Demo: Mobile Plus: Mobile Platform for Transparent Sharing of Functionalities Across Devices

Sangeun Oh Hyuck Yoo Daelyong Jeong Sooyoung Park
Duc Hoang Bui Sungsoo Moon Insik Shin
School of Computing, KAIST, South Korea

1. INTRODUCTION

Smart devices rapidly grow in popularity and diversity from smartphones and tablets to smart TVs and wearables. At the same time, mobile app ecosystem continues to grow and mature rapidly, offering a wide variety of functionalities, such as SNS, shopping, entertainment, and healthcare. Such trends present that there must be exciting opportunities if those smart devices are incorporated together. For example, it is likely to be insecure to carry out a payment service when purchasing goods via a shopping application on a public device (e.g., smart TV). However, as illustrated in Figure 1, we can safely enjoy shopping on the public device if the payment is processed on a user's personal device (e.g., smartphone) and then the result is returned to the shopping application. In a similar way, we can also enjoy motion sensor applications on a smart TV without sensors if a smartphone shares its sensor functionality.

In order to bring them into reality, mainly two design requirements must be considered. Firstly, additional engineering efforts for cross-device interaction will strongly discourage needs and accessibilities of such features. Therefore, the interaction between the devices has to be transparent; app developers should be able to use functionalities from other devices as if they were working on the same device. When supported, this transparency allows existing applications to employ functionalities from other devices even without modification. Secondly, it is desirable to support as broad range of functionalities as possible for cross-device sharing. Thus, it is preferable to adopt a functionality class-agnostic mechanism that is general enough to support both existing and emerging functionalities.

We present a novel platform M+ (Mobile Plus), an extension of Android that allows transparent sharing of both application and system functionalities[1] cross multiple mobile devices, in order to fulfill the two design requirements. M+ extends one of the Android's design principles a step further. Android allows an application to invoke a method on a different process as if the method were inside the caller process boundary. To this end, Android takes care of all underlying inter-process communications (IPC) with its Binder IPC mechanism. M+ leverages and extends this mechanism to allow a method on a different device to be called as if it were inside the same process boundary. M+ mediates cross-device sharing at the platform layer, intercepting Binder messages (called *Parcels*) and forwarding them to different devices. Since Binder is used for IPC communication between any kind of processes (either applications or system services), it naturally suits intercepting functionality class-agnostic requests to support transparent cross-device sharing of both application and system functionalities.

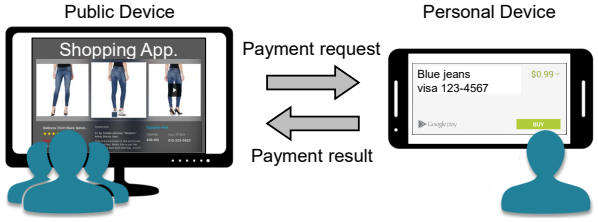

Figure 1: Payment scenario example

M+ is not the first work to share functionality between mobile devices. As an example, Rio [1] proposed a kernel-level solution that provides cross-device sharing of I/O functionalities. It can be achieved by intercepting file operations for I/O device drivers in the kernel. On the other hand, M+ adopts a platform-level solution that intercepts IPC messages. It allows a great opportunity to share a wider range of functionalities, including application as well as I/O functionalities.

2. DEMONSTRATION

In this demo, we prove M+ concept with a prototype implementation on Nexus 6. Our demo setup consists of two Nexus 6 devices which are connected through the same WiFi AP. One is a client device that demands functionalities from a different device, and another is a server device that provides functionalities requested by the client. We demonstrate that M+ allows existing applications on different devices to successfully share application functionalities such as Facebook login, Google market payment, and PDF browsing, as well as system functionality such as sensor. During the demo, guests can arbitrarily experience some of these functionalities. In addition, the demonstration will be accompanied with a short demo video which shows how other functionalities can be shared across devices.

3. REFERENCES

[1] A. A. Sani, K. Boos, M. H. Yun, and L. Zhong. Rio: A System Solution for Sharing I/O between Mobile Systems. In *Proc. ACM MobiSys*, 2014.

[1]In this paper, we refer application functionality to as a service, content, or resource which is provided at the application layer, such as in-app payment and SNS login. System functionality refers to a service, content, or resource which Android platform provides, such as camera and sensors.

MobiSys'16 Companion June 25-30, 2016, Singapore, Singapore

ACM ISBN 978-1-4503-4416-6/16/06.
DOI: http://dx.doi.org/10.1145/2938559.2938568