EmerGent

EMERGENCY MANAGEMENT IN SOCIAL MEDIA GENERATION

Deliverable 3.3 Interaction Design Patterns for Using EmerGent in Social Apps

Rajendra Akerkar[1], Christian Reuter[2], Massimo Cristaldi[4], Thomas Ludwig[2],
Andrew Muddiman[3], Federica Toscano[4], Federico Sangiorgio[4]

Western Norway Research Institute[1], University of Siegen[2],
Oxford Computer Consultants Limited[3], Intelligence for Environment & Security[4]

June 2014

Work Package 3

Project Coordinator

Prof. Dr.-Ing. Rainer Koch (University of Paderborn)

7th Framework Programme

for Research and Technological Development

COOPERATION

SEC-2013.6.1-1: The impact of social media in emergencies

Distribution level	Public (P)
Due date	30/06/2014 (M3)
Sent to coordinator	30/06/2014
No. of document	D3.3
Title	*Interaction Design Patterns for using EmerGent in Social Apps*
Status & Version	*Final*
Work Package	*3: Analysis of Social Media in Emergencies Today and Tomorrow*
Related Deliverables	*D3.2, D3.4, D7.5*
Leading Partner	*Western Norway Research Institute*
Leading Authors	*Rajendra Akerkar, Western Norway Research Institute (Sections 1– 4 – 6)* *Christian Reuter, University of Siegen (Section 1.1, 3, 6)* *Thomas Ludwig, University of Siegen (Sections 1.1, 3, 6)* *Andrew Muddiman, Oxford Computer Consultants Limited (Section 2)* *Federica Toscano, Intelligence for Environment & Security (Section 5)* *Federico Sangiorgio, Intelligence for Environment & Security (Section 5)* *Massimo Cristaldi, Intelligence for Environment & Security (Section 5)*
Contributors	*Reynold Greenlaw, Oxford Computer Consultants Limited (Section 2)* *Simon Scholl, University of Siegen (Section 1.1, 3, 6)*
Reviewers	*Tony O'Brien, EENA* *Alexis Gizikis, EENA*
Keywords	*Design patterns, technical analysis, mobile EmerGent app.*

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 608352.

Table of contents

List of Figures

Glossary

Abbreviation	Expression
EmerGent	Emergency Management in Social Media Generation
EMS	Emergency Management Services
HCI	Human Computer Interaction
OSN	Online Social Networks
DPA	Data Protection Act
ICO	Information Commissioner's Office
API	Application Programming Interface - Application Programming Interface - specifies how some software components should interact with each other
MAU	Monthly Active Users
SA	Situation Awareness
UI	User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
REST	Representational State Transfer
SDK	Software Development Kit - a set of software development tools that allows for the creation of applications for a certain software package
OAuth	An open protocol to allow secure authorization in a simple and standard method from web, mobile and desktop applications.

1. Introduction

1.1 Abstract

In this task a technical analysis of how EmerGent would interact with various Online Social Networks (OSN) will be performed. We focus on those with the highest active user numbers. The identification of the systems is based on the MAU-Value (Monthly Active Users), which counts unique users over 30 days. The MAU-Value is an approximate value to analyze the intensity of use. There is no clear definition of how an "active" user is characterized. Basically, a user is considered to be active if they are unique and have logged on to the system at least once in the month. [Ball13] shows the numbers of the MAU-Value in 2013 for the largest social networks, as well as the distance between the values of the individual systems.

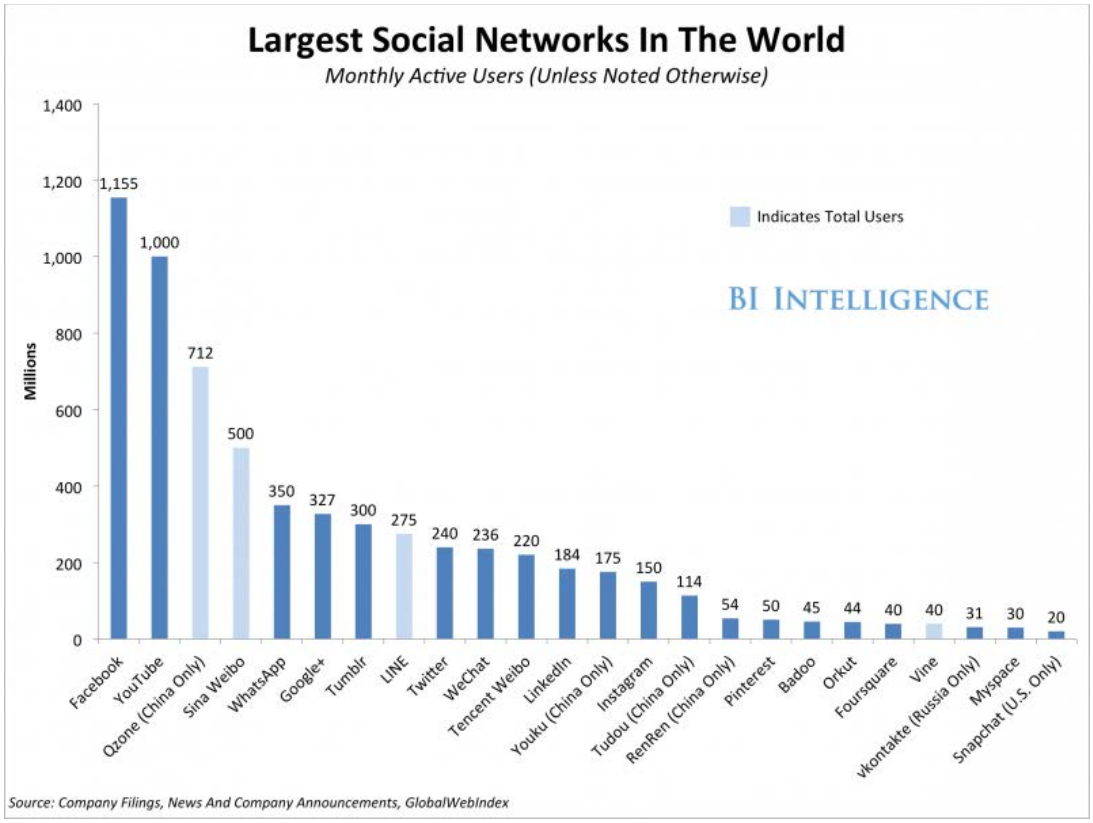

Figure 1: Largest Social Networks in the World1

The process of the increasing usage of social media will continue in 2014 Stat14]. In the context of EmerGent the focus of social media is only on systems which are well-known in Europe. An additional restriction is the minimum limit of the MAU of 100 million. This will provide a view on the most used media. This results in the following list:

- **Facebook** (1.15 billion)
- **YouTube** (1 billion)
- **WhatsApp** (350 million)
- **Google+** (327 million)

[1] http://www.businessinsider.com/the-worlds-largest-social-networks-2013-12

- **Tumblr** (300 million)
- **Twitter** (240 million)
- **LinkedIn** (184 million)
- **Instagram** (150 million)

1.2 Purpose of the Document

The focus of the Task 3.3 will be on the continuous analysis of social media in emergencies in order to identify the potential of new communication media at all phases in the EMS. This analysis is an important precondition for the definition of target social networks and for the definition of the needed API for accessing data streams.

In this regard, the main purpose of this document is to analyse:

- how to interact with users' profiles taking into account user privacy issues (e. g. access to private data streams) and technical limitations of each social network.
- how to provide the means for supplying to citizens targeted information both on social networks applications and on the mobile EmerGent app.
- how to interface the mobile EmerGent app to offer citizens novel methods for capturing and sharing relevant information for EMS, e.g. audio, photo or video.

Therefore we will first review interaction with user profiles to understand how selected Online Social Networks (OSNs) implement privacy and data protection (Section 2). Then we will review technical shortcomings of existing social networks (Section 3). Further we will review design patterns for citizens targeted information (Section 4). Finally we will analyse an interface for the mobile EmerGent apps (Section 5). Finally we will summarize our findings.

D3.3 provides insights into the technical analysis of social media in emergencies; this includes:

- a review of different OSN's policies for both end users and developers (Section 2)
- characterization of technical limitations and potentials of existing social networks (Section 3.2)
- design issues based on existing limitations (Section 3.3)
- a situation aware design in mobile apps (Section 4.1)
- a useful overview of strategies for design patterns and interaction design (Section 4.3)
- a constructive discussion on capturing and sharing relevant information for EMS using OSN (Section 5.2)
- the need to provide direct communication channels between the citizens and EMS (Section 5.3).

This analysis will, in some measure, lead to define the requirements for the project.

1.3 Target Audience

- Researchers in social media for emergencies
- Companies producing software tools/apps for emergencies

2 Interaction with User Profiles

This section deals with how selected Online Social Networks (OSNs) implement privacy and data protection, by informed consent on the part of the user and control through User Profile settings.

The policies and user profiles of Facebook, Twitter and Google (YouTube and Google+) were chosen for analysis as a representative sample of the OSN identified in section 1.

Common characteristics have been identified across the OSNs' policies for end users and developers and the result is categorised as requirements and assigned to a pertinent data protection principle where appropriate.

2.1 Privacy and Data Protection

Article 8 of the European Convention of Human Rights[2] provides the following right in respect of private and family life.

1. Everyone has the right to respect for his private and family life, his home and his correspondence.
2. There shall be no interference by a public authority with the exercise of this right except such as is in accordance with the law and is necessary in a democratic society in the interests of national security, public safety or the economic well-being of the country, for the prevention of disorder or crime, for the protection of health or morals, or for the protection of the rights and freedoms of others.

The EU Data Protection Directive[3] (1995) requires Members States to protect Article 8 rights with specific attention to the processing and handling of personal data.

In the United Kingdom this is enshrined in law by the Data Protection Act[4] 1998 (DPA) and we have taken the Act as a representative implementation to be used within the context of this analysis. Legislation from other member states could also be investigated to provide a wider picture. However, the UK's Information Commissioner's Office (ICO) principles could serve as providing generic criterion in which to see the OSNs' terms and settings. These principles[5] are listed as;

1. Personal data shall be processed fairly and lawfully and, must
   (a) have legitimate grounds for collecting and using the personal data;
   (b) not use the data in ways that have unjustified adverse effects on the individuals concerned;
   (c) be transparent about how you intend to use the data, and give individuals appropriate privacy notices when collecting their personal data;
   (d) handle people's personal data only in ways they would reasonably expect; and
   (e) make sure you do not do anything unlawful with the data.

[2] http://www.legislation.gov.uk/ukpga/1998/42/schedule/1
[3] http://eur-lex.europa.eu/legal-content/en/ALL/?uri=CELEX:31995L0046
[4] http://www.legislation.gov.uk/ukpga/1998/29/contents
[5] http://ico.org.uk/for_organisations/data_protection/the_guide/the_principles

2. Personal data shall be obtained only for one or more specified and lawful purposes, and shall not be further processed in any manner incompatible with that purpose or those purposes.
3. Personal data shall be adequate, relevant and not excessive in relation to the purpose or purposes for which they are processed.
4. Personal data shall be accurate and, where necessary, kept up to date.
5. Personal data processed for any purpose or purposes shall not be kept for longer than is necessary for that purpose or those purposes.
6. Personal data shall be processed in accordance with the rights of data subjects under this Act.
7. Appropriate technical and organisational measures shall be taken against unauthorised or unlawful processing of personal data and against accidental loss or destruction of, or damage to, personal data.
8. Personal data shall not be transferred to a country or territory outside the European Economic Area unless that country or territory ensures an adequate level of protection for the rights and freedoms of data subjects in relation to the processing of personal data

In Deliverable 7.5, we set out the structures that will contribute to good data governance through an Ethics Advisory Committee and a Privacy Impact Assessment. The extent to which each OSN's terms and associated software permissions control the effective data scope will need to be monitored to ensure that changes to either do not affect D7.5's structures.

2.2 Accessing Private Data Streams

2.2.1 Analysis of the OSN Policies

The review of each of OSN's policies showed that there were common elements running through each which we have categorised for both end users and developers. For example, each policy stated how the OSN and application developer should handle data acquired through normal usage, how it should be kept current and retained. We have assigned these to the requirements Data Currency and Data Retention Policy respectively and also assigned them to the data protection principle appropriately (4 and 5).

OSN End Users

Requirement	DPA Principles	Google	Twitter	Facebook
OSN seeks publishing consent from the End User	1	Yes	Yes	
OSN provides a defined Privacy Policy	1	Yes	Yes	Yes
OSN states a data retention policy and allows End Users to delete and download their personal data	5	Yes		Yes
OSN seeks consent to transfer and process data in USA	8		Yes	Yes
OSN commits to applicable Safe Harbors of the Digital Millennium Copyright Act	8		Yes	Yes
End User gives licence to use Intellectual Property of their content to OSN	1,2	Yes	Yes	Yes

Applicable laws and regulations	1,6	California	California	California (German)

OSN Developers

Requirement	DPA Principles	Google	Twitter	Facebook
The Developer is required to seek explicit publishing consent from the End User	1			Yes
The Developer shall supply an appropriate Privacy Policy	1		Yes	Yes
The Developer ensures that the End User can opt out of specific activity tracking	3			Yes
The Developer ensures that only data that is in relational to the stated aims of the app will be stored	3			Yes
The Developer ensures that the usernames and passwords are not collected, directly or via proxy	3	Yes	Yes	Yes
The Developer will provide the End User with the option of opt out to leave the app	3			Yes
Any cached data will be relevant and current.	4			Yes
The Developer will publish data retention policies.	5		Yes	Yes
Data security is ensured by protecting static or transferred information from unauthorised access or use	7	Yes		Yes
The Developer will ensure to protect third party interaction & confidentiality (especially tokens, ids, keys, etc.)	7	Yes		Yes
Developer commits to applicable Safe Harbors of the Digital Millennium Copyright Act	8	Yes		Yes
Developer to seek from End User gives licence to use Intellectual Property of their content	1,2	Yes	Yes	Yes
Applicable laws and regulations **Some specific clauses have been added to for the German market*	1,6	California	California	California (German*)

1.2.2 Applicable User Profile Perm*issions*

We reviewed User Profile settings and how it relates to what is available through the OSN's respective API. The table that follows lists the available information, in the light of the assignments that we have described in the previous section and is applicable to Emergent.

Facebook	
Public Profile[6] (Default) Information always available	This is information that is always publically available as well as information that the User has chosen to be publically available. Facebook describes this information as; • can be associated with you (i.e., your name, profile pictures, cover photos, timeline, User ID, username, etc.) even off Facebook; • can show up when someone does a search on Facebook or on a public search engine; • will be accessible to the Facebook-integrated games, applications, and websites you and your friends use; and • will be accessible to anyone who uses our APIs such as our Graph API The fields that will always Publically Available • Name • Profile pictures and Cover Photos • Networks • Gender • Username • User Id
Data shared with a Facebook app.[7]	In addition to the Public information, Facebook will also share the User's information; • the language • country • age range In addition, the list of the User's friends (and their Ids) who also use the app will be shared. Whilst installed, an app can access, store and update information that the User has shared. *Note*: «apps may still be able to access your information when the people you share with use them. And, if you've removed an application and want it to delete the information you've already shared with it, you should contact the application. If you want to completely block applications from getting your

[6] https://www.facebook.com/about/privacy/your-info
[7] https://www.facebook.com/about/privacy/your-info-on-other

	information when your friends and others use them, you will need to turn off all Platform applications »
Extended Profile Properties[8]	These properties are available upon permission of the User, and include; • Birthday • Profile About statement • Relationships • List of activities • Groups for which the user is a member • Likes • Home town • Location • Status • Photos • Videos • Tagged Places • Interests • Education History • Work history • List of events User is attending
Extended Permissions - Read[9]	These permission give read access to extended permissions. • Custom lists used to manage friends • Insights data for pages, applications and domains • The User's Facebook inbox • Provides access to all posts in a User's news feed.
Google+	
Privacy Profiles	Google+ privacy settings are categorized into various sections (Work, Education, Places, Contact Information, etc.) with fields defined for each one. For Contact Information, there are Home and Work settings for; • Phone • Email • Mobile • Fax • Pager • Chat

[8] https://developers.facebook.com/docs/facebook-login/permissions/v2.0#reference-extended-profile
[9] https://developers.facebook.com/docs/facebook-login/permissions/v2.0#reference-extended-read

	• Address Visibility is controlled by Permission settings that are grouped by Circles such as Only You, Your Circles (friendship groups), the Public (everyone), Extended Circles (friendship groups and their circles) and Custom settings. Depending upon how the Google+ API has been authenticated determines how the fields are returned. If an agent can access a resource (such as a file) but the Account's email address does not fall with the scope of permissions then the email field will not be returned with a blank or null value, but will be totally absent. A full list of the Google+ api can be found here [https://developers.google.com/+/api]
Twitter	
Twitter Interfaces	Twitter provides a number of programming interfaces: **Search API (v1.1)** The search is undertaken using OAuth authentication and polls data through Twitter's search facility by passing keywords (eg. usernames, places, activities, etc.) and returning json formatted data. Twitter describes[10] it as: "It allows queries against the indices of recent or popular Tweets and behaves similarly to, but not exactly like the Search feature available in Twitter mobile or web clients, such as Twitter.com search. Before getting involved, it's important to know that the Search API is focused on relevance and not completeness. This means that some Tweets and users may be missing from search results." **Streaming API** The Streaming API requires the developer to register a set of search criteria (as with the Search API) and accept data (in the form of tweets) as it is pushed from Twitter. The pushed data is only a subset of all the tweets matching the registered criteria. The returned data adheres to the Twitter rate limits. Twitter describes their streaming API[11] as:

[10] https://dev.twitter.com/docs/using-search
[11] https://dev.twitter.com/docs/api/streaming

	"The set of streaming APIs offered by Twitter give developers low latency access to Twitter's global stream of Tweet data. A proper implementation of a streaming client will be pushed messages indicating Tweets and other events have occurred, without any of the overhead associated with polling a REST endpoint." **Firehose** The Twitter Firehose is an API similar to the Streaming API, but guaranteeing that the pushed data will contain 100% of public tweets. Twitter provides its historical and real time firehose information through specialist partners who act as wholesale agents. Examples include: **Gnip**[12] "Gnip provides enterprise-grade access to the full Firehose of public Tweets so partners can build Twitter-enabled applications. Partners can access the complete historical archive of public Tweets as well as the full Firehose in real-time". **DataSift**[13] "DataSift provides both realtime and historical access to the full Twitter firehose, along with the unique capability to provide robust augmentations to the data that go beyond just keywords to include sentiment, influence, language detection, demographics". More can be found at the following url: http://business.twitter.com/partners/list/certified-products
Profile Settings	By default, the Twitter permissions are set to allow the public see the various profiles fields including (name, email address, etc). A full list of available User fields can be found here: https://dev.twitter.com/docs/api/1/get/users/search The list of objects returned through the APIs can be found here: https://dev.twitter.com/docs/platform-objects Twitter employs controls with a low level of granularity. Tweets can be Protected or Not Protected. Protected tweets will only be available to those who have been approved. However, even if accounts are protected, tweets gathered from those

[12] http://gnip.com/products/realtime/firehose
[13] https://business.twitter.com/partners/certified-products/datasift

	accounts are available and leave it up to the developer to not disclose data. Twitter states that Protected Tweets have the following restrictions[14]: • People will have to request to follow you; each follow request will need approval. • Your Tweets will only be visible to users you've approved. • Other users will not be able to re-tweet your Tweets. • Protected Tweets will not appear in Google search; protected Tweets will only be searchable on Twitter by the account holder and approved followers. • @Replies you send to people who aren't following you will not be seen by those users (because you have not given them permission to see your Tweets).

2.3 Currency of User Profiles

It is an important feature of the project as detailed in T7.5 that the consortium monitor the User Profile settings in respect of the current state of Data and Privacy Protection legislation. A secondary but significant task is to monitor changes to the terms of use and privacy requirements for each OSN.

Google end user terms maybe modified or terminated with reasonable notice and changes to their API terms are undertaken with 7 days' notice. Whereas Twitter makes no guarantee stating "the form and nature of the Services that Twitter provides may change from time to time without prior notice to you". Facebook also may change their API terms without notice but allow for 7 days' notice for users. The notice will appear on their site governance page they recommend that users employ the Facebook *like* mechanism to keep informed of updates.

There is not a standard to which each OSN adheres and so the consortium will need to carefully take note of individual OSN notification procedures and notice periods to ensure compliance with data protection and privacy laws.

[14] https://support.twitter.com/articles/14016-about-public-and-protected-tweets#]

3 Technical Review of Existing Social Networks

3.1 Introduction

Based on the list of the aforementioned different social media systems, the following section aims at the comparability and characterization of their technical limitations and potentials [ReSc14]. Almost all of the big players offer the possibility to access public data. Access to private user data can also be accomplished in most cases by asking for user authorization. In our consideration the only exception is WhatsApp as it does not support access via API or other development tools by default.

3.2 Technical and Business oriented Possibilities and Limitations

3.2.1 Facebook (Social Networking Service)

Graph API	
Supported systems	The API can be called by HTTP requests or by using the PHP, JavaScript, iOS or Android SDK which wraps the HTTP calls.
Functionality / accessible data	You can access all data via several root nodes in Facebook. The permissions to these are regulated by tokens. A lot of user content, such as messages, is privately shared and requires the use of a specific user token to access even those data. If you request a node not all available fields get returned automatically. It is necessary to define the parameters, which are relevant for your response, to filter the result set. E.g.: *"graph.facebook.com/bgolub?fields=id,name,picture"* returns just the id, the name and the picture of the requested profile. An overview on all possible API-calls and fields can be viewed on the Facebook developer page [WWW5].
License / security limitations	To use the Graph API you have to generate an access token which will temporarily provide secure access. This token is able to identify a user, app or page. If an app is using such a token, there could be several use cases with different token types: 1. User Access Token: If the app calls the API to read or modify personal data, the app needs a user access token. 2. App Access Token: This is used to request the Graph API for an application, not for a user. With this it is possible to change settings and parameters of an app, additionally the users can be managed, too. 3. Page Access Token: Based on the user access token, API calls to read and modify a page get accessed by the page access token. To generate such a token an admin has to grant extended permission for the page. 4. Client Access Token: This token is an identifier for mobile or desktop

	apps and gives access to a subset of API calls of the app access token. The token lifetime gets managed by the Facebook SDK. One limitation for using the Graph API is the rate of 600 queries in 600 seconds [WWW30]. A request has to pass several limitation layers: 1. User Level Rate Limiting: Each request gets assigned a score; once the entire score has reached the maximum limitation the call is blocked. 2. Specific API Rate Limiting: An allowed limit for API calls per second exists. If a user has reached the request limit for an app, the call fails. An allowed limit for API calls per second exists. If a user has reached the request limit for an app, the call fails.
FQL	
Supported systems	Facebook Query Language (FQL) enables to use a SQL-Style interface to query the data exposed by the Graph API.
Functionality / accessible data	The requested data are gained from the Graph API.
License / security limitations	“FQL will not be available after version 2.0. Please migrate your applications to use Graph API instead of FQL. ” [WWW21].
Keyword Insights API	
Supported systems	Data are queried from an FQL table and include aggregated, anonymous insights.
Functionality / accessible data	Analysis layer on top of all Facebook posts that enables you to query aggregated, anonymous insights about people mentioning a certain term.
License / security limitations	“Access to the Keyword Insights API is restricted to a limited set of media publishers and usage requires prior approval by Facebook. You cannot apply to use the API at this time.” [WWW22]
Public Feed API	
Supported systems	No HTTP API endpoint; updates are sent to a server through a dedicated HTTPS connection.
Functionality /	Provides a stream of posted user status updates and page status updates

accessible data	
License / security limitations	“Access to the Public Feed API is restricted to a limited set of media publishers and usage requires prior approval by Facebook. You cannot apply to use the API at this time.” [WWW23]
Chat API	
No support anymore: “The service and API this document covers has been deprecated with the release of Platform API v2.0. Once version 1.0 is deprecated, chat.facebook.com will no longer be available.” [WWW20]	

3.2.2 YouTube (Video-Sharing Website)

YouTube Data API (v3)	
Supported systems	It is possible to access several resource types, such as channel, video, playlist and so on. In connection with these, the API offers standard operations on the resources like list, insert, update, delete. It is possible to request the API via HTTPS.
Functionality / accessible data	It offers the possibility to integrate YouTube functionality into applications: Fetch search results and insert, update and delete videos or playlists.
License / security limitations	A Google account is required to use the API. Beside this, you need an API-Key and you have to register your application on Google. After that the “YouTube Data API” must be activated as a service for the project. Requests to the API are limited by a quota. Every request gets a quota assigned. The value depends on the requested resource type and the selected operation. Different operation types have varying quotas. E.g. a simple read operation has a cost of 1 unit, a write operation has a cost of 50 units and a video upload costs 1600 units. The entire quota per day is at 50.000.000 units[WWW11].
YouTube Analytics API	
Supported systems	Included in the Google APIs Client Library for .NET, Java, JavaScript, C, PHP, Python, Go, Google Web Toolkit, Node.js, Ruby
Functionality / accessible data	YouTube Analytics API offers the possibility to retrieve viewing statistics, popularity metrics, and demographic information for YouTube videos and channels.
License /	Each API request retrieves data for a particular YouTube channel or content

security limitations	owner. The quota is at 50,000 requests per day[WWW17].
YouTube Live Streaming API	
Supported systems	Part of the YouTube Data API
Functionality / accessible data	YouTube Live Streaming API aims at creating, updating, and managing live events on YouTube. The API connects events (broadcasts) and video streams, which represent the actual broadcast content.
License / security limitations	Some API requests need user authorization via OAuth 2.0. Since the API is available in the Google-Developer area, the 1000 subscriber limitation for channels to use the API seems to have been cancelled [WWW12].

3.2.3 WhatsApp (Mobile Instant Messenger)

It is not allowed to access data from WhatsApp. There have been some Open-Source-Projects like DISA or WhatsAPI, but they were removed from GitHub because they might violate the copyrights of WhatsApp [Heis14].

3.2.4 Google+ (Social Networking Service)

Google+ API	
Supported systems	Supports: Android, iOS, C#/.Net, Go, Java, JavaScript, PHP, Python, Ruby. The core is formed by a REST-API, which can be queried by HTTPS.
Functionality / accessible data	The API provides access to public data. Therefore all calls require a token based on the OAuth 2.0 Protocol or an API key. Based on a REST-Architecture the API offers the retrieving and manipulating of resources via HTTP-Requests. Resources which can be accessed are data about people, activities, comments and moments. If developers are using the API, they can define app activities as moment resources to reflect user actions.
License / security limitations	Users of an application have to allow the access to their user data via OAuth 2.0. The request rate is limited to 10.000 calls per day and 5 calls per second. Calls which belong to Google+ API (Sign-in) have a limit of 20,000,000 requests per day. Google+ API (Sign-in) covers calls out of the scope "*https://www.googleapis.com/auth/plus.login*". Via the API-console developers can request a higher limit of API calls. Additionally developers can define a limit for user requests per second to their own application [WWW08].

Hangouts-API	
Supported systems	Can be requested with a JavaScript interface.
Functionality / accessible data	Special API for Google+ Hangouts Video calls. Accesses data about hangout participants, shares data between app instances during a hangout session and controls the required hardware components.
License / security limitations	Because this API offers functionality around the Hangout-Services, there are no public limitations for using it. It has no request limitations in contrast to the Google+ API[WWW03].
Google+ JavaScript API	
Supported systems	The API is easy to integrate into webpages, via script loading. The API also supports multilingualism and asynchronous API loading.
Functionality / accessible data	Google offers Plugins for a bundle of functionalities. It is possible to integrate the +1-Button similar to Facebook's "Like"-Button. For the use of such plugins it is necessary to comply with the google guidelines, which regulate Privacy and Design aspects of the elements. The box is a further plugin resource, which provides the displaying of google+ profiles. Additionally, Google+ - login can be integrated for the authentication and authorization of users, Google+ - sharing provides user interaction with Google+ on custom websites and snippets can display a rendered link preview of an URL.
License / security limitations	No direct limitations; the requested data come from Google+ API. So the default rate limits also apply for this API [WWW07].
Google+ Play SDK/ Google+ iOS SDK	
Supported systems	Supports iOS and Android application development.
Functionality / accessible data	The SDK provides integration of UI-elements, API calls for the JavaScript API. Additionally, it is possible to request other Google services such as location and activity services.
License / security limitations	No direct limitations, the requested data come from Google+ API. So the default rate limits also apply for this API [WWW09, WWW10].

3.2.5 Tumblr (Microblogging and Social Networking Service)

Tumblr-API	
Supported systems	Supports: JavaScript, Ruby, PHP, Java, Python, C
Functionality / accessible data	Tumblr offers a JSON-based API, which provides reading and writing permissions. The API makes distinctions between blog-data and user-data. Based on the authentication-level the access is regulated.
License / security limitations	The API uses three different levels of authentication. Public methods do not need any authentication, some methods need a generated API key and other methods need a signed request based on the OAuth 1.0a Protocol. In addition to that Tumblr limits each account to 250 posts per day, across all of the owned blogs. Within those 250 posts and re-blogs, it is possible to create 75 150 original photo posts per day. The Queue staggers publishing of your posts — with a queue limit of up to 50 posts per day [WWW32]. Tumblr allows you to like up to 1,000 posts per day from your account. [Nova14]

3.2.6 Twitter (Microblogging Service)

REST-API	
Supported systems	The API is based on the REST-Design, which allows requests based on HTTP/HTTPS.
Functionality / accessible data	The REST-API offers the possibility to query and modify a user account. Modifying an account requires the user permission via OAuth 2.0
License / security limitations	A rate limit of 15 minutes' duration defines how many requests for an API-Call are allowed per user and per application. For instance, requesting the friends list of a given user can be called 15 times by a single user and 30 times by an application within 15 minutes [WWW03]. An overview on all the different calls can be found at the Twitter-developer page [WWW01].
Streaming-API	
Supported systems	Similar to a push service, the streaming-API requires a persistent HTTP connection.
Functionality /	The streaming-API delivers live tweets based on a search term or specific

accessible data	users which get requested.
License / security limitations	The streamed tweets are limited by streaming cap. This cap is calculated based on the total amount of tweets at a certain time. If there are more tweets, that match your filter criteria, then you receive, you will get the number of the not streamed tweets. E.g. you are able to get 1% from 30,000 tweets for a given set of hashtags. [WWW02]

3.2.7 LinkedIn (Business Oriented Social Networking Service)

LinkedIn-Rest-API	
Supported systems	Supports HTTP requests over REST; for development purposes, LinkedIn offers a JavaScript wrapper via the default REST-API. The API supports XML and JSON response format.
Functionality / accessible data	The available information, which is provided by the LinkedIn API, is divided into several resources: people, companies, groups, connections, jobs and separate resources for searching people, companies, and jobs. This results in Sub-APIs which describe the access to the resources individually. Based on a large set of different parameters, it is possible to request explicit data and metadata about information resources. For example, the request "http://api.linkedin.com/v1/people-search?keywords=hoffman&company-name=linkedIn" searches for members with the keyword "hoffman", who have worked in a company named "linkedIn". This allows making complex requests to the API in an easy way. Besides the read access to LinkedIn data, some Sub-APIs offer special functionalities. The connections-API gives the possibility to exchange messages between several members, the invitation-API can be integrated which allows users to invite other people to LinkedIn out of custom applications and the share-API lets users distribute shares on LinkedIn while the custom app defines the appearance of the content.
License / security limitations	LinkedIn divides data into public and private. For example, a requested profile can be called as standard or public. The public request returns all public user data, based on the privacy settings of the profile owner. The standard requests filter the response even on the privacy settings of the owner, but also on the privacy setting of the requestor and the relationship between requestor and owner. In addition to using OAuth for generating access tokens for API calls, LinkedIn has a defined throttle for applications, users and developers.

	The rate limits of API-calls are available under "https://developer.linkedin.com/documents/throttle-limits". The usage of the capacities gets cleared daily. For instance, posting shares has a daily limit per application of 125.000 calls, per user 25 and per developer 100 calls [WWW04].

3.2.8 Instagram (Mobile Photo and Video Sharing Service)

Instagram-API	
Supported systems	Based on a REST-architecture the API-endpoints can be requested via HTTPS. For special support they offer a python and a ruby library.
Functionality / accessible data	The API supports access on data about users, relationships, media, comments, likes, locations.
License / security limitations	Especially Instagram refers to core terms for the use of the API. Defined API-terms are [WWW25]: 1. Instagram users own their media. It's your responsibility to make sure that you respect that right. 2. You cannot use the Instagram name in your application. 3. You cannot use the Instagram API to crawl or store users' media without their express consent. 4. You cannot replicate the core user experience of Instagram.com 5. Do not abuse the API. Too many requests too quickly will get your access turned off. Other limitations are 5000 calls per authenticated user (access token or client) per hour. API actions such as commenting, following/unfollowing and liking have its own limit of 350 calls per hour [WWW27]. The standard calls are signed with a default scope. Scopes provide permissions to advanced API calls. It is possible to extend permission with scopes for creating and deleting comments, scopes for following and unfollowing a user and to like and unlike items [WWW26].
Instagram-RealTime-API	
Supported systems	Supports the subscription via server.
Functionality / accessible data	Works with an integrated PUSH-Service to notify applications about new content. For that, developers can subscribe Instagram-objects, like users, tags, and locations.

License / security limitations	Limited to 30 real-time-subscriptions.

3.3 Design Considerations Based on Current Limitations

The most common architecture to request data from social media is the use of REST-APIs. These APIs provide an easy way to extract heterogeneous datasets. To provide a secure communication between the requester and data source, a large number of requests are based on HTTPS. In addition to the safety of use, the user privacy is one significant aspect for the data access.

A standard way for APIs is distinguishing between public and private data. A public user profile contains basic information and such information, which is filtered by the privacy settings of the user. An application which wants to use this information does not need any special authorization for making the requests. But if more information is needed the requestor has to ask for user's authorization. A common way for implementation is the OAuth 2.0 protocol that allows applications to access each other's data. The following figure shows how a custom application requests data from Facebook for authentication.

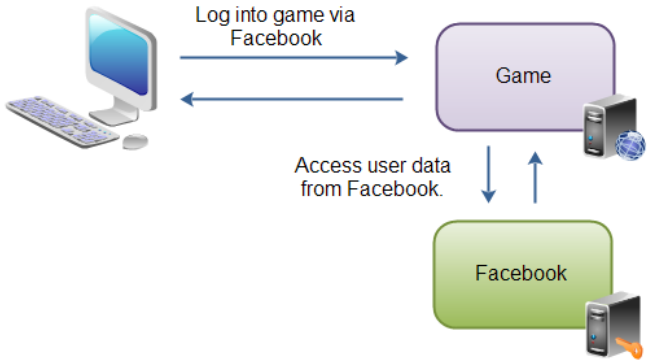

Figure 2 Example of how OAuth 2.0 is used to share data via applications. (http://tutorials.jenkov.com/images/oauth2/introduction.png)

Much of this functionality is wrapped in the provided SDKs and APIs for different programming languages to offer a simple way for integrating social media into new applications or technologies. Besides the default functionality for requesting and posting data, the social media provide many standard elements out of their systems for integration. For example, the Facebook "Like" button can be integrated in every website or application. This aims to strengthen the distribution of the underlying social media. LinkedIn offers the opportunity for inviting new members to the network out of any application. An application which integrates these features from different media types can build crossover the existing systems. For instance, a Facebook-App can invite people to LinkedIn. The limitations for using APIs and SDKs are nearly equal, because all functionality which is provided has an underlying layer to query the systems' APIs.

Requests are regulated by rate limits, which define a specific amount of calls for a user, an application or another requestor in a given time interval.

1. In Facebook the rate limitation for using the Graph API is set to 5M MAU's and 100M API calls per day or 50M impressions per day. [WWW06] (see section 3.2.1).
2. YouTube's Data API set a limitation of 30,000 units/second/user and 50,000,000 per day. Google defines the unit and gives an explanation on the developer website [WWW16] (see section 3.2.2).
3. WhatsApp does not allow accessing data – however it is possible (see section 3.2.3).
4. Google+ API sets a limitation of 10,000 requests per day and 5 requests per second. To view the rate limitation it is necessary to log into the developer console [WWW14] (see section 3.2.4).
5. Tumblr limits the amount of posts per day for all owned blogs for each account to 250 posts per day (see section 3.2.5).
6. Twitter makes distinctions between several requests in a 15-minute interval. E.g.: it is possible to query 180 tweet-searches in 15 minutes per user and 450 tweet-searches in 15 minutes per app (see section 3.2.6).
7. LinkedIn sets daily limits with the same distinction as Twitter. So you can request in the role as application, developer or user. For example, an application can make 100,000 status updates per day, a developer and a user only can make 250 updates per day (see section 3.2.7).
8. Instagram sets a defined limit of 5,000 requests per hour per access-token (see section 3.2.8).

An application which reaches this limit has the option to request a higher rating. A defined pricing table is not available, even Google just offers pricing tables for services like "App Engine" [WWW13]. Using other elements like login buttons has to deal with the specific guidelines for integration, which are dictated from social media owners. Those guidelines define the look and functionality of given resources.

A new trend seems to be streaming- and live-APIs that allow developers to integrate live information out of social media. Requestors subscribe specific content like YouTube channels or tweets and get notified about new content. This is "the next step" to build rich social media apps, which not only are able to request and interact with other social media services, but also may observe content and react on specific events. Besides this, another trend is based on the increasing usage of mobile devices. The amount of users which access the internet via mobile devices will further raise in 2015 [WWW18]. Social media are aware of the mobile context and offer functionality supporting the integration of their platform into third-party mobile apps.

This support is not limited to application development: to assess user mobile context it is possible to request metadata beside the default data and display contextual dependencies between user behavior and external events, e.g. location-based services. A use case could be the mobile use of Facebook. The user moves through its environment, which is subject to the influence of different factors. These may be characterized by the current location of the user, the social interaction or the time of use. This has an impact on the use of Facebook and similar networks, since these factors form the mobile use situation of the user and so are critical to the sharing of information in such a system [ApHJ13]. In a crisis, the proximity to the event could affect the use so that the user is the information agent itself or to retrieve certain information. The reaction to such behavior offers the possibility of selectively disseminating or extracting information.

4 Design Patterns for Citizens Targeted Information

In the field of emergency management, application design knowledge is dispersed and available in discrete ways. Thus, the purpose of this chapter is to discover and collect the available and rather hidden design knowledge.

Emergency response demands fast and effective action, often in life-threatening situations. It requires collaboration between citizens and authorities: the personnel at an incident site, in the emergency vehicles, at the command and dispatch centers, at hospitals, etc.

Interaction design plays a key role in the design of application user experiences. Good interaction design:

- effectively communicates the scope and type of an application's functionality through its primary contexts for interaction, as well as its menus or navigation system
- defines and evidently presents both simple and complex workflows and, consequently, eases users' tasks
- provides easily apparent opportunities for interactivity through interactive elements
- defines user interactions that are consistent with best practices and easy to use
- specifies behaviours that clearly communicate an application's responses on targeted information
- dynamically prevents user error

This chapter provides a general introduction to design patterns and the types of patterns in the area of Human Computer Interaction (HCI). Design patterns for citizen targeted information in the perspective of emergency management are also discussed.

4.1 Citizens Targeted Information & Situated Awareness

4.1.1 Design for citizen targeted emergency response

Designing interactive systems within the area of emergency management requires special care, since the end users might be in a stressful situation where lives and property are at stake. General design patterns from available collections are not necessarily applicable for safety critical environments and emergency response.

Design patterns are used to describe best practices and effective design solutions, and for capturing and sharing design knowledge with other people faced with the same problem and context. The solution proposed by a design pattern should be generic rather than specific, so that it can be implemented in numerous different ways.

In this domain, usual HCI design solutions run the risk of resulting in hard to use products although their foundation represents building blocks for system design in emergency response systems. The transfer to this domain has to be performed carefully. Specifically in the field while on duty, new approaches must be created that simplify the interaction of citizens with the EMS. These interaction models must support the user and help them achieve a natural interaction with the device. For example, emergency units often focus on physical stimuli and do not want to be distracted by other stimuli originating from a software application

[HCDM09]. The focus of attention must be captured by the software system in order to avoid interruptions in the user's current task.

4.1.2 Current user interfaces and interaction design

Across Europe diverse social and cultural contexts work to diverse logics and people find it difficult to mobilise technological potential with existing interfaces and interaction paradigms. Semantic interoperability is a barrier [AlKN13], experiences of information overload impede appropriation when emergency situations require responders to identify relevant information [RaNj09], there are privacy and security issues, and the different agencies involved have different priorities, information models and approaches [AlKN13]. While network assisted and network centric organizational models and forms of 'agile response' are emergent in the USA [WSSJ07, Harr06] and the Netherlands [BoWW10], much of Europe finds it challenging to introduce new approaches to emergency management and technologies for greater interoperability and collaboration, because there is a shortage of consideration to the social and organizational practices of making services interoperable and a shortage of support for the rendition of these practices into more interoperable contextures at all levels of design, specifically interface and interaction design.

Methods for handling gestures are very common in the area of emergency management. The methodology of [BaMT08] expresses an emergency management system with a tabletop display where a map of the affected area is displayed. Due to cameras above the display, gestures from the hand can be recognized. This enables the user to perform actions such as selecting, zooming, and similar, by pointing at or sweeping over the appropriate area. Besides the gesture recognition, the system also facilitates intra- and inter-team communication. This can be achieved by offering a personalized view, which can be used by a single-person or a group of people working together.

Icons are used for interaction between different team members. [FiLe07] illustrate an example of an icon language. The work of [AcDA10] describes the development of a design pattern collection for web-based emergency management. This design catalog created based on the different phases of emergency with respect to first responders and the command control stands: Preparedness, Emergency Response, Recovery, and Mitigation.

[Dene11] presents a pattern language describing the social structures within firefighting units. Nonstandardized but still often applied processes on duty, self-made tools and courses of action are described that are usually not documented but learned while on the job. Such patterns help in understanding the domain, its processes and stakeholders.

[BeLa09] shows that video is a very important source for information, and that it can be a positive contribution to the work practice. [CaTu07] describe parallels between emergency responders and gamers. They base this metaphor on the notion that gamers, just like EMS, must have a full overview about what is going on. Thus, best practices from playing or designing video games can be a worthy source. Moreover, there are standard design patterns in the context of multimedia that may be used in the conventional design process to find commonly valid solutions. The multimedia focused patterns are relevant for displaying images, sets of images, videos and similar. Such patterns can also be transferred to the design processes for mobile apps.

A pattern collection for mobile apps is provided in the Flaminco design pattern created library by Nilsson [Nils09]. Based on the assumption that humans can only process four items simultaneously, [HCDM09] suggest that the screen area should be divided into four sections, where each of the four sections should be colorised according to the content it holds.

[KeBl98] presents the integration of a participatory design approach that involves the user at a very early stage in the design process.

[GrBl08] explains the consequent application of design patterns as a step in the software system engineering process. It also gives framework for identifying and validating design patterns within the field of emergency management through a user-centered design process. They extend the user interface design process beyond the traditional ones by introducing periodic evaluations of the pattern validity.

4.1.3 Phases to design emergency support tools/apps that support effective situation awareness

In emergency communication, tools and apps can be designed to assist emergency responders accomplish effective situation awareness (SA). This can be divided into three phases –

- **Discover which citizen targeted information required to make effective decisions during emergencies.** Understanding the information needs of citizens is especially vital to accomplishing high-quality shared SA during large scale emergencies where people from different services must cooperate to achieve common objectives. While analyzing the information needed to achieve effective SA is the crucial precursor to the next two design phases, however it is often ignored phase.
- **Discover the sources where the citizen targeted information can be acquired**. While the technical community focuses on sensor technologies, much valuable information must also be acquired from other people.
- **Determine how that information can be presented in a way that best supports user goals**. Displaying information requires careful design to avoid requiring citizens to spend unnecessary time searching for and deciphering information. If the design is good then it helps responders prevent information overload.

4.1.4 Situation aware design

Situational awareness (SA) concerns “knowing what is (and has been) going on”, being aware of what is happening around you in the environment and having a shared understanding of the information.

Situational awareness is expected to be a crucial factor of team performance [BCGS05; Ends95]. In multidisciplinary settings situational awareness information is affected by abilities of individual members, their interaction with other authorities, and the environment in which they collaborate [BCGS05]. Various factors affect individual situational awareness formation: environmental (physical location, display arrangement and size etc.) and group aspects (communication, use of collaboration tools, team processes etc.). In order to assess SA during

evaluation of collaborative interfaces or awareness displays, specific factors need to be identified relevant to a particular domain.

Theory of situational awareness [Ends95] suggests that SA can be accomplished by linking an objective state of the world to its mental analogue on three main levels:

1. Level 1 of SA, is perception of relevant elements in the environment. It is an active process whereby individuals extract salient cues from the environment.
2. Level 2 is comprehension of the meaning of these cues. It involves integration of information in working memory [SPBS95] to understand how it will impact the individual's goals and objectives.
3. Level 3, projection, consists of extrapolating this information forward in time to determine how it will affect future states of the operating environment.

The third level of SA combines what the individual knows about the current situation with his or her mental model of similar events from previous experience, to be prepared for what might happen next.

4.1.5 Constituting situation awareness in mobile apps

It is fundamental that the apps provide location information to the user, enabling basic situational awareness. A user needs to be able to identify their own location, and find the locations of authorities quickly and easily. The locations of dangerous or suspicious activities, or other areas in need of Public Safety support should also be available to users through the application. Locations disseminated by this service should be error-proof, to prevent false location information from interfering with Public Safety operations.

A set of design principles should be developed based on a theoretical model of the mechanisms and processes involved in acquiring and maintaining SA in dynamic complex systems [Ends95]. These guidelines are focused on a model of human cognition involving dynamic switching between goal-driven and data-driven processing and feature support for limited operator resources, including:

1. Direct presentation of higher level SA needs (comprehension and projection) is recommended, rather than supplying only low level data that operators must integrate and interpret manually.
2. Goal-oriented information displays should be provided, organized so that the information needed for a particular goal is co-located and directly answers the major decisions associated with the goal.
3. Support for global SA is critical, providing an overview of the situation across the operator’s goals at all times (with detailed information for goals of current interest) and enabling efficient and timely goal switching and projection.
4. Critical cues related to key features of schemata need to be determined and made salient in the interface design. In particular those cues that will indicate the presence of prototypical situations will be of prime importance and will facilitate goal switching in critical conditions.
5. Irrelevant information not related to SA needs should be removed.

4.2 Design Pattern in Emergency Management

General design patterns are not necessarily applicable for safety critical environments and emergency management due to their general description and usage. In order to be applicable, design concepts or patterns must be translated and set into the proper context. Furthermore, they must be collected in a structured manner and validated periodically based on progress in the current domain.

[GrBl08] describes a framework for identifying and validating design patterns within the field of emergency management through a user-centred design process. The process also facilitates creation, update and reuse of already known patterns. They widen the user interface design process beyond the conventional ones by introducing periodic evaluations of the pattern validity. The process starts with the creation of an early draft of the UI. This step is based on analysis of the requirements with the user. During the design phase, the design is evaluated with the users based on their expectation. This process therefore results in an iterative manner to have an ongoing improvement of the UI until no more changes are needed and the UI design process halts. Within the design phase, design experts have the possibility to access a pattern library, containing important patterns identified from previous experiences (projects). The pattern library can be browsed by the experts and the results are analyzed. As a result of the analysis new patterns are created if no existing pattern fits, updated, if the existing pattern can be more generalized, and reused, if the pattern fits the expectation of the experts. This results in an up-to-date pattern library containing the experiences of different UI designs in safety critical environments.

The pattern form used by [GrBl08] is an improved version of the original work of [Alex77].

4.2.1 Categories of design patterns

The usefulness of design patterns has been recognized within the field of HCI, taking the form of a variety of pattern sub-types, such as: User interface design patterns, Interaction design patterns, User experience patterns, Usability patterns, Web design patterns, and Visualization patterns. However, there seem to be a great overlap between the different pattern types, and generally a lack of consensus regarding how they differ from one another.

In a review of design for safety-critical systems, [GrBl08] proposed the following categorization of patterns:

1. Interaction design patterns
2. User interface design patterns
3. Information visualization patterns.

4.2.2 Online pattern sets

The following pattern sets contain design patterns for UI and interaction design – basically they are intended towards web design. For use in emergency management the patterns must be carefully incorporated and validated according to the user and the context.

- Android Patterns http://www.androidpatterns.com/
- Mobile UI Patterns: http://mobile-patterns.com/
- Yahoo! Design Pattern Library http://developer.yahoo.com/ypatterns/
- UI Patterns by categories http://ui-patterns.com/patterns
- Welie.com Pattern Library http://www.welie.com/
- Endeca Pattern Site: http://patterns.endeca.com/content/library/en/home.html
- UI Anti Pattern Library: http://ui-patterns.com/blog/User-Interface-AntiPatterns
- Android Anti-Pattern: http://www.androiduipatterns.com/p/android-ui-pattern-collection.html
- Interface Design Patterns http://www.patternbrowser.org/
- Patternry http://patternry.com/patterns/
- Quince Pattern Library http://quince.infragistics.com/
- HCIpatterns.org http://www.hcipatterns.org/patterns
- The Design of Sites: http://www.designofsites.com/design-patterns/
- Designing Web Sites: http://designingwebinterfaces.com/explore

4.3 Strategies for Design Patterns and Interaction Design

Emergency management involves a great variety of different personnel, ranging from tactic personnel working in the field, to operative and strategic personnel working at EMS. All the individuals involved have different requirements and needs according to their role, tasks and the equipment they use. Thus, for mobile EmerGent apps specific characteristics must be considered, both for the EmerGent interaction in the field, and the more fixed interaction taking place in the EMS.

[CaTu07] highlights the following aspects for implementing successful emergency management system/apps:

- Exchange of information including information from in the field: Exchange of information is significantly essential in order to obtain/maintain a common situational awareness across the involved personnel.
- Just-in-time decision support: All information must be summarized to save time and to enable efficient decision-making.
- Converging user's attention on most important facts without the feeling of interruption: Emergency personnel that are occupied with carrying out orders and/or saving lives should only be interrupted when this is strictly necessary.
- Place for creativity to find a solution by the user: Due to the dynamic and complex nature of emergency situations, it is important that the systems enable its users to improvise when solving a given problem.
- Trust building between team members: Response teams working in large-scale emergency situations are typically assembled ad-hoc. As team-members may not have worked together in the past, trust-building becomes essential.
- Facilitate workflow dependent communication: gain additional information over time: For certain emergency situations predefined workflows exist that should be adopted by the emergency response personnel. A system should therefore facilitate workflow-dependent situations.

Moreover, [FWBZ08] emphasizes the following aspects in the design of user interfaces:

The UI should reduce complexity and facilitate an overview of the situation and simplify communication of information

- Focus on tasks and devices in the UI design: Different roles and tasks needs specialized equipment. Emergency management teams should be supported through devices they know well and suited for working in the field (e.g. smart phones)
- Operable by novice and experts: The apps should be designed for the usage by both professionals and novices
- Flexible UI for different situation (environment and device): The full range of the application area is not assessable from the very beginning, so flexibility in UI design is needed
- Security and privacy: Sensible data must be protected for unauthorized access, and decision making must be traceable

While these principles are applicable for various kinds of applications, they are particularly important and sensible for emergency management.

The design of EmerGent apps for emergency management needs special design activities due to specific characteristics of a crisis situation. [TCVX04] presents a framework for the design and development of emergency management systems, based on a vast literature review. They present the following design principles for emergency response systems:

1. System Directory: Hierarchical structure for all the data to browse information with a text search and to group related information.
2. Information Source and Timelines: Data must be specified by its source, time, state, location and links to information already present within the system to reach the maximal overview.
3. Open Multi-Directional Communication: Communication structure open for all interacting parties.
4. Content as Address: User specifies events as important for their tasks; content-based deliverables.
5. Up-to-date information and data displayed
6. Link of data objects: Information which is interrelated must be semantically linked
7. Authority, Responsibility, and Accountability: Information must be visible to the right persons, who have the authority and the overview of what is allowed and what is not allowed (or what is conflicting and what is not conflicting)
8. Psychological and sociological factors: Encourage and support the psychological and social needs of the crisis response team to build trust and reduce stress.

5 Interface for the Mobile EmerGent Apps

In recent years, thanks to the mass diffusion of smartphones, several applications have been designed to help facing different kind of emergencies. This need arises from the limitations of voice only emergency communication as well as from the wishes of providing novel communication methods between citizen and EMS ([GòJu14]).

Use of such applications may ensure fast methods to exchange relevant information between the caller and the EMS in an improved way (taking advantage of technologies integrated in smartphones, such as data connection, cameras, and position sensors).

5.1 Existing Interfaces

Taking into account the current situation in terms of communication between citizens and EMS (and vice-versa) and the need to improve it examples of existing interfaces have been analysed, highlighting actual methods of data sharing and interfacing with the EMS. A general guideline for emergency apps has been developed by EENA. EENA 112 emergency app document ([WWW33]) describes how an application for smartphones that interacts with the European emergency number 112 should be developed, being compliant with the requirements.

The resulting requirements arise from a questionnaire [VaLu11] which was sent out to nineteen countries-representatives of the European Union. Main results are:

- 18 countries agreed that a 112 app is a valuable supplement for location information
- 18 countries agreed that accurate GPS location information should be made available for 112 calls
- 19 countries agreed that further action should be taken and coordinated on European level
- 19 countries agreed that a communication standard (MSD) is needed to cooperate with third parties on national and international level
- 19 countries think that 112apps can be complementary as a tool for users with a communication handicap

These general guidelines have been, in practical terms, implemented by some applications existing in the market. Most of them are isolated efforts and uses non-standard way of interfacing with EMS command and control rooms software. Although there are many apps targeted to the emergency field (see, i.e. [WWW28] and [WWW29]), a sample of these has been considered hereafter with the scope of underlining how these are used in EMC phases and highlighting some of the technologies used.

The FRESS APP (http://www.fress.org)

FRESS gives citizen the capabilities to establish a communication channel with the EMS which provides more information than a traditional phone call (based on text, photo or video with communication of geolocation) and provides immediate access to the necessary answering points to respond to their emergency.

In addition, the FRESS app will allow citizens to create their customized private social emergency response network (e.g., doctor or health care group) and personal emergency response network (e.g., family and friends). It also allows:

- the citizen to customize what personal information they desire to be shared in the case of an emergency and well as who they want to notify when they have an emergency,
- the EMS to customize the form and nature of communication they are prepared to receive from the citizen.

This app is designed to help the citizens and EMS to communicate together (using innovative methods) during the response phase of EMC.

112 Denmark (http://www.112app.dk)

With Denmark's official 112 App citizens can initiate a call to the PSAP and satellite positioning system coordinates will be sent simultaneously.

This app addresses the response phase of EMC.

Echo112 (http://www.echo112.com)

Echo112 is a lifesaving smartphone app that sends citizens' location to local emergency services, worldwide.

Echo112 will know which country citizens are in and will select and call the correct emergency number; citizens' position is transmitted to the emergency operator which will be able to see where they are.

Echo 112 works also without Internet connection sending information by SMS.

This app addresses the response phase of EMC.

112 Iceland (https://itunes.apple.com/it/app/112-iceland/id530144261?mt=8)

It is the official App for Iceland's emergency service 112. The App sends an SMS to the Icelandic emergency service 112, with the phones satellite positioning system location, before calling 112

This app addresses the response phase of EMC.

115 FOR DEAF (http://www.iessolutions.eu/en/products/products-security/control-room/115-for-deaf-en)

115 for deaf allows deaf people to interact with Italian Fire Brigades emergency number (115).

The system 'user side' consists of a web application that can be used from any fixed location and mobile devices (smartphones, tablet PCs, etc.) with an Internet connection, and requires the user to complete the registration before being used. The procedure for distress call is facilitated through presentation to the user of a series of buttons associated with some types of danger, therefore the user will simply have to click the button for the type of request and possibly write text to provide more information. When the request is sent, the operations room of the Fire Department receives notification of the alert and the intervention will take

place as if the request was received by phone. Once the user submits his request, he'll wait for the Fire Brigade reply, shown on the screen.

The Control Rooms that receive such type of alerts make use of JIXEL. JIXEL allows interoperability and data exchange using internationally adopted open standards, such as the CAP (Common Alerting Protocol) used worldwide for alerts exchange and officially adopted in Italy as the protocol for incident data sharing. The JIXEL CAPGenerator and CAPViewer are used to view and take charge of the calls received by deaf people, and to generate alerts and reply messages to such people. 115-for-DEAF has been recently used in the frame of the civil protection exercise "TWIST" held in Salerno in October 2013 (the video of the event is available here: http://youtu.be/GQZvO5EOV2U).

This app addresses the response phase of EMC.

Gencat APP (http://www.gencat.cat/mobils/eng/appgencat.htm)

Gencat APP brings together mobile services created by the Generalitat de Catalunya or by third parties based on open data made available by this institution.

A host of different services are included:

- Protection Civil: emergencies and advices.
- Transport: cameras and road reports, timetables and itineraries on public transport.
- Weather: weather forecast.

Public facilities: museums, hospitals, and libraries.

This app offers notifications and methods to inform citizens about what may can occur in prevention, preparedness and recovery of EMC, and provides a fast way to contact EMS especially during the response phase. Interactions with EMS is seldom supported.

FEMA (https://itunes.apple.com/us/app/fema/id474807486)

The FEMA App contains preparedness information for different types of disasters, an interactive checklist for emergency kits, a section to plan emergency meeting locations, information on how to stay safe and recover after a disaster, a map with FEMA Disaster Recovery Center locations (one-stop centers where disaster survivors can access key relief services) and Shelters, general ways the public can get involved before and after a disaster, and the FEMA blog.

This app is useful in prevention ad preparedness phases of EMC

These examples show how smartphones are becoming an essential tool to provide information for emergency services in an automatic and fast way overcoming the limits of traditional tools. The types of information obtained are related to the common technologies present on smartphones and are specially designed to ensure the EMS to get the information that allows them to assess the full extent of reporting. In addition to the traditional methods of interfacing, such as voice calls, the aim is to use other methods, such as SMS and information sent via data connection based on the use of open standards (e.g. OASIS CAP or EDXL).

Many of these applications take advantage of some smartphones capabilities that allow new communication methods based on text, photo and video messages. Geolocation services are often part of these Apps and allow (Assisted) GPS based user location identification. At the current state applications generally cover the third phase of the EMC (response); the other phases are generally addressed by the use of notifications that inform citizens about what is happening at their position. Generally the interaction with EMS applications is seldom supported. What is clear from the previous analysis is the fact that is difficult to find applications covering each EMC phases and this should be taken into account when designing the EmerGent mobile app.

5.2 Capturing and Sharing Relevant Information for EMS using OSN

The analysis of the state of the art highlights how emergency-oriented applications mainly focus on managing the third phase of the EMC (response phase). A new generation of these applications should, therefore, enable communication also in other phases of the EMC, in order to facilitate coordination and information exchange between relevant stakeholders before, during and after emergencies.

By the use of novel applications (i.e. EmerGent mobile App), citizen would become able to cooperate with the EMS, sharing information, even simultaneously, to diverse OSNs, in order to help the prevention of possible incidents, and to contribute on returning back to normality. Information such as incident location and multimedia files, can be acquired using traditional peripherals today included in most smartphones (e.g. cameras), and then shared through social media.

In order to bring citizens to “work with” EMS, through information exchange on social media, it is necessary to perform appropriate awareness campaigns and trainings (foreseen also in EmerGent) and provide them with a “tangible” benefit, such as personalized streams of information. Thus, EMS could receive relevant information, and at the same time access multimedia and geo-location data and decide “what to share” with collaborating citizens (Authority to Citizen - A2C).

During a critical emergency situation, conscientious citizens can share, in OSNs (using EmerGent mobile app), relevant information which may allow EMS to manage the emergency from several points of view, miming a "Big Brother" effect. Photo and video sharing through social media, could provide EMS with a real time, constantly updated view of the situation, and allow them to monitor how it changes over time. If photos provide a static point of view that can be consulted quickly, the adoption of technologies as video broadcast, streaming and podcast, could give further help to EMS acquiring a more accurate view on what is occurring. (Citizen to Authority - C2A)

5.3 Interface for Supporting Emergent Collaboration using direct channels

Although one of the main features foreseen in EmerGent is the automatic analysis of big data coming from the social media to provide additional, relevant sources of information to EMS (which in turns can send feedback to citizens), we must not forget (see the description of

existing emergencies Mobile Apps in Section 5.1), the need to provide direct communication channels between the citizens and the EMS.

As a first example, thinking to a near future, health related information, both historical (e.g. case history on the therapeutic-diagnostic path of an individual) and actual (e.g. heart rate, body pressure, hydration, body temperature), could be a class of information of interest for EMS. Actual health information could be retrieved using biometric sensors sending data to smartphones; these sensors are starting to popup in some mobile devices. As an example in the recent years, mobile applications have been developed allowing heart rate measuring using smartphone camera and flash. Thanks to the technology progress, in the future, general purpose biometric sensors could be worn by people, and potentially integrated inside smartphone applications. It is understandable that healthcare information would be best targeted at EMS directly, without the need of social media sharing: thus the EmerGent APP should be designed keeping in mind different sharing needs related to locally sensed data. In Wp5 information collection should also take into account the need for classifying data sensed by the EmerGent APP ([LMBMR11], [ChRP08]).

As second example, NFC or Bluetooth technologies, already integrated in new generation smartphones, could replace the currents "dog tag" (http://en.wikipedia.org/wiki/Dog_tag), giving the possibility to obtain information about people (name, surname, blood group). The TILEAPP (http://www.thetileapp.com/) uses small Bluetooth devices to help people keep track of objects. "Sensordone" (http://www.sensordrone.com) is small sized portable device able to make different environment measurements that will be sent to a mobile application. Potentially this information could be captured and sent to the EMS (C2A) (e.g. temperature, humidity, pressure, blood alcohol level, air quality, light intensity and possible gas leaks).

As a third example in case that, for a given emergency, the EMS does not have any information of what is happening in a specific area (e.g. no posts or media files are available), they could ask citizens directly to help.

A way to do that could be using Social Media and the EmerGent mobile app, informing citizens about what kind of feedbacks are needed. To interface this information with the EmerGent App, it is necessary to build structured metadata that will be embedded and broadcasted inside social media posts feedback. If a citizen that uses the application is within the area of interest, he/she will be informed about that and told to capture specific data requested by EMS.

It is clear that those examples show some of the potentialities of a combined use of Social Media, a dedicated EmerGent mobile app for next generation smartphones and arising companion devices. It is out of the scope here to define the needed actual functionalities from EMS perspective but it is rather important to highlight them to imagine potential scenarios of “social media enabled” communication.

6. Conclusion

In this document we have presented a technical analysis of OSNs in emergencies in order to discover the potentials of new communication media at different phases in the EMS.

At the outset, we analysed how selected OSNs implement privacy and data protection. Common characteristics have been identified across the OSNs' policies for end users and developers and the result is categorised as requirements and assigned to a pertinent data protection principle where appropriate. This analysis is beneficial for the Task 7.5.

Technical shortcomings of selected existing OSNs are described in section 3. Access to information from social media is generally possible. Besides mere statistical data, such as profile information, it is possible to request additional information about the connection of users and groups. The technical possibilities for the integration of other applications are widely diversified by the provider; especially in the mobile sector the trend goes towards mobile SDKs, which make standard functionalities of the social networks accessible to each type of program similar to Facebook's "Like" button. The access to internal data is subjected to security measures such as the OAuth protocol. The protection of personal data is in the foreground, so that access to it must be approved by the users themselves. In addition to safety aspects, the use of these services is only possible in the context of the granted query rate. If, for example, the use a search query met the capacity in a specific time interval, then the use would only be possible after the end of this interval. These rate limits are dynamic for many requests and systems; if the given amount of possible requests is not sufficient, it will be necessary to make a separate request to the system owner. It is required to provide detailed information to the relevant application and provide access to this. However, the limitations allow the connection of different services of the individual networks, so the exchange between them is possible and information according to their quality and relevance can be collected and disseminated.

In section 4 design patterns for citizens targeted information in the view of emergency management are discussed. Design patterns are used to describe best practices and effective design solutions, and for capturing and sharing design knowledge with other people faced with the same problem and context. Supporting collaboration across, authorities and citizens in complex and time-critical situations is extremely challenging. Designing suitable interfaces and interactions is a vital aspect for enabling actors to deal with this complexity. Moreover, significance of situation awareness in designing mobile apps is discussed in this section. In T3.4, design patterns and interaction issues related to mobile devices are combined with the gathered requirements to scope out the most relevant requirements for the intended purpose.

Finally, chapter 5 describes need for distinctive interface for the mobile EmerGent apps. From the state of the art of current emergency applications, it is clear that there aren't applications covering at the same time all phases of EMC, and that the one existing generally introduce new methods for sharing information taking advantage of last generation smartphones.

Thinking to the EmerGent scenario, some technical suggestions have been provided and could be taken into account during the EmerGent mobile app design for what concerns information sharing on OSNs and direct communication between EMS and citizens (in all phases of EMC and for each communication direction).

References

[AHHST12] Abel, F., Hauff, C., Houben, G.-J. , Stronkman, R., Tao, K. (2012). Twitcident: Fighting Fire with Information from Social Web Streams, in Proceedings of the 21st International Conference Companion on World Wide Web, ser. WWW '12 Companion: New York, NY, USA: ACM, pp. 305–308.

[AcDA10] Acuna, P., Diaz, P., and Aedo, I. (2010). Development of a Design Patterns Catalog for Web-based Emergency Management Systems. In Proceedings of the 7th International ISCRAM Conference. ISCRAM.

[Alex77] Alexander, C. (1977). A Pattern Language: Towns, Buildings, Construction. Oxford University Press, New York, New York, USA.

[AlKN13] Allen, D. K., Karanasios, S., & Norman, A. (2013). Information sharing and interoperability: the case of major incident management. European Journal of Information Systems. Advance online publication 18 June 2013 doi:10.1057/ejis.2013.8.

[ApHJ13] Apaolaza, A., Harper, S., & Jay, C. (2013). Understanding Users in the Wild Categories and Subject Descriptors.

[BaMT08] Bader, T., Meissner, A., & Tscherney, R. (2008). Digital Map Table with Fovea-Tablett®: Smart Furniture for Emergency Operation Centers. *Proceedings of the 5th international ISCRAM Conference.* Washington, DC, USA, 679–688.

[Ball13] Ballve, M. (2013). The World's Largest Social Networks. Retrieved May 20, 2014, from http://www.businessinsider.com/the-worlds-largest-social-networks-2013-12#ixzz30MjZz5TK

[BeLa09] Bergstrand, F. and Landgren, J. (2009). Information Sharing Using Live Video in Emergency Response Work. In Information Sharing Using Live Video in Emergency Response Work. Gothenburg, Sweden.

[BoWW10] Boersma, K., Wagenaar, P., & Wolbers, J. (2010). Organizing Emergent Safety Organizations: The Travelling of the Concept 'Netcentric Work' in the Dutch Safety Sector. In Iscram (Ed.), Proceedings of the 7th International Conference on Information Systems for Crisis Response and Management.

[BCGS05] Bolstad, C.A., Cuevas, H., Gonzalez, C., and Schneider, M. (2005). Modeling shared situation awareness. In Proceedings of the 14th Conference on Behavior Representation in Modeling and Simulation (BRIMS), Los Angles, CA, USA.

[Borc01] Borchers, J. (2001). A Pattern Approach to Interaction Design 1st Ed. John Wiley & Sons.

[CaTu07] Carver, L. and Turoff, M. (2007). Human Computer Interaction: the human and computer as a team in emergency management information systems. Commun. ACM 50, 3, 33-38.

[ChRP08] Chan, V., Ray, P. and Parameswaran, N. (2008). Mobile e-Health monitoring: an agent-based approach.

[Dene11] Denef, S. (2011). A pattern language of firefighting frontline practice to inform the design of ubiquitous computing. Ph.D. thesis, Delft University of Technology.

[Ends95] Endsley, M.R. (1995). Measurements of situation awareness in dynamic systems. Human Factors, 37(1):65–84.

[FiLe07] Fitrianie, S. and Leon, R. (2007). A Visual Communication Language for Crisis Management. The International Journal of Intelligent Control and Systems, 208–216.

[FiRo09] Fitrianie, S., & Rothkrantz, L. (2009). Computed Ontology-based Situation Awareness of Multi-User Observations. *ISCRAM 2009: International Community on Information Systems for Crisis Response and Management*, (S. 13--21). Gothenburg, Sweden.

[FiRo07] Fitrianie, S., & Rothkrantz, L. J. (2007). A Visual Communication language for Crisis Management. *The International Journal of Intelligent Control and Systems*, S. 208-216.

[FiYR08] Fitrianie, S., Yang, Z., & Rothkrantz, L. (2008). Developing Concept-Based User Interface Using Icons for Reporting Observations. *ISCRAM 2008: International Community on Information Systems for Crisis Response and Management.* Washington DC, USA.

[FWBZ08] Flentge, F., Weber, S. G., Behring, A., & Ziegert, T. (2008). Designing Context-Aware HCI for Collaborative Emergency Management. *Workshop on HCI for Emergencies.*

[GóJu14] Gómez, B., Juiz, C. (2014). Emergency Web App for Accessing the Medical Emergency Services:http://www.thinkmind.org/download.php?articleid=etelemed_2014_1_30_40185.

[GrBl08] Grill, T. and Blauhut, M. (2008). Design Patterns Applied in a User Interface Design (UID) Process for Safety Critical Environments (SCEs). In HCI and Usability for Education and Work, A. Holzinger, Ed. Lecture Notes in Computer Science Series, vol. 5298. Springer Berlin / Heidelberg, 459–474.

[Harr06] Harrald, J. R. (2006). Agility and Discipline: Critical Success Factors for Disaster Response. The ANNALS of the American Academy of Political and Social Science, 604, 256-272.

[HCDM+09] Humayoun, S., Catarci, T., De Leoni, M., Marrella, A., Mecella, M., Bortenschlager, M., and Steinmann, R. (2009). The WORKPAD User Interface and Methodology: Developing Smart and Effective Mobile Applications for Emergency Operators. In Universal Access in Human-Computer Interaction. Applications and Services, C. Stephanidis, Ed. Lecture Notes in Computer Science Series, vol. 5616. Springer Berlin / Heidelberg, 343–352.

[Heis14] Heise, ?. (2014). *WhatsApp kickt Open-Source-Projekte von Github*. URL http://www.heise.de/open/meldung/WhatsApp-kickt-Open-Source-Projekte-von-Github-2112486.html

[KeBl98] Kensing, F. and Blomberg, J. (1998). Participatory Design: Issues and Concerns. Computer Supported Cooperative Work (CSCW) 7, 3, 167–185.

[KSKS02] Krahnstoever, N., Schapira, E., Kettebekov, S., & Sharma, R. (2002). Multimodal Human-Computer Interaction for Crisis Management Systems. *Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision*, (S. 203-208). Washington, DC, USA.

[LMBM+11] Luxton, D., McCann, R., Bush, N., Mishkind, M. and Reger, G (2011). mHealth for Mental Health: Integrating Smartphone Technology in Behavioral Healthcare.

[MiYa07] Micire, M., & Yanco, H. (2007). Improving Disaster Response with Multi-Touch Technologies. *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007.*, (S. 2567 -2568).

[Nils09] Nilsson, E. G. (2009). Design patterns for user interface for mobile applications. Advances in Engineering Software 40, 12, 1318–1328.

[Nova14] Novak, M. (2014). Unwrapping Tumblr. Retrieved May 20, 2014, from http://unwrapping.tumblr.com/tagged/tumblr-limits

[RaNj09] Rake, E., & Njå, O. (2009). Perceptions and performances of experienced incident commanders. Journal of Risk Research, 12, 5, 665–685.

[ReSc14] Reuter, C., & Scholl, S. (2014). Technical Limitations for Designing Applications for Social Media. In Workshop-Proceedings der Tagung Mensch & Computer 2014 (pp. 1–8). München, Germany: Oldenbourg-Verlag.

[SPBS95] Salas, E., Prince, c., Baker, D.P., and Shrestha, L. (1995). Situation awareness in team performance: Implications for measurement and training. Human Factors, 37(1):123–136.

[ScLu07] Schümmer, T. and Lukosch, S. (2007). Patterns for Computer-Mediated Interaction. John Wiley & Sons, Chistester, West Sussex, England.

[Stat14] Statista, ?. (2014). Leading social networks worldwide as of January 2014, ranked by number of active users (in millions). Retrieved May 28, 2014, from http://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

[TCVX04] Turoff, M., Chumer, M., van de Walle, B., & Xiang, Y. (2004). The Design of a Dynamic Emergency Response Management Information System (DERMIS). *The Journal of Information Technology Theory and Application (JITTA)*, (S. 1-35).

[VaLu11] Van Hunnik, C., Lumbrera, C. (2011). EGEA Position paper 112 emergency-app's for smartphones.

[WSSJ07] Walker, G. H., Stanton, N. A., Salmon, P. M., & Jenkins, D. P. (2007). A Review of Sociotechnical Systems Theory: A Classic Concept for New Command and Control Paradigms. Retrieved from http://www.hfidtc.com/research/command/c-and-c-reports/phase-2/HFIDTC-2-1-1-1-2-command-paradigms.pdf

[WWW01] dev.twitter.com. (2014a). REST API v1.1 Limits per window by resource. Retrieved June 12, 2014, from https://dev.twitter.com/docs/rate-limiting/1.1/limits

[WWW02] dev.twitter.com. (2014b). The Streaming APIs. Retrieved June 11, 2014, from https://dev.twitter.com/docs/api/streaming

[WWW03] dev.twitter.com. (2014c). The Twitter REST API. Retrieved June 11, 2014, from https://dev.twitter.com/docs/api

[WWW04] developer.linkedin.com. (2014). API Overview. Retrieved June 11, 2014, from http://developer.linkedin.com/apis

[WWW05] developers.facebook.com. (2014a). Graph API Reference. Retrieved June 12, 2014, from https://developers.facebook.com/docs/graph-api/reference/v2.0

[WWW06] developers.facebook.com. (2014b). Things you should know. Retrieved June 12, 2014, from https://developers.facebook.com/policy/

[WWW07] developers.google.com. (2013a). Google JavaScript-API. Retrieved June 11, 2014, from https://developers.google.com/+/web/api/javascript?hl=de

[WWW08] developers.google.com. (2013b). Google+ API. Retrieved June 11, 2014, from https://developers.google.com/+/api/?hl=de

[WWW09] developers.google.com. (2013c). Google+ Platform for Android. Retrieved June 11, 2014, from https://developers.google.com/+/mobile/android/getting-started

[WWW10] developers.google.com. (2013d). Google+ Platform for iOS. Retrieved June 11, 2014, from https://developers.google.com/+/mobile/ios/getting-started

[WWW11] developers.google.com. (2013e). YouTube Data API (v3). Retrieved June 11, 2014, from https://developers.google.com/youtube/v3/?hl=de

[WWW12] developers.google.com. (2013f). YouTube Live Streaming API. Retrieved June 11, 2014, from https://developers.google.com/youtube/v3/live/?hl=de

[WWW13] developers.google.com. (2014a). App Engine Pricing. Retrieved June 11, 2014, from https://developers.google.com/appengine/pricing?hl=de

[WWW14] developers.google.com. (2014b). Google Developer Console. Retrieved June 12, 2014, from https://code.google.com/apis/console

[WWW15] developers.google.com. (2014c). Hangouts-API-Referenz. Retrieved June 11, 2014, from https://developers.google.com/+/hangouts/api/

[WWW16] developers.google.com. (2014d). Quota usage. Retrieved June 12, 2014, from https://developers.google.com/youtube/v3/getting-started#quota

[WWW17] developers.google.com. (2014e). YouTube Analytics API. Retrieved from https://developers.google.com/youtube/analytics/?hl=de

[WWW18] eMarketer. (2014). Smartphone Users Worldwide Will Total 1.75 Billion in 2014. Retrieved May 28, 2014, from http://www.emarketer.com/Article/Smartphone-Users-Worldwide-Will-Total-175-Billion-2014/1010536

[WWW19] Extracting information nuggets from disaster related messages in social media [online], available under: http://chato.cl/papers/imran_elbassuoni_castillo_diaz_meier_2013_extracting_information_nuggets_disasters.pdf [retrieved: 2014/04/27]

[WWW20] facebook.com. (2014a). Chat API. Retrieved May 20, 2014, from https://developers.facebook.com/docs/chat

[WWW21] facebook.com. (2014b). Facebook Query Language (FQL) Overview. Retrieved May 20, 2014, from https://developers.facebook.com/docs/technical-guides/fql/

[WWW22] facebook.com. (2014c). Keyword Insights API. Retrieved May 20, 2014, from https://developers.facebook.com/docs/keyword_insights

[WWW23] facebook.com. (2014d). Public Feed API. Retrieved May 20, 2014, from https://developers.facebook.com/docs/public_feed/

[WWW24] Humanitarian Computing Library: http://humanitariancomp.referata.com/wiki/Welcome [retrieved: 2014/04/29]

[WWW25] instagram.com. (2014a). API Terms of Use. Retrieved from http://instagram.com/developer/#

[WWW26] instagram.com. (2014b). Instagram API Endpoints. Retrieved May 20, 2014, from http://instagram.com/developer/endpoints/#

[WWW27] instagram.com. (2014c). Real-time Photo Updates. Retrieved May 20, 2014, from http://instagram.com/developer/realtime/#

[WWW28] missionmode.com (2012), 15 Disaster and crisis apps for iphone and ipad , http://blog.missionmode.com/blog/15-disaster-and-crisis-apps-for-iphone-and-ipad.html

[WWW29] missionmode.com (2012), 15 disaster and crisis apps for android http://blog.missionmode.com/blog/15-disaster-and-crisis-apps-for-android.html

[WWW30] stackoverflow.com. (2012). Facebook graph API rate limit and batch requests. Retrieved June 11, 2014, from http://stackoverflow.com/questions/8805316/facebook-graph-api-rate-limit-and-batch-requests

[WWW31] Statista. (2014). Leading social networks worldwide as of January 2014, ranked by number of active users (in millions). Retrieved May 28, 2014, from http://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

[WWW32] tumblr.com. (2014). Tumblr API. Retrieved June 11, 2014, from http://www.tumblr.com/docs/en/api/v2

[WWW33] 112 Apps EENA Operations Document (2014): http://www.eena.org/ressource/static/files/2014_02_25_apps_v1.0.pdf