

The SHARC Framework: Utilizing Personal Dropbox Accounts to Provide a Scalable Solution to the Storage and Sharing of Community Generated Locative Media

Trien V. Do

School of Computing and Communications
Lancaster University
t.do@lancaster.ac.uk

Keith Cheverst

School of Computing and Communications
Lancaster University
k.cheverst@lancaster.ac.uk

ABSTRACT

The emergence of personal cloud storage services provides a new paradigm for storing and sharing data. In this paper we present the design of the SHARC framework and in particular focus on the utilization of personal Dropbox accounts to provide a scalable solution to the storage and sharing of community generated locative media relating to a community's Cultural Heritage. In addition to scalability issues, the utilization of personal Dropbox storage also supports 'sense of ownership' (relating to community media) which has arisen as an important requirement during our on-going 'research-in-the-wild' working with the rural village community of Wray and involving public display deployments to support the display and sharing of community photos and stories. While the framework presented here is currently being tested with a particular place-based community (Wray), it has been designed to provide a general solution that should support other place-based communities.

Author Keywords

Dropbox; cloud storage; cloud service; locative media; authoring tool; framework; community

ACM Classification Keywords

H.2.4. Systems [Distributed databases], H.3.5. Online Information Services [Data sharing]

INTRODUCTION

Recently, Community Informatics related research [16] has become an active domain in HCI. Technologies and applications deployed in projects from this domain can be trialed and evaluated 'in-the-wild' [30] over long periods of time through actual day-to-day use. The work presented in this paper is being carried out under the SHARC project (Investigating Technology Support for the Shared Curation

of Local History in a Rural Community) and represents our current focus in a longitudinal research 'in-the-wild' project that has involved the co-design, deployment, and evaluation of situated displays within a rural village community. This rural community of Wray is an example of a place-based community (i.e. a community that is tied together by a particular place, e.g. village or town). Wray village has a population of approximately 500 residents and is situated in the North of England.

Our main goal within the SHARC project is to explore the design and use of digital tools (e.g., mobile apps, web apps) that can support the curation and sharing of photos, stories, and narratives relating to the Cultural Heritage of Wray. The shared content (taking the form of locative media [14]) will be consumed by both residents and visitors. In order to achieve this main goal, our current focus is to create a framework (the SHARC framework) that allows the rapid development and deployment of the aforementioned digital tools. While the framework has been developed to facilitate our work in Wray, its general design should also support projects with other place-based communities.

In this paper, we describe the design of the SHARC framework with particular focus on the utilization of personal Dropbox (dropbox.com) accounts to provide a scalable solution to the storage and sharing of community generated locative media relating to the Cultural Heritage of a given place-based community. In addition to scalability issues, the utilization of personal Dropbox storage may also support 'sense of ownership' (relating to community media) which has arisen as an important requirement for our public display deployments supporting the display and sharing of community photos.

The remainder of this paper is structured as follows. In the next section, we summarize our past work and current motivation for supporting the community creation and sharing of locative media relating to Cultural Heritage. Next, we present background and related work. Following this, we describe the design of the SHARC framework and in particular focus on the utilization of personal Dropbox accounts to provide a scalable solution to the storage and sharing of community generated materials. The evaluation of the storage solution is then presented. Following this, we describe our implementation experiences, discuss the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

EICS'15, June 23 - 26, 2015, Duisburg, Germany

© 2015 ACM. ISBN 978-1-4503-3646-8/15/06...\$15.00.

DOI: <http://dx.doi.org/10.1145/2774225.2774841>.

solution, and provide guidelines based on our experiences with using Dropbox within the framework. Finally, we present our conclusions and future work.

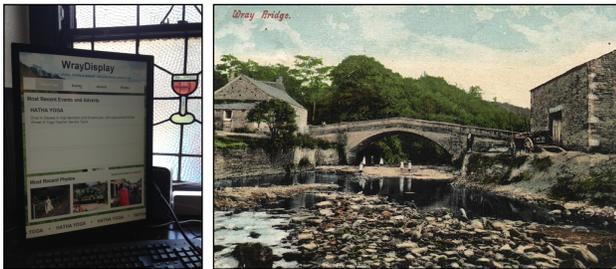
COMMUNITY SHARING OF MEDIA RELATING TO THE CULTURAL HERITAGE OF WRAY VILLAGE

Display Deployments in Wray Village

The Wray PhotoDisplay [31][33] is a community photo display system which was co-designed with the residents of Wray using a technology probe [18] based approach (see Figure 1.a) and first deployed in Wray in 2006.

As of January 2015, 2639 photos over 10 categories (such as “Wray Flood”, “Old Photos”, etc.) have been uploaded to the PhotoDisplay. Photos within the “Old photos” category particularly relate to the local history of the village. Approximately, 80% of these photos are related to different historic locations in Wray (e.g., Wray Bridge which was partly destroyed by a flood in 1967, a key event within the history of the village). Figure 1.b is a photo of Wray Bridge before the flood submitted to the “Old photos” category.

The actual PhotoDisplay application is written in Java and its GUI is designed for touch-screen displays with a size of approximately 20 inches. Currently, all content is stored on a virtual server maintained by the research team’s institution. The research team has responsibility for data backup and recovery and data security.



(a)

(b)

Figure 1. a) A typical Wray PhotoDisplay deployment in the village pub, b) A historic photo of Wray Bridge uploaded to the Wray PhotoDisplay by a resident of the village.

One of the key decisions from an early co-design workshop was to allow residents in the village to have control (e.g. moderation) over photo content and content categories. This would also help facilitate residents to have a sense of ownership regarding the PhotoDisplay system.

Feedback from users of the system has been received through e-mail, three co-design workshops, a focus group, and via a comments book. Feedback has, in general, been very positive and the following comment highlights the pride felt by many residents for the village and the role that sharing historic content can play in supporting this:

“What a superb idea, especially for those who are new to the village and a delight for those who were born here and want to go down memory lane. It’s a tribute

duly deserved for the wonderful village of Wray and its inhabitants”.

Current Motivating Scenario

In order to inform the design of the SHARC framework, we consider the following motivating scenario (and associated personas [5]). This scenario has arisen from our longstanding work with the Wray village community. It should also be noted that the persona of Sarah was informed by an interview and guided-walk through the village with a local historian who has researched the history of Wray extensively (this involved both authors and took place in July 2013).

Sarah is a local historian and recently retired university lecturer who has a depth of knowledge about the village’s history. She maintains an archive of photos (in both digital and non-digital forms) and stories relating to the Cultural Heritage of Wray. Her digital photo archive is backed up across her Dropbox and Google Drive storage. Sarah occasionally takes visitors (such as local history groups) for guided walks around the village. She tends to use a specific route which takes 30 minutes and includes Wray’s key historic points of interests (POIs). She enjoys engaging with her audience during the walks but cannot always spare her time as much as she would like. Sarah uses an authoring tool (part of the SHARC framework) to design and publish a locative media *experience* for Wray that can be consumed by visitors using their smartphones or tablets at their leisure. This experience leads visitors through her 30 minute walk and pushes the appropriate locative media (e.g., photos, audio narrative, videos, textual description, etc.) as points of interest (e.g. Wray Bridge) are approached. The tool also sends Sarah notification emails when visitors submit media responses (e.g. an audio comment or a photo) to her locative media experience that she can either approve (and so become part of her locative media experience) or reject.

Mary is in her early thirties and recently moved to Wray from London. She has always been interested in history and regrets not having had the opportunities to attend university and study history. As Mary settles in Wray, she would like to learn more about the history of the village. In the village post office she sees an advert for a guided walk led by a local historian called Sarah. Unfortunately, she is unable to attend on the advertised day but notices that the advert includes a website where she can download the guided walk onto her tablet. Mary goes to the website and reads an overview of the walk which looks interesting and so she follows instructions to install a mobile app with the guided walk onto her tablet device. The next morning she starts the tour accompanied by a friend. When they reach the first POI (Wray Bridge), Mary’s tablet starts playing an audio narrative and presents various photos of the bridge (e.g. Figure 1.b). Mary observes the bridge and notices its different texture layers. She discusses these with her friend and they decide to submit a response in the form of a new

photo highlighting the layers and a comment. The mobile app informs Mary that her response has been created successfully and will be uploaded when internet connectivity is available (the village of Wray has very poor 3G coverage). When Mary gets home after the walk, she uploads her response and is informed that the response is awaiting moderation by the creator of the experience. Meanwhile, Sarah receives an email detailing the submitted response and requesting approval for it, which Sarah is happy to provide.

John is a keen cyclist in his early forties who enjoys spending his weekends cycle touring through the counties of Yorkshire and Lancashire. He particularly enjoys breaking his journeys with good meals at local pubs. One day he cycles through Wray and decides to stop by the village pub for lunch. He notices a public display in the pub showing what appear to be historic photos of the village. He starts interacting with the photos and is particularly interested in the ones relating to a flood event which occurred in 1967. While browsing the photos and reading some of the stories associated with the flood he notices a feature that enables him to download a guided walk relating to this flood event. He explores some of the details of the walk using the display (e.g. whether he will be able to cycle the route) and decides to give it a go. Following the instructions given, he connects his smartphone with the free Wi-Fi in the pub and downloads the guided tour. He is given the option to have the locative media such as audio narratives play automatically on his smartphone or to receive alerts when they are available. John selects the latter and sets off. When John approaches the first POI en route, which is Wray Bridge, his phone beeps. He stops and looks at his phone to see options for playing various locative media associated with Wray Bridge and the flood of 1967. He selects to play the audio narrative and looks at associated photos of the bridge (e.g. Figure 1.b) while it is playing. He also sees the response submitted by Mary.

Following on from the motivating scenario and associated personas we can identify two broad categories of user, i.e. *designers* and *consumers*. Sarah is a designer of her locative media experience while both Mary and John are consumers of the locative media. However, Mary is a consumer who also responds (to Sarah's locative media experience) with her own media.

BACKGROUND AND RELATED WORK

Two areas of background and related work are applicable to the work presented in this paper, namely: locative media authoring tools and Community Informatics related systems. This section summarizes primary functionality and approaches to data storage (e.g., local storage, central server storage, cloud storage, personal cloud storage) of these tools and systems.

Locative Media Authoring Tools

Locative media authoring tools enable non-programmers to create locative media experiences. The MediaScape [17]

framework is perhaps the most widely used system for supporting the development and playing of locative media experiences. It was released in 2004 and comprises three components: Mscape Maker, Mscape Tester, and Mscape Player. The Mscape Maker is a desktop authoring application that allows designers to design locative media experiences by enabling the author to draw geographic trigger zones and associate media with these trigger zones. Created locative media experiences are stored locally on the computer running Mscape Maker and can then be tested using the Mscape Tester desktop application which includes a GPS simulator. Finally, locative media experiences are transferred via a cable to the PDA running the Mscape Player application (i.e. network connectivity was not required during execution of a locative media experience). The Mscape Player was implemented for PDAs with Windows Mobile 2003 only.

More recent tools include the IVO [28] and 7Scene (7scenes.com) frameworks. These provide: (1) Web-based authoring tools for designers to create context-aware [6] applications, and (2) iOS and Android mobile "player" apps for downloading applications created by the authoring tools from servers to operate as locative media apps.

Community Informatics Related Systems

In recent years, Community Informatics related research [16] has provided several examples of innovative applications that utilize community generated content. One example is StoryBank [13] which supported the creation and sharing of audiovisual stories in an Indian village. Users could create stories in a simple manner using their camera phone and these stories could then be transferred to a digital library which was attached to a village display. StoryBank stored data locally and did not require any form of network connection.

Deployed in the urban setting of Oulu in Finland, CLIO [29] enabled communities within the city to capture and share memories via text, images, audios, and videos. CLIO stored text in a relational database, photo and audio files in a folder on a central server, and links of videos hosted on social media websites. Users explored shared memories on public displays and via mobile apps.

CrowdMemo [2] was designed to support the preservation of local Cultural Heritage in a town in rural Argentina. Using digital cameras, the community created videos about personal memories then associated them with locations in the town either in-situ via QR codes or virtually on Google Maps embedded on the project website. The project was setup in such a way that it could be continued by the community after researchers had left. This sustainability was achieved by requiring the community to upload videos to YouTube first and then associate links of these videos with QR codes or markers on Google Maps.

FieldTrip GB [3] is a mobile mapping and data collection app. It allows users to collect data against high quality

cartographic maps. These data can then be uploaded to users' personal Dropbox accounts. Later, users can access their collected data from a web application for further analysis. A community (e.g., a group of biology students) can collect and share data (e.g., samples of plants) by using the same Dropbox account.

THE SHARC FRAMEWORK

In this section, we first describe the data storage requirements of the SHARC framework, next we introduce its architecture and then focus on the utilization of personal Dropbox accounts to provide a scalable solution to the storage and sharing of community generated locative media. The motivating scenario in Wray is used as a running example to illustrate how the framework works.

Data Storage Requirements

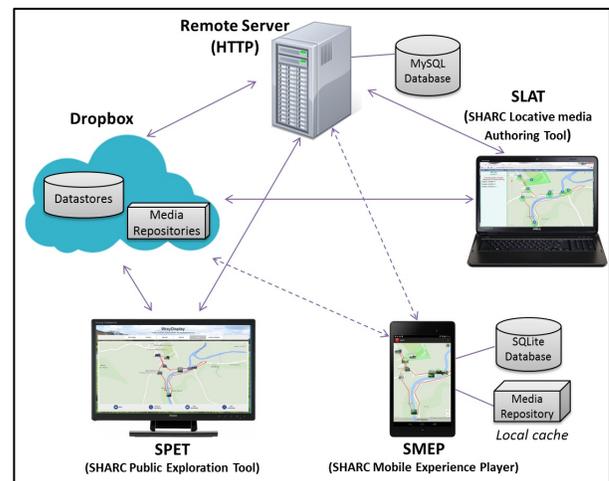
The motivating scenario in Wray reveals the need for a flexible and scalable approach to the storage of locative media experiences and associated media responses. The framework should also be capable of supporting multiple place-based communities. Furthermore, the framework should provide individual users with a strong 'sense of ownership' regarding the content that they have submitted. On one occasion it was necessary to provide our 'Champion' (or human access-point [23]) within the Wray community with a back-up tool for transferring all submitted content to her PC. This was required because, at the time, the research project funding the support of the PhotoDisplay deployments had come to an end and no follow-on funding had been obtained (at the time). This and other issues related to the 'sustainability' of deployments encountered when conducting our 'research-in-the-wild' with communities are described in [32]. Furthermore, as noted in [32] the architecture of the Wray PhotoDisplay system was such that displays relied on a university web server to host the display's content and website.

Today, personal cloud storage services (e.g., Dropbox, Google Drive) have gained widespread recognition and adoption as a new way of storing and sharing data. Most of these services offer each registered user a certain volume of storage space without any fee (e.g., 2GB on Dropbox, 15GB on Google Drive) and users can choose to pay a monthly fee to have more storage space. These services also provide APIs for programmers to write applications which can store and retrieve data on the end-users' cloud storage accounts. We have investigated the potential of personal cloud storage services (e.g., Dropbox) to store data for the SHARC framework. The approach means that a research team can design and develop a scalable and reliable storage solution for the SHARC framework without having to financially invest in data storage space and associated maintenance. Therefore the framework can support a large number of communities even after the funding of the given research project comes to an end. Using personal cloud storage also shifts the responsibility of data administration (e.g., data security, backup and recovery) from the research

team to cloud service providers. Cloud storage also helps protect the research team from certain legal liability issues associated with the content of data (e.g., copyright). From the users' perspective, the utilization of personal cloud storage also supports 'sense of ownership' because data are stored in folders in their own cloud accounts. Among a number of popular cloud storage services, Dropbox has been chosen for the framework because of its widespread popularity [15].

Architecture

The SHARC framework adds significant new features (e.g., authoring functionality) to the LoMAK framework [7] which was specifically designed to enable non-programmers to generate locative media experiences from KML files. The overall architecture of the SHARC framework is illustrated in Figure 2. It comprises three primary components: SLAT, SPET, and SMEP.



Key
 ↔ Constant internet connectivity required
 ⇄ Internet connectivity required to download and upload data

Figure 2. The overall architecture of the SHARC framework.

SLAT (SHARC Locative media Authoring Tool)

SLAT is a web-based authoring tool which allows designers (e.g., Sarah in the motivating scenario) to create locative media experiences.

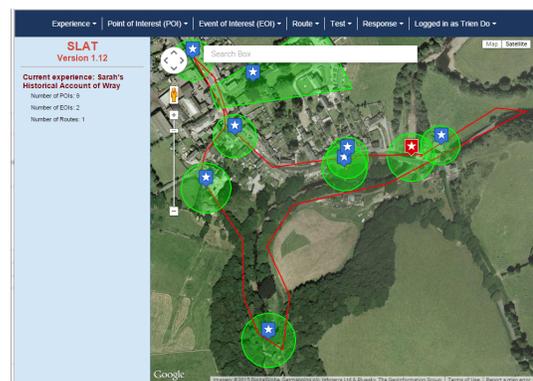


Figure 3. Sarah's locative media experience designed using the SLAT authoring tool.

Each experience can contain a number of entities (e.g., POIs, Events of Interest or EOIs and routes) and locative media can be associated with each entity. For example, Sarah can create an experience for Wray comprising one route, two EOIs, and nine POIs. One of these POIs is Wray Bridge (e.g., the red marker in Figure 3) with photos and an audio narrative. Figure 3 is an example of a locative media experience designed for Wray. SLAT also lets designers moderate responses for POIs submitted by visitors.

SPET (SHARC Public Exploration Tool)

SPET is another web application which allows visitors (e.g., Mary, John) to browse and explore available locative media experiences. Visitors can either plan their trips in advance and access SPET at home or interact with SPET on a public display. In the Wray scenario, John interacts with the Java PhotoDisplay application (that has the SPET component embedded within it) in order to select, view and then download (to his smartphone) Sarah's locative media experience (see Figure 4).



Figure 4. Sarah's locative media experience viewed on a Wray public display.

SMEP (SHARC Mobile Experience Player)

SMEP is a mobile app which allows visitors to explore, load and then 'play' locative media experiences created by SLAT. One key requirement for SMEP (informed by the poor 3G coverage in Wray) is the support for off-line operation. Consequently, SMEP has been designed to cache data on local storage so it can work offline.



Figure 5. The SMEP Mobile Experience Player: a) A locative media experience for Wray in SMEP, b) Photos associated with Wray Bridge, c) Responses.

SMEP detects the visitor's current location in order to push

appropriate media to them when a given trigger zone is entered. The visitor can also add responses to POIs. For example, when Mary approaches Wray Bridge, photos of the bridge are pushed to her (see Figure 5.b). Mary is interested in different texture layers of the bridge and submits a response with a new photo and an associated comment. When Mary gets home, she uploads her response (see Figure 5.c).

Utilization Of Personal Dropbox Accounts

The SHARC framework utilizes personal Dropbox accounts to provide a scalable solution for storage and sharing of community generated locative media. The key idea of the solution is that if designers or consumers generate any data (i.e., Sarah creates an experience and Mary adds a response), these data are stored on their Dropbox accounts (see Figure 2). Then the framework makes these data accessible to other Dropbox accounts (e.g., the experience for Wray designed by Sarah is accessible to Mary, John and any other consumers). Consequently, a designer is required to have a Dropbox account while a consumer is only required to have one if s/he wants to submit response(s).

The SHARC framework stores details regarding Dropbox accounts (e.g. account id) and experiences (e.g. designer id) on a MySQL database on the remote server managed by our university (see Figure 2). The framework interacts with the Dropbox personal cloud service through Dropbox APIs. These APIs are supported in different programming languages such as Java, JavaScript, PHP, Android, and HTTP. The Dropbox datastore APIs were first released in July 2013. However the shared datastore feature had not been introduced until September 2014 [9]. This feature enables our cloud storage solution. One point to notice is that Dropbox datastores are NoSQL databases [27] so they provide weak atomicity, isolation, and durability (ACID) guarantees compared to traditional relational databases (e.g., MySQL, SQL Server, Oracle).

To enable the SHARC framework to store data on Dropbox, the research team had to create a Dropbox app in Dropbox's App Console [10]. As part of the process, appropriate permissions for the app were selected. The Dropbox app for the SHARC framework uses core APIs, datastore APIs, and limits its access to only files it creates. This app is assigned a pair of keys (*App key* and *App secret*) by Dropbox. SLAT, SMEP, and SPET use these keys to access the Dropbox app of the SHARC framework.

Dropbox Storage Solution With SLAT

Assume that Sarah is at home (and has good internet connectivity). In order to create an experience, Sarah needs to sign into SLAT with her own Dropbox account details. This sign in process goes through the following steps. First, SLAT directs Sarah to the login page of Dropbox with the *App key* and *App secret* parameters. If Sarah logs into Dropbox successfully, then Dropbox will ask if Sarah grants SLAT permissions (see Figure 6) to access her Dropbox account. SLAT needs these permissions to create

and manage datastores and to work with files in a folder created by the Dropbox app of the SHARC framework (e.g., the “Media Repository” folder in Figure 7). If Sarah grants these permissions, Dropbox redirects Sarah back to SLAT and provides SLAT with an *access token*. SLAT uses this token together with the *App key* and *App secret* to work with the Dropbox account of Sarah.

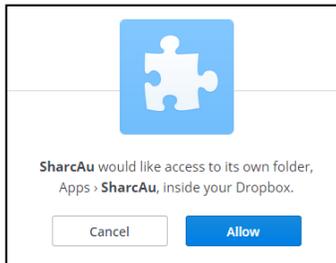


Figure 6. SLAT asking designers for permissions to access their Dropbox account.

When the login process finishes, details about Sarah’s Dropbox account (e.g., Dropbox account ID, name, and email address registered to Dropbox) are stored in the “Users” table of the MySQL database on the remote server (see Figure 7). This information is primarily used to notify Sarah via email when responses to her experience are submitted.

After selecting the “create new experience” option from the appropriate SLAT menu, Sarah names her experience ‘Sarah’s Historical Account of Wray’. This action performs two tasks relating to her Dropbox account. First, a new datastore (identified by a unique ID) is created in Sarah’s Dropbox account (e.g., the datastore inside “Account 3” in Figure 7) to store content relating to the experience (e.g., POIs, routes, etc.). However, by default this datastore is private and can only be accessed by Sarah’s Dropbox account details. To enable other Dropbox accounts (e.g., Mary’s) to access Sarah’s experience and add responses, this datastore must be shared with the *editor* permission. SLAT performs this task by calling a Dropbox API (i.e. using the *datastores/put_delta* endpoint) to insert a new record to the access control table of this datastore. Each record in the access control table defines who (e.g., *public*, *team*, *private*) can access the datastore and what permissions (e.g., *viewer*, *editor*) they are granted. Once this datastore is shared, any other Dropbox accounts can access it with the appropriate datastore ID.

Secondly, details about the experience (e.g., the ID of the associated Dropbox datastore) are stored in the “Experiences” table of the MySQL database (see Figure 7). This table maintains a list of experiences and where these experiences are stored in Dropbox accounts. The table also allows SPET and SMEP to retrieve these experiences and present them on a Google Maps canvas for consumers to explore.

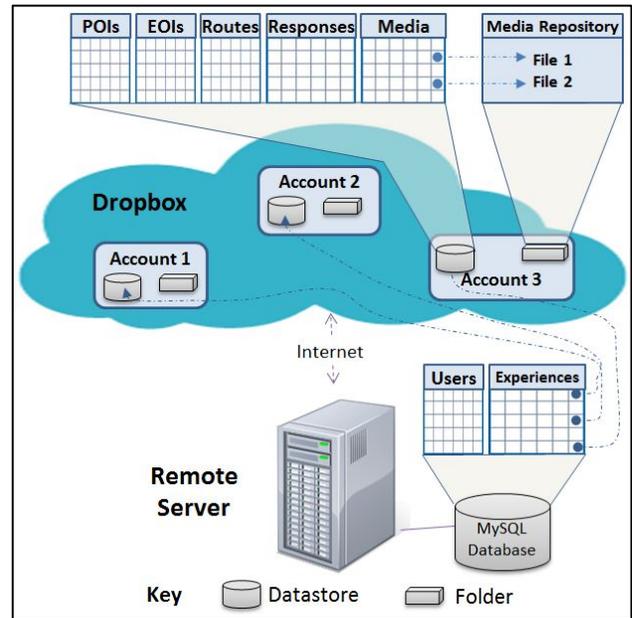


Figure 7. Data stored on Dropbox and in a MySQL database.

The details relating to Dropbox accounts and experiences could also be stored in another datastore on Dropbox. However, to access the contents of a datastore, Dropbox requires its users to sign in with their account (which is performed in a secure fashion via HTTPS). From an interaction design point of view, we did not want the SPET and SMEP components of the framework to require consumers to sign in just to explore available locative media experiences. Consequently, we have currently chosen to have the aforementioned details relating to Dropbox accounts and experiences stored on a MySQL database because this approach means that consumers are only required to sign in if they want to submit responses.

Next, Sarah can start adding POIs to her experience and associating media items to these POIs. For example, Sarah creates a POI for Wray Bridge and then associates with it the photo of Wray Bridge before the flood (see Figure 1.b). Information about the Wray Bridge POI (e.g., name, location) is stored in the “POIs” table of the datastore relating to her experience and the image file itself is uploaded to a folder on Sarah’s Dropbox account (see Figure 7). Similar to a datastore, by default, this image file is private. SLAT needs to invoke another Dropbox API method (i.e. */shares*) in order to share this file and retrieve a public URL of the file. Unlike a datastore, a shared file can be read by any users even if they do not have a Dropbox account. The information about the image file (e.g., name, public URL) is stored in another table of the datastore (e.g., the “Media” table in Figure 7). SLAT resizes all images to the maximum size of 900 x 900 pixels and compresses them (using the *toDataURL* method of HTML canvas [34]) before uploading them to Dropbox in order to minimize the storage burden on personal Dropbox accounts.

Note that the experience designed by Sarah is not visible to

consumers. Once Sarah finishes the design, she can publish it. This action makes the experience visible to consumers. Content in the datastore of the experience is then exported to a JSON file in her Dropbox account. We term this JSON file a snapshot of an experience. SMEP and SPET can load snapshots without requiring consumers to sign in with their Dropbox account. Designers can create new snapshots after they edit experiences (e.g., adding new POIs, approving responses).

To summarize, when a designer creates a locative media experience, SLAT creates a datastore to store contents such as names of POIs, EOIs, etc. and a media folder to store the actual media files, such as image files, in the Dropbox account of the designer. Both datastores and media files are then shared with other Dropbox accounts. Currently, a MySQL database on a remote server is also used to store details regarding the designers' Dropbox accounts and locative media experiences. This database enables consumers to explore and download experiences without having to sign into SMEP and SPET with their Dropbox accounts. Note that in order to be fully independent from the University, a community such as Wray can elect to rent a web hosting service to host SLAT. Furthermore, data currently stored in the MySQL database on the remote server can be moved to a flat file in Dropbox.

Dropbox Storage Solution With SPET

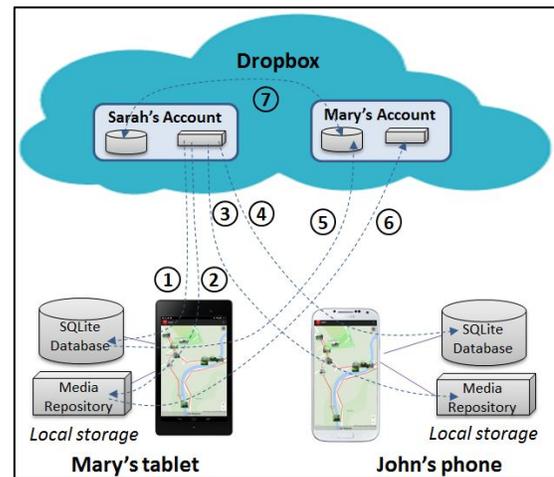
Relating to the Dropbox storage solution, the SPET exploration tool needs to connect to the remote server (see Figure 2) in order to retrieve available locative media experiences from the MySQL database. These experiences can then be presented as markers on a Google maps canvas for consumers to explore. When an experience is selected, SPET loads the snapshot of this experience from Dropbox. With the Wray scenario, the SPET component is embedded within the PhotoDisplay application and presents the 'Sarah's Historical Account of Wray' experience when John interacts with the public display in the village pub (see Figure 4). In the more general case, consumers can utilize SPET in order to browse available experiences on a map.

Dropbox Storage Solution With SMEP

Similar to SPET, the SMEP mobile experience player first connects to the remote server (see Figure 2) in order to load into local storage a given locative media experience (this is achieved by SMEP loading the snapshot of the chosen experience from the designer's Dropbox account).

When consumers add responses (e.g., Mary's question about different texture layers of Wray Bridge) to POIs, these responses are initially stored locally on the consumer's smartphone or tablet. Later, these responses are uploaded to Dropbox when internet connectivity is available (see Figure 5.c). Note that, details of the responses (e.g., the associated POI) are added to the datastores of experiences but the actual media files (e.g., the photo of Wray Bridge taken by Mary) are stored in the Dropbox account of the consumer who submitted the responses.

To summarize, with SMEP, consumers actually access experiences from the Dropbox accounts of designers (and other consumers for media files relating to responses). In the Wray scenario, Mary and John download the locative media experience stored in Sarah's Dropbox account to their tablet and smartphone respectively; media files in Mary's response are stored in Mary's Dropbox account while content of the response is stored in the datastore created by Sarah (see Figure 8). John can also view media files of Mary's response.



- ①④ Downloading a snapshot of an experience ⑦ Synchronizing datastores
 ②③ Downloading media files ⑤⑥ Uploading a response's content and media files

Figure 8. Data flows among devices and Dropbox.

TESTING THE FEASIBILITY OF THE CLOUD STORAGE SOLUTION WITHIN THE FRAMEWORK

The feasibility of the Dropbox storage solution has been tested in the SHARC framework. In more detail, using the local historian's guided walk around Wray as a motivating scenario, we designed an experience using SLAT. The experience has nine POIs, two POIs, and one route (see Figure 3). In total, 68 photos and four audio narratives are associated with these entities. The experience consumes less than 50MB on the Dropbox account of a member of the research team.

To estimate the scalability of the Dropbox storage solution, consider the storage space required for 1000 locative media experiences designed by 1000 designers. As described in the previous section, the MySQL database on the remote server (see Figure 2) is only used to store details about Dropbox accounts of designers and locative media experiences. The database needs less than 200 bytes to store details about a Dropbox account (id, name, email) and less than 800 bytes for experience (id, name, account id, description, location). Therefore, to store details about 1000 Dropbox accounts and experiences, the database will require no more than 1MB (1000 x (800 bytes + 200 bytes)) of storage on the remote server.

All datastores and media files of locative media experiences are distributed across the Dropbox accounts of designers

and consumers. The utilized image compression and resizing algorithm used by SLAT produce images of less than 1MB. On average, one minute audio (256kbps mp3) needs less than 2MB [1], and one minute of video captured by a Nexus 7 needs less than 70MB. Therefore, a one hour locative media experience with 200 images, 50 minutes of audio, and 10 minutes of video will occupy less than 1GB in a designer's Dropbox account. Currently, each Dropbox user is allocated 2GB of storage without fee. To create more locative media experiences, a designer can either register a new Dropbox account or purchase more storage.

In summary, the approach of utilizing personal Dropbox accounts to store data enables the SHARC framework to support a potentially large number of place-based communities in curating and sharing locative media.

EXPERIENCES, DISCUSSION AND GUIDELINES

This section describes our experiences implementing the Dropbox storage solution, discusses the solution, and provides guidelines for the EICS community.

Implementation Experiences

Many of the challenges we encountered when implementing the cloud storage solution with Dropbox were associated with SLAT. The first challenge concerned authentication. OAuth is an open standard which provides a method for clients to access server resources on behalf of a resource owner [19]. Dropbox supports both versions of OAuth (OAuth 1.0 [19] and OAuth 2.0 [20]). OAuth 2.0 requires an HTTPS server. Consequently, SLAT has been implemented with OAuth 1.0 because its hosting server (the remote server in Figure 2) is an HTTP server. Note that HTTP is typically offered when renting a remote server.

When using the Dropbox APIs we have found that, on occasion, documentation is unclear. For example, it took considerable development time to work out how to generate a valid ID for a datastore or how to share a datastore. On these occasions, we emailed Dropbox's technical support team to ask for further information.

Cloud Storage Discussion

This paper focuses on the utilization of personal Dropbox accounts to provide a scalable solution to the storage and sharing of community generated locative media. This solution can also be applied with other personal cloud storage services as long as they support both datastore and file sharing (e.g., Dropbox and Google Drive). With personal cloud storage services that only support file sharing such as SugarSync (sugarsync.com) our solution can still be utilized by storing the content relating to experiences in flat files instead of datastores.

We plan to extend our cloud storage solution to include a component to abstract over the specifics of a particular cloud storage solution. Returning to the Wray scenario, this would allow the framework to support Sarah in utilizing both her Dropbox and Google Drive accounts.

Our utilization of personal Dropbox accounts is different from that of Fieldtrip GB [3]. The approach adopted by Fieldtrip GB is to have all members of the community (working towards the same data collection goal) share the same group Dropbox account. Consequently, they have the same set of permissions on data in that Dropbox account. In addition, people outside the community cannot access the community's data. With the SHARC framework, each designer/consumer uses their own personal Dropbox account and has different permissions on shared data. For example, only designers (e.g., Sarah) can modify their locative media experiences. Responses need their approvals to be part of their locative media experiences. Any consumers (even without a Dropbox account) can explore and consume shared experiences but they cannot edit them.

While, the utilization of personal cloud storage offers many opportunities there are also pitfalls. As discussed in P-LAYERS [24], from the developers' perspective, there are a number of inter-related challenges when working with third party services. For example, Dropbox has the right to change its policies and services at anytime [21]. Currently, Dropbox lets developers write applications on top of their platform without fee and doesn't limit the number of users accessing these applications [11]. Again, Dropbox could change these policies at any point. Developers might then be required to either pay to continue using the Dropbox platform or redesign and change their application architecture. Similarly, currently, Dropbox provides a new user with their first 2GB storage for free and users only have to pay to have additional storage space. In the future, they may charge money for this first 2GB storage [21]. In that case, developers would lose existing users who are not willing to pay for Dropbox storage service. Another issue is that Dropbox can change their API interfaces with little or no notice. This would suddenly break applications and require debugging and bug fixing from developers. Last but not least, as users can directly delete and edit files on their Dropbox accounts, developers need to be more cautious when writing applications which read/write data to files.

From the end-users' perspective, the use of their cloud storage account by a framework such as SHARC may cause some anxiety about security and safety [21]. For example, end-users may hesitate when using their Dropbox username and password to sign into other applications because they do not know which information and permissions they are providing to these applications. The sign-in process of these applications also takes more steps and so requires additional effort/time on behalf of end-users. For example, to sign into SLAT with her Dropbox account, Sarah first needs to enter her username and password and then she has to confirm that she permits SLAT to access her Dropbox account. Using cloud storage, developers also put end-users in charge of managing their own data storage space (i.e., users have to buy more storage space themselves) but as discussed earlier this may help foster an important 'sense of ownership'.

Guidelines

We have found that the Dropbox storage solution offers a good fit for the SHARC framework. Based on our implementation experiences, we provide the following guidelines for developers who may be considering using Dropbox to store data for their community applications.

Assess The Suitability Of Using Dropbox In Projects With Strong Data Integrity Requirements

The personal Dropbox storage solution is suitable with applications in which end-users want to create and share data with others. It is especially a promising candidate for research projects because it can be problematical to maintain a large volume of storage space when these projects end and in this respect the solution supports the sustainability of research-in-the-wild deployments.

Datastores in Dropbox are not relational databases. Advanced features such as foreign keys, stored procedures, triggers, and transaction management are not provided. Developers have to manage data integrity by writing code in applications. So compared to relational databases (e.g., MySQL) more work is required from developers.

Consider Data Security And Privacy Policies

Data security and privacy are the most often-cited barrier to the implementation of cloud storage [35]. These data can represent personally identifiable information, sensitive information, etc. [26]. Developers should consider what type of data needs to be stored. For example, applications in the health domain particularly need to protect user's data privacy [21][22]. The cloud storage solution is suitable for the SHARC framework because all data and files are submitted in order to be shared with others.

Check Dropbox's News Regularly

Currently, there is no universal standard for cloud APIs. Dropbox warns developers that APIs' interface can be changed in the future. Potential changes from Dropbox need to be tracked so applications can be updated quickly.

Pay Attention To Datastore Size Limits

Dropbox has limits on the maximum size for each record, maximum number of records per datastore, maximum size for a datastore, and maximum size of a single transaction [12]. Such limits need to be kept in mind when deciding to use Dropbox cloud storage. With the SHARC framework, we encountered problems with the size limit for a record when storing a route in SLAT. The particular route contained more than 1000 GPS coordinates captured by a smartphone and was too large to be stored in a record. We solved this issue by simplifying the complexity of routes using the Douglas-Peucker algorithm [8].

CONCLUSION AND FUTURE WORK

In this paper we have presented the design of the SHARC framework and in particular focused on the utilization of personal Dropbox accounts to provide a scalable solution to the storage and sharing of community generated locative media relating to a community's Cultural Heritage. This

solution enables the framework to support not only the village community targeted by the research project but also other place-based communities. In our experience, one of the key factors that increases the likelihood of a Community Informatics research project being successful is to encourage the community to feel a sense of ownership of both the deployed systems and generated content [32]. In addition to scalability issues, we anticipate that the utilization of personal Dropbox storage will also support sense of ownership. This is one of the key design decisions motivating our use of Dropbox but the extent to which this sense of ownership is felt by the community will require longer term deployment-based evaluation in order to gain ecologically valid insights [4]. Consequently, we are currently deploying the SHARC framework in Wray to gauge the community's opinion regarding the use of personal Dropbox accounts and whether the solution generally promotes a feeling of ownership regarding content or whether users feel that significant ownership is instead given to Dropbox. We anticipate that the deployment will require support from the research team's 'champion' in the village. This was certainly the case with our earlier deployments where the champion was available to assist members of the community with uploading photos. It is important to note that we anticipate that a relatively small number of Wray's residents will actually create new locative media experiences but we expect that a larger number will contribute content to these experiences once created (thus reflecting participation in the PhotoDisplay system and the so-called "90-9-1 principle" [25]).

The framework currently has the flexibility to exploit the fact that users can choose to increase their Dropbox storage dynamically. However, looking ahead, we plan to increase the flexibility of the framework to support multiple cloud storage vendors and to enable users to aggregate content across their multiple cloud storage accounts.

We are also planning to test the suitability of the framework by working with another rural community based in the Lake District where again internet connectivity is poor and there is a rich set of cultural heritage materials to be shared.

REFERENCES

1. AudioMountain. Audio file size calculations. <http://www.audiomountain.com/tech/audio-file-size.html> (Accessed May 2015).
2. Balestrini, M., Bird, J., Marshall, P., Zaro, A., and Rogers, Y. Understanding sustained community engagement: a case study in heritage preservation in rural Argentina. *Proc. CHI 2014*, ACM Press (2014).
3. Butchart, B., Pope, A., King, M., Hamilton, G., Terzis, P., and Koutroumpas, M. Fieldtrip GB: Creating a customisable mapping and data capture app for the HEFE community. *GISUK 2013*, (2013).
4. Carter, S., Mankoff, J., Klemmer, S., and Matthews, T. Exiting the cleanroom: On ecological validity and

- ubiquitous computing. *Human-Computer Interaction* 23, (2008), 47–99.
5. Cooper, A. and Saffo, P. *The inmates are running the asylum*. Macmillan Publishing Co., Inc., 1999.
 6. Dey, A.K., Salber, D., Abowd, G.D., and Futakawa, M. The conference assistant: Combining context-awareness with wearable computing. *Proc. ISWC 1999*, IEEE Computer Society (1999).
 7. Do, T. V., Cheverst, K., and Gregory, I. LoMAK: A framework for generating Locative Media Apps from KML files. *Proc. EICS 2014*, (2014), 211–216.
 8. Douglas, D.H. and Peucker, T.K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica* 10, 2 (1973), 112–122.
 9. Dropbox. Developer blog. <https://www.dropbox.com/developers/blog> (Accessed May 2015).
 10. Dropbox. App Console. <https://www.dropbox.com/developers/apps> (Accessed May 2015).
 11. Dropbox. Dropbox Platform developer guide. <https://www.dropbox.com/developers/reference/devguide> (Accessed May 2015).
 12. Dropbox. Datastore HTTP API Documentation. <https://www.dropbox.com/developers/datastore/docs/http> (Accessed in May 2015).
 13. Frohlich, D.M., Rachovides, D., Riga, K., et al. StoryBank: mobile digital storytelling in a development context. *Proc. CHI 2009*, ACM Press (2009).
 14. Galloway, A. and Ward, M. Locative media as socialising and spatializing practice: Learning from archaeology. *Leonardo electronic almanac* 14, 3 (2006).
 15. Goncalves, G., Drago, I., Silva, A.P.C. da, Almeida, J.M., and Vieira, A.B. Characterizing and modeling the Dropbox workload. *Proc. SBRC*, IEEE (2014).
 16. Gurstein, M. Community informatics, community networks and strategies for flexible networking. In *Community Informatics: Shaping Computer-Mediated Social Networks*. Routledge, 2001, 263–283.
 17. Hull, R., Ben, C., and Tom, M. Rapid authoring of Mediascapes. *Proc. UbiComp 2004*, Springer Berlin Heidelberg (2004).
 18. Hutchinson, H., Hansen, H., Roussel, N., et al. Technology probes: inspiring design for and with families. *Proc. CHI 2003*, ACM Press (2003).
 19. IETF. The OAuth 1.0 Protocol. <http://tools.ietf.org/html/rfc5849> (Accessed May 2015).
 20. IETF. The OAuth 2.0 authorization framework. <http://tools.ietf.org/html/rfc6749#section-1.3> (Accessed May 2015).
 21. Ion, I., Sachdeva, N., Kumaraguru, P., and Čapkun, S. Home is safer than the cloud! Privacy concerns for consumer cloud storage. *Proc. SOUPS 2011*, ACM Press (2011).
 22. Löhr, H., Sadeghi, A.-R., and Winandy, M. Securing the e-health cloud. *Proc. IHI 2010*, ACM Press (2010).
 23. Marsden, G., Maunder, A., and Parker, M. People are people, but technology is not technology. *Philosophical Transactions of the Royal Society A* 366, 1881 (2008), 3795–3804.
 24. Memarovic, N., Langheinrich, M., Cheverst, K., Taylor, N., and Alt, F. P-LAYERS - A layered framework addressing the multifaceted issues facing community-supporting public display deployments. *ACM ToCHI* 20, 3 (2013), 1–34.
 25. Nielsen, J. The 90-9-1 rule for participation inequality in social media and online communities. 2006. <http://www.nngroup.com/articles/participation-inequality/> (Accessed in May 2015).
 26. Pearson, S. Taking account of privacy when designing cloud computing services. *Proc. CLOUD 2009. ICSE Workshop*, IEEE (2009), 44–52.
 27. Pokorny, J. NoSQL databases: a step to database scalability in web environment. *Proc. iiWAS 2011*, ACM Press (2011), 278–283.
 28. Realinho, V., Dias, A.E., and Romão, T. Testing the usability of a platform for rapid development of mobile context-aware applications. *Proc. Interact 2011*, Springer Berlin Heidelberg (2011).
 29. Ringas, D. and Christopoulou, E. Collective city memory: field experience on the effect of urban computing on community. *Proc. C&T 2013*, ACM Press (2013), 157–165.
 30. Rogers, Y. Interaction design gone wild: striving for wild theory. *Interactions* 18, 4 (2011), 58–62.
 31. Taylor, N., Cheverst, K., Fitton, D., Race, N.J.P., Rouncefield, M., and Graham, C. Probing communities: Study of a village photo display. *Proc. OZCHI 2007*, ACM Press (2007), 17–24.
 32. Taylor, N., Cheverst, K., Wright, P., and Olivier, P. Leaving the wild: lessons from community technology handovers. *Proc. CHI 2013*, ACM Press (2013), 1549–1558.
 33. Taylor, N. and Cheverst, K. Social interaction around a rural community photo display. *International Journal of Human-Computer Studies* 67, 12 (2009), 1037–1047.
 34. W3Schools. HTML 5 Canvas. http://www.w3schools.com/html/html5_canvas.asp (Accessed May 2015).
 35. Zetta. Cloud storage adoption trends, barriers and expectations - a Survey. http://pages.zetta.net/rs/zetta/images/Zetta_Cloud_Storage_Survey.pdf (Accessed May 2015).

