emergencies.

### 1.0.3 Workflow

My responsibilities would start from the ideation phase of the project, with researching needs and requirements from both stakeholders and end-users. After a first specification has been made, research into technologies and implementation methods can be done. Following a rapid iteration process, several low fidelity prototypes should be designed and made, after which they can be evaluated on end-users. Depending on the scope and time constraints of the realisation phase, a final prototype of relatively high fidelity can be made, which could possibly be integrated into One2Many's existing architecture for evaluation and further development. Along with the internship, One2Many's staff and software developers can help guide me, and provide resources, data and technical support in the research, design and development process.

# 2 Background

This project is a result of several factors and stakeholder, and came to be based on research already done. This sections will describe the different stakeholders, and discuss how this project came to be, starting with the most direct source of previous work, project ENGAGE.

## 2.1 ENGAGE

ENGAGE is an EU-funded project as part of the Horizon 2020 program. Horizon 2020 is EU's research and innovation funding programme from 2014 to 2020 and has a budget of almost 80 billions euro. After bidding, the European Commission awarded ENGAGE with a grant of around 4.8 million euros[1], with the goal of innovating and researching on societal resilience. The ENGAGE consortium itself consists of more than 14 partners, Everbridge being on of them, and utilises the grant to research more than 200 solutions towards societal resilience.

Societal resilience can be defined as "the ability of human communities to withstand and recover from stresses, such as environmental change or social, economic or political upheaval"[2]. More concrete, how societies deal with emergencies and disasters, from terrorist threats to floods and landslides. In the context of societal resilience, ENGAGE focuses on "improving the interaction between first responders, authorities and civil society"[3]. They do this by creating and evaluation a catalogue of solutions based on research, and presenting them on their knowledge platform. They also support the Knowledge and Innovation Community of Practice, which aims at involving end-users in the field, both in validation as well as research phases.

As part of the communication technology research, ENGAGE partners like Tel-Aviv University and SINTEF Norway lay the groundwork for the chatbot project with deliverable D3.2[4]. Here they describe the directions of social media and communication technology, and their applications in societal resilience. They describe issues and opportunities, both for end-users as well as technological capabilities, and propose a chatbot blueprint, as part of the ENGAGE solutions catalogue. Key takeaways from their research, and the concrete blueprint, will be explained in the following sections.

### 2.1.1 Research takeaways

Authorities use chat-bots and AI technology very conservatively. With many fields adopting the fast improving technology, societal resilience is lagging behind. Reasons for this include fear of the extending reach of AI chat-bots and their knowledge field, and the risk of unpredictable results during delicate and critical moments. Doubts over the different technologies are also quoted. Stakeholders and authorities should feel as comfortable as possible, so the level of AI

[1] https://cordis.europa.eu/project/id/882850
[2] https://www.stockholmresilience.org/research/resilience-dictionary.html
[3] https://www.project-engage.eu/knowledge-platform/

complexity has to be taken into account in the development process. On top of this, the authorities should also be informed (to a degree) in the workings of the chatbot, to facilitate understanding, in order to overcome organisational barriers.

More concrete, N. Stolero et al[4] provide suggestions for implementation of a chatbot prototype, relevant for this project. In their blueprint, a combination of closed-form and open questions are recommended as input types for users to interact with the chatbot. They also suggest using multimedia, such as audio and video, as input types. This comes with a side note "However, as the audio analysis may be less accurate concerning a voice of a person the system did not train on, audio inputs should be limited to a predefined set of requests (e.g., recording the sound of water flow after being asked to)."[4, p40]. Another interesting suggestion in the technical blueprint considers bidirectional communication and data-flow. In a practical sense, this means the general public can supply emergency operators and organisations with realtime data during an emergency, such as photos and a GPS location.

## 2.2 One2Many Everbridge

One2Many as a company is formed in 2007, but its components are active in the industry since 1964. They first introduced Cell Broadcast technology, which is used for many Public Warning solutions today. After successful international deployment and growth, One2Many was bought up by Everbridge in 2020. Everbridge is a global player in resilience for companies, offering a range of tech solutions. With its acquisition of One2Many and others, it establishes itself in disaster communication technologies. Within One2Many, there are a variety of departments, see

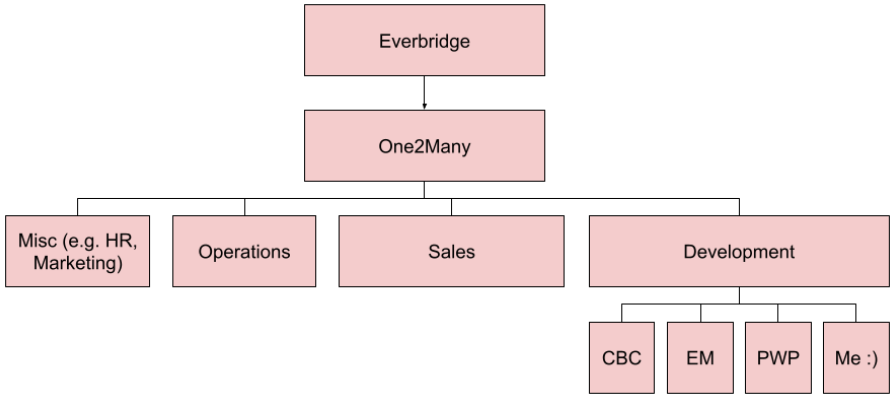


Figure 1: Simplified organisational chart of One2Many

figure 1. These are divided into two spaces within the office; Development, and everything else. The development team, around 14 people, are then dis-

tributed over a few product teams. The ENGAGE chatbot project falls outside these product teams, as its design and development is relatively isolated from other One2Many products. EU-funded research projects like ENGAGE gives One2Many an innovation advantage, as well as placing themselves close to EU policy makers.

# 3 Specification

This section sets out to define the context of the proposed solution, what are the requirements? What are the limitations given to the development and ideation process?

## 3.1 Authority

If the proposed solution is to be implemented in the public warning system, the product will undoubtedly be associated with the party responsible for sending the alert message, such as national emergency authorities or the police. While the discussion if this might be an issue is out of scope for this project, it still adds important responsibilities for designing the system. This responsibility is even greater because the information the system can provide is used in critical and dangerous situations, and can potentially be life-saving. This information must therefore always be true and trustworthy. This can be done by using as much official data as possible, for instance the data provided by the authorities in the form of a Common Alerting Protocol (CAP). Now matter how advanced the design of a Machine Learning powered conversation agent is however, there is inherently a degree of unpredictable results, which cannot be verified to always be trustworthy. The amount of randomness in the chat-bot's output should be kept to a minimum, and the training data used to construct the conversational model should be imported from trusted sources.

## 3.2 Privacy

Another important requirement for the completion of this project is the protection of sensitive private information of the users. As seen in the field of public warning systems, platforms are susceptible to data-leaks and hacks[4], which can be detrimental for the roll-out of such applications, as authorities are already very cautious with new technologies, see section 2.1. Therefore, the protection of user's privacy is of crucial importance. This means not only implementing high grade security and encryption standards, but also limiting the amount of Personally Identifiable Data (PID) collected. The application should process the data locally as much as possible, and only sending PID to the cloud, or server-sided processing, when absolutely necessary.

[4] https://www.rtlnieuws.nl/nieuws/nederland/artikel/5108016/grapperhaus-nl-alert-lek-app

## 3.3 Responsiveness

Because the situations in which the proposed solution is used are potentially dangerous and highly time-sensitive, users should be able to get the required information in a matter of seconds. This means that it should be very clear to users how to launch and use the application. Usability testing and User Experience (UX) design should be done to ensure actual end-users are able to use the product and accomplish their goals as fast as possible. On top of this, the platform itself should meet performance standards as to save time, and not get in the way of users accomplishing their goals. This means that an important rubric should be the launch time of the application, measured on a range of supported devices.

## 3.4 Accessibility

Resilience during emergencies and disasters is important for everybody. We cannot afford to exclude people from using technology and applications like the proposed solution. Accessibility means making the solution work for as wide a range of people as possible, including people with impairments. One of the main advantages of the proposed solution is being able to adapt to these people much more effectively than a single Cell Broadcast Public Warning message can. These adaptations should be made for non-Dutch speakers, people with visual impairments or colour-blindness, people with auditory impairments or deafness and possibly more. On top of this, there is a large group of target users that lack digital literacy, such as senior citizens. While it is impossible to develop a solution that works perfectly for everybody, efforts should be made to make the system as clear as possible for these users.

## 3.5 Limitations

As the proposed solution needs to be accessible through the public warning system provided by One2Many-Everbridge, the design is bound by the interfaces of said system. This means that, especially during later implementation stages, the prototype has to be integrated using existing channels.

# 4 Method

## 4.1 User journey

User interactions are at the centre of the product. To start designing a specific feature set, a clear overview of the user interactions are needed. Who are our users? What are their goals? How do they use the product?

### 4.1.1 User profile

In the context of public alerting, it is very clear that the target user is the general public. Considering the scope of the project, this is narrowed down to

citizens of the Netherlands specifically. This does not out rule future expansion to other regions, as the target audience is overlapping substantially, but does allow for easier implementation of region specific features like translation. It also enables more convenient evaluation procedures with local participants.

While the general public, in the broadest sense of the word, ideally includes every single citizen of the Netherlands, this is an unrealistic expectation. Nevertheless, as section 3.4 describes, efforts should be made to reach as many people as possible. This includes senior citizens, non-Dutch speaking citizens and citizens with visual or auditory impairments.

#### 4.1.2 User goals

As described by Stolero et al[4] in previous work, user needs can be prioritised as follows;

1. Cognitive needs (information)
2. Unidirectional flow of communication needs (fast information)
3. Integrative needs
4. Affective needs
5. Escapist needs

In the context of chatbot technology and public warning, a set of more concrete user goals can be conceived.

- Glance quick information on current emergencies
- Get elaborate information & instructions on current emergencies
- Ask questions and get tailored answers & information regarding current emergencies
- Connect with authoritative sources and human contact points

#### 4.1.3 Interaction flow

From this list of goals, a user journey can be defined in more detail, with the desired interactions between user and product. This can then be used to construct a specific set of features for the conceptual design of a chatbot. With a sample user profile, the journey will explain how the proposed solution should be used, how users can interact with it, and goals it will accomplish. It can be seen in figure 2.

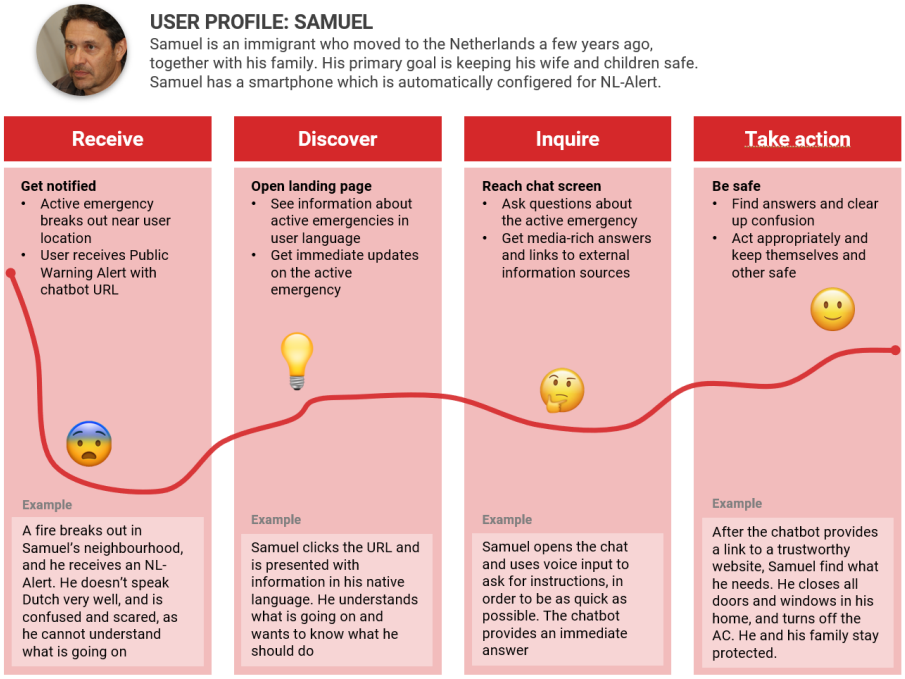


Figure 2: User Journey

## 4.2 Feature set

To support this flow, and keep to the specification, a concrete feature set for a conceptual prototype can be conceived, based on end-user and stakeholder needs. There are more aspects to consider for a feature list, such as scope and technical & human resources. These trade-offs however, are different propositions for the level of production. While the scope of the internship itself, in 3 months, will be too small to accomplish every desired feature, recommendations should be made for future work. So that, if desired, future prototype iterations can build on the presented concept, all the way towards production-ready. Concrete recommendations for future work can be found in section **??**. For the reasons presented above, all features are prioritised on importance and feasibility.

Table 1: Prioritised feature set

| Priority | Feature | Description | Included in prototype |
|---|---|---|---|
| 1 (Required) | Automated text chat | Basic Machine Learning chatbot functionality, and the basis for the project | yes |

| | | | |
|---|---|---|---|
| 2 (Required) | Public Warning Integration | Attaching the product to the existing public warning dissemination technologies, e.g. through a URL, to connect users with the application | yes |
| 3 | Overview page | An addition to chatbot functionality, where the most important emergency information and metadata is displayed without needing to explicitly ask for it | yes |
| 4 | CAP support | Adherence to the CAP standard, for importing and displaying universal emergency metadata | yes |
| 5 | Multi language support | Translating the content and conversation in multiple language | yes |
| 6 | Embedded multimedia | Rich content, like embedded interactive maps, video, audio, etc | yes |
| 7 | Voice Input | Speech-to-Text (STT) functionality for asking questions to the chatbot | no |
| 8 | Analytics | System for logging user behaviour | no |
| 9 | Narration | Text-to-Speech (TTS) functionality for reading out incoming messages and the overview page | no |
| 10 | Visual Accessibility | Colour-blind and Visual impairments display options to increase accessibility | no |
| 11 | Showcase website | Static website showcasing the chatbot features, include (technical) documentation & reporting, and possibly embed an interactive prototype | yes |

## 4.3 The prototype

The prototype can be divided into two parts, the back-end and the interface. The interface should be responsible for interacting with users, while the back-end should include the conversational agent, and be responsible for relating queries to contextual answers.

For the interface platform there are a few options. The chatbot can be run as a mobile application, website, or be integrated into existing messaging frameworks, like Whatsapp and Facebook messenger, 'Omni-channel'. All have there own advantages and disadvantages, but a significant distinction has to made between custom building the front-end interface or utilising third parties. While, the latter has significantly lower costs and solid UX and scaling possibilities, and is recommended by Stolero at al[4, p36], it lacks customizability. As many custom features like voice support, multi-language and embedded multimedia are needed, the prototype includes a custom interface, which should also integrate far better with One2Many's Public Warning system, compared to social media channels. Specifically, the interface runs as a Progressive Web App (PWA).

Being a combination of a mobile application and a website, a PWA is mobile-first but still features good compatibility with desktop and legacy devices. It

should be faster than a traditional website, but there is no need to install it first, which would cripple discoverability and ease of access. There are plenty of software stacks to create a PWA, but for this prototype, flutter was chosen. While its performance, especially for static pages, is slightly worse than other frameworks, it works well enough for more complex and data-heavy web-apps. One additional advantage, is it's cross-platform native compilation capabilities, making it possible to create performant apps running on both mobile, desktop and web environments.

The back-end, arguably the most important part, will be heavily Machine Learning (ML) and Natural Language Processing (NLP) focused. The implementation of this conversational agent be will more elaborately described in section 4.5.

## 4.4 Prototype Integration

One of the main features of the proposed solution is the integration in the Public Warning System of One2Many Everbridge. It provides the solution with an unprecedented level of access and discoverability, which is unmatched on the open web. The goal is to include a reference or link to the application inside publicly broadcasted warning messages coming from One2Many's Public Warning Portal (PWP). These messages will be directly visible on smartphones in the affected area, and allows users to be forwarded to the chatbot platform.

### 4.4.1 Platform

To reach the largest audience possible, and to not exclude users, as described in section 3.4, the solution should run on a wide range of hardware. This includes low-end devices, and devices with a variety of input methods, such as touch or mouse and keyboard. This leads to the choice for a website as target platform. While not necessarily the only build platform, having web as a main launch target means very wide compatibility, and easy access without an installation process. The big disadvantage is the requirement for an active internet connection, as well as slightly worse performance compared to native platforms. Especially the internet connection can be a crucial flaw, as network coverage is all but guaranteed in disaster areas.

To combat this, alternative hosting methods can be considered, compared to traditional public webservers. One such method is using Multi-access Edge Computing (MEC), an initiative from ETSI[5]. A node close to the user in a MEC network, such as a local telecom station, can keep communicating with the user's device using eNodeB communication, even if the telecom station is no longer connected to the global internet. Because these nodes are still a part of the network, and internet connection can still be utilised, but with an offline version as backup. This added redundancy is extremely useful during active disasters.

[5] https://www.etsi.org/technologies/multi-access-edge-computing

### 4.4.2 Data

Somehow, the chatbot should then 'know' what the corresponding active emergency is (there can be multiple concurrent), so it can use it's metadata. This metadata is also transmitted by the PWP, and takes the form of a Common Alerting Protocol (CAP) message. This open-source protocol is developed and maintained by OASIS, and the specification can be found online [5]. It states a universal XML-based file format, containing standardised fields for use in PW, such as 'severity', 'instruction' and 'area'. Adapting the solution to the CAP standard ensures compatibility with most PW systems, in addition to removing ML training barriers, as implemented algorithms can use widely available historic CAP datasets.

### 4.4.3 Communication

To get these data files to the chatbot, and show them to the user, some rerouting has to be done. The first issue is sending the CAP files from the PWP to the proxy server. Luckily, the PWP already includes multiple output streams for these files, using technologies such as *HTTP Post* and *Kafka*. As there can be multiple concurrent emergencies, the transmitted CAP files should be indexed and stored for a short period of time.

To accomplish this, the Node-RED based proxy server has been set up to run on a MEC node, and listen for HTTP calls. The HTTP server stores the incoming CAP file with a custom file name, containing a 5-digit identifier. This identifier has been sent by the PWP and will also be used as a parameter in the URL, which is included in the warning message. The web-based chatbot can then use this ID parameter to send out an HTTP GET request to the proxy server, to receive the CAP for the current emergency. This means the URL can be launched from anywhere and will lead directly to the chatbot, pre-configured for the current emergency. An overview of this communication protocol, including the ML back-end, can be found in the sequence diagram, see figure 3.

## 4.5 Conversational Agent

As mentioned in section 3.1, the conversational agent in the proposed solution should produce answers which are as predictable as possible. One way of minimising the number of unpredictable results is by restricting the ML model to not use any Natural Language Generation (NLG) techniques, and pre-defining the output to human-written answers, which some degree of dynamic variables and entities. This method is called a retrieval-based model. In contrast to Generative models, they limit the Machine Learning implementation to input parsing only, instead of dynamically generating the Conversational Agent's output, as described by Ramesh et al[3]. This method works by defining a set of pre-configured responses, which are matched to the user's input using a process called 'Intent Matching'. In applying this to the Public Warning chatbot, a list

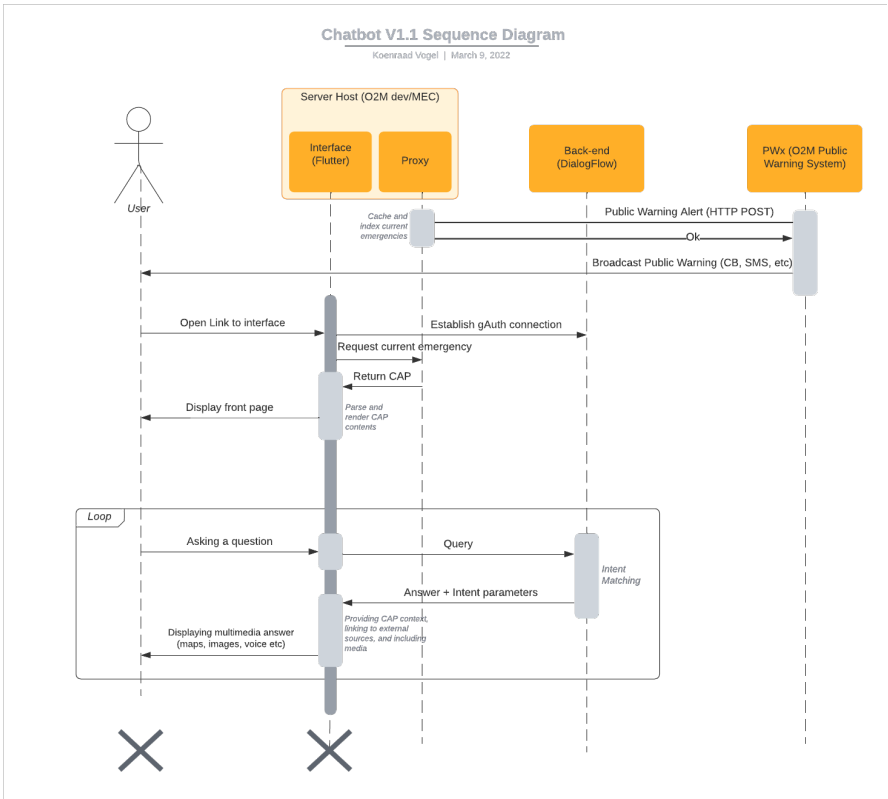


Figure 3: Sequence diagram for one user session

of concrete responses can be formulated, which will be matched to the user's input. The chatbot will therefore not be able to respond anything other than the preconfigured answers, including the typical "I'm sorry, I do not understand the question, can you rephrase?".

Another key consideration in typical conversational agent design is the breadth of the knowledge domain. A general distinction can be made between closed domain and open domain conversational agents. In short, a closed domain design is made only to understand specific conversation types, such as sales chat-bots; meanwhile, an open-domain conversational agent is broadly designed to understand conversations in any domain. In application to the proposed solution, the domain is rather narrow. Besides answering questions specifically on the ongoing emergency, the conversational agent does not need to anticipate an off-topic conversation.

### 4.5.1 Implementation

Within these specifications, the conversational agent can be implemented in two ways. It can be built out with Neural Networks and NLP techniques on a Python-powered software platform, or be offloaded to cloud chatbot technologies, such as google DialogFlow. The latter being a far simpler and less customisable implementation, but with the significant advantage that no large data-sets are needed for training, as their internal advanced NLP models are utilised. It will also be less time and resource-intensive to implement on a small scale, and run much faster. Even though outsourcing this functionality will cost money per user, for the early development and evaluation phases the trade-off makes sense, to speed up iterative prototyping. As described in section 7.3, the conversational agent can always be improved or substituted in future iterations. Modularity of the entire chatbot software platform should therefore be kept in mind.

Table 2: List of conversational intents, with examples

| Intent ID | Training Phrase Examples | Response Template |
|---|---|---|
| 0 | - (Fallback) | "Sorry, I didn't get that, can you rephrase?" |
| 1 | "Where is the flood?" | "The flood is located at the Bergpoortstraat in Deventer. Here it is on the map:" |
| 2 | "How close am I to the flood?" | "You are 2.3 km away from the flood. It is located at the Bergpoortstraat in Deventer." |
| 3 | "What should I do?" | "Find safe shelter right away, do not walk, swim or drive through flood waters. Remember, just 15cm of moving water can knock you down, and one foot of moving water can sweep your vehicle away" |
| 4 | "Can I speak with a human" | "In need of immediate help? Call 112. You can also contact the Dutch police at 0900-8844" |
| 5 | "I don't speak English"/"Kan je Nederlands praten?" | "Natuurlijk, laten we verder in het Nederlands praten" |
| 6 | "How dangerous is the flood?" | "The severity of the flood is *Extreme*" |
| 7 | "When is it over?" | "Unfortunately, it is not know when the event will end. Check the front page to stay up-to-date." |

| | | |
|---|---|---|
| 8 | "Where am I?" | "You are 2.3 km away from the flood. It is located at the Bergpoortstraat in Deventer. You can see an overview on the map:" |
| 9 | "Are you sure?"/"How trustworthy is this information?" | "The certainty of the flood is *Observed*." |
| 10 | "Can you give more information?" | "Thursday afternoon 12:07 a fire broke out at the bergpoortstraat in Deventer, several buildings are affected. Fire brigade is at location, the fire is expected to be put out in an hour. The origin of the fire is still unknown." |
| 11 | "What can you do?" | "I will help answer all your questions about the current flood. For an example try one of the suggestions below. However I cannot replace first responders, please call 112 if you require immediate aid." |
| 12 | "Is there an English translation?" | "Ofcourse, let's continue speaking in English." |

## 4.6 Brand Identity

As the target audience of the proposed solution is the general public, creating a strong brand identity is required. While there is no need for discoverability in an online environment, as the solution is integrated in the public warning system, a strong identity still helps the users recognise, trust and connect with the chat-bot.

### 4.6.1 Name

The name of the proposed solution is a crucial aspect of the brand identity, and should conform to a few requirements. First, it should reflect the core functionality and use-case, which is providing information and asking questions on active emergencies. Secondly, is has to be short and concise. Not only to be easier to remember, but also because it should be reflected in an accompanying top-level domain. This domain will be used to host the web-server running the proposed solution, and could be a way for users to reach the application. As the URL of the domain is also visible in public warning messages as part of the integration, it should contain as few characters as possible. After a brainstorm session considering many name combinations, see appendix B.1, *Crisis.Chat* was chosen, after which the accompanying domain has been purchased.

### 4.6.2 Colours

To convey the brand identity of Crisis.Chat, the design of the application should conform to a consistent colour scheme, as well as cohesive UI and UX guidelines. Colours are an essential aspect of brand identity, and can convey emotions in the user, as described by Singh and Srivastava[1]. For example, red is often associated with passion and danger, while green is perceived as more calm, and associated with nature.

In the context of this project, colours should not play a calming role in the design, as the user interactions are critical and time-sensitive. Therefore the primary colour in the colour scheme is based around red. Additional colours should be complementary, and are matched together using basic colour theory basics[6]. Another basic design guideline taken into account is the 60-30-10 rule[7], which dictates the use and coverage of the colours used, as seen in table 3.

| Colour | Coverage | Code | Preview |
|---|---|---|---|
| Accent | 10% | #FC534F | |
| Complementary | 30% | #2F4858 | |
| Neutral | 60% | #FFFFFF & #E0E0E0 | |

Table 3: Colour scheme of the proposed solution

### 4.6.3 Logo design

To represent the product online, a logo is essential. It can be seen as a favicon on browser tabs, as a launcher icon on android and iOS home-screens, and in marketing material. Logo's should be simple in design, with minimal amount of colours or complex shapes, to make it easily recognisable. To connect the logo to the product goals, different disaster/chat-bot related elements like outline, shapes and colours are layed out, see figure 4, after which they can be combined in a series of design iterations, see figure 5. The final logo can be seen in figure 6.

# 5 Evaluation

In iterative design processes like this, users, stakeholders and outside experts should be involved as early as possible in the development cycle. They provide not only useful feedback on a finished prototype [2], but also guide the development by making sure the prototype aligns with user goals. During development phases, stakeholders such as ENGAGE partners and members of the One2Many development team gave intermediate feedback during prototype &

[6] https://www.canva.com/colors/color-wheel/
[7] https://www.flowmapp.com/blog/glossary-term/60-30-10-rule

Figure 4: Different disaster/chat-bot related design elements, in order: shape outline, colour scheme, iconography

Figure 5: A variety of logo design iterations

Figure 6: Final iteration of the logo design

design presentations, which are continuously integrated into the prototype. In addition, the solution will be evaluated on a few small metrics, as described in this section.

## 5.1 UX

While the design should ideally be evaluated with usability testing methods, lack of time and more importantly, data-sets, prevent an extensive user testing study. Nevertheless, some specific recommendations and usability metrics will be described in section 7.2. Instead of involving end-users through usability testing, the solution was presented to online communities of practitioners, specifically on the field of UI design. This does not cover the full range of possible UX issues but should offer insights on some quick improvements to make. After gathering feedback[8], recommendations can be summed up as follows. Note that for some

[8] https://www.reddit.com/r/UI_Design/comments/u8ipi8/developing_a_chatbot_for_active_emergencies_im/

issues, solutions have already been found and implemented.

Table 4: UI design feedback

| Priority/ frequency | Issue | Implemented solution |
|---|---|---|
| 1 | Inconsistent spacing for software buttons and notched devices | Implemented safe area around widgets |
| 2 | Imperfect typography | Solution not (yet) implemented |
| 3 | Lack of prominence and prioritisation of updates | Solution not (yet) implemented |
| 4 | Lack of profile picture for chatbot messages | Solution not (yet) implemented |

### 5.2 Cost-effectiveness

The road from prototype to production will be long, with many steps until the product reaches end-users, such as localisation, deployment and possibly certification. While it is far from certain experiments like crisis.chat will be approved internally for productization, providing clear specifications and work to be done, will help it along. A more in-depth road-map and description of future work can be found in section 7.3, but a clear cost analysis should help the process. Using currently implemented technologies, a quick cost analysis can be made, seen in figure 5.

Table 5: Cost overview

| Service | Functionality | Cost |
|---|---|---|
| Google DialogFlow | Intent Matching | $0.002 per request |
| Google Maps API | Interactive Maps | $0.007 per map load |
| Cloud Firestore | Analytics | $0 for first 20.000 writes per day, then $0.0000018 per write |
| Amazon AWS VM | Hosting | - |

## 6 Conclusion

Within three months' time, the idea of a Public Warning chatbot is thought out and designed, building on the work of ENGAGE research. On top of this, the first iteration of a prototype implementation is built on the design, using iterative development methods. This implementation, build as a PWA in flutter and DialogFlow, includes a working conversational agent, and a significant subset of the proposed features. The prototype has basic integration with One2Many's Public Warning products, making end-to-end demonstration possible, and is

currently in a state ready for evaluation and further productisation. The final result can be seen on the included website[9], built alongside the prototype.

# 7 Discussion

While overall evaluation and reception has been positive, there are some downsides to the proposed method, and improvements which can be made. This section will describe the limitations of the prototype and method, and suggest improvements to be made in future work.

## 7.1 Limitations

One of the most flawed aspects in the implementation of the proposed design is the conversational agent. Not only is the amount of content (list of intents) limited, making the chatbot less functional than it could be, the response accuracy is not perfect, and query parsing is generalised. This means the intent matching algorithm, while trained on custom training phrases, is built to handle generic English language. This works in most cases, but in a crisis situation, user linguistics might differ slightly, prompting the need for a more tailored solution.

All these stated issues depend heavily on the back-end platform, currently based on DialogFlow. During prototype development, the back-end possibilities were heavily constrained due to a lack of data-sets, which are hard to come by. When quality data-sets are available, namely emergency call centre conversation transcripts, the prototype could presumably be improved significantly by building a second, custom iteration of the Back-end. The next section will elaborate by providing concrete recommendations for a framework.

## 7.2 Future Work – Evaluation

While a significant effort has been made in evaluating the proposed solution based on a variety of factors, the notably exception is end-user usability testing. Arguably the most important evaluation method, usability testing allows us to qualitatively and quantitatively validate the design and functionality of the prototype by exposing it to participants (the general public), preferably early in the design process. This could not be accomplished due to lack of time and data-sets, and was substituted by other evaluation methods, see section 5. In follow-up development phases, usability testing should be a priority, specifically along the following recommendations.

First, the feature list should be optimised, which can be effectively done using survey and card-sort techniques, as described by Baxter et al [2]. Features can be prioritised based on a variety of factors, such as perceived usefulness, relevance, and ease of use. While this can be empirically done, minority target users

[9] https://crisis.chat/

such as the visually and auditory impaired, should be kept in mind. Similar techniques can be used to tweak the User Interface design.

Then, observatory user research techniques, such diary studies, field experiments, or lab observations, should be employed to get qualitative data on general usability issues, thus exposing possibilities for improvement.

Finally, additional evaluation exercises are already set in motion, scheduled for after the termination of the internship project. These large-scale deployment exercises, integrated in disaster & public warning drills, both in Rome and Trondheim, have a couple of purposes. As direct observation of users is infeasible for the scope of up to 30.000 concurrent users, more indirect information-gathering techniques can be used, such as the unfinished analytics system. Data on usage patterns and intent matching accuracy can be gathered, all by collecting no Personally Identifiable Data. In addition, these exercises are great in testing wide-scale deployment and scaling of the platform implementation. How does the system hold up? How many users can it simultaneously handle? How much does it cost?

## 7.3 Future Work – Prototype

The first obvious improvement on the next version of the prototype should be implementing the full suggestion list, as described in section 4.2. This includes voice support, accessibility improvements, and finishing the work-in-progress analytics system.

On top of this, a different framework might be considered to run the prototype interface. More low-level languages and frameworks such as Angular, React or bare HTML, CSS and JavaScript, would theoretically be more efficient, faster, and integrate better with the back-end, in an effort to connect the two more tightly. Such an approach, however, would be far more time and resource-intensive, and would lose the prototype's ability to compile multiple platforms. This feature, while not necessary for deployment, provides flexibility in iterative prototyping.

The last, and most important improvement to be made for a second version of the prototype, is a custom back-end solution. Offloading Machine Learning processes from DialogFlow flow to a custom solution would provide more efficient communication to the front-end interface, as well as taking over much, if not all, functionality that now would be handled by the proxy server, thereby simplifying the whole system. In addition, the custom back-end should include a custom Intent-Marching algorithm, which would be fairly complex to create, but should provide better accuracy, lower costs, and increase customisation.

More concrete, a good setup could be built in Python, utilising frameworks such as NLTK, to create the conversational agent. First, incoming queries would be analysed using constituency parsing, stemming, dependency parsing, and other NLP methods to understand human language. Once a relatively solid understanding model has been created, conversation transcript data-sets should be matched (either by adapting existing data-sets, or create additional data-sets from model), and intent ID's should be assigned manually for items in the

database. Then, using packages like Keras and PyTorch and TensorFlow, to train for an extended period of time, and fine tune hyper parameters. Iterate by splitting data-sets into testing and training sets, and aim for increasing the accuracy.

In order to deliver a more concrete plan to the One2Many team for future development, a list of prioritised targets was made, see appendix section C. Above mentioned goals are included, as well as a short MoSCoW analysis, in order to assess prioritisation. The list of targets forms a road-map up to the next significant milestone for the project, the ENGAGE landslide exercise in 2023, as mentioned in section 7.3.

# References


[1] N. Singh and S. K. Srivastava, "Impact of colors on the psychology of marketing — a comprehensive over view," *Management and Labour Studies*, vol. 36, no. 2, pp. 199–209, May 2011. DOI: 10.1177/0258042X1103600206.

[2] K. Baxter, C. Courage, and K. Caine, *Understanding Your Users: A Practical Guide to User Research Methods*, 2, reprint. Morgan Kaufmann, 2015, ISBN: 9780128006092.

[3] K. Ramesh, S. Ravishankaran, A. Joshi, and K. Chandrasekaran, "A survey of design techniques for conversational agents," in *Information, communication and computing technology*, S. Kaushik, D. Gupta, L. Kharb, and D. Chahal, Eds., ser. Communications in computer and information science. Springer Singapore, 2017, vol. 750, pp. 336–350, ISBN: 978-981-10-6543-9. DOI: 10.1007/978-981-10-6544-6_31.

[4] N. Stolero, T. Simon, M. Bodas, K. Peleg, and B. Adini, *Deliverable 3 . 2 – Exploration of innovative use of communication and social media technologies*. Sep. 2021.

[5] [Online]. Available: http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2.html.

# A Detailed Log

| Week | Date | Tasks |
|---|---|---|
| Week 1 | 31-01-2022 | Getting to the office at 9am and getting settled. I'm getting my own desk :). Discussing with Menno to what extend I will be part of the scrum processes in the office |
| Week 1 | 01-02-2022 | The first daily stand-up meeting, where I introduced myself. Also an IT introductory meeting, where someone from IT remotely helps me set up the laptop. No admin rights though. |
| Week 1 | 03-02-2022 | First weekly update R&D meeting for One2Many, introducing myself to the larger One2Many team. |
| Week 2 | 08-02-2022 | Meeting with Rachelle Gianfranchi and a few other ENGAGE members, getting to know the larger European initiative. Also in the evening, Jeroen came by to retrieve the Everbridge monitors I had lying at home. |
| Week 2 | 10-02-2022 | Finished Figma designs of chatbot plans |
| Week 3 | | |
| Week 4 | 21-02-2022 | Presented the concrete plans and designs for the, still unnamed, chatbot to ENGAGE members (Rachelle, Bruria, Nathan). They are positive, and we agreed on Everbridge's participation within ENGAGE, with the chatbot project. |
| Week 4 | 25-02-2022 | Some disagreements through email concerning the Helsinki BUILDers conference on march 10th. The conclusion is that we are not presenting the chatbot prototype at the conference |
| Week 5 | 28-02-2022 | Launcher icons, and worked on slides for the CERIS conference |
| Week 5 | 01-03-2022 | Researched Kafka vs https protocol standards for receiving CAP alerts from the PWP. |
| Week 5 | 02-03-2022 | worked on adding a map URL launcher for location related responses. started looking at http server implementations for flutter. Confirmed with Rachele that I will be going to Brussels for the CERIS conference on 23 of march :) |

| | | |
|---|---|---|
| Week 5 | 03-03-2022 | Got access from Herwin to a vnc to the dev server, worked on desktop support and scaling. Meeting with Bruria and Nathan about CERIS presentation (Nathan will record prerecorded 2.5m presentation about D3.2) and about a meeting in San Sebastian (where I will presumably NOT be a part of) where the prototype can go through a quick evaluation session with a panel of end users. |
| Week 5 | 04-03-2022 | Working on google maps integration |
| Week 6 | 07-03-2022 | Working on google maps integration |
| Week 6 | 08-03-2022 | Working on google maps integration |
| Week 6 | 09-03-2022 | Documentation. Also brainstormed with Menno about hosting server options during the implementation into PW. Possible Patent options discussed. Also, I will no longer be going to Brussels, as Menno is needed :( |
| Week 6 | 10-03-2022 | Documentation. Also implemented full HTTP request system to retrieve current CAP from the hosting server using a 5 digit identifier placed in the chatbot URL as a parameter |
| Week 6 | 11-03-2022 | Day off |
| Week 7 | 14-03-2022 | I got diagnosed with Covid over the weekend, so working from home for the best part of this week. |
| Week 7 | 17-03-2022 | Got better just in time for the team-building drink at the local Deventer brewery Davos. Was fun to socialize more with the team. |
| Week 7 | 18-03-2022 | Documentation. Also implemented new colour scheme and logos across the stack. |
| Week 8 | 21-03-2022 | Menno tested positive for Covid, so last-minute Rachele asked me to fill in for him on a conference in Brussels, of course I said yes :) |
| Week 8 | 22-03-2022 | Travel to Brussels |
| Week 8 | 23-03-2022 | CERIS conference on disaster resilience by the European commission. Rachele gave the presentation, I was part of the expert panel and answered 1 question on the challenges of advanced communication technologies in emergency organisations. Afterwards seeing a bit of Brussels and going out for dinner with Rachele. She mentioned a possible job opportunity :). |

| | | |
|---|---|---|
| Week 8 | 24-03-2022 | Visited europarlementarium and travelled back home. |
| Week 8 | 24-03-2022 | Re-structuring code & preparing evaluation session in San Sebastian |
| Week 9 | 28-03-2022 | Preparing for the evaluation in San Sebastian. Also reworked intent matching to be identifier-based. Added two additional intents |
| Week 9 | 29-03-2022 | Travelling to San Sebastian and preparing for the meeting with Rachelle and Menno. Also meeting the whole SINTEF team. |
| Week 9 | 30-03-2022 | Whole day meeting with the ENGAGE consortium. Very interesting to finally meet a lot of people in real life, and get to know the project more. The goal is to get the chatbot into validation exercises and raise overall awareness |
| Week 9 | 31-03-2022 | Chatbot presentation and evaluation by KI-CoP with Menno & Nathan. Useful feedback gathered. |
| Week 9 | 01-04-2022 | Day off |
| Week 10 | 04-04-2022 | Day off |
| Week 10 | 05-04-2022 | Processing feedback from ENGAGE meetings, adding sender, website url and timestamp to homepage |
| Week 10 | 06-04-2022 | Starting with analytics system, brainstorming and deciding on firebase, created list of log properties |
| Week 10 | 07-04-2022 | Working on firebase analytics. At the afternoon drink, discussion possible part-time job opportunity with One2Many-Everbridge, Menno is in favour, but we need to discuss it with Morten Seliussen from the Norway department |
| Week 10 | 08-04-2022 | Working on firebase analytics, struggling with a bug caused by a conflict with multiple google service API's |
| Week 11 | 11-04-2022 | Improving CAP interface by adding all required and optional fields as indicated by the PWP CAP output profile |
| Week 11 | 12-04-2022 | Improving CAP interface by adding language support, communicating with PWP team about CAP questions |
| Week 11 | 13-04-2022 | Multi-language support |
| Week 11 | 14-04-2022 | Multi-language support; ARB language files |

| | | |
|---|---|---|
| Week 12 | 18-04-2022 | Multi-language support; Multimedia templates |
| Week 12 | 19-04-2022 | Multi-language support; State management |
| Week 12 | 20-04-2022 | Redrawing suggestion chips as to not include already-asked questions |
| Week 12 | 21-04-2022 | Start designing the end presentation. gathering UX and UI feedback online |
| Week 12 | 22-04-2022 | Bug-fixing, discussing collaboration with Italian health services for rome heatwave exercise |
| Week 13 | 25-04-2022 | Reporting and implementing UI and UX feedback in prototype |
| Week 13 | 27-04-2022 | Report and presentation |
| Week 13 | 29-04-2022 | Designing slides for end presentation |
| Week 14 | 02-05-2022 | Practising and finalising presentation |
| Week 14 | 03-05-2022 | Final presentation, for both my internal and external supervisors, as well as my colleagues from the One2Many team |
| Week 14 | 04-05-2022 | Discussed results of the presentation with team, and discussing possibilities for follow-up part time position |
| Week 14 | 05-05-2022 | Written the full experience report |
| Week 14 | 06-05-2022 | Documenting working on internship report |

# B Brand Identity

## B.1 Naming scheme brainstorm

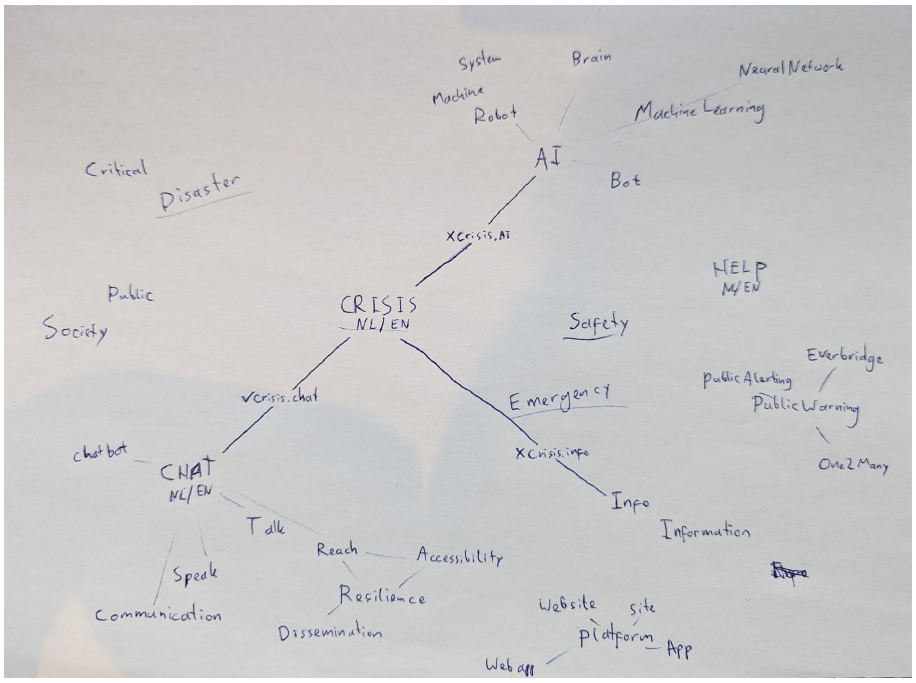


Figure 7: Brainstorm for short & disaster-related chatbot domain names

# C Roadmap

Koen Vogel
21-06-2022

ENGAGE Landslide Exercise

# **Chatbot** targets

## Prototype 2.0 requirements

| Must Have | Should Have | Could Have | Won't Have |
|---|---|---|---|
| • Deployment & Hosting environment<br>• Urgent bugfixes<br>• Prepare CAP and tailor intents | • Analytics system<br>• Norwegian language support | • Voice support (TTS & STT)<br>• Updated intents using call center datasets | • Custom Python back-end |

## Task priority

| | |
|---|---|
| **1. Urgent bugfixes** | • Language related crash in flutter 3.0.2<br>• Crash when returning to home page while loading response<br>• Instructions message crashing when no <instructions> tag is included in CAP file |
| **2. Deployment** | • Research hosting environment that fits our needs (CCU, maximum peak load)<br>• Deploy prototype to hosting instance, and make sure external connections work<br>• Integration into Norwegian PWC, including CAP profile compatibility<br>• Include remote config for controlling rollout and prototype behaviour on the fly |
| **3. Prepare CAP** | • Prepare custom CAP for exercise, with abundance of data<br>• Tailor conversation intents to work well with landslide events |
| **4. Analytics System** | • Building a logging system for gathering useful user data during the exercise<br>• Making the system accessible online during the exercise |
| **5. Norwegian language support** | • Adding support for Norwegian, by collaborating with UMS |
| **6. Updated intents** | • Getting access to call center conversation transcripts, preferably related to landslide emergencies<br>• Tailoring the conversational intents to actual user data and landslide events |
| **7. Voice support** | • Text-to-speech support<br>• Speech-to-text support |

Figure 8: Roadmap for requirements and future goals of the chatbot prototype, up to the ENGAGE landslide exercise in Trondheim in 2023