Phone Proxies: Effortless Content Sharing between Smartphones and Interactive Surfaces

Alexander Bazo
Media Informatics Group
University of Regensburg
alexander.bazo@ur.de

Florian Echtler
Media Informatics Group
University of Regensburg
florian.echtler@ur.de

ABSTRACT
We present *Phone Proxies*, a technique for effortless content sharing between mobile devices and interactive surfaces. In such a scenario, users often have to perform a lengthy setup process before the actual exchange of content can take place. Phone Proxies uses a combination of custom NFC (near-field communication) tags and optical markers on the interactive surface to reduce the user interaction required for this setup process to an absolute minimum. We discuss two use cases: "pickup", in which the user wants to transfer content from the surface onto their device, and "share", in which the user transfers device content to the surface for shared viewing. We introduce three possible implementations of Phone Proxies for each of these use cases and discuss their respective advantages.

Author Keywords
casual interaction; smartphone; mobile device; interactive surface; NFC

ACM Classification Keywords
H.5.m Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION & RELATED WORK
As large-scale interactive tables are increasingly appearing in public settings, the number of people having the opportunity to interact with such devices also grows. Due to today's pervasive availability of smartphones, most of these people can also be expected to own a personal mobile device. As these mobile devices are rapidly becoming the default storage location for personal media such as pictures, music or contact data, users will desire an easy way to share this content on the interactive surface. However, all current solutions to this task require a complex setup process involving various combinations of stick-on marker tags, custom app installations and wireless connection procedures. Due to these lengthy setup requirements, scenarios involving casual interaction from first-time users are rare.

EICS 2014, June 17–20, 2014, Rome, Italy.

ACM 978-1-4503-2725-1/14/06...$15.00.
http://dx.doi.org/10.1145/2607023.2610276

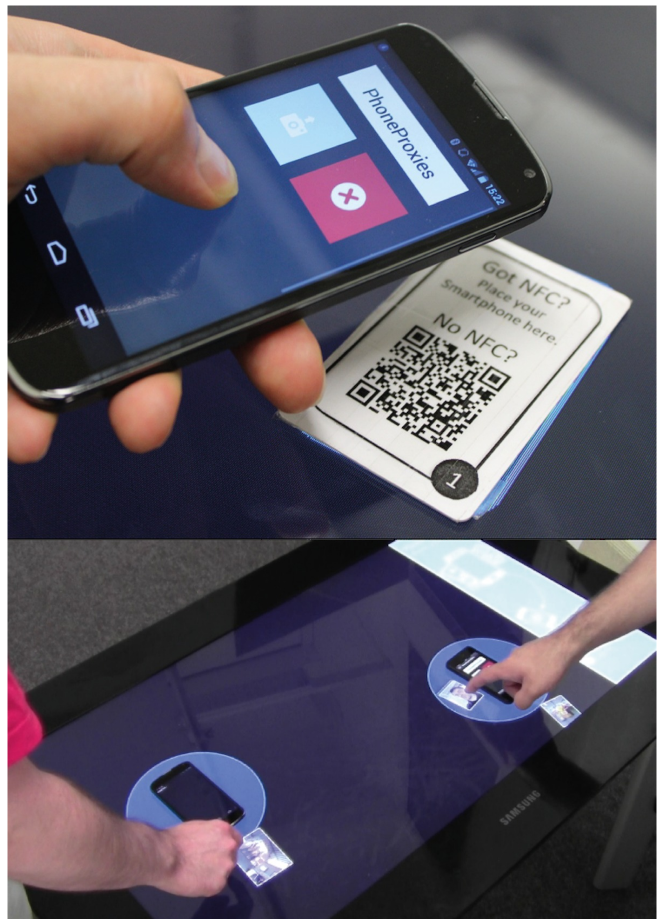

Figure 1. Connecting a mobile device to an interactive tabletop through a *Phone Proxy* (top). Transferring media between multiple phone proxies (bottom). Numbers on cards are used to visually identify the different tags.

Our approach to this problem is called *Phone Proxies*, shown in Figure 1, which leverages the NFC reader integrated in more and more mobile devices in conjunction with the optical marker tracking available on many interactive surfaces. Devices access custom URLs embedded into the NFC tags, thereby allowing association between devices placed on the tags and the tags' position on the tabletop. For complex scenarios - where a custom app is still required - the NFC tag can also automate a major part of the app installation process and consequently minimize the user's initial effort.

We motivate this approach through two usage scenarios. In the first scenario, called "pickup", the user wants to transfer media shown on the tabletop surface to their own mobile device. In the second scenario, called "share", the user wants to transfer selected media from their own device to the tabletop and show it on the surface (with the additional option of later using this media in the "pickup" scenario of another user).

Numerous researchers have investigated the topic of establishing a connection between mobile devices and interactive tabletops. Most related work relies on a combination of optical detection of device presence on the surface with a secondary data channel for disambiguation. For example, Wilson et al. use infrared or visible light flashes [12] generated by the device's screen or IrDA port. Echtler et al. use Bluetooth signal strength information [5] in conjunction with knowledge about the receiver's antenna sensitivity, while Schmidt et al. present PhoneTouch which relies on acceleration data generated by touching the surface with the device [8]. More recently, Boring et al. [2] have focussed on connecting phones and public displays by means of QR (quick response) codes detected by the device's camera. However, all of these approaches require the prior installation of a custom application on the phone in order to access the various hardware sensors or output ports. While one variant of Phone Proxies still also requires a custom app, care is taken to reduce the user interaction required for installation to an absolute minimum.

Employing NFC tags for simplifying the connection process between mobile devices and larger displays in general has also been a topic of research. Several researchers have used a large grid of individually encoded NFC tags mounted behind a larger screen for coarse position sensing. Seewoonauth et al. have investigated this approach with a laptop display [9], while Broll et al. applied this concept to a large projection screen [3]. A broader overview of these techniques involving multiple tags and/or readers can be found in [4].

There is surprisingly little work, however, which attempts to directly merge interactive surfaces with NFC interaction. MobiSurf by Seifert et al. [11] uses a technique similar to PhoneTouch for coupling mobile devices with an interactive surface and uses NFC for direct data exchange *between* the mobile devices (similar to Android Beam[1]), but does not integrate the tags with the surface. EPawn, a French company, presents a flatscreen which claims to integrate NFC-like sensing functionality directly into the display[2] but provides little detailed information. A similar product, the "Dynamic NFC Screen" from think&go[3] also integrates individual NFC tags at fixed screen locations, but does not offer a way to dynamically move the mobile device during interaction.

IMPLEMENTATION VARIANTS

We present three implementations variants for Phone Proxies, each with their own specific drawbacks and advantages. We will discuss each of these implementations in the context of our "pickup" and "share" use cases.

All variants have been implemented on the Nexus 4 Android smartphone and the iPhone 5 (where applicable), a Samsung SUR40 (PixelSense) interactive tabletop system and credit-card-sized "Mifare Ultralight C" NFC tags (see figure 2). The software running on the SUR40 is written using the MT4J framework in combination with the Surface2TUIO adapter. The NFC tags carry a optical marker for the PixelSense tracker on their bottom side. When such a marker is detected by the SUR40 sensor, a halo is shown around the tag's position which acts as a "drop zone" in the pickup scenario and in which the shared images appear in the share scenario (see also figure 1). Data in the NFC tags is formatted according to the NDEF standard [7].

Bluetooth Tags

This implementation is based on NFC tags which contain NDEF "Bluetooth Out-Of-Band Pairing" data, storing the Bluetooth hardware address of the host device. When such a tag is read by the mobile device, a connection is opened to the stored hardware address and the Bluetooth pairing process is initiated. For security reasons, this process requires the user to enter a 4-digit PIN code which is printed on top of the proxy tag and which has been pre-defined on the host side. Although the NDEF standard also allows the tag to directly store the required encryption data and perform the pairing process without further user interaction, this feature is currently unsupported on Android.

Depending on the usage scenario, it may be desirable to differentiate between multiple tags which can be in use simultaneously. Since the only identifying information the standard allows to be stored in the NFC tag is the Bluetooth hardware address, this requires the use of one separate Bluetooth adapter per tag on the host side. However, as an USB Bluetooth adapter is currently priced at about 5 US$, this is possible with moderate additional cost even for a larger number of tags.

Once the Bluetooth connection has been established, the host can now initiate a transfer of arbitrary media data

[1] https://developer.android.com/guide/topics/connectivity/nfc/nfc.html

[2] http://www.epawn.fr/products-2/

[3] http://www.thinkandgo-nfc.com/?page_id=472

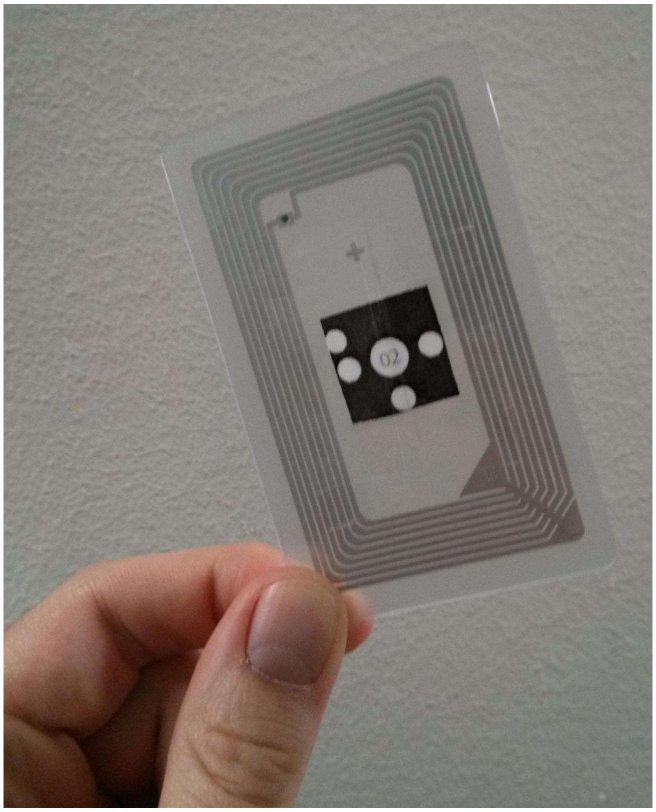

Figure 2. A *Phone Proxy* tag (bottom view). The optical PixelSense tag is glued to the tag's center, while the NFC antenna and the NFC chip itself (top left) are visible through the transparent plastic of the tag.

to the device by means of the *OBEX Push* protocol [1], thereby fully supporting the "pickup" scenario on nearly every Bluetooth-enabled device. Although the Bluetooth/OBEX standards also specify the more complex *OBEX FTP* protocol which would allow the host to automatically browse and retrieve stored media on the mobile device in the "share" scenario, very few devices actually implement this protocol due to security concerns. Should a suitable implementation be available (either natively on the device or by means of a third-party app such as BlueFTP[4]), this second scenario can also be supported.

Custom App

The second implementation of our concept is the most complex, but also most flexible one. Advanced use cases will require additional functionality on the mobile device which can only be achieved through a custom app. However, installing such an app usually is a cumbersome process for the user which involves opening the app store, searching for the correct app and performing the actual installation.

[4]https://play.google.com/store/apps/details?id=it.medieval.blueftp

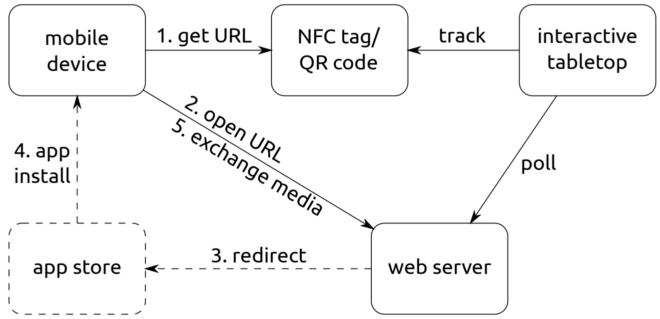

Figure 3. Data flow in an application using Phone Proxies. Steps 3 and 4 are only relevant for the app-based implementation variant.

We accelerate this process by using pre-defined URLs integrated with the server infrastructure shown in figure 3. The URL pointing to this web server is encoded as a standard NDEF message on the NFC tag. When the URL from the NFC tag is scanned for the first time, it will automatically open in the mobile browser. The web server detects this special case through lack of additional POST data and redirects the mobile device to a URL starting with `https://play.google.com/...`, thereby directly launching the Android app store page for the custom app. The user then has to perform three actions in total: tap "Install", tap "Confirm" (for app permissions) and re-scan the NFC tag. As the final action (scan the tag again) is not self-evident and many users will just start the app directly after installation, this case triggers a message asking the user to re-scan the tag.

For devices without NFC support, it is possible to use a QR code on top of the tag and access the URL using a barcode app. To also support iOS devices, the web page at the initial URL could detect the browser type and redirect either to the correct iTunes store page or to a custom URL scheme which will then launch the app.

When the tag is scanned for the next time, the custom app will now start instead of the browser. This functionality is achieved by means of an URL filter registering the app as default handler for all URLs starting with the address of our local web server. At this point, our sample app implementation will then automatically push the three most recent images from the mobile device to the server without requiring any further user interaction for the "share" scenario. Obviously, this is a security issue and is only meant to demonstrate the capabilities of a custom app approach. In a possible real-world usage, the user may (pre-) select a certain folder or album, from which images - when placing the mobile device on to the NFC tag - are automatically pushed to the server. In addition, the app will monitor the server-side directory for images that have not been uploaded by the app itself and, if such images are detected, download and display them on the mobile device to support the "pickup" sce-

Table 1. Feature matrix for different implementation variants of Phone Proxies.

Implementation variant	*supported scenarios*	*support for multiple tags?*	*OS support*	*Requirements (mobile device)*	*Interaction for first-time setup*	*Interaction for normal usage*
Bluetooth Tags	Pickup[1]	yes[2]	Android	NFC, Bluetooth	PIN entry	-
HTML5 Sharing	Pickup, Share	yes	Android, iOS[3]	Internet access	-	image selection
Custom App	Pickup, Share	yes	Android, iOS[3]	Internet access	confirm install	-

[1] "Share" scenario only supported using third-party software
[2] extra Bluetooth dongle per tag required
[3] iOS support uses QR codes instead of NFC tags

nario. The data flow for this implementation variant is illustrated in Figure 3 involving steps 1 to 5.

To avoid having to upload multiple images over the potentially slow and costly mobile data connection of the device, our app provides optional support for the automatic connection to a local wireless network. The URL encoded in the tag can contain additional credentials (network name and passphrase) for a WPA2-secured WLAN. If these credentials are provided, the app will automatically enable the device's WLAN adapter, connect to the network and restart the synchronization process with the server.

HTML5 Sharing

The third - and most promising - implementation of Phone Proxies makes use of built-in functionalities of modern mobile devices and also relies on access to a web server like the previous variant. The stored URL points to a web page containing an HTML5 file upload form, accessed by one large touch-friendly image button. When this button is tapped, a file selection dialog opens, allowing the user to choose which files to share. Afterwards, the upload process will start immediately. The URL in every tag also contains an unique ID as additional parameter which is used to distinguish the different devices and to select the correct upload directory on the server side. When new images appear in the directory associated with a tag, they are displayed next to the tag on the interactive surface. For the "pickup" scenario, media dragged onto the tag's halo on the surface are placed into the tag's directory on the server and pulled onto the webpage by a custom JavaScript program. The user can then save individual images to the device using the browser's built-in dialog.

This implementations also supports older devices without an NFC reader or iOS-based devices by using additional QR codes printed on top of the proxy tag. Both variants, NFC and QR, encode the same URL. Scanning the code with a suitable app before placing the device on the tag will also direct the browser to the HTML5 page, thereby enabling a similarly seamless interaction scenario as with NFC-equipped devices. The data flow for this implementation variant is illustrated in Figure 3 involving steps 1,2 and 5.

To be used in this scenario, the user's device is only required to support web access and either NFC or QR capability. The possible field of application is limited, not by the the device's technical specifications, but its browser's support for different actions. We have chosen the example of image sharing, as it is supported by both Android's and iOS's stock browser. For security reasons most platforms do not support browser access to other providers such as contacts or calendar. However, this data may be accessible by integrating cloud services - e.g.: *Google Calendar* - into the web client.

Summary & Discussion

In table 1, we summarize the different features of our implementation variants for Phone Proxies.

The Bluetooth-based implementation is somewhat limited in terms of use cases, supported operating systems and flexibility. However, as Bluetooth adapters are nearly ubiquitous in today's computers, this variant is perhaps the easiest to set up. When a simple PIN such as "0000" is chosen and printed on top of the proxy tag, the one-time setup effort for each individual user is also quite small. For scenarios involving only one-way transfer of data to the mobile device, this variant is consequently a valid option.

The HTML5-based variant has the considerable advantage of not requiring any initial setup by the user. As soon as the tag is scanned, the corresponding web page will open on the mobile device and enable the user to exchange media with the tabletop system. Although some interaction is required to choose the media on the mobile side or save shared media to the device, this variant perhaps offers the best balance between complexity and features. However, it requires a high-bandwidth internet connection to be available on the mobile device.

For more complex scenarios, the implementation variant centered around a custom app is likely the best choice. Although a small amount of user interaction is required for the first-time installation of the app, all further interaction can be automated. In particular, it is also possible to automatically switch connectivity to a local wireless network, thereby increasing transfer speed and reducing bandwidth consumption on the mobile data link.

EXPERT REVIEW

In order to evaluate our concept and its prototypical implementations, we performed a short expert review with colleagues and students from our local university department. We chose to focus on the two advanced implementation variants (HTML5 & app) due to their higher flexibility and broader OS support. We collected opinions & feedback through semi-structured interviews from 4 participants using their own, unmodified devices (1 x Galaxy S3 and 3 x iPhone 5). Participants exchanged and discussed pictures with the interviewer who was using a Nexus 4 device.

All participants (P1-P4) expressed highly positive opinions of our concept, but also suggested enhancements which we aim to address in future work. For example, one participant expressed concerns over privacy ("Are these pictures being posted on Facebook right away?", P1) and a desire to explicitly delete pictures from the surface again. Two other participants requested a better visualization of pictures' origins, either through matching color-coded frames and haloes (P2) or through animations during transfer (P3). Interestingly, only one of the participants (P4) was bothered by the manual interaction required for sharing media in the HTML5-based variant. This suggests that the other reviewers were fine with this kind of interaction as it provides explicit control over which pictures are shared, as opposed to the automatic sharing afforded by the app-based variant. Finally, P4 also suggested to visualize information about albums or folders on the mobile device as segments of the halo.

In summary, we observe that nearly all feedback from our reviewers focused on the interactive surface and not on the mobile devices. At the very least, this suggests that our goal of unobtrusive sharing was achieved on the mobile side.

CONCLUSION & FUTURE WORK

We have presented Phone Proxies, a technique for effortless connection between mobile devices and interactive tabletop systems. By using the NFC reader increasingly available in mobile devices, various scenarios such as retrieving and sharing media on the tabletop can be supported. For more complex use cases, the automated installation of a custom app is also possible. Source code for tabletop and mobile app is available at https://github.com/alexanderbazo/portals.

As future work, we will also implement and test our third, app-based implementation variant with iOS. While some minor modifications to the URLs are required (redirect either to iTunes or to a custom URL scheme such as "phoneproxy://"), the general concept is still applicable. Some additional directions for future investigation are outlined below.

We intend to evaluate the quality and acceptance of our prototypical implementation by user tests. As our project emphasizes an easy-to-use and seamless approach to share content between personal mobile devices we suggest to study its capabilities in a natural environment. The utilized tabletop computer is easy to move and can be placed in a public and well-frequented place. We are planning to let random passers-by use both, the native app and the HTML5-based variant, to share images from their smartphones. For this purpose, we will modify the application to prevent accidental sharing of personal data. We hope to gain a deeper insight into how usable the proposed technique is and how well it is accepted by actual users.

In order to preserve users' privacy when they are no longer present at the table, it would be desirable to delete all shared data from the server when the mobile device is removed from the proxy tag. Removal of a device can be detected using the optical tracking system: as the device is noticeably larger than the tag, the secondary object outline detected in addition to the tag itself will rapidly shrink in area when the phone is picked up, thereby triggering a disconnection event for the tag in question. The same also applies if the tag is picked up together with the phone. Techniques such as Shield&Share [10] might also be employed here to provide users with better control over privacy-related issues.

Alternatively, it is also be possible to apply a sticky silicone coating to the top side of each proxy tag which will temporarily affix the tag to the phone on contact. While it should still be easily possible to peel off the tag from the phone's back, the tag will have to be picked up with the phone first. This action would remove the marker from the surface, too, thereby also triggering media deletion. If a coating with sufficiently strong adhesion is applied to both sides of the tag, it might even be possible to attach the entire phone-tag-combination to a vertical interactive surface, similar to the concept of *Vertibles* introduced by Hennecke et al. [6].

REFERENCES

1. Bluetooth SIG. Bluetooth Core Specification 4.0. https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737, 2012. Accessed 2013/07/31.
2. Boring, S., and Baur, D. Making public displays interactive everywhere. *IEEE Computer Graphics and Applications 33*, 2 (2013), 28–36.
3. Broll, G., Reithmeier, W., Holleis, P., and Wagner, M. Design and evaluation of techniques for mobile interaction with dynamic NFC-displays. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, TEI '11, ACM (New York, NY, USA, 2011), 205–212.
4. Broll, G., Vodicka, E., and Boring, S. Exploring multi-user interactions with dynamic NFC-displays. *Pervasive and Mobile Computing 9*, 2 (Apr. 2013), 242–257.
5. Echtler, F., Nestler, S., Dippon, A., and Klinker, G. Supporting casual interactions between board games on public tabletop displays and mobile devices. *Personal and Ubiquitous Computing 13*, 8 (2009), 609–617.
6. Hennecke, F., Wimmer, R., Vodicka, E., and Butz, A. Vertibles: using vacuum self-adhesion to create a tangible user interface for arbitrary interactive surfaces. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, TEI '12, ACM (New York, NY, USA, 2012), 303–306.
7. NFC Forum. NFC Data Exchange Format. http://www.nfc-forum.org/specs/spec_list/, 2006. Accessed 2013/07/31.
8. Schmidt, D., Chehimi, F., Rukzio, E., and Gellersen, H. PhoneTouch: a technique for direct phone interaction on surfaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, ACM (New York, NY, USA, 2010), 13–16.
9. Seewoonauth, K., Rukzio, E., Hardy, R., and Holleis, P. Touch & connect and touch & select: interacting with a computer by touching it with a mobile phone. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '09, ACM (New York, NY, USA, 2009), 36:1–36:9.
10. Seifert, J., Dobbelstein, D., Schmidt, D., Holleis, P., and Rukzio, E. From the private into the public: Privacy-respecting mobile interaction techniques for sharing data on surfaces. *Personal Ubiquitous Comput. 18*, 4 (Apr. 2014), 1013–1026.
11. Seifert, J., Simeone, A., Schmidt, D., Holleis, P., Reinartz, C., Wagner, M., Gellersen, H., and Rukzio, E. MobiSurf: improving co-located collaboration through integrating mobile devices and interactive surfaces. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces*, ITS '12, ACM (New York, NY, USA, 2012), 51–60.
12. Wilson, A. D., and Sarin, R. BlueTable: connecting wireless mobile devices on interactive surfaces using vision-based handshaking. In *Proceedings of Graphics Interface 2007*, GI '07, ACM (New York, NY, USA, 2007), 119–125.