GraaspBox: Enabling Mobile Knowledge Delivery into Underconnected Environments

Andrii Vozniuk
École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
andrii.vozniuk@epfl.ch

Adrian Holzer
École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
adrian.holzer@epfl.ch

Jorge Mazuze
Médecins Sans Frontières
Geneva, Switzerland
jorge.mazuze@geneva.msf.ch

Denis Gillet
École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
denis.gillet@epfl.ch

ABSTRACT

Interacting with knowledge in a timely fashion is critical for the success of humanitarian missions. At the same time, the places where humanitarian action is needed are often those where Internet connection is poor or not available at all, making digital knowledge access difficult. In this paper, we propose a novel knowledge delivery model that relies on a peer-to-peer middleware and uses low-cost computers for local knowledge replication. We have developed a system implementing the model and evaluated it during eight deployments in Médecins Sans Frontières missions. The evaluation demonstrated knowledge delivery abilities of the system and its usefulness for the field staff.

CCS CONCEPTS

• **Human-centered computing** → **Computer supported cooperative work**; • **Computer systems organization** → **Distributed architectures**; • **General and reference** → *Reliability*;

KEYWORDS

Content Delivery, Knowledge Management, Poor Connectivity, Information System, Rural Area, Disaster, NGO

ACM Reference format:
Andrii Vozniuk, Adrian Holzer, Jorge Mazuze, and Denis Gillet. 2017. GraaspBox: Enabling Mobile Knowledge Delivery into Underconnected Environments. In *Proceedings of ICTD '17, Lahore, Pakistan, November 16–19, 2017,* 11 pages.
https://doi.org/10.1145/3136560.3136572

1 INTRODUCTION

Knowledge is regarded as one of the most valuable resources available in humanitarian organizations [14]. Hence, organizations need to be able to access critical knowledge timely and reliably and build and share knowledge efficiently between the headquarters (HQ) and multiple, often geographically dispersed, field teams.

When sharing knowledge in areas with broadband Internet access, organizations can rely on mainstream cloud services such as *Evernote*, *Google Drive*, *Dropbox*, and others that combine knowledge management and social media features. At the same time, the field conditions are challenging and often missing a reliable Internet connection, especially for organizations that operate in crisis situations, making these cloud-based tools unreliable. There is currently a lack of compelling mainstream solutions enabling knowledge delivery into underconnected areas.

Médecins Sans Frontières (MSF or Doctors Without Borders) is one of the leading organizations in the humanitarian and medical fields and improving internal knowledge sharing was identified as one of the strategic objectives of MSF by the direction of the operational center in Geneva (OCG). Graasp[1] (also known as Graspeo) [17] is a social media web platform that was designed and developed from the ground up based on the knowledge sharing requirements of Humanitarian NGOs [17], including MSF, the user interface is presented in Figure 1. The main concepts of the platform were inspired by earlier works in supporting collaboration in online learning communities. The central concept in Graasp is Space that embeds the shared knowledge in the form of text files, videos, topic-based discussions, etc. A space in Graasp can be loosely compared to a folder with associated permissions. The HQ typically uploads essential documents into the appropriate space, and the field staff downloads them when needed. As an example, Figure 1 shows the MALARIA space containing content relevant to malaria thematic.

The high-level architecture of Graasp is explained in [17]. In this paper, we present the novel knowledge delivery component for underconnected settings, making use of Graasp for proof of concept implementation. This paper aims to improve the knowledge access situation by addressing the following research question:

RQ: *How to improve access to digital knowledge in underconnected environments?*

To answer this research question, this paper proposes the following contributions:

(1) a novel knowledge delivery model from an Internet server to a user device in an underconnected environment;
(2) a novel proof-of-concept implementation of the model in a knowledge sharing platform;

ICTD '17, November 16–19, 2017, Lahore, Pakistan

ACM ISBN 978-1-4503-5277-2/17/11.
https://doi.org/10.1145/3136560.3136572

[1]https://graasp.net/

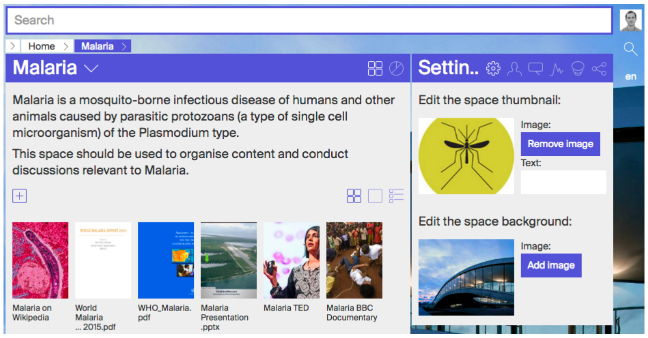

Figure 1: The MALARIA space in Graasp. The sidebar shows the space settings.

(3) an evaluation of the proposed approach in the laboratory and the field with Médecins Sans Frontières.

Concerning the delivery model, we propose to use (1) a peer-to-peer (P2P) synchronization middleware for content delivery and (2) low-cost computers as intermediate local nodes deployed in the field. Such nodes can serve as peers in the P2P delivery network and also provide access to the content via a local web server accessible with a direct WiFi connection. Since the online data is replicated on the local node, it is fast to access. Thanks to P2P protocol the data is synchronized once to a local node and then distributed to all other personal user devices (smartphones, laptops, tablets) locally hence saving the costly external bandwidth. Due to low power consumption of the local node, it can run on a battery. And because of its small pocketable size, it can be taken wherever the content access is required (e.g., a remote hospital in a rural area) not depending on the network or power grid.

This paper is structured as follows. In Section 2 we discuss our learnings from the field experience and state requirements for a content delivery system. Section 3 discusses related work addressing the challenges of underconnected environments. Section 4 presents our proposal and Section 5 describes the implementation of our general model in Graasp. Section 7 presents evaluation results based on tests done in the laboratory and our experience of deploying the solution in the field. Finally, Section 8 wraps up with conclusions and future work.

2 BACKGROUND

To understand the knowledge delivery requirements and come up with a suitable approach, we have worked closely with MSF. In a nutshell, MSF staff typically need to access essential documents during field trips. These essential documents cover thematics (i.e., malaria, sanitation) and geopolitical subjects (i.e., the situation in Yemen) and are frequently updated by the HQ staff as new information appears.

At the same time, the Internet connection type, speed, and reliability can differ substantially from mission-to-mission impacting the content access experience. As an example, Figure 2 shows the setup used to provide Internet access in the Kampala mission in Uganda. In Kampala, the mission is connected to the Internet with a

Figure 2: A 3G mobile Internet stick mounted on the wall provides Internet access in the MSF mission in Kampala, Uganda.

3G modem and afterward the connection is provided in the mission via a WiFi access point.

2.1 Delivery Requirements

To better understand the field knowledge delivery requirements, our colleagues from MSF conducted in-depth field studies that consisted of a total of 145 hours of interviews in Geneva, Niger and Swaziland between June and August 2013. Relying on this process, we have identified together with MSF key field delivery requirements listed below for a suitable field content delivery system:

Req 6.1 – Enable Autonomous Content Access: Situations without Internet or power supply are still common in rural areas [10]. Hence, the system should be capable of operating in an entirely autonomous mode.

Req 6.2 – Provide Fast Data Access: High latency when working with information technology is known to be a source of user frustration [5] and can lead to user inefficiency hindering the technology adoption [11]. Slow knowledge access is particularly severe in emergency cases where fast decision-making is critical. This requirement makes it necessary for the system to enable content access with minimal latency.

Req 6.3 – Be Bandwidth Efficient: The field staff often relies on a satellite or mobile Internet connection. These types of connections are usually costly, have high latency and limited bandwidth. Addressing these limitations, the system should be designed to use as little bandwidth as possible.

Req 6.4 – Be Portable: The systems should be deployable wherever it is needed, outside of the main base and on the road. Hence, it should be easy for a single person to carry around.

Req 6.5 – Enable Content Availability on the User Device: According to our field experience, most of the field staff have a personal mobile device and prefer to have content on the device, so it is available to them anytime wherever they go.

Req 6.6 – Provide Up-to-date Information: The system should update the content whenever it is possible, in order to provide access to the latest version available.

Req 6.7 – Require No Infrastructure Change: Changing IT infrastructure on the scale of MSF is labor-demanding and costly. The required solution should be just drop-in without the need to redesign the local IT infrastructure already in place.

A system meeting these requirements addresses the key challenges when accessing content from the field, namely latency, bandwidth, infrastructure, and the data cost.

3 RELATED WORK

As highlighted in the Introduction, mainstream knowledge management services are suboptimal in underconnected environments since they rely on a centralized cloud-based infrastructure which requires the clients to exchange the data (download and upload) with remote servers in the cloud. There is currently a lack of compelling mainstream solutions enabling knowledge delivery into underconnected areas.

When content is delivered from an Internet server to an end-user device in the field, be it a tablet, a smartphone or a computer, it typically passes through a number of steps schematically represented in Figure 3. We identified three main groups of solutions in existing research literature aiming to improve knowledge delivery. Solutions of **the first group** rely on mechanical content delivery (e.g., by a car or a helicopter) on a portable storage and are suitable for cases when there is no network connection available at some part of the delivery chain. For instance, some humanitarian agencies employ portable storage technologies, such as USB flash drives or USB hard disk drives. Field staff would download the relevant content to the portable storage before going to the mission, then take it to the mission and use access content offline with their laptop. This type of technology has also been proposed to update software in the field [6]. Such solutions require manual upfront content download and afterward manual update and usually can be accessed only by a single user at a time.

Solutions of **the second group**, target the delivery problem when the Internet connection is available but is of poor quality, for instance, unreliable, with high latency or small bandwidth. These solutions are based on intermediate servers (e.g., [12], [10], [13]) and caching. In this case, a local server plays the role of a proxy or relay to the Internet. When the connection is available, the server caches the downloaded information so that it can be easily accessible locally using high local network bandwidth with small latency. One of the common drawbacks of these solutions is that the hardware and software involved (e.g., a proxy server on a PC) often make them not portable and inoperable in the field. Solutions of **the third group** aim to improve the end-user experience by applying techniques on the personal user device, for instance by using caching and offline features in mobile apps or by relying on browser local storage in case of web apps (e.g., [16]). Below, we review existing solutions in more details and highlight their advantages and limitations motivating our approach.

3.1 Group 1. Mechanical Delivery with Portable Storage

Historically, mechanical content delivery was done with various media depending on its cost and availability. Just during the last two decades, first, the CDs were employed for delivering content, afterward DVDs, and more recently USB memory sticks and USB hard drives were used. Hereafter, we outline some of the existing cases of mechanical delivery relying on various types of storage medium.

CDs and DVDs. DVDs were used in [1] for video content delivery in a low-resource classroom. In 2010, DVD players and TVs were already widespread in developing areas, so it was reasonable to use interactivity features provided by DVD players to deliver interactive encyclopedias, language tutoring materials, and medical decision systems on a DVD without the need of having a computer which was considerably more expensive at that time. The core idea of that project was for local teachers to use for instructions in rural schools video content produced by more experienced teachers from better schools. This approach should expose the students to better quality content and in theory, improve their learning outcomes. The authors deployed Digital StudyHall (DSH) software making educational videos available on interactive DVDs having two applications: (1) displaying a PowerPoint presentation and (2) showing books for children. These two applications were evaluated in cases studies done in Indian schools. Based on the experience, the authors concluded that it might be reasonable to deliver high-quality cloud-based MOOCs to low-resource schools to study them together and improve the learning outcomes. In the case of MSF, the content delivery problem was first approached by issuing a CD to an MSF employee before going to the field with prerecorded files relevant to their upcoming mission. Later, a USB Stick with the files was given.

USB sticks. Differently from often read-only CDs and DVDs, USB sticks are writable. This feature is employed by some existing proposals to allow updating the content delivered with a USB stick. For instance, MSF uses a system called LogKey to store relevant content on a USB stick and access the content by running a web application on a user computer. The web application has an option to update the content when an Internet connection is available if the user manually clicks the Update button. Another USB stick delivery example is presented in [6], where the authors propose to use USB flash drives to distribute software updates (in their case anti-virus bases) based on FlashPatch software developed by them. The insight is that since the sticks move around and are plugged into multiple computers, they can serve as a way to spread software updates across the infrastructure. When a drive is plugged into a computer with FlashPatch installed, it offers to write software updates to the drive and automatically monitors and applies new software updates delivered on the stick. With this approach, the updates will move around naturally as drives get plugged into computers.

HDDs. When the content size is becoming large, it is reasonable to employ hard disk drives (HDDs) having bigger capacity. For instance, HDDs were used in Cuba to deliver content locally without

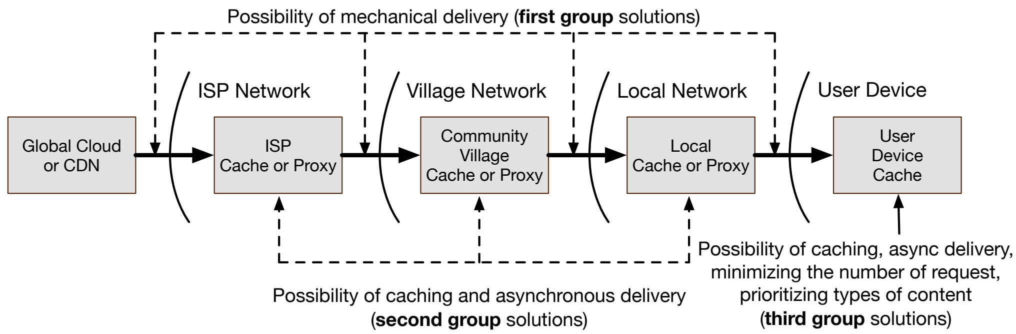

Figure 3: A schematic representation of a typical content delivery chain from a server on the Internet to an end-user personal device in the field. Three groups of solutions are highlighted.

the Internet[2]. Mechanical delivery of HDDs can also be a reasonable option if the Internet connection speed is too low compared to the size of the content being transferred (e.g., AWS ImportExport Snowball[3]).

The described mechanical delivery solutions can be used in situations **when no network is available at all**, and the only option is to deliver content mechanically. The typical small form factor of the storage makes it easy to carry around. Unfortunately, most of the discussed storage solutions are challenging or not possible to connect to mobile phones or tablets since they usually lack a USB port or require special adapters for the connection. Another limitation is related to the fact that such storage solutions can usually be accessed only by a single user at a time since the access requires to attach the storage to the user device physically. The storage-based solutions are also often limited to providing access only to the content, while it can be beneficiary for the users to have access to services built on top of the delivered content. Finally, the content on the storage needs to be updated manually to the latest version every time the Internet connection is available. We aim to keep the highlighted benefits and address the mentioned limitations in our proposal.

3.2 Group 2. Intermediate Servers

When **the network is available but of poor quality** (unreliable or with small bandwidth or large latency) at some part of the delivery chain in Figure 3, a common approach used by a number of solutions is to deploy an intermediate server that will cache the data close to the end-user allowing several users to benefit from already downloaded data. The server can as well allow handling requests asynchronously, downloading content on behalf of the user allowing her to come back when the downloading is done and saving the waiting time. Below, we considered several of existing solutions in more details.

TroTro [12] is a web application running in the browser that integrates with a shared proxy server. The web app can operate in three modes: (1) online (2) poor connectivity and (3) offline. In the poor connectivity mode, the web pages being downloaded are added to a queue, allowing asynchronous download and making the user aware of the downloads in progress and expected waiting time. When accessing the Internet content through the app, the data is being cached on a shared proxy server possibly benefiting the whole local community. In the offline and poor connectivity modes, the user can navigate and search through the content available locally (downloaded before to the proxy), making it possible to find and browse some pages in zero-connectivity settings. The users are informed of the current mode through clear indicators in the web-interface. TroTro is particularly suitable for communities that share the same interests (for instance, cafes and school classes) due to the increased chance of cache hits.

Kwaabana [10] addresses the problem of unreliable, slow and costly Internet in developing areas when sharing content via a social network, in their case Facebook. Kwaabana consists of two parts: (1) a global Kwaabana server where the files shared outside of the village are put and (2) a village server storing the files shared locally between the members of the same village (located on the same network). Based on the social graph, the system can identify to which of those servers a particular shared file should be uploaded in the end. The files are always first uploaded to the village server, but if the sharing is with an external member, then the files are transferred to the global server. Kwaabana employs several techniques when transferring files. It uses a queue to order the uploads and make sure that eventually files get transferred even in the case of connectivity or power grid disruptions. Also, Kwaabana shapes the traffic minimizing the negative impact of long-running uploads on interactive (browsing) traffic. Together with the data in files, Kwaabana synchronizes the metadata stored in the database (users, their location, files metadata) by shipping the database SQL statements in a text file. As a result, the system speeds up the sharing and saves the bandwidth when sharing happens locally (often the case according to the social graph structure) and provides a reliable way to transfer files for global sharing even in the case of unreliable network and the power grid. The Kwaabana authors pointed out that a P2P architecture may be a promising future direction, but no details of the idea were presented.

Offline Downloading [13] is a popular approach employed in China to overcome low download bandwidth. According to [13], there are two broad types of offline downloading approaches in

[2] Cuba Offline https://www.theguardian.com/world/2014/dec/23/cuba-offline-internet-weekly-packet-external-hard-drives (last accessed 10 May 2017)
[3] AWS ImportExport Snowball https://aws.amazon.com/importexport/ (last accessed 10 May 2017)

China: (1) when an Internet Service Provider (ISP) offers a service for offline downloading and (2) using software installed on smart WiFi access points providing a download manager. In both cases, the idea is that the user would submit a download request specifying the file to be downloaded and the download would be handled by the ISP server or by the access point. After the download is finished, the user can have fast access to the file without no further waiting. The recommended approach to be used in a particular situation depends on several factors [13], including the quality of the connection between the end-user device and the server handling the offline download.

Content Delivery Networks (CDN) [15] replicate the data to the edge servers all over the world (usually, several servers per continent) aiming to improve the download speed by redirecting the client to the server with the faster connection to that client, usually the one located physically the closest. Technically, when a DNS request is made to download a web resource (for instance, an image), the hostname of the resource is resolved to IP address of the server that will be the best in serving the resource, normally the one located in the proximity to the requesting device.

3.3 Group 3. Approaches on Personal Device

The third group of solutions applies various **techniques on the user device**, including in-app caching, asynchronous content download and upload and minimizing the number of requests to improve the response time. We review below some of the proposals.

COCO [16] is a framework allowing to improve the user experience when working with web applications in underconnected settings. COCO uses a local cache based on Google Gears[4] to allow the web app to operate completely offline in case of no Internet connection or reduce the latency by reducing the number of requests when operating online. The solution is particularly suitable for data submission scenarios when the recorded data is accumulated locally and pushed to the server when the connection becomes available. When version conflicts occur, COCO relies on a built-in mechanism for their resolution.

ODK Submit [4] targets the data collection and submission problem. ODK Submit uses contextually available information about the data (like the importance of the data, availability, and price of different types of network connection) to identify which channel should be used to transfer which type (or part) of the data. For instance, information about the number of doctors in the hospital is not heavy and is important while their photos are heavy and not that important. So, in this case, it makes sense to send the number of doctors even via SMS (or pricey 3G when available), while the photos can be delivered later for free via WiFi when it is available. The Submit framework developed by the authors can run on Android as a service that abstracts the networking and can be used by other Android applications when they need to send the data. The service allows to use 3G, SMS and supports P2P data transfer with QR codes, two versions of NFC, Bluetooth and WiFi Direct to transmit the data. In practice according to the user studies, WiFi Direct seemed to be a preferable solution. WiFi Direct can take a bit longer to setup compared to Bluetooth, but offers better speeds and is particularly suitable for files more than 1MB in size as demonstrated by the authors. The authors have shown that Submit can act according to configured policies, for instance, it was able to conserve 3G traffic by sending only essential data, and the rest of the data was sent using WiFi.

3.4 Discussion

Based on the presented review of the delivery solutions, in our proposal, we aim to offer benefits of the first (Mechanical Delivery) and the second (Intermediate Servers) proposal groups. As in the first group, we strive to make our proposal portable allowing mechanical content delivery and as in the second group we want to reduce latency and save bandwidth by integrating intermediate servers into our proposal. In the next section, we describe the delivery model we propose.

4 MOBILE KNOWLEDGE DELIVERY: P2P AND LOW-COST COMPUTERS

To offer a system suitable for field conditions satisfying the requirements formulated in Section 2.1, we have revisited the assumptions underlying the design of existing knowledge delivery systems. Two prominent trends are underway in the recent years. The **first trend** is the ongoing proliferation of personal computing devices including tablets, laptops and, particularly, smartphones. The personal devices are already ubiquitous in developed countries reflected by the Bring your own device (BYOD) policy and are becoming pervasive even in developing countries. At the same time, mobile phone subscriptions penetration in developing regions is already approaching 100% [18]. While the personal computing devices availability is still far from 100%, the ongoing spreading of smartphones allows expecting that the smartphones will become ubiquitous in developing regions in the near future, like mobile phones recently did. In practice, the majority of MSF mission employees are equipped with at least one personal computing device, a laptop provided by MSF, and also often possess a smartphone or a tablet. The **second trend** is the availability of low-cost small yet powerful general-purpose computers, particularly single board computers (SBCs). It is now possible to buy a general-purpose SBC under 30 USD, an equivalent of a USB flash drive cost a few years ago. Such low-cost computers enable to build a set of services running locally and accessible in the underconnected areas. Our proposal builds on these two trends.

To facilitate knowledge delivery into underconnected settings and answer RQ, we propose a novel model to deliver knowledge from an Online System on the Internet to the end-user Personal System using: (1) a P2P synchronization middleware, (2) a Local System running on a low-cost computer deployed in the field and (3) a Personal System providing content access on personal mobile devices. The model is schematically presented in Figure 4. Hereafter we discuss each of the components and their interplay more closely.

Online System. When a high-quality Internet connection is available, the Online System is the preferred way of accessing organizational knowledge. The Online System can benefit from the computation power offered by the cloud and provide useful but at the same time computationally demanding services including search, analytics, and recommendations that are challenging to implement on low-cost computers. The HQ staff would typically

[4]Google Gears https://en.wikipedia.org/wiki/Gears_(software) (last accessed 10 May 2017)

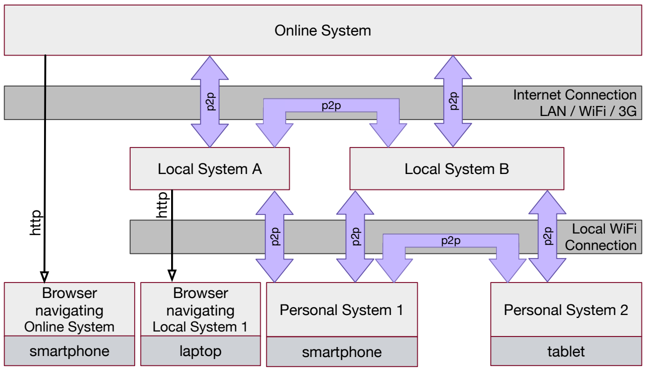

Figure 4: The mobile delivery model with P2P and low-cost computers. The Local System is located in an underconnected environment (e.g., an MSF mission) and acts as a local replica containing relevant content.

use the Online System since good Internet connection is usually available in the HQ.

Local System. The Local System serves two primary purposes: (1) it acts as a local peer with the relevant content that can be synchronized from it and (2) it provides a web-access to the content in the field. It is possible to connect to the Local System (using an ethernet cable or wirelessly) and using a web-browser even when the Personal System is not installed. To provide knowledge access in various field situations, possibly off-the-grid and meet the delivery requirements stated in Section 2, the Local System should have a small form factor, low power consumption, and be able to run on a battery. Since often the field bandwidth is limited and expensive, the Local System should download the content from the Online System only once and then locally distribute it to Personal Systems. This is different from the mainstream cloud solutions where each client downloads from the cloud its own copy of the content. The Local System would be typically used by the field staff when accessing content in a permanent or temporary base, usually as part of a team.

Personal System. The Personal System is an application (mobile or desktop) running on a personal user device. It has two functions: (1) handle P2P synchronization of the relevant content and (2) provide a UI for the personal device content access. Once the initial synchronization is done, the Personal System is the most portable option allowing the user to access relevant content at any time. The content can be updated by synchronizing with the Local System or the Online System depending on the available connection. The Personal System would usually be used by the field staff on the go, when away from a base. In case the user does not have the Personal System installed, she can still use the device to connect to the available Local System and navigate the content with the browser. This enables the content downloading but not synchronization.

P2P Middleware. Using a P2P middleware enables resilient and reliable content delivery on the global scale. For instance, CERN relies on P2P to deliver Large Hadron Collider experiment data to its partners worldwide. Because of its resilient properties, a P2P middleware is a suitable foundation for content delivery into underconnected environments. Thanks to the P2P middleware, the Online System, Local Systems and Personal Systems can all act as nodes in a P2P content delivery network. Because of the local peer discovery, the Local System and the Personal Systems can find each other and synchronize directly in the field (shown with horizontal arrows in Figure 4), without the need for transferring data via the central server, benefiting from a faster local connection. The P2P middleware can adapt the delivery network structure to the connection speeds allowing to synchronize the data from the Online System to the Local System in the field only once and then distribute it to the Personal Systems hence minimizing the external bandwidth usage and saving often costly data.

5 IMPLEMENTATION: THE GRAASPBOX

Following the model proposed in the previous section, we have developed a proof-of-concept solution for MSF using Graasp as the Online System, a low-cost computer running custom software (called GraaspBox) as the Local System and Bittorrent Sync[5] (BtSync) as P2P middleware.

Online System: Graasp. MSF relies on Graasp for their knowledge sharing as explained in Section 1. In Graasp, each space is mapped to a folder in the file system that can be synchronized to the Local System and the Personal System. To enable delivery of some non-traditional content types such as Space descriptions or online threaded discussions, we generate HTML files so the content can be viewed in browsers available virtually on any personal device.

A space owner can enable synchronization of a space by pressing the "Enable peer-to-peer sync" button as shown in Figure 5 and disable using the "Disable peer-to-peer sync" button (see Figure 6). When synchronization is enabled, the space content synchronizes to Local Systems according to the model in Figure 4. To start content synchronization to specific Personal System (e.g., a smartphone or a laptop) the user should click the "Sync space content to my device" button in the browser of her device (see Figure 6). When the Internet connection is absent, the synchronization to the Personal System can be set up in a similar way by using the interface of the Local System.

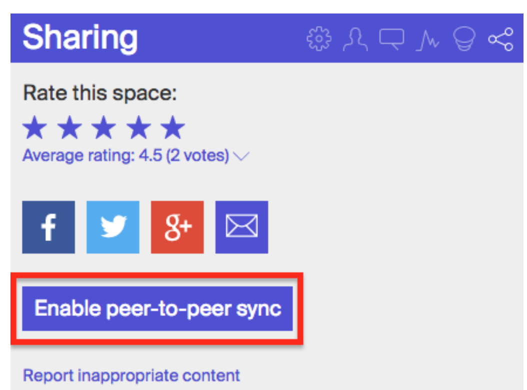

Figure 5: A space owner can enable synchronization of the space by pressing the "Enable peer-to-peer sync" button on the sidebar.

[5]http://www.bittorrent.com/sync (last accessed 10 May 2017)

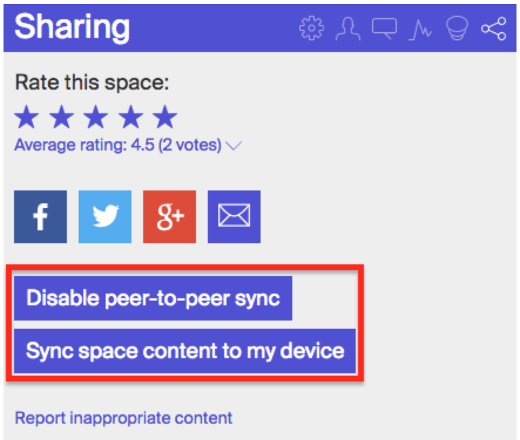

Figure 6: A space owner can disable synchronization of the space by pressing the "Disable peer-to-peer sync" button on the sidebar. To start synchronizing the space to her personal device, she can press the "Sync space content to my device" button.

P2P Middleware: BtSync. We choose to use BtSync as a P2P middleware over open-source alternative including Hive2Hive[6] or Syncthing[7] since, according to our benchmarks, BtSync looked more production-ready in terms of the APIs maturity and general stability compared to the open-source solutions. Also, the Bittorrent protocol that BtSync is built on has a proven resilience and performance. Still, our general delivery model is not coupled with a particular P2P middleware implementation. When syncing with BtSync, each space in Graasp has an associated secret Sync key accessible by the Local System. To manage the synchronization, the Online System and the Local System interact with BtSync process via its APIs. The P2P BtSync middleware is aware of the network performance and makes sure that Personal Systems are downloading content from the Local system (GraaspBox), minimizing external Internet bandwidth usage.

Local System: GraaspBox. In essence, GraaspBox is a Raspberry Pi single board computer (SBC) shown in Figure 7 running the Local System software. SBCs are low-cost, portable and usually energy efficient general purpose computers. When choosing a device for running the Local System, we have evaluated five different SBCs including Banana Pro, Orange Pi, Orange Pi Plus, Orange Pi Plus 2, and Raspberry Pi 3 and settled on Raspberry Pi 3 due to its adoption and community support. Reliability of the device hardware and software is crucial when deploying technology into the field since in case an issue occurs it is not always possible to fix it remotely. Higher adoption increases chances that errors are discovered and fixed early. In addition, there are already cases of employing Raspberry Pi in developing regions, for instance, when teaching practical Geography with sensors in a secondary school in Kenya [7]. Thanks to the low power consumption of Raspberry Pi, GraaspBox can run on a battery chargeable via USB so that it can be recharged in a car or using a solar panel. Due to its small pocketable size, the GraaspBox can be taken wherever the content is required (e.g., a hospital in a rural area or a car on its way to a remote mission) not relying on the network or power grid. To enable the local WiFi access of the content, we run a local DNS server with *dnsmasq* that resolves http://graasp.box to the local web server. This web server, built with Node.js, interacts with the BtSync process managing the Sync keys and making sure that required content is synchronized. It as well allows the users to browse already synced content as shown in Figure 8 (1).

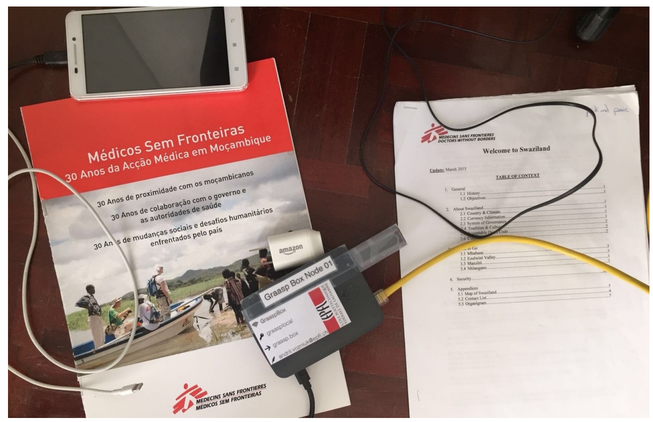

Figure 7: A GraaspBox node in the Maputo mission, Mozambique. At this point, the node is connected to the Internet using the yellow LAN cable.

Personal System. We use the free BtSync Android, iOS and Windows applications[8] to enable P2P synchronisation and provide content access on users' personal devices (see Figure 8 (2)). When the BtSync app is not available, a web-browser can be used to browse the content stored on the GraaspBox (see Figure 8 (1)).

In the next section, we explain how GraaspBox can be used in the field to improve content access.

6 GRAASPBOX USAGE SCENARIO

When there is a **fast and reliable Internet connection available**, the user can access, view and download content directly on the Online System (Graasp). In addition to just content access, the user can benefit from other services provided by the Online System including advanced search, analytics, recommendations and a rich set of interaction options.

When the **Internet connection is slow but still present**, the user can use a hybrid scenario. In this scenario, she continues to navigate, search and interact using the online platform that is often cached by the browser, but in case she needs to download a large file, she may download it quickly from the local GraaspBox by clicking on the "GraaspBox Download" button as shown in Figure 9. Pressing this button will redirect the user browser from the online content address starting with http://graasp.net/ to the GraaspBox content address starting with http://graasp.box/. For this redirect to work the user must be directly connected to the GraaspBox or

[6]http://hive2hive.com/ (last accessed 10 May 2017)
[7]http://syncthing.net/ (last accessed 10 May 2017)
[8]BtSync Apps https://getsync.com/download (last accessed 10 May 2017)

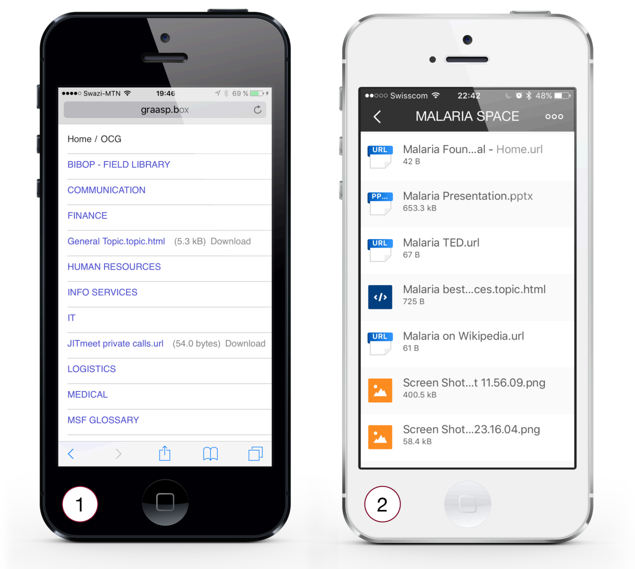

Figure 8: (1) The MALARIA space on the GraaspBox accessed with a browser. (2) The MALARIA space on the phone managed with BtSync client.

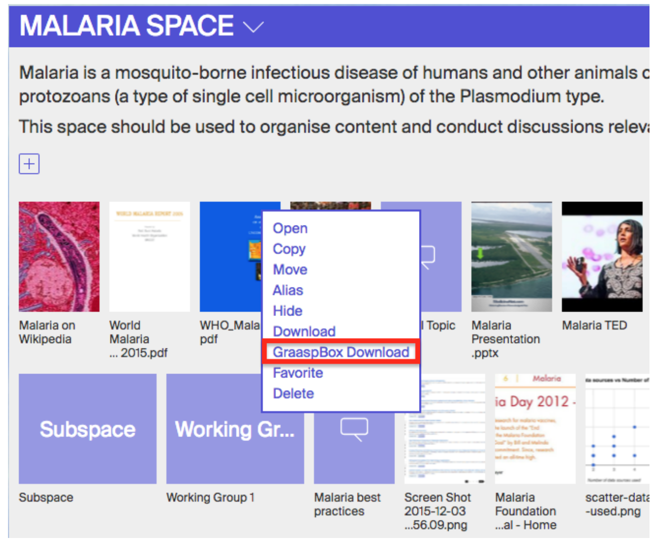

Figure 9: A user can navigate the content using the Online System and at the same time download it from the Local System by clicking the "GraaspBox Download" button highlighted in the contextual menu.

the router in the user local network should have a corresponding DNS record in place.

If the Internet quality further degrades and the **connection becomes unreliable or not available at all**, the user can connect to the GraaspBox (via ethernet or wirelessly), navigate and access the content there offline. In case if the user would like to have the content available on her personal device, she can set up the synchronization with GraaspBox and afterward interact with the content locally on the device even when disconnected from GraaspBox.

The next section focuses on the GraaspBox deployment in the field and user feedback.

7 EVALUATION

To evaluate the GraaspBox, we conducted a series of laboratory functional and performance tests and deployed the devices in eight MSF missions during field trips.

Tests in the Laboratory. Before sending the device into the field, we have conducted a series of tests in the laboratory setting simulating the usage scenarios expected in the field. In summary, the GraaspBox was able to synchronize the content and provide local access to it via WiFi without the Internet access. It was able to act as a local peer for the peers connected to it via WiFi. When we enabled the Internet connection again, the GraaspBox downloaded updates from the online server. The maximum online synchronization speed observed in our tests was around 90 Mbps when GraaspBox was connected by cable to LAN. The download and upload speeds observed when using the WiFi stayed close to 20 Mbps. In our tests, the Raspberry Pi was able to sustain maximum 32 simultaneous WiFi connections. The device was able to work continuously for nearly 45 hours (workload-dependent) on a single charge of a 20000 mAh battery. These results validate that GraaspBox has sufficient capabilities to support an MSF team in typical field scenarios.

Field Network Performance. To obtain the results reflecting the actual field condition the most accurately, the tests from the field were necessary. Realistic data is of particular importance since the Internet connection in the field is usually much less stable than in the HQ, even in the case where there are a couple of redundant connections available in the missions. The network performance tests were conducted by MSF field staff representing two typical Graasp access scenarios. The first one measured the Graasp landing page loading time for not logged in users and the second measured loading time when the user was logged into her account on Graasp seeing her home page. The tests were done in August 2015 in Swaziland, Myanmar, Niger, Chad, Uganda, Kyrgyzstan and Mozambique. The test results are summarized in Table 1. Not surprisingly, in most of the cases, the measured loading times strongly correlated with the Internet connection bandwidth. It is worth noting that the obtained values varied during the day due to the Internet connection usage by the other users in the missions. Although the results of these tests are not completely reliable, they still provide useful insights. For instance, when a VSAT (satellite) connection is used (as in Yangon, Myanmar, and N'Djamena, Chad), the response is slow due to the network latency, even when the bandwidth is sufficient. As for the other locations, when using an optical fiber or a DSL connection the Graasp website loads quickly. During the tests, we did not find a relation between the computer model and the loading times verifying that Graasp loading is not CPU bound. Interestingly, during the test the smallest loading time values were obtained on the oldest computer due to its good Internet connection. Regarding the tests generalization, it is worth keeping in mind that all the tests were conducted from the locations where the Internet connections are usually the best available to MSF staff in the country. Apart from the most developed countries, such as Kyrgyzstan, it is reasonable to expect that the most often Internet connection type available in the field is a VSAT connection.

Location	Connection Type	Download speed	Upload speed	Landing page loading time (seconds)	Home page loading time (seconds)
Kyrgyzstan Bishkek	Optical Fiber	10M	In country: 10 Mbps Out country: 1 Mbps	2.56	2.56
Myanmar Yangon	VSAT DSL	512 Kbps 1 Mbps	51 Kbps 512 Kbps	20.09	
Niger Niamey	Optical Fiber Wi-Max	1 Mbps 64 Kbps	1 Mbps 256 Kbps	17.13	12.45
Swaziland Nhlangano	DSL DSL	5 Mbps 1 Mbps	508 Kbps 252 Kbps	4.91	
Chad N'djamena	VSAT	2 Mbps	256 Kbps	34.50	
Uganda Kampala	Optical Fiber	1 Mbps	1 Mbps	3.24	8.89

Table 1: Internet connection type and performance in MSF missions. The gray cells denote the data not obtained from the field.

Field Surveys. GraaspBox was deployed and evaluated by one of the authors during a total of eight field trips to Mozambique (Maputo and Gaza), Swaziland (Nhlangano, Mbabane, Hlathikhulu, Matsanjeni), Uganda (Kampala), and Chad (N'Djamena). Two examples of the field study with the GraaspBox are presented in Figure 10. At the beginning of the trips, the manager demonstrated to the field staff several scenarios accessing the MSF content locally with the GraaspBox as explained in Section 6. He showed how to connect to the device, open content and establish synchronization with a personal device. Then he encouraged the mission staff to use the GraaspBox for several days for their daily needs.

During these trips, the GraaspBox was used by in total 45 people, out of them individual interviews were conducted with 27 field employees, and of those $N = 14$ (11 - Male and 3 - Female; age ranged from 26 to 45 with median 31) have completed our questionnaire. The questionnaire consisted of three parts: (1) evaluating the current situation with the Internet access; (2) collecting the needs regarding the knowledge delivery and (3) evaluating the field staff overall GraaspBox experience with the system usability scale (SUS) questionnaire [3]. We report the results below.

Internet Access. The survey results have confirmed that slow and unreliable Internet connections were regular in the field. Six respondents (43%) said that they had a slow connection all the time, six (43%) witnessed it daily, and two (14%) - a few times each month. Unreliable connections were reported to be encountered all the time by six people (43%), daily - by five (36%), less often - by two (14%), and never - by one (7%). Moreover, five people (36%) reported to work all the time in situations without any Internet connection, four (29%) - work daily, three (21%) - a few times each month, and two (14%) - always have some way to access the Internet. Thirteen people (93%) expressed a strong agreement that they wanted to access the MSF content faster than they could. These numbers

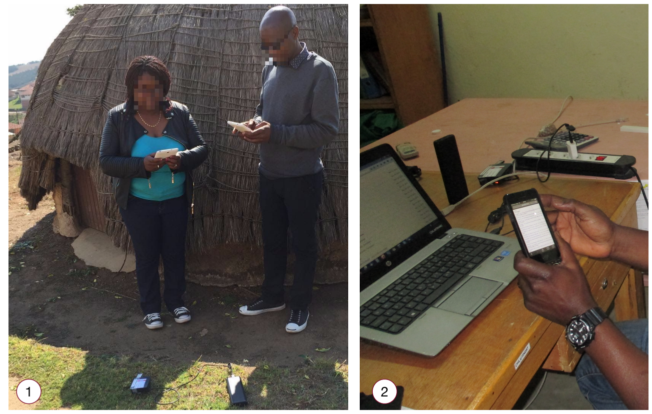

Figure 10: (1) The MSF mission members are accessing content on a battery-powered GraaspBox in Maputo, Mozambique. (2) A mission member accessing the GraaspBox content using his phone and laptop in Chad.

indicate the challenging access conditions in the field and are in line with the requirements formulated in Section 2.1. After using the GraaspBox, 11 people (79%) expressed a strong agreement that MSF content access became faster than it was with the online platform.

Delivery Needs. According to the survey results, different ways of accessing the content are complementary, since users expressed a strong agreement with having relevant content available on their personal devices, on the GraaspBox and online (correspondingly, $\mu = 4.93$, $\mu = 4.57$ and $\mu = 4.50$ on the 5-point Likert scale). When asked regarding the single preferred access type, the majority of respondents picked their personal device (10 responses out of 14).

System Usability Scale. Based on the SUS questionnaire responses, the final GraaspBox SUS score was equal to 78, meaning that the system has a GOOD, close to EXCELLENT, usability score [2]. The Figure 11 presents an overview of answers to individual questions of the SUS. The users reported that they were eager to use GraaspBox frequently ($\mu = 4.86, \sigma = 0.53$), they did not find it unnecessary complex ($\mu = 1.79, \sigma = 1.05$), they thought that it was easy to use ($\mu = 4.50, \sigma = 0.85$), they thought that they would not need the support of a technical person ($\mu = 2.00, \sigma = 1.04$) they found various functions to be well integrated ($\mu = 2.86, \sigma = 0.66$), they neither agree nor disagree regarding the system inconsistency ($\mu = 3.00, \sigma = 1.18$) they imagined most people would learn to use it quickly ($\mu = 4.79, \sigma = 0.58$). It was not cumbersome to use ($\mu = 1.57, \sigma = 1.02$), they felt rather confident using it ($\mu = 4.79, \sigma = 0.58$) without the need to learn a lot of things before getting going ($\mu = 2.29, \sigma = 1.59$). In addition to these positive results based on the survey responses, we have obtained positive feedback during face-to-face interviews with the field staff. The aspects of the system that received the lowest scores were the integration of different components (SUS Q5) and overall consistency (SUS Q6). This outcome is not surprising since the system is still fairly new providing opportunities for iterative improvements. In the future, we are planning to conduct more detailed user interviews and think-aloud sessions to identify parts of the system that should be made more consistent or better integrated.

Figure 11: GraaspBox system usability score survey results.

8 DISCUSSION AND CONCLUSIONS

Our proposal allows a humanitarian organization to deliver and keep up to date knowledge in their underconnected missions. Implementing our model, other knowledge sharing platforms can as well benefit providing offline access to content hosted there. It is worth highlighting an additional benefit of our model. Since the data is replicated from the Online System to multiple Local Systems in different locations, this provides additional data backup and improves the overall data storage reliability.

The focus of our study was on delivering organizational knowledge for humanitarian organizations, where field workers are exposed to underconnected environments on a daily basis. At the same time, the proposed model and its implementation with GraaspBox can be used to deliver training and educational resources (especially, cloud-based MOOCs) to educational institutions in developing countries located in underconnected areas [8]. Also, our model can be employed to deliver software updates and improve overall IT infrastructure security in the field [6].

Although GraaspBox fulfills all of the field delivery requirements formulated in Section 2.1, our field experience has uncovered some shortcomings of our approach discussed below.

Complexity of P2P. We observed that for regular users it was hard to understand principles of P2P networks. Accessing and synchronizing data from a local node and not from the cloud is something they are not used to. In most of the cases, GraaspBox just worked as expected. But when some issues occurred (e.g., the updated file was not synced yet), it was challenging to explain to the user's specific technical problems underlying the issue. In the future versions of the GraaspBox, we are planning to make it more transparent to the end-user how the content gets delivered by explaining it better in the documentation.

P2P App Installation. P2P synchronization to the user device requires an app (Personal System) to be installed. Since the installation often needs to take place in the field without a fast Internet connection, we store the apps on the GraaspBox so they can be downloaded locally. It is possible to install the apps in this way on Android and Windows or Linux, but not on iOS (without the jailbreak). iOS allows installing apps only from the App Store, for which the Internet connection is a must. When the app cannot be installed, the GraaspBox provides a way to access the content using the web-browser available on virtually any platform. But in such case, the content is not synchronized to the personal device. Another promising direction is to enable P2P synchronization directly in the user browser without the need to install a standalone application. Several browser-based P2P architectures exist built on top of the WebRTC standard[9], but still, not all of the features are supported even by the latest versions of popular browsers. Usually, browsers used in the field can be outdated lacking the standard support at all.

Below, we discuss some of the promising directions for the future research.

Knowledge Delivery From Underconnected Environments. In this study, we targeted knowledge delivery to underconnected environments. We would also like to be able to deliver knowledge from the field to the HQ. Uploading data from poor network environments is a known problem, where it was measured that more than 75% of uploads fail in such conditions [10]. The GraaspBox users in the field have suggested using GraaspBox to manage the unreliable uploads so the users can just put the files on the personal device and have them eventually uploaded. Technically, our architecture also allows uploads, but uploads bring challenges related to handling conflicted versions of the same content that was modified by multiple users. As part of the future work, we plan to enable one-way content delivery from the field to the HQ for some Graasp spaces. In such a case the Local System (GraaspBox) can serve as a staging node where data is stored until the Internet connection becomes available. Afterward, the accumulated content is eventually synchronized to the server in the HQ.

Delivering Services. We continuously extend GraaspBox to provide not only local access to the content but also to run a set of services locally in the field. Such services can be beneficial for local collaboration and communication. For instance, one of the services we deployed and evaluated on GraaspBox is SpeakUp [9], a system allowing anonymous brainstorming that can elicit the best ideas in a group or best questions in a Q&A session. We run SpeakUp server on GraaspBox and provide local access to it by forwarding requests coming from SpeakUp mobile applications to GraaspBox.

To summarize, in this paper we studied how to improve digital knowledge access in underconnected environments (*RQ*). We proposed a novel knowledge delivery model that answers this question by enabling knowledge delivery to mobile devices using peer-to-peer middleware and low-cost computers. We also presented a novel proof-of-concept implementation of the model called GraaspBox for the knowledge sharing platform Graasp. Finally, we evaluated the implemented model in the laboratory through a series of tests and in the field by deploying the GraaspBox in eight MSF missions

[9]WebRTC Project https://webrtc.org/ (last accessed 10 May 2017)

in Mozambique, Swaziland, Uganda, and Chad. The evaluations showed that the proposed model and the corresponding implementation allow to retain knowledge access in cases when there is no Internet or power grid connection and speed up the access when the connection is slow.

ACKNOWLEDGMENTS

This research has been partially funded by the European Union in the context of Go-Lab and Next-Lab projects (FP7 and Horizon 2020 Programmes, grant agreement nos. 317601 and 731685). Also, part of the research was supported by the EPFL Innogrant and via the collaboration with Médecins Sans Frontières.

REFERENCES

[1] R Anderson, C Robertson, E Nabi, U Sahni, and others. 2012. Facilitated video instruction in low resource schools. *ICTD* (2012).

[2] Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An Empirical Evaluation of the System Usability Scale. *Int. J. Hum. Comput. Interact.* 24, 6 (29 July 2008), 574–594.

[3] John Brooke. 1996. SUS: A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.

[4] Waylon Brunette, Morgan Vigil, Fahad Pervaiz, Shahar Levari, Gaetano Borriello, and Richard Anderson. 2015. Optimizing Mobile Application Communication for Challenged Network Environments. In *Proceedings of the 2015 Annual Symposium on Computing for Development*. ACM, 167–175.

[5] Jay Chen, Saleema Amershi, Aditya Dhananjay, and Lakshmi Subramanian. 2010. Comparing web interaction models in developing regions. In *Proceedings of the First ACM Symposium on Computing for Development*. ACM, 6.

[6] Henry Corrigan-Gibbs and Jay Chen. 2014. FlashPatch: Spreading Software Updates over Flash Drives in Under-connected Regions. In *Proceedings of the Fifth ACM Symposium on Computing for Development*. ACM, 1–10.

[7] Francesco Cuomo, Eric Mibuari, Komminist Weldemariam, and Osamuyimen Stewart. 2013. Leveraging Raspberry Pi for interactive education. In *Proceedings of the 4th Annual Symposium on Computing for Development*. ACM, 16.

[8] Gérard Escher, Dimitrios Noukakis, and Patrick Aebischer. 2014. Boosting Higher Education in Africa through Shared Massive Open Online Courses (MOOCs). In *Education, Learning, Training*. Brill, 195–214.

[9] Adrian Christian Holzer, Sten Govaerts, Jan Ondrus, Andrii Vozniuk, David Rigaud, and Benoit Garbinato. 2013. SpeakUp - A Mobile App Facilitating Audience Interaction. In *The 12th International Conference on Web-based Learning*.

[10] David L Johnson, Elizabeth M Belding, and Consider Mudenda. 2013. Kwaabana: File sharing for rural networks. In *Proceedings of the 4th Annual Symposium on Computing for Development*. ACM, 4.

[11] Hee-Woong Kim and Atreyi Kankanhalli. 2009. Investigating user resistance to information systems implementation: a status quo bias perspective. *Management Information Systems Quarterly* 33, 3 (2009), 9.

[12] Laura Dan Li and Jay Chen. 2013. TroTro: web browsing and user interfaces in rural Ghana. In *Proceedings of the Sixth International Conference on Information and Communication Technologies and Development*. ACM, 185–194.

[13] Zhenhua Li, Christo Wilson, Tianyin Xu, Yao Liu, Zhen Lu, and Yinlong Wang. 2015. Offline Downloading in China: A Comparative Study. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*. ACM, 473–486.

[14] Christina Matschke, Johannes Moskaliuk, and Ulrike Cress. 2012. Knowledge exchange using Web 2.0 technologies in NGOs. *Journal of Knowledge Management* 16, 1 (2012), 159–176.

[15] Stefan Saroiu, Krishna P Gummadi, Richard J Dunn, Steven D Gribble, and Henry M Levy. 2002. An Analysis of Internet Content Delivery Systems. *Oper. Syst. Rev.* 36, SI (Dec. 2002), 315–327.

[16] Saureen Shah and Apurva Joshi. 2010. COCO: a web-based data tracking architecture for challenged network environments. In *ACM DEV*.

[17] Andrii Vozniuk, Adrian Holzer, Sten Govaerts, Jorge Mazuze, and Denis Gillet. 2015. Graspeo: a Social Media Platform for Knowledge Management in NGOs. In *ICTD*.

[18] Susan Wyche. 2015. Exploring mobile phone and social media use in a Nairobi slum: a case for alternative approaches to design in ICTD. In *ICTD*. ACM.