Project Proposal

Co-located mobile media experiences through localized cloud

Mutakha, Jarvis
MTKJAR001@myuct.ac.za

Mahlaza, Zola
zmahlaza@cs.uct.ac.za

June 17, 2014

Contents

1 Introduction

In 2001, Kraut et al[3] coined the term "strong ties". This was to refer to the things which strengthen human relationships such as frequent contact, and deep feelings of affection and obligation. This project aims to investigate the use of mobile devices and cloudlets that can also be used to strengthen human relationships. Cloud computing has been a powerful computing paradigm which has changed the way we deliver services to users. We make use of cloud services because they provide us with unique ways to collaborate, stay in touch, and engage with media socially. These services have been used to connect and make sharing easier for people across the globe. The same technologies that power cloud service providers, namely web servers, data stores, and databases, can also run at a smaller scale on embedded systems, such as the Raspberry Pi. Computer hobbyists have revealed that the Raspberry Pi could make as an attractive solution for a small lightweight server because of its solid state storage, no noise, small form factor and low power consumption[2]. This project aims to make use of a Raspberry Pi extended with a battery, WiFi radio and local storage to provide a group of co-located friends with opportunities to share and engage with media amongst themselves. These cloudlets can be used to exhange files, and document collaboration. In essence, they can be used to provide all cloud services on a lower scale. The advantage being that there is less network latency. This research is also interested in the human-computer interaction aspects, that is, which services can be offered to enhance co-located device use.

2 Significance

The significance of this project is that it will continue on the work done by Reitmaier and Benz[5]. They have shown that people believe that cloudlets can be effective, however, they have concerns over privacy and data control. This project will address those issues, and then evaluate cloudlets to see if they are fully viable for enhancing co-located mobile device use. In addition, they will test the feasibility of using embedded systems as the backbone of a cloudlet.

Currently, sharing can be done through a variety of technologies such Bluetooth, Wifi direct, etc. These technologies focus solely on sharing files on a one-to-one basis, that is, transfer of files from one device to another. The proposed system aims to offer a multitude of services, file sharing being only one of them. We aim to provide a system that can offer services such file suggestions from connected friends using the file history, sharing media files with multiple people at once, temporary sharing of data files similar to Snapchat, etc. We aim to provide a system that can support a number of services, it is important to mention that the goal of this project is not for us to develop and provide all these services but create a platform that support them.

3 Problem Statement and Research Questions

3.1 Problem statement

The cloud is generally used for data storage and service provision. Cloudlets can also be used to provide the mentioned things. The provision of a large number of cloud services in a cloudlet scenario would be beneficial, however, for this project a few number of services will be offered due to time constraints. There are numerous aspects the current cloud computing paradigm is lacking in, these limitations carry over to cloudlets. The following are the aspects which the project attempts to focus on.

- **Information Ownership, Control and Security**: Data centers for cloud computing service providers can be located in any country in the world. Users may be concerned about where

their information is stored. Users may want the ability to control where their data is stored and who has access to it. This may be due to privacy laws and government organizations in different countries.
- **Cellular networks**: Connecting to the Internet via cell phone networks is generally slower than using the Internet, especially in non-first world countries. They also introduce high charges from cell phone providers in some cases. These two factors therefore limit the use of cloud services by users of mobile devices.

3.2 Research Questions

- Can we create an effective common sandbox for data?
- What interface conceptual metaphors are effective in conveying the properties of the cloudlets as ephemeral data stores.
- Can an embedded system be practical for creating cloudlets, taking in consideration battery consumption and network range?

4 Brief Literature Review

The literature that has been studied focuses on offloading computation, co-located device use, file sharing in ad-hoc network for mobile devices, providing collaboration services for mobile devices.

The use of technology is massively affected by the infrastructure that is available to us. We have observed online media sharing become popular with the advent of services such as Dropbox, Google+, Google Drive, etc to people who have access to the Internet. Walton et al[8] have shown, however, that in places such as Khayelitsha (Cape Town, South Africa) co-located phone use surpassed online sharing. The media stored on the mobile devices became public personae, that is, played the same role as "profiles" in social media[8]. An important finding they presented was that since people already knew each other to an extent, this changed their data storing patterns. In particular, it meant that they gave each other access to files as they became their 'profiles' but they still required privacy. This means a trust model that takes into account that users know each other is important.

Ah Kun and Marsden[4] implemented an application for photo sharing in a co-located scenario. Their goal was to study the interaction between the users. This application allowed simultaneous access to the same file. They explored using a number of access control mechanisms, the interesting ones being:
- **Ad-hoc**: This is a policy whereby any user can control a resource at any time. Here, there are no software locks. A social protocol is expected to arise from the users[4].
- **three-seconds**: This is a policy whereby control is passed around among the users. A user acquires control and performs an action. When there is no action from the user on a given time limit (three seconds), control is passed to another user. A user will request control by attempting to perform an action[4].

These mechanisms had advantages and disadvantages. For instance, in the three-second policy, the users adapted the policy to meet their needs and it showed no signs of turning chaotic. This means that coordination of resources can be done through software locks and psychosocial behaviour of the parties involved. Policies such as the three-second policy are more attractive as they give control to users and yield positive results.

A more prevalent application of cloudlets, however, is the offloading of computation. In 2009, Satyanarayanan et al[6] proposed that less portable computing devices such as desktop computers, could be used to carry out computationally intensive processes in the fields of natural language processing; speech recognition; computer vision; machine learning; and augmented reality, on behalf of "resource-poor" mobile devices. They went on to suggest a new architecture where mobile devices use virtual machine technology to run on cloudlets that are within physical proximity of the device and are accessible via a wireless local area network. Each cloudlet would contain at least one computer with relatively high processing power, memory and is available for use by mobile devices within the area of the LAN. This application of cloudlets is not within the scope of this project.

Our work is closely related to the work done by Reitmaier and Benz et al[5]. They have looked into various methods in which co-located people share information with each other. Their conclusion was an intuitive albeit a subtle one: people's sharing patterns changed according to the situation they were in even if they were sharing with the same people. This point was supported with the Mobiphos study done by Clawson et al[1], whereby photos are automatically shared to people in a predefined group. The conclusion was that people's sharing behaviours changes based on the individuals in a group[3]. Psycho-social behaviour like this has to be taken into account especially for this project because the success of the implementations may hinge heavily on this.

5 Ethical, Professional and Legal Issues

The Raspberry Pi boards (including accessories) and mobile devices have been bought therefore the Computer Science department (University of Cape Town) has full ownership of the hardware. In an attempt to avoid any legal issues and intellectual property rights infringements surrounding the use of software libraries; all software libraries used will be open source, in particular, they will have GPL-compatible licenses. The user testing will require focus groups, therefore ethical clearance will be required.

6 Research approach

The proposed system's architecture will be divided up into two parts; front-end and back-end. This division implies that different research approaches will be employed. The front-end is concerned with providing intuitive user interfaces, more generally, it's interested in user experience (UX) thus a qualitative research methodology will be followed for it's development. Development of the back-end will follow a quantitative approach. This will help in understanding which database is optimal for a server in the form of an embedded system, whether or not the use of access points offer faster file transfer speeds than Bluetooth, etc.

7 Methods

7.1 Front-end

This is the part of the system which the user directly interacts with i.e. the mobile device application. The system will be designed to run on Android mobile devices, specifically the Samsung Galaxy Pocket which runs Android 2.3 (codenamed Gingerbread). The literature review reveals that users would like an application with very simple and intuitive interfaces with a short learning curve and a few number of interactions per function. Users also want a system to allow them control of their shared information by dictating: who has access to it; where it is stored; who owns it; and the persistence of it. The application has three main goals. It first has to be able to connect to the cloudlet server. Next, it should allow users to find other users who are connected to the same cloudlet using the application. Finally, it should allow users to select files of any format, stored in the mobile devices, and send those files or allow visibility of them to other selected devices and also be able to store them in the central server.

7.2 Back-end

The back-end of the system can be split into two different logic layers; file storage and network architecture. The raspberry Pi will act as a central server and use ArchLinux as the operating system. The detailed operating system information is as follows:

Arch Linux
A lightweight Linux distribution
Release date: **2014-05-01**
URL: archlinuxarm.org
Kernel version: **3.10**

7.2.1 Network architecture

The front-end of the system is an android application. The Android operating system does not support ad-hoc networking. This restriction means that other means need to be employed for connecting mobile devices with the central server. The Raspberry Pi will be used as an access point by equipping it with a wireless 11n USB adapter for networking. The access point will be created using hostapd , isc-dhcp-server and other necessary software packages.

7.2.2 File Storage

There are currently two potential locations for file storage. The files can be stored on a flash drive or/and SD-card. The information pertaining to file ownership and sharing will be stored on a SQLite database or other database systems. The reason for choosing SQLite as the primary database is because an SQLite database requires little or no administration, SQLite is a good choice for devices or services that must work unattended and without human support. SQLite is a good fit for use in cellphones, PDA's, set-top boxes, and/or appliances. It also works well as an embedded database in downloadable consumer applications[7]. However, other database systems will be investigated to improve performance of the embedded system.

8 Anticipated challenges and testing

8.1 Challenges

8.1.1 Front-end

- The interface needs to be intuitive, that is, it should easily communicate which files have been shared with who, how secure a file is on the server, etc. In order to implement this correctly,the user interface (UI) will be designed and tested with a focus group. Best practices will also be investigated.
- Designing the application to work in very different physical settings may prove difficult because it is being developed in a lab under controlled settings.

8.1.2 Back-end

- The Android operating systems may have other networking restrictions (e.g It does not support ad-hoc networking). Other forms of networks will be investigated.
- SQLite may present high latency thus result in high battery consumption. Other database manegement systems will be used.
- Storage management logic might have flaws. The validity of file sharing is vital thus unit tests will be conducted to ensure reliability.
- General algorithmic inefficiency will also lead to high battery consumption. Best practices will be employed.

8.2 Testing

The testing of the the system will be broken down into several parts based off the main development goals. The front-end and back-end need to satisfy different requirements. This is due to the fact that development of the two layers will follow different methodologies.

8.2.1 Front-end

An important aspect of the front-end is the usability of the interface. Focus groups will test the application to assess if it is intuitive to use. Also, the application should allow the mobile device to connect to the server. Packets will be sent and received from both ends to test the stability of the connection. The application should allow more than one mobile device to connect to the same server at the same time - simple tests will be carried out to check if this is possible. Additionally, multiple devices connected to the network should be able to identify each other or at least know that they are in the presence of other devices. Different ways of presenting this information on the user interface will be investigated and tested. Finally, these devices should obviously be able to send data across the network to each other, this will be tested.

8.2.2 Back-end

JUnit, a testing framework for Java will be used to test data management and file sharing logic. A rule of thumb presented in the SQLite website is that one should avoid using SQLite in situations where the same database will be accessed simultaneously from many computers over a network filesystem. This is because it presents high latency. A benchmark will be conducted to investigate which database is best to use between SQLite, Berkeley Database, NoSQL and other available database management systems.

8.2.3 Security

Cloud services usually have security strategies implemented to mitigate security risks. Some of these security risks also apply to a cloudlet setting. Although security is an important aspect of the technology, it is not the focus of this project and therefore it will not be tested.

9 Work allocation

Development of the project has been broken down naturally into two parts: the front and back end. The front-end is the development of an android application to interface with the server and testing it's usability through focus groups. The back-end is the creation of a server, data storage and development of data management protocols. A slightly detailed account for the front-end and back-end is given in section 7. The front-end will be developed by J. Mutakha. The back-end will be developed by Z. Mahlaza.

10 Anticipated outcomes

10.1 The system

The expected tangible products are a portable central server that will instantiate a cloudlet and coordinate media sharing between users. The second product is an android application targeted at Android 2.3 (Gingerbread) and above with the key features being: connecting to a nearby cloudlet; storing data on the cloudlet and sharing media with other connected devices.

10.2 Expected Impact of the Project

The developers expect the system to work with the bare minimum requirements. That is, the requirement for instantiating a cloudlet is a single Raspberry Pi board coupled with the developed software solution. Users can then share media using any phones compatible with Android 2.3 and above. The expected and intangible outcome is a cloudlet which caters solely for personal social interactions. Additionally, a cloudlet which can also be easily extended to provide more cloud services on a smaller scale.

10.3 Key Success Factors

There are a handful of success factors which the developers deem necessary for the project to be successful. Firstly, having the right software and networking protocols running on the back end is critical.The structure of the cloudlets should cater for the psychosocial aspects of co-located file sharing and allow users the control they would like over sharing and viewership of data. In essense, the cloudlet should address all the ascpects identified in section 3. Tests will be conducted to confirm performance of both system sections; front-end and back-end. A qualitative approach will be taken to test the usability of the front-end. A quantitative test will be carried out to test the suitability of the back-end for an embedded system. A sucessful project will be one where users find the front-end intuitive and one that has an optimal back-end.

11 Project Plan

11.1 Resources Required

- **Development environment**: Linux Ubuntu 12.10/13.04, Windows 7/8, Eclipse 4.3.2 IDE for Java SE Developers (with various plugins), Monitor with HDMI support, wireless 11n USB adapter, SD-card and additional memory.
- **Version Control** - Git. Github will be used for hosting git repository. SourceTree as the GUI client will be used where necessary.
- **Test Mobile Device** - Samsung Galaxy Pocket (GT-S5300) running Android v2.3.6
- **Embedded systems** - Intel Galileo board and Raspberry Pi Model B (with accessories).
- **Database** - SQLite, Berkeley Database and other database management systems.
- **End users** - Focus groups are required to assess the usability and viability of the system. The members of these focus will likely be university students, however, in an ideal world; the focus groups will be diverse to observe the different patterns in use.

11.2 Risks

Any software development project faces risks which may cause it to fail. This project is not different, it has and will be subject to risks until completion. The project started with three members, a project member was lost which has lead to project work being reassigned. The project still faces a number of risks. Table 1 (page 10) shows the risks pertaining to the project's success including each risk's likelihood of occurrence, it's impact on the project and the strategies the team plans on using to mitigate each risk. It is important to understand that the factor is derived by multiplying the likelihood and impact together. The purpose of the factor variable to highlight the overall impact of the risk towards the project. The table is formatted in the following manner.

The risk will be written here	
The mitigation strategy will be written here.	1 Likelihood(1-10): x 2 Impact(1-10): x 3 Factor(1-100): x

Table 1: Risks, their impact and mitigation strategies

Risk / Mitigation	Assessment
Little to no user involvement in design and test process	
Involve potential users in the whole process including design. Regular meetings should be held with them and other stakeholders to ensure there are no wrong assumptions made with the development of the system.	1 Likelihood(1-10): 7 2 Impact(1-10): 8 3 Factor(1-100): 56
Failure to complete all features of the system	
Set priorities on all the features and implement the most important first. Use vertical prototyping where possible over horizontal.	1 Likelihood(1-10): 4 2 Impact(1-10): 9 3 Factor(1-100): 36
Inexperience with working hardware results in delays	
Make use of June/July and September vacation to catch up.	1 Likelihood(1-10): 7 2 Impact(1-10): 9 3 Factor(1-100): 63
Scope creep	
Ensure all the features are prioritised and that any new feature will be added after completion of the initial aspects.	1 Likelihood(1-10): 7 2 Impact(1-10): 9 3 Factor(1-100): 63
Android networking limitations	
Android has shown limitation with ad-hoc networking. If limitations arise with other network modes, we will investigate using other available modes. In the case where none is available, Bluetooh will be used as a fail-safe.	1 Likelihood(1-10): 5 2 Impact(1-10): 9 3 Factor(1-100): 45
Not receiving appropriate hardware in time	
The current available embedded system is the raspberry Pi, the Intel Galileo is not yet available. Development will carry on and focus on the Raspberry Pi	1 Likelihood(1-10): 8 2 Impact(1-10): 1 3 Factor(1-100): 8
Failure to complete all features of back-end or front-end	
Both project members will collaborate to complete remaining features	1 Likelihood(1-10): 5 2 Impact(1-10): 9 3 Factor(1-100): 45

11.3 Timeline

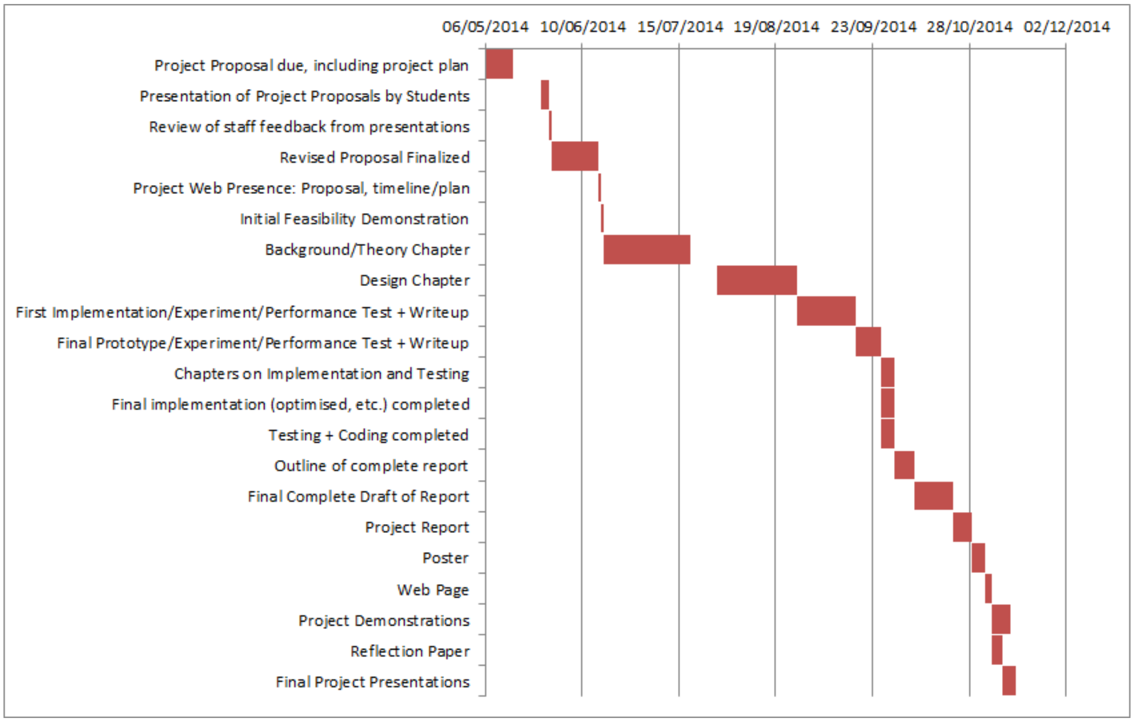


Figure 1: Project Timeline

11.4 Deliverables

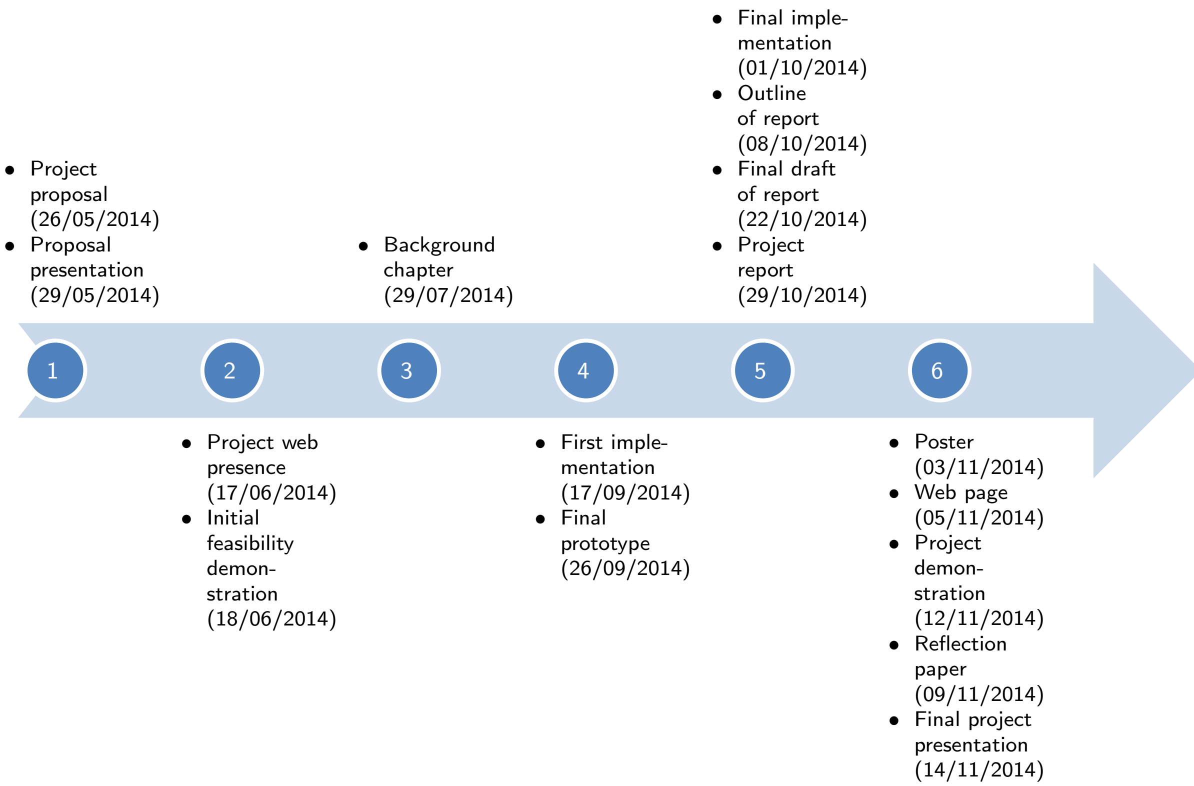
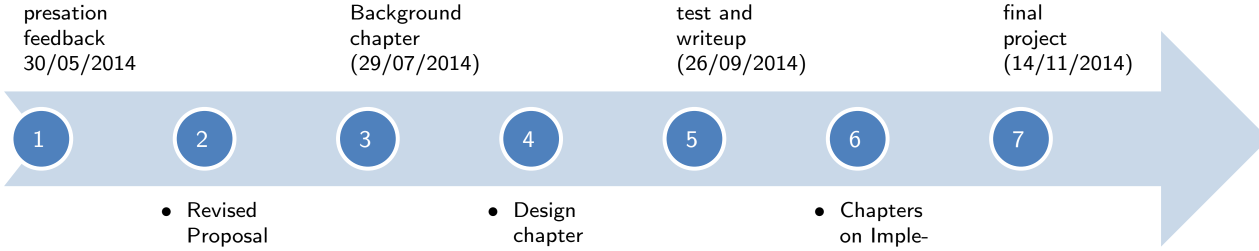


Figure 2: Project deliverables

.

11.5 Milestones



1.
- Project proposal presentation (29/05/2014)
- Review of presation feedback 30/05/2014

2.
- Revised Proposal Finalized (16/06/2014)
- Project Web Presence: Proposal, timeline/plan (17/06/2014)
- Initial Feasibility Demonstration (18/06/2014)

3.
- Background chapter (29/07/2014)

4.
- Design chapter (27/08/2014)

5.
- First Implementation, experiment, performance test and writeup (17/09/2014)
- Final Prototype, experiment, performance test and writeup (26/09/2014)

6.
- Chapters on Implementation and Testing (01/10/2014)
- Complete final implementation (01/10/2014)
- Complete testing (01/10/2014)
- Outline of complete report (08/10/2014)
- Final Complete Draft of Report (22/10/2014)
- Submit project report (29/10/2014)

7.
- Complete poster (03/11/2014)
- Publish web page (05/11/2014)
- Demonstrate project (12/11/2014)
- Submit reflection paper (09/11/2014)
- Present final project (14/11/2014)

Figure 3: Project milestones

.

References

[1] James Clawson, Amy Voida, Nirmal Patel, and Kent Lyons. Mobiphos: a collocated-synchronous mobile photo sharing application. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 187–195. ACM, 2008.

[2] Chris Clay. Raspberry pi: 11 reasons why it's the perfect small server. `http://www.zdnet.com/raspberry-pi-11-reasons-why-its-the-perfect-small-server-7000025206/`, January 2014. [Online; accessed 14-May-2014].

[3] Robert Kraut, Sara Kiesler, Bonka Boneva, Jonathon Cummings, Vicki Helgeson, and Anne Crawford. Internet paradox revisited. *Journal of Social Issues*, 58(1):49–74, 2002.

[4] Leonard M. Ah Kun and Gary Marsden. Co-present photo sharing on mobile devices. In *Proceedings of the 9th international conference on Human computer interaction with mobile devices and services*, pages 277–284. ACM, 2007.

[5] Thomas Reitmaier, Pierre Benz, and Gary Marsden. Designing and theorizing co-located interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 381–390. ACM, 2013.

[6] Mahadev Satyanarayanan, Paramvir Bahl, Ramón Caceres, and Nigel Davies. The case for vm-based cloudlets in mobile computing. *Pervasive Computing, IEEE*, 8(4):14–23, 2009.

[7] Appropriate uses for sqlite. `http://www.sqlite.org/whentouse.html`, 2014. [Online; accessed 24-May-2014].

[8] Marion Walton, Gary Marsden, Silke Haßreiter, and Sena Allen. Degrees of sharing: Proximate media sharing and messaging by young people in khayelitsha. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, pages 403–412. ACM, 2012.