Using Google Glass in Crisis & Disaster Management

Richard Prinz (richard.prinz@frequentis.com), Corporate Research, Frequentis AG

2014-12-09

1 Introduction

In Q1-2014 Frequentis; Corporate Research was in the fortunate position to get hold on one of the first **Google Glass (Glass)**[1] Explorer Editions available in Austria. At the time of this writing (Q4-2014), Glass starts to become publicly available.

Looking for alternate user interfaces to make users life easier, products like **Microsoft Kinect**[2] where already evaluated in the past. With the availability of Glass we continued our exploration in order to find areas where the usage of Glass can provide benefits for users.

This document describes the efforts made by us to find use cases for Glass in some of Frequentis; Corporate Research projects in the Crisis & Disaster Management area.

[1] https://www.google.com/glass/start/

[2] http://www.microsoft.com/en-us/kinectforwindows/

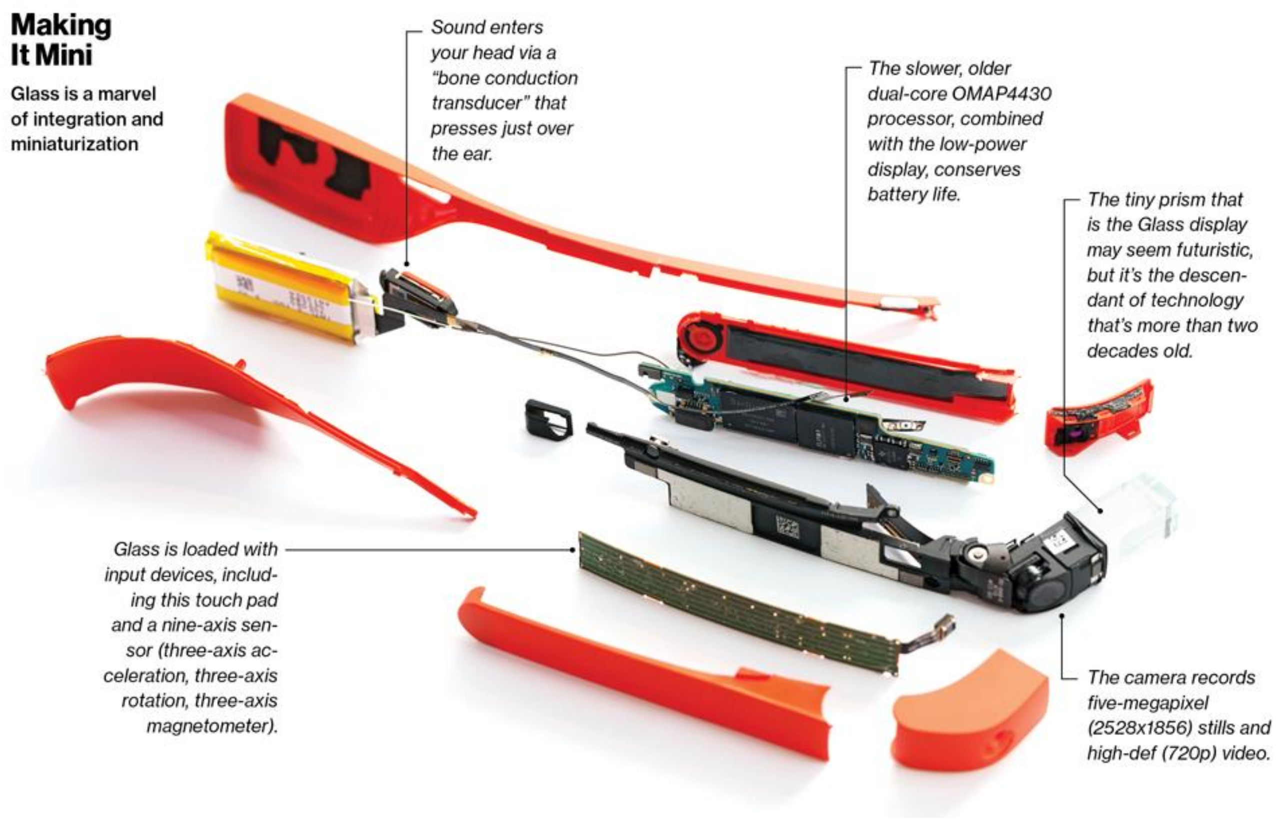
Contents

List of Figures

2 Google Glass

Introduced in mid-2012, Glass, belonging to the category of "wearables", provides a form of mobile **Head-up-Display (HUD)**[3] based on standard eyeglasses with a 640 x 360 pixel display at the top right corner of the right eye.

The Glass hardware matches that of a mid-range Android phone. A 5 Megapixel camera for taking pictures and videos, microphone, various sensors (compass, temperature, gyroscope, accelerometer etc.), 16GB memory, a dual-core OMAP 4430 SoC CPU and a Lithium Polymer battery (2.1 Wh) are tightly packed in the right temple.



Making It Mini

Glass is a marvel of integration and miniaturization

Sound enters your head via a "bone conduction transducer" that presses just over the ear.

The slower, older dual-core OMAP4430 processor, combined with the low-power display, conserves battery life.

The tiny prism that is the Glass display may seem futuristic, but it's the descendant of technology that's more than two decades old.

Glass is loaded with input devices, including this touch pad and a nine-axis sensor (three-axis acceleration, three-axis rotation, three-axis magnetometer).

The camera records five-megapixel (2528x1856) stills and high-def (720p) video.

Figure 1 Google Glass components[4]

To use all of Glass functions an Android, IOS or PC companion device (PC, Phone or Tablet) connected via Bluetooth to Glass running Google's **MyGlass App**[5] is necessary.

The MyGlass App provides setup (e.g. configuring WiFi networks), management (e.g. installing GlassWare), and feature functions (e.g. navigation route calculations, Glass screen and picture sharing).

[3] http://en.wikipedia.org/wiki/Head-up_display
[4] http://www.catwig.com/google-glass-teardown/ (Creative Commons BY NC SA 3.0)
[5] http://www.google.com/glass/help/myglass/

Glass is powered by standard Android 4.x operating system with a **special designed user interface**[6]. Based on a timeline with card decks the user can navigate with Glass's touchpad to past, present and future information. Card decks consist of one or more cards displaying static information (e.g. HTML content) or of native Android applications which provide dynamic content or functions.

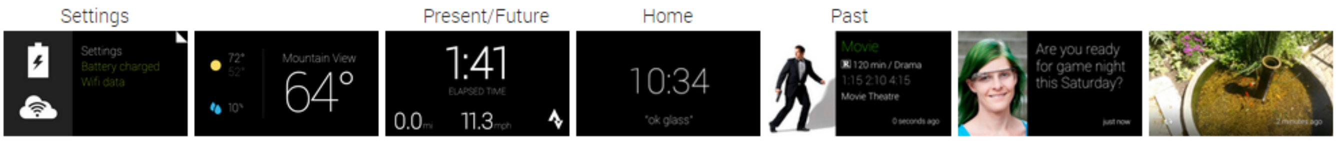


Figure 2 Google Glass user interface – Timeline

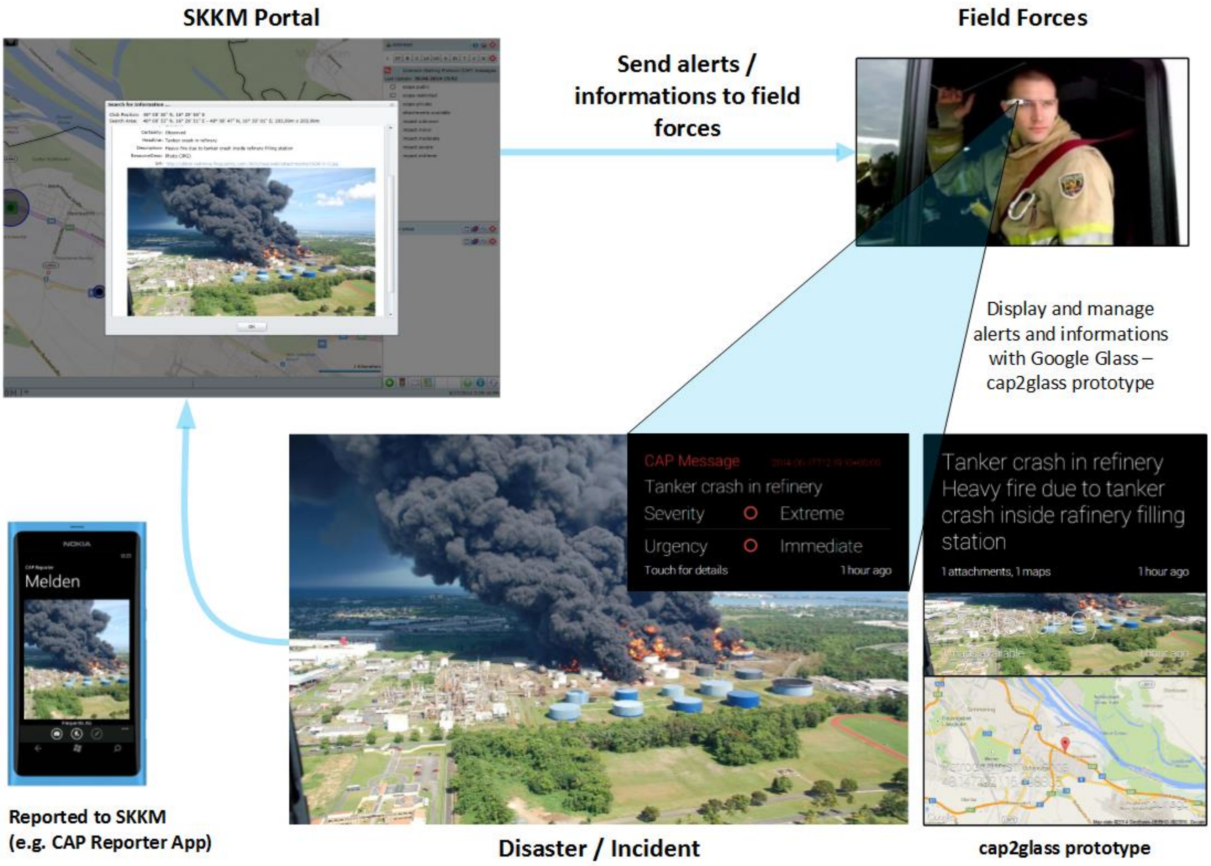
3 Projects

In the past there were already experiments performed with alternate user interface technologies like Microsoft Kinect for controlling application functions using body/hand gestures. The following sections describe two of our research projects where Glass was tested. Aim of these projects was to provide disaster field forces a fast and coherent view of accumulating information.

[6] https://developers.google.com/glass/design/ui

3.1 SKKM

One of these past experiments was done in the **SKKM** customer project (Austrian Ministry of Interior). SKKM - the Austrian national crisis and disaster management system - provides a **Common Operational Picture (COP)**[7] based on digital maps for visualizing, analysing and annotating geographical data (e.g. X-Ray / water level gauge measurements, temperature or rainfall). Also Glass was tested first with SKKM and an example prototype was developed to check out general feasibility. This first prototype was able to visualize incident details on Glass display *(cap2glass)*. Incidents were recorded using a companion Android or Windows Phone App *(CapReporter)* as illustrated in Figure 3.


SKKM Portal
Field Forces
Send alerts / informations to field forces
Display and manage alerts and informations with Google Glass – cap2glass prototype
Melden
Reported to SKKM (e.g. CAP Reporter App)
Disaster / Incident
cap2glass prototype

CAP Message
Tanker crash in refinery
Severity ○ Extreme
Urgency ○ Immediate
Touch for details 1 hour ago

Tanker crash in refinery
Heavy fire due to tanker crash inside rafinery filling station
1 attachments, 1 maps 1 hour ago

Figure 3 SKKM / Glass integration overview

[7] http://en.wikipedia.org/wiki/Common_operational_picture

3.2 IDIRA

Actual experiments are performed in the **IDIRA**[8] research project. IDIRA also provides a COP much like SKKM but expands its visualization capabilities with workflow functions like managing resources and needs or assigning tasks etc. A prototype was developed in cooperation with **Salzburg Research**[9] to use Glass as alternative user interface to IDIRA to further explore its capabilities and possible areas of application. Also Glass now not only visualizes incident information but also records incidents with a native GlassWare app *(CapCapture)*.

During the IDIRA large scale exercises in Wels (A), Dresden (D) and Athens (GR) we tested Glass's practical usability in real life scenarios.

Using Glass's **Text-to-Speech (TTS)**, voice recognition, head movement detection and eye tracking it is possible to handle and manage information without being distracted from the actual task which provides field forces hands free, up to date information about current crisis state.

Field forces can record incidents *(Figure 9)* with Glass including pictures of the scene *(Figure 7)*, geographic position, a first rating of the severity and additional textual notes in a very easy and fast way. Text is recorded by Glass **Speech-to-Text** functionalities which work quite well even in loud environments.

These incident recordings are then transmitted to the IDIRA **Mobile Information and Communication Structure (MICS)** *(Figure 5)* backend using the standardized **Common Alerting Protocol (CAP)**[10] and visualized on the IDIRA COP web *(Figure 11)* and native Android *(Figure 12)* applications.

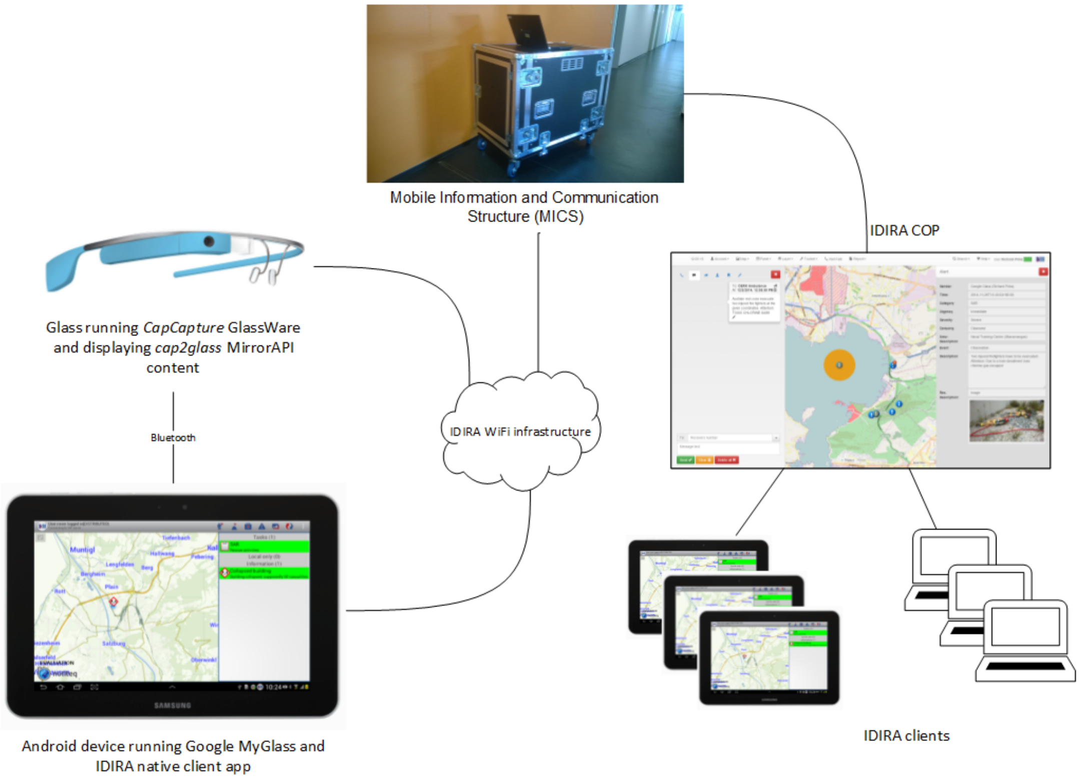

Mobile Information and Communication Structure (MICS)
IDIRA COP
Glass running *CapCapture* GlassWare and displaying *cap2glass* MirrorAPI content
Bluetooth
IDIRA WiFi infrastructure
Android device running Google MyGlass and IDIRA native client app
IDIRA clients

Figure 4 IDIRA / Glass integration overview

[8] http://www.idira.eu
[9] http://www.salzburgresearch.at/
[10] http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html

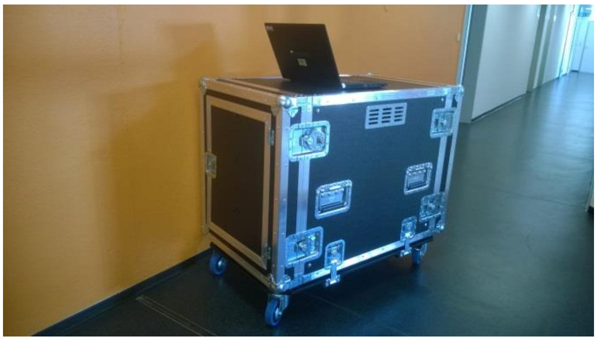


Figure 5 IDIRA MICS

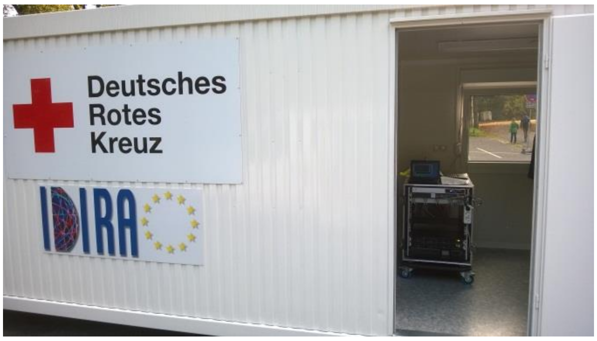


Figure 6 MICS deployed at disaster site

It’s not only possible to record incidents and share them but also to receive incidents from other sources. These incidents are visualized with all relevant details on Glass’s display *(Figure 13)*. Glass Text-to-Speech can read out incident details to the user. Additional incident information like pictures *(Figure 16)* or maps *(Figure 15)* are provided, real time navigation to (or away from) the incident is also possible.

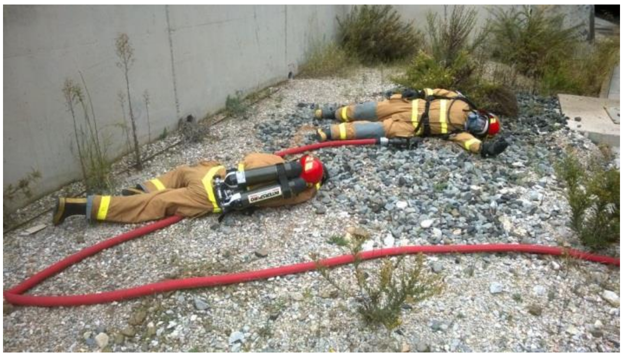


Figure 7 Incident scene

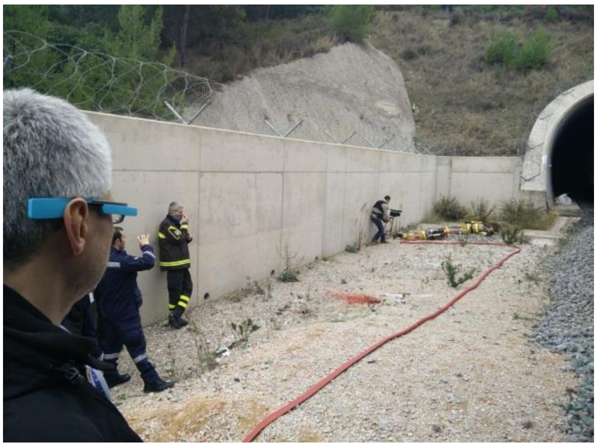


Figure 8 Incident recorded with Glass

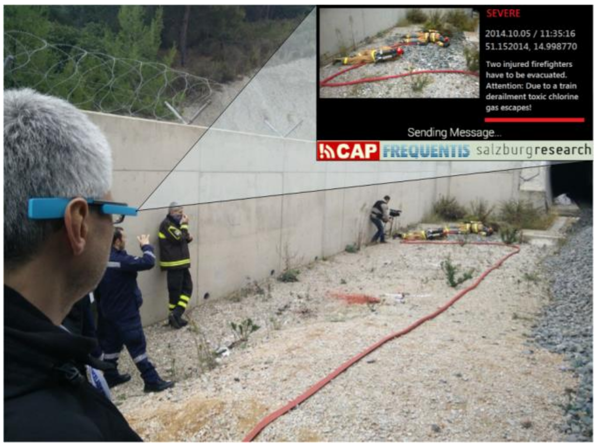
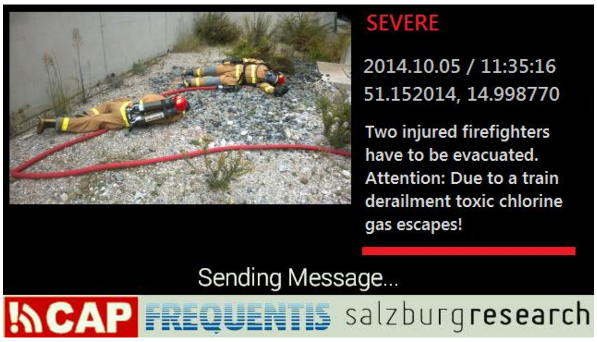

SEVERE
2014.10.05 / 11:35:16
51.152014, 14.998770
Two injured firefighters have to be evacuated. Attention: Due to a train derailment toxic chlorine gas escapes!
Sending Message...
CAP
FREQUENTIS
salzburgresearch

Figure 9 Incident seen with Glass



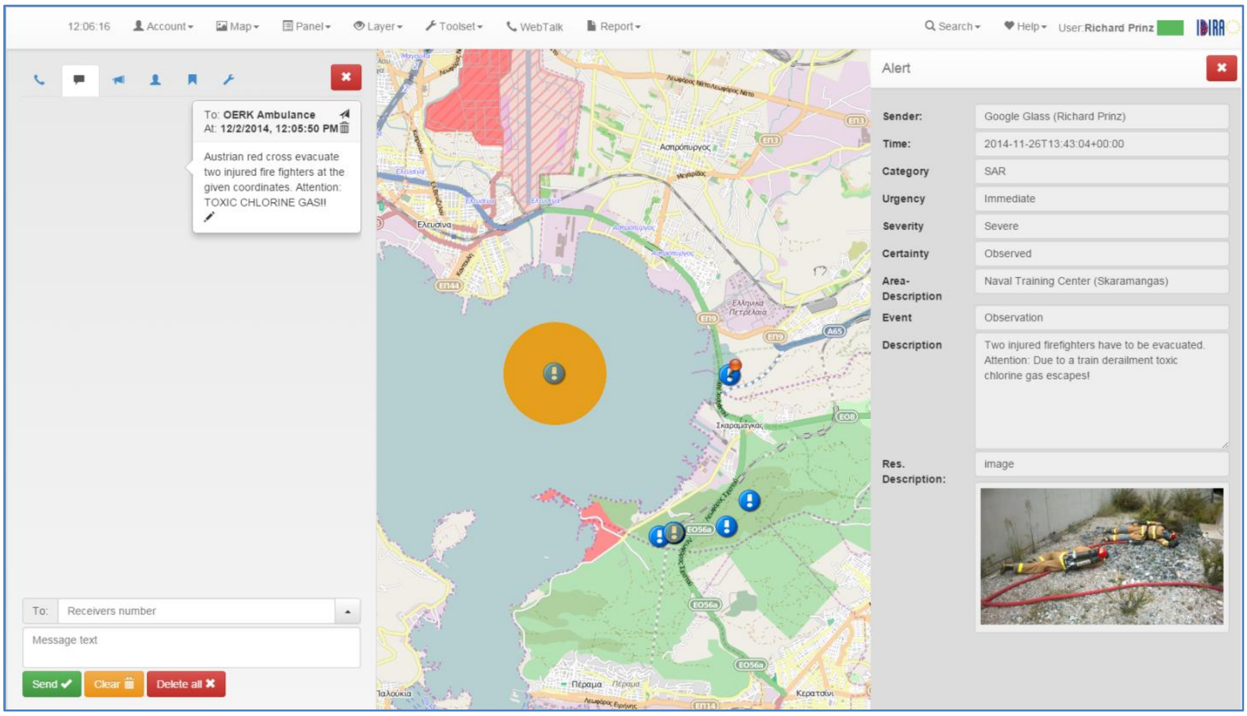
SEVERE

2014.10.05 / 11:35:16
51.152014, 14.998770

Two injured firefighters have to be evacuated. Attention: Due to a train derailment toxic chlorine gas escapes!

Sending Message...

Figure 10 CapCapture Glass screenshot



To: **OERK Ambulance**
At: **12/2/2014, 12:05:50 PM**

Austrian red cross evacuate two injured fire fighters at the given coordinates. Attention: TOXIC CHLORINE GAS!!

To: Receivers number

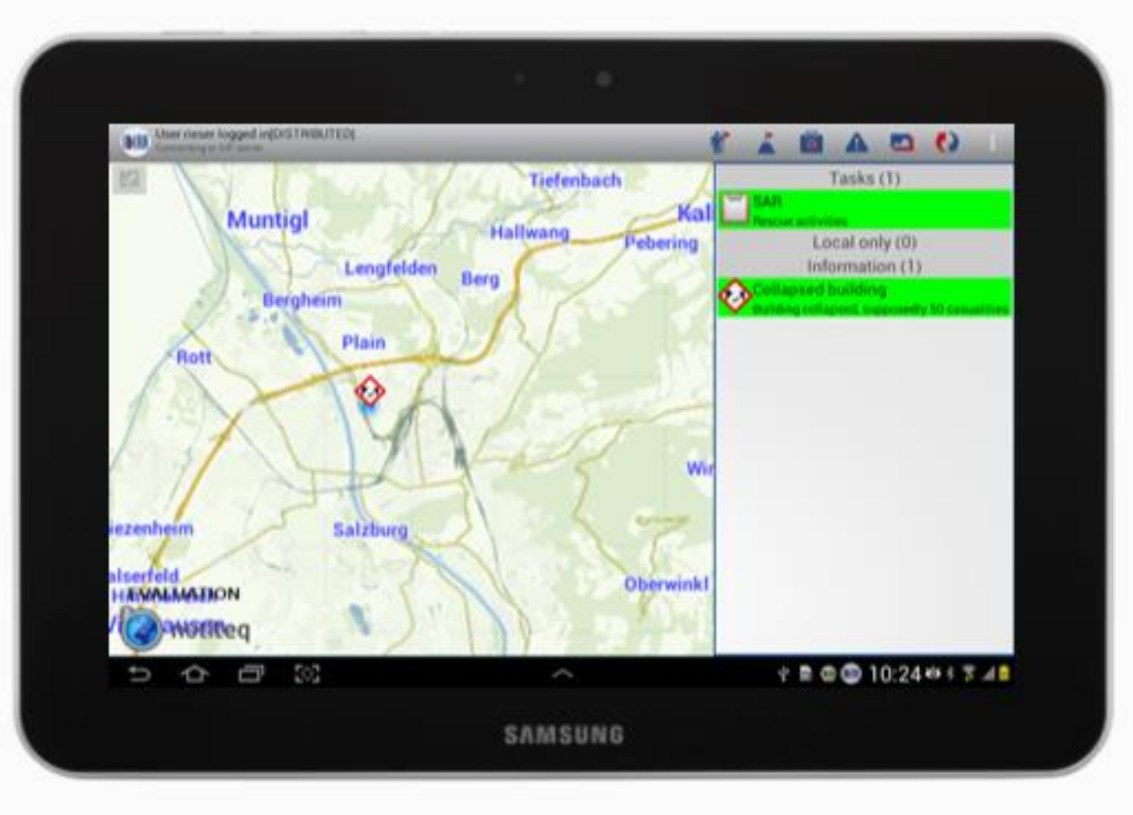
Message text

Send ✓ Clear Delete all ✖

Alert

Sender:	Google Glass (Richard Prinz)
Time:	2014-11-26T13:43:04+00:00
Category	SAR
Urgency	Immediate
Severity	Severe
Certainty	Observed
Area-Description	Naval Training Center (Skaramangas)
Event	Observation
Description	Two injured firefighters have to be evacuated. Attention: Due to a train derailment toxic chlorine gas escapes!
Res. Description:	image

Figure 11 Incident shown on IDIRA COP Web Application



User rieser logged in[DISTRIBUTED]

Tasks (1)

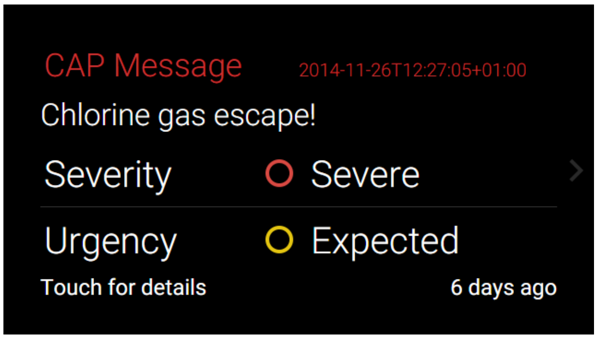
SAR
Rescue activities

Local only (0)

Information (1)

Collapsed building
Building collapsed, supposedly 50 casualties

Figure 12 Incident shown on IDIRA native Android App



CAP Message 2014-11-26T12:27:05+01:00

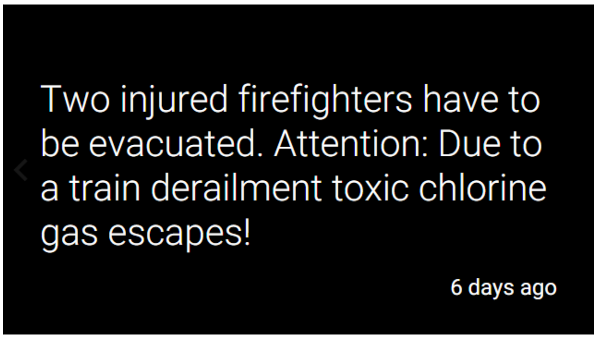
Chlorine gas escape!

Severity ○ Severe

Urgency ○ Expected

Touch for details 6 days ago

Figure 13 cap2glass received incident overview



Two injured firefighters have to be evacuated. Attention: Due to a train derailment toxic chlorine gas escapes!

6 days ago

Figure 14 Received incident details

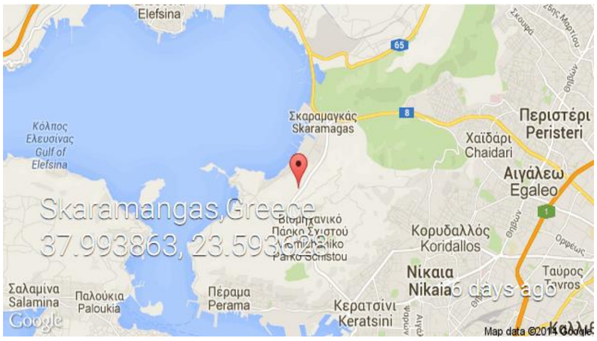


Figure 15 Received incident location

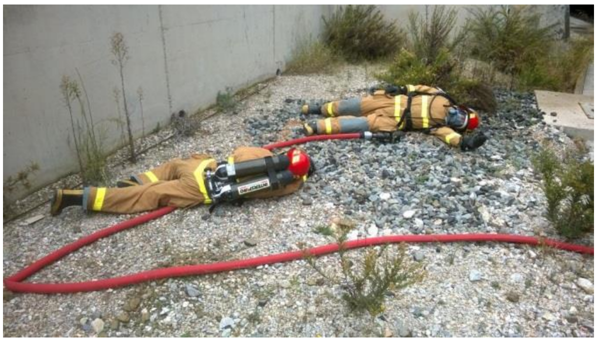


Figure 16 Received incident picture

Tests were performed by Frequentis but also by field professionals like the Austrian Red Cross or Greek Fire Fighters. From just getting used to Glass *(Figure 19, Figure 20)* over documenting incidents *(Figure 17)* to coordinating field forces *(Figure 18)* users were able to try out Glass during the tests.

Practice has shown that many users when first using Glass were somehow distracted and had to get used to Glass which required some time. After that almost all users were able to use Glass and our applications without problems.

Figure 17 Frequentis testing Glass

Figure 18 Users testing Glass I

Figure 19 Users testing Glass II

Figure 20 Users testing Glass III

4 Development

Technically there are three ways to develop solutions for Glass[11]:

1. **Native GlassWare**:
   This is more or less standard Android development using the **Glass Development Kit (GDK)**[12] resulting in an **Android Application Package (APK)**[13] file which must be installed on Glass either through Google Play Store or locally via USB.

2. **Mirror API[14]**:
   This form of development all takes place on a Server. Content (e.g. CAP incident information) are sent to Glass via Google's server infrastructure and received by Glass with built in functionalities. There is no need to install anything on Glass.

3. **Hybrid**:
   This combines Native GlassWare and Mirror API. In our IDIRA Glass integration prototype we used this approach. First to evaluate both types of development and second to split send and receiving functions. Receiving incident information might be provided to more Glass users without installation and by using Google services (much like news broadcast). Sending incident information is provided by fewer, trained and trusted Glass users and requires installation and receiving backend services.

   Incidents are recorded using a native GlassWare application *(CapCapture)* which acquires the location, controls the camera, records and converts voice to text and send all these information using CAP to the MICS backend.

   Receiving of incident information is done using a lightweight server component based on **node.js**[15] *(cap2glass)* which receives CAP messages from the MICS backend, converts it into JSON objects compatible with Google's Mirror API and sends these to Glass via Google services which presents the information to the user.

[11] https://developers.google.com/glass/develop/overview
[12] https://developers.google.com/glass/develop/gdk/index
[13] http://en.wikipedia.org/wiki/Android_application_package
[14] https://developers.google.com/glass/develop/mirror/index
[15] http://nodejs.org/

During out tests we came in contact with lots of professions working in various areas in Crisis & Disaster Management. Being it medical teams, fire fighters, rescue dog squads or telco emergency response units all of them developed some ideas how Glass could be used to support their work. Some of these ideas like using Glass for medical support might be further developed in other research projects.

5 Conclusion

Results for development and tests are so far:

1. Developing for Glass is rather easy and straight forward. Every Android developer should be able to write native GlassWare. Mirror API is a bit more involved due to the need of a server component talking to Googles servers which could be implemented in various ways and programming languages. Development time was about a month for *cap2glass* and *CapCapture*.

2. Finding use-cases with real improvements for users is not as easy as it sounds. Google Glass is still a cool Gadget but that alone is not enough. Real world use-cases like displaying a patients vital parameters during surgery[16,17] or displaying building information and floor plans for fire fighters[18] provide improvements and might be standard someday.

3. Glass's hardware was somehow weak in first versions (and still is) but has improved in late versions. Especially battery life and robustness are key factors for practical usage. Next generation Glass might have more memory, a faster a CPU[19] or other hardware changes which could make new applications possible.

4. Practical test with end users have shown that acceptance is very different. Some users accustomed very fast and easy to Glass whereas others had some difficulties and were distracted when wearing Glass and having information displayed in their field of view.

[16] http://www.kleinezeitung.at/s/lebensart/gesundheit/4136204/Forschung_Erste-OP-mit-GoogleBrille
[17] http://www.journal-surgery.net/article/S1743-9191%2814%2900914-5/pdf
[18] http://mashable.com/2014/01/21/google-glass-firefighter-app/
[19] http://www.zdnet.com/article/new-google-glass-to-run-on-intel-chips/