INSTITUTIONEN FÖR DATAVETENSKAP | LUNDS TEKNISKA HÖGSKOLA

EXAMENSARBETE Comparison of Web and Native approaches to implementing Distributed User Interfaces for Android
STUDENT Joakim Ahle
HANDLEDARE Sven Gestegård Robertz (LTH), Patric Lind (Sony Mobile Communications)
EXAMINATOR Klas Nilsson (LTH)

A comparison of frameworks for building Distributed User Interfaces

POPULÄRVETENSKAPLIG SAMMANFATTNING **Joakim Ahle**

In 2014, Google launched an operating system for wearable devices (Android Wear) and a TV dongle (Chromecast) that we can stream media to. When a smartphone app supports both of these devices, a distributed user interface (DUI) can be created, but how does it compare to other approaches?

Introduction
In our digital life, we interact with many different devices. Many of our devices are capable of doing the same thing, but we usually prefer to use them differently. This is because of the device's different characteristics.

When viewing media, we usually do it on a TV, since they have a large screen. Because of the distance between the screen and the viewer, the TV is usually operated with a remote control.

The smart wrist watch is ideal for showing the user a small amount of information and for making simple responses to that information.

By making it possible for the user to use these devices together in the same context, in a way that utilizes the strength of each device, we have by definition created a distributed user interface.

Implementing DUIs is a complex process where programmers face many challenges doing so. The Android Wear and Chromecast frameworks provide developers with solutions to several of these challenges, but until now it has been unclear exactly which ones.

The features and limitations in Google's frameworks have been analyzed and compared to a Web based approach called Distributed Web Components. When a user loads a web page containing Distributed Web Components, each component can be sent to another device through a central server, which gives a similar conceptual behaviour.

Experiment
The analysis Google's frameworks was conducted as a case study where a media player was implemented. The media player worked as any other media player until the user chose to distribute the UI. If the respective devices were available, the media playback was moved to the Chromecast and the media controller was moved to the smart watch.

When Sony was experimenting with Distributed Web Components, a similar app was written. That application was used to compare the implementation aspects of the different approaches.

Some of the most important challenges developers face when implementing this type of functionality is keep the devices synchronized. Current play times, media files and user preferences have to be the same on all of the devices at every moment.

Conclusions
Distributed web applications can be used by web applications, in contrast to Google's frameworks that requires a native implementation. It has the advantage that you don't have to implement one version of the app for each mobile OS (Android, iOS etc.). The same app works on all of these devices - which saves us implementation time. Google's solution works with Android and it's limited (though wide) user range. The native application can utilize all of the UI components of Android and it's actually quite easy to extend an existing application into supporting Android Wear and Chromecast. However, we do meet some constraints. We cannot properly handle devices coming online and offline during runtime - So all the sudden the watch might run out of battery and we can no longer properly pause the movie. The conclusions presented should help developers choose a concept to implement their applications with (each concept have their strengths) and help Distributed Web Applications overcome its limitations.