diretto: A Toolkit for Distributed Reporting and Collaboration

Benjamin Erb, Stefan Kaufmann, Tobias Schlecht, Florian Schaub, Michael Weber

Institute of Media Informatics, Ulm University, Germany

Abstract

The goal of the diretto project is the creation of an extensible infrastructure and easy-to-use toolset for distributed on-site media reporting and collaborative event coverage in real-time. It empowers collocated users to participate dynamically in event reporting, and facilitates collaboration with remote users. For example, to cover public events or support disaster relief missions with on-site information. The diretto platform focuses on scalability to support large crowd participation. Our platform currently supports smartphone clients, a reporting solution for SLR cameras, and a rich web application for remote collaborators. diretto is easily extensible and can be tailored to mission-specific requirements.

1 Introduction

The popularity of smartphones with Internet connectivity has facilitated data transmissions from almost any point on earth. Camera phones allow instant uploads to photo sharing sites; journalists with high-end cameras only require a portable netbook in order to send pictures and commentary to any news desk worldwide. Already, public events are captured by a plethora of devices while they unfold. While emergency and police services might record their deployments to preserve evidence, normal people take snapshots and videos to preserve their impressions and increasingly share them online. Sometimes, citizens even provide the first distributed images of significant events, e.g., the 2008 Mumbai terrorist attacks (Arthur 2008). Eye-witnesses documented the events on social media services long before traditional reporters arrived to provide their viewpoint of the situation.

By making use of the extensive sensor capabilities found in current smartphones people can be seen as mobile nodes interacting in a multimedia sensor network. Cuff et al. term this *urban sensing* (Cuff et al. 2008). Sheth defines a citizen sensor network as "[an] interconnected network of people who actively observe, report, collect, analyze, and disseminate information via text, audio, or video messages." (Sheth 2009). But as long as participants know little about each other and conduct their coverage only within their own scope, network synergy effects may be lost. Furthermore, if reported information is spread across different platforms and sharing services, often with limited or none context information, obtaining a comprehensive view of the situation is inherently difficult. As a result, information that might have been critically important for involved parties directly after its creation, may fail to reach them in due time. Data may only be distributed hours later, or may be missed by parties that would benefit from it due to the number and heterogeneity of service outlets. Therefore, a citizen sensing network for collaborative distributed reporting should support the submission and collection of multimedia data annotated with machine-readable metadata, aggregate the provided information, and present a coherent picture of the situation.

In this paper, we propose *diretto, the distributed reporting toolkit* as such a scalable solution for collaborative event coverage. diretto is an extensible platform that can support a variety of use cases. diretto offers multimedia reporting capabilities from mobile devices, supporting image, video, audio, and text reports. Reported media items are annotated with context metadata and are instantaneously available to other users. The diretto platform offers multiple mechanisms to facilitate collaborative evaluation and interpretation of reported data between eye witnesses and remote users. Additionally, diretto also supports the live integration of external services like flickr or Twitter.

In the rest of this paper, we first discuss potential use cases and derive corresponding requirements in Section 2. In Section 3, we present the diretto architecture, followed by a description of our open source reference implementation in Section 4. Section 5 discusses related work and Section 6 concludes the paper.

2 Use Cases and Requirements

Collaborative reporting systems can be employed in different scenarios. In the development of diretto we focused on two general use case categories distinguished by their criticality and involved parties.

Rescue and relief missions. Rescue workers and police forces can use collaborative reporting to support coordination of help after a disaster, e.g. an earthquake. Currently, rescue organizations rely mainly on radio dispatches. Images and videos could give the control center a detailed picture of the situation that may be otherwise hard to convey with oral reports. If media items are automatically annotated with spatiotemporal information, dispatchers could be relieved from manually cataloguing and mapping radio dispatches. Field workers and civilian remote users can also collaboratively create and update information in a concerted effort, as shown by the Haiti mapping effort after the earthquake in 2010[1]. With a collaborative reporting system in place, these efforts can be properly supported. Furthermore, chains of events and their relations can be analyzed to inform decisions on where aid needs to be

[1] http://wiki.openstreetmap.org/wiki/WikiProject_Haiti

prioritized. A challenge for such systems is the potential destruction of communication infrastructure in the course of a natural disaster.

Public live events. Public events, such as festivals, demonstrations, or sports matches, commonly attract large crowds, but are also of interest to people not present. The utilization of collaborative reporting tools can offer an immersive experience to remote spectators, if content is available almost in real-time and presented in an accessible manner. Live coverage and real-time availability is also of interest to media corporations. So far, runners had to collect memory cards from mobile photographers and rush them to the news desk in order to achieve real-time publishing of imagery. With a collaborative reporting system, the picture desk would have instant access to shot material and could benefit from meta information supplied with the footage. Participants could also provide supplementary material from different vantage points.

In both use cases, information might also be of interest retrospectively. With data being tagged spatiotemporally, chains of events and probable causes can be reconstructed forensically. Possible applications include the evaluation of measures taken by relief organizations and their effects, or the documentation of police activities during demonstrations.

From these use cases, we can generalize requirements for a collaborative reporting platform:

- *Multimedia support.* Submitted reports may consist of text, photos, audio or video footage. The platform should provide generic mechanisms to handle various media types. Users should be able to augment raw footage with improved versions, e.g. video transcripts.
- *Spatiotemporal metadata.* Each media item should be associated with spatiotemporal metadata about its origin and the context of its creation to facilitate data analysis and presentation. The platform needs to take inaccurate metadata into account, and should provide mechanisms to enhance metadata quality a posteriori.
- *Live availability:* A collaborative reporting platform should provide almost instant access to reports while events are unfolding in order enable users to evaluate ongoing events.
- *Collaboration.* Users should be supported in contributing additional information to raw data, collaboratively analyzing and evaluating data, and enriching gathered information.
- *Accessible interfaces.* Gathered information must be easily accessible to be of use for users. Appropriate visualizations should facilitate exploration of reports within their spatiotemporal context. Query and filtering capabilities should also be provided.
- *Robust and lightweight communication.* Due to mobility of on-side reporters and potentially congested communication infrastructure, communication protocols should be robust against intermittent connectivity. Protocols should also be lightweight because of resource constraints of mobile devices, i.e., battery life and bandwidth.
- *Scalability.* Momentary situations on-site can cause short reporting bursts and high activity peaks. A platform must cope with highly concurrent submissions and requests and scale with large user bases in large-scale deployments.

3 diretto Architecture

Our architecture design takes the outlined requirements into account and focuses on a scalable, extensible and versatile distributed platform. Our modular platform design can be adaptable to various use cases and their domain-specific requirements. The diretto architecture only takes into account the application layer. While our communication protocols are robust against intermittent connectivity, we assume the existence of an underlying operational network infrastructure. The problem of providing such an infrastructure in disaster response scenarios, is being addressed by others (Manoj & Baker 2007), e.g., based on wireless ad-hoc communication (Portmann & Pirzada 2008).

Our architecture follows the client-server model. On-site clients collect and upload documents to a backend infrastructure, which manages user-provided content and handles redistribution to on-site and remote users. This limits local traffic for mobile nodes and allows outsourcing of expensive operations, such as content transcoding, to the backend.

A media report is represented by a *document*. A document consists of multiple *attachments* and *metadata*. The first attachment is the original unaltered footage. Alternative variants, such as color-corrected photos or stabilized video, can be added to the document as attachments to improve document quality. The metadata contains spatiotemporal context information, as well as the document's MIME type, file size, original author, and upload time. The diretto platform supports *metadata correction* to address inaccuracies. Users can suggest improved metadata information, eventually leading to more accurate values. For example, inaccurate position information can be updated based on objects depicted in the media item.

3.1 Collaboration Features

Collaboration support is the core feature of the diretto platform. Users can assign *tags* to any document. Based on tags, the large amount of gathered documents can be categorized and sorted. The tagging system allows arbitrary strings to support dynamic forming of folksonomies, rather than requiring strict ontology adherence. While adjacency can be derived from spatiotemporal document origins, diretto also supports *links* to model causal relations between documents.

In a system where users can add information to any document, the question of reliability of information arises, although a document's original data and metadata can only be amended and not changed. The diretto platform employs *voting* as a relatively simple yet powerful mechanism to assign credibility to user generated data. Users can vote whole documents, individual attachments, and metadata items, including tags and links, up or down, in order to convey their trust in the item's accuracy. Through the collaborative voting process valuable entries can emerge over time.

Further collaboration features support coordination and evaluation. Adding *comments* to documents facilitates discussion about the document's meaning. *Messages* can be exchanged independently of documents to enable organization of smaller groups and coordination of their efforts. *Tasks* extend on this concept by requesting reporting or media acquisition from on-site users. For example, reports on a distinct position or specific objects of interest not covered before.

3.2 Scalability and Communication

Assuming that the per device uplink bandwidth is sufficient to transport multimedia data, the platform only requires scaling on the server-side. We use a modularized server design to ease scalability. Document storage, metadata storage, a service endpoint for client-server interaction, and a client notification service are realized as independent modules. The storage services are built around operations with append-only semantics to facilitate convergence towards eventual consistency in replicated services deployment. Documents can be stored in local or distributed file systems; commercial cloud storage services could be used in large-scale deployments.

Our communication API for information exchange between clients and server is based on a HTTP-based web service following the REST paradigm (Representational State Transfer), which is more lightweight than “big” web services. See (Pautasso et al. 2008) for a detailed comparison. The advantage of a HTTP-based solution is the fact that most mobile platforms support it also enables reuse of caching components or reverse proxies in the backend when needed. Messages are encoded using a JSON format, also in accordance with our requirements for lightweight protocols.

The diretto platform uses a publish/subscribe mechanism to notify clients about new content in a timely manner. Due to the asynchronous nature of publish/subscribe, the paradigm is very suitable for mobile environments with intermittent connectivity (Huang & Garcia-Molina 2004). New documents and user generated entries are published into channels triggering instant notification of subscribers. Filtering functions and dedicated channels can be used for slicing document streams, i.e. depending on the spatial origin or the media type.

The publish/subscribe mechanism also enables auxiliary clients to carry out tasks automatically when new documents have been uploaded. Thus, auxiliary clients can encapsulate predefined workflows, such as creating thumbnail attachments for photo uploads. By outsourcing tasks and specific business logic to auxiliary clients, the core platform remains lean and manageable. Similarly, a registry for endpoints of third-party services enables integration of additional features and services. For example, a map provider that supplies specialized overlays.

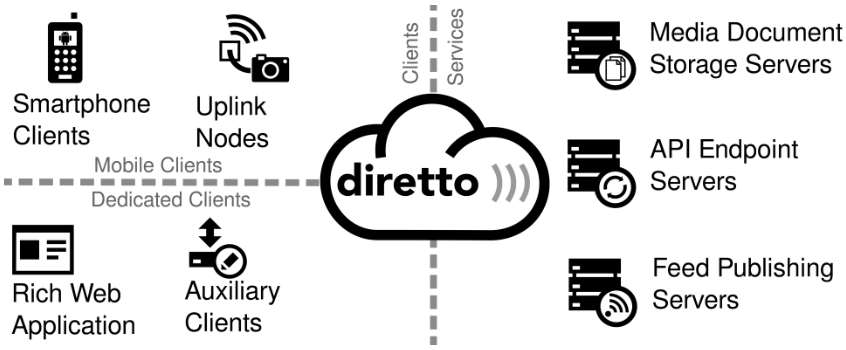

Figure 1: The architecture of the diretto platform. Icons courtesy of picol.org

4 Reference Implementation

In the following, we outline the diretto reference implementation. Our architecture is mainly based on standard web technologies to simplify integration of new features, services, or clients. The reference implementation is released open source on the diretto website.[2] Figure 1 illustrates the implementation of the diretto architecture.

4.1 Backend Services

The diretto server implementation consists of API, media storage and feed nodes. The API nodes expose the RESTful web service to clients. Media nodes accept, store, and provide access to documents. Feed nodes implement the publish/subscribe system and offer Atom and KML feeds for different channels. The nodes have been developed with node.js, an asynchronous I/O framework for highly scalable network applications.

The modular architecture allows for both horizontal and vertical scalability. For small deployments all three nodes and the database can be deployed on one server. Thus, relief workers could carry their own backend system with them, using a laptop as a dedicated diretto server. In large-scale deployments, on the other hand, heavy load can be met by adding additional servers and spawning new node instances. We achieve storage scalability by storing document metadata, tags, links, comments, and messages in CouchDB, a non-relational, eventual consistent database that supports replication between backend nodes. The media node currently stores documents in the local file system, but it would be straightforward to integrate an external cloud storage or content delivery network. The feed nodes use the open PubSubHubbub protocol for efficient and scalable notification dissemination to subscribers.

[2] diretto website: http://www.diretto.org

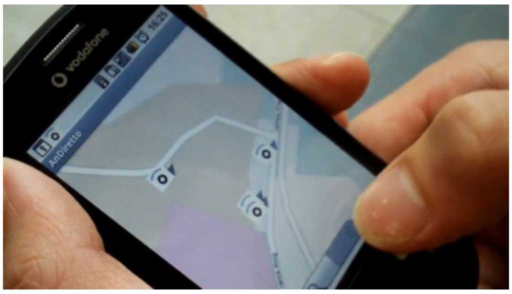

Figure 2: AnDiretto - Android-based smartphone client.

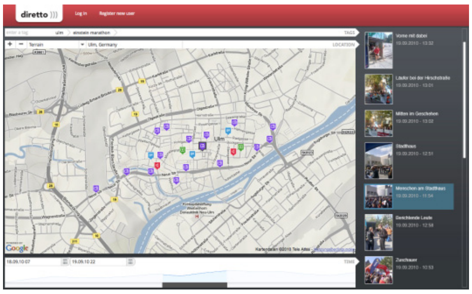

Figure 3: User interface of the rich web application.

4.2 Mobile Clients

Mobile clients facilitate collaboration of on-site users. Currently, the diretto reference implementation offers smartphone clients for Android and Java ME, as well as an upload node for SLR cameras. Mobile clients make use of a Java library that abstracts from diretto API interactions and eases development.

The mobile clients enable users to record and upload photos, video, and audio documents. Documents are automatically tagged with the current location and creation date. Users can also add relevant tags, tags used by other participants in proximity are provided as suggestions. The applications also provide a location-based view on the ongoing reporting process. Figure 2 depicts the Android client (Andiretto).

The SLR upload node is a prototype for the integration of professional cameras with diretto. An off-the-shelf digital SLR camera connected via USB to the uplink device in the photographer's backpack. When an image is taken it is automatically copied by the uplink node, tagged with geospatial metadata and uploaded to diretto. Currently, the uplink node runs on a Linux-based netbook with UMTS and GPS adapters. Inside the backpack, a foam padded metal case protects the uplink device against shocks and physical stress.

4.3 Rich Web Application

The rich web application enables collaboration with remote users. Besides uploading and displaying data, users can review, vote, and comment on documents and their metadata. Filtering capabilities help users to cope with massive data sets. Displayed documents can be reduced by applying dicing operations on an underlying data cube with the dimensions location, time, and tags. An example of the user interface's prototype is shown in Figure 3.

5 Related Work

Multiple systems exist that allow users to upload and share user-generated content or report incidents, but provide not the versatility offered by diretto. The microblogging service Twitter has proven itself as a valuable tool for fast sharing of information in crisis contexts, for instance during the 2008 Mumbai bombings (Arthur 2008). In order to make reports more accessible for evaluation, a prescriptive syntax for such events has been proposed (Starbird & Stamberger 2010), while others resort to analyzing the complete textual conversation for this purpose (Nagarajan et al. 2009). As a drawback, Twitter only supports short text messages, requiring links to multimedia items scattered on third party services, which makes it difficult to obtain a comprehensive view of a situation just from Twitter. In contrast to diretto, incorrect information is hard to correct once it has spread, since messages cannot be altered retroactively and might have been "re-tweeted" without verification by influential Twitterers with many followers. Mobshare (Sarvas et al. 2004), CoMedia (Jacucci et al. 2007), and mGroup (Jacucci et al. 2005) are systems focused on cooperative photo gathering and sharing. Micro-Blog (Gaonkar et al. 2008) is another collaborative system supporting audio and video uploads, which takes battery constraints of client devices into account. All three systems put little emphasis on the subsequent collaborative interpretation of the gathered data. In contrast, diretto allows remote participants to play a vital role in the evaluation and, if necessary, correction of the incoming data while the event still unfolds or from a forensic perspective.

In the support of humanitarian missions and disaster response, several projects have been focusing on the development of dedicated systems to facilitate collaborative response to disaster situations. Palen et al. describe a comprehensive vision for systems integrating public respondents and official response organizations (Palen et al. 2010). A notable real-world application in this area is the crisis mapping tool Ushahidi, which first received greater attention when Kenyan citizens used it to share information during the 2007-2008 Kenyan crisis (Okolloh 2009). Users can submit reports via text message or email to a Ushahidi server which displays them on an interactive map. Disasters2.0 (Camarero Puras & Iglesias Fernández 2009) is a crisis management system which already fulfills many of our requirements (cf. Section 2). But Disasters2.0 is specifically tailored to disasters and, therefore, built around entities like "firemen" and "injured persons", making it hard to adapt for other types of collaborative reporting. Also, apart from the specified entity types, no other visual information recorded on-site can be submitted.

This short overview of related work shows that many systems address specific aspects of collaborative reporting but do not provide a comprehensive and versatile approach. The diretto system provides a unified platform for collaborative reporting which could also integrate other systems, such as Twitter, as external information sources.

6 Conclusions

The diretto platform provides a generic toolkit for distributed reporting and collaboration between users on-site and remote ones. We based diretto on requirements generalized from the use cases public live event coverage and rescue and relief mission support, and believe that it can also be applied to other scenarios. The diretto architecture is designed for high scalability. A lightweight web service provides robust communication with backend services.

Publish/subscribe channels ensure instant notification of clients in case of new content. Automated processing of incoming data is performed by auxiliary clients. Mobile clients for smartphones and even SLR cameras already exist, and our rich web application supports participation of remote users in collaborative reporting activities. Users can enhance documents with comments, tags, and links to add semantic interpretation and organize available data. Different representations of the same item are encapsulated as attachments in a document. A rating system facilitates crowd-based relevance and credibility assessment of recorded information.

Our open-source reference implementation is a proof of concept of our design. Now, real world deployments and field studies are required to evaluate the performance in critical situations. We also plan to extend the architecture with domain-specific plugins in order to provide tailored services on top of the generic platform. We are currently developing such a plugin for disaster response scenarios. We plan to use collective intelligence and data mining approaches to support and automate extraction of relevant events and patterns from gathered data. Another important aspect is enhancing the presentation and visualization of gathered information and providing focused feedback to on-site users.

References

Arthur, C. (2008). How Twitter and Flickr recorded the Mumbai terror attacks. *The Guardian, 27 Nov 2008.*

Camarero Puras, J. & Iglesias Fernández, C. A. (2009). Disasters 2.0: Application of web 2.0 technologies in emergency. In *Proc. 6th Intl. ISCRAM Conference*. Gothenburg.

Cuff, D., Hansen, M. & Kang, J. (2008). Urban sensing: out of the woods. *Communications of the ACM, 51*, 24–33

Gaonkar, S., Li, J., Choudhury, R. R., Cox, L. & Schmidt, A. (2008). Micro-blog: sharing and querying content through mobile phones and social participation. In *Proc. Conference on Mobile systems, applications, and services (MobiSys'08).* ACM, New York, NY, 174–186.

Huang, Y. & Garcia-Molina, H. (2004). Publish/subscribe in a mobile environment. *Wireless Networks - Special issue: Pervasive computing and communications, 10,* 643–652.

Jacucci, G., Oulasvirta, A., Salovaara, A. & Sarvas, R. (2005). Supporting the shared experience of spectators through mobile group media. In *Proc. 2005 intl. ACM SIGGROUP conference on supporting group work.* ACM, New York, NY. 207–216.

Jacucci, G., Oulasvirta, A., Ilmonen, T., Evans J. & Salovaara A. (2007). Comedia: mobile group media for active spectatorship. In *Proc. conference on Human factors in computing systems (CHI'07).* ACM, Gaithersburg, MD, 1273–1282.

Manoj, B. & Baker, A. (2007). Communication challenges in emergency response. *Communications of the ACM, 50(3),* 51–53.

Nagarajan, M., Gomadam, K., Sheth, A., Ranabahu, A., Mutharaju, R. & Jadhav, A. (2009). Spatio-temporal-thematic analysis of citizen sensor data: Challenges and experiences. In *Web Information Systems Engineering (WISE'09).* Posen. S. 539–553.

Okolloh, O (2009). Ushahidi, or “testimony”: Web 2.0 tools for crowdsourcing crisis information. In *Change at Hand: Web 2.0 for development.* Nottingham: Russell Press. S. 65–70.

Palen, L., Anderson, K. M., Mark, G., Martin, J., Sicker, D., Palmer, M., & Grunwald, D. (2010). A vision for technology-mediated support for public participation & assistance in mass emergencies & disasters. In *Proc. ACM/BCS 2010 Conference on Visions of Computer Science.* British Computer Society, Edinburgh.

Pautasso, C., Zimmermann, O. & Leymann, F. (2008). Restful web services vs. “big” web services: making the right architectural decision. In *Proc. 17th international conference on World Wide Web.* ACM, New York, NY. 805–814.

Portmann, M. & Pirzada, A. (2008). Wireless mesh networks for public safety and crisis management applications. *IEEE Internet Computing, 12(1)*, 18–25.

Sarvas, R., Viikari, M., Pesonen, J. & Nevanlinna, H. (2004). MobShare: controlled and immediate sharing of mobile images. In *Proc. 12th Annual ACM international Conference on Multimedia.* ACM, New York, NY. 724–731.

Sheth, A. (2009). Citizen sensing, social signals, and enriching human experience. *IEEE Internet Computing, 13(4)*, 87–92.

Starbird, K. & Stamberger, J. (2010). Tweak the tweet: Leveraging microblogging proliferation with a prescriptive syntax to support citizen reporting. In *Proc. 7th International ISCRAM Conference.* Seattle.

Contact Information

Benjamin Erb, Stefan Kaufmann, Tobias Schlecht, Florian Schaub, Michael Weber
Institute of Media Informatics, Ulm University
James-Franck-Ring, 89073 Ulm, Germany
Email: <firstname>.<lastname>@uni-ulm.de