A Platform Enabling 2nd Screen Functionality for Mobile Applications

Aleksander Kovačić, Jože Guna
LTFE, Faculty of Electrical Engineering,
University of Ljubljana, Tržaška cesta 25, 1000 Ljubljana
E-mail: aleksander.kovacic@ltfe.org

Abstract

We address the problem of presenting an arbitrary content from the mobile device on additional screen. A television is usually considered as such example but it is also possible to use other devices as an additional or second screen. A working platform that enables 2nd screen functionality is presented and compared to some of the most known related technologies, such as DLNA, AirPlay, Anymote and specific solutions from TV manufacturers.

1 Introduction

In the last years we have seen a tremendous growth of smartphones and tablet computers in the mobile market. Important factors that enabled this growth are platform specific digital distribution platforms, where users can buy and download different kinds of applications or multimedia content (music, video, books, etc.). 2nd screen applications have proved to be very popular among users, which is not very surprising since researches have showed that more than 50% of people are using a mobile device for browsing the Internet while watching television [1]. The term 2nd screen usually refers to the kind of application, which runs on a mobile device and provides additional information about the current show on the television to the user [2]. Applications can automatically identify the show, which is currently playing, by different means (audio watermark detection, time schedule, proprietary connection to the broadcasting system, etc.). In addition to providing additional information about the show to the user, these applications can also provide interactivity (voting, polls, gambling, etc.), social networking (rating, chatting, sharing), advanced advertising, and other similar services.

TV manufacturers are now following the business model from the mobile domain. Connected television sets and set-top boxes offer applications and additional content to the user through manufacturers’ digital distribution platforms. For this reason the connected TVs are also called Smart TVs. Increased interconnectivity between smart television sets and mobile devices has lead to another kind of 2nd screen applications where the television is considered as the second (additional) screen for the mobile applications.

In this research paper we present a platform, which enables mobile applications to use Smart TVs (or any other device with Internet connectivity and web browser) as the 2nd screen where applications can present arbitrary content in HTML format. A pilot application has been developed, which uses presented 2nd screen platform (Figure 1). In this tablet application the user searches and selects a video content, which is than played on the second screen.

The main advantages of our solution are:

- Easy integration of 2nd screen functionality into mobile applications for various operating systems.
- Mobile applications can use any device with a web browser as a second screen.

![Figure 1: Pilot application using 2nd screen platform](image)

The rest of the paper is organized as follows. The architecture and belonging technical implementation is presented in Chapter 2. In Chapter 3 similar technologies (DLNA, AirPlay, Anymote and specific solutions from different TV manufacturers) are compared to our solution. Pros and cons for each technology compared to our solution are presented in separated subchapters. We conclude in Chapter 4 with a short discussion and future plans.

2 Architecture and technical realization

In order to use the mobile application’s 2nd screen functionality, the user must first open the corresponding application on the television and login with the same...
credentials as on the mobile application. After that, the mobile application can use the television as the second screen.

The mobile device and the 2nd screen (usually television) are not connected directly in our solution, but instead they communicate through the server (Figure 2). The need for the server derives from the demand that television applications are realized as web pages, which reside on publically accessible web servers. The server maintains open connections with the mobile application and the 2nd screen browser and it pushes requests from the mobile application to the 2nd screen browser. The mobile application presents the information on the second screen in the form of HTML, because TV applications are basically web pages that run inside web browsers. The content presented on the second screen can be dynamically changed by means of sending Javascript commands to the browser on the television.

Because all users are using the same server, different devices are sorted in user groups based on login credentials. All the devices, which are logged in with the same user name, belong to one user group. This is necessary for the server to differentiate between user devices, so that it sends the requests from the mobile application to the 2nd screen device that belongs to the same user. It is also possible for the user to use more than one 2nd screen device. The mobile application can present its content on multiple 2nd screen devices, which are part of the same user group.

2.1 Server architecture

The main requirement, which must be fulfilled by the server, is the capability to maintain the connection with the 2nd screen, so as to be able to push data to it, when requests from the mobile application arrive. It is possible to satisfy this requirement with traditional web servers (such as Apache web server), but this solution does not scale well in our example. Traditional web servers create a new thread for every maintained connection and when the number of connections increases the server resources are exhausted very quickly. To overcome this weakness we chose the Javascript server Node.js [3] with integrated NowJS framework [4] (Figure 3). Node.js is an event driven server where all requests are managed by a single thread, which fires off asynchronous events for requests. Multiple concurrent connections for event driven servers do not represent much burden as they do for the traditional web servers, what makes them more scalable [5, 6]. NowJS framework maintains open connection between the server and the client with the help of Socket.IO library [7] and provides a simple API for calling javascript functions on the client browser from the server (and vice versa).

2.2 Mobile application architecture

Mobile application is realized (Figure 4) as a native application on top of mobile OS (iOS, Android, etc.). The application uses client side Javascript library of the NowJS framework for connecting to the server and sending requests.

2.3 2nd screen application architecture

The application, which runs on the 2nd screen device, is a web page, so basically every device with a web browser can be used as the 2nd screen (Figure 5). This web page also uses client side Javascript library of the NowJS framework for maintaining open connection with the server. The server pushes requests, which contain changes to the web page to the 2nd screen application over the open connection.
3 Comparison with existing similar technologies

2nd screen is a popular concept and many technologies that are similar to our platform already exist. In this chapter we compare our solution to some of the most known related technologies: DLNA, AirPlay, Anymote and specific solutions from TV manufacturers.

3.1 Comparison with DLNA

The Digital Living Network Alliance (DLNA) is a non-profit trade organization started by Sony in 2003. DLNA defines interoperability guidelines to enable sharing of digital media between various consumer devices such as computers, printers, cameras and cell phones over local (wireless or cable) network [8]. DLNA is designed to act as a bridge between different consumer devices from various manufacturers, so a user can easily watch a film from the PC on the TV, play an MP3 file from a smartphone on the stereo, or send photographs from the camera to the printer. DLNA guidelines are built upon existing public standards, but the guidelines themselves are only available for a fee.

At first glance it seems that DLNA provides the same functionality as our 2nd screen platform with the benefit that DLNA works completely over local network and it does not require any Internet connectivity. But a huge difference and the main advantage of our platform over DLNA is, that applications can use 2nd screen to present arbitrary content, which they format in HTML format. For example the 2nd screen can be used to present video, pictures and some additional information simultaneously, as opposed to DLNA, which can only stream video, audio, or images to the other device one at a time. DLNA protocol is also difficult to implement in applications and a fee is required to obtain the DLNA guidelines.

3.2 Comparison with solutions from TV manufacturers

TV manufacturers are starting to realize the added value of the 2nd screen functionality in television sets. They are developing frameworks, which enable mobile applications to connect to televisions over local network and transmit control commands, essentially acting as a remote control. As an example, applications can navigate to a particular menu on the television, change the channel, volume, source, etc. or obtain the information about the current state of the television and user settings. One of the most advanced frameworks is the jointSPACE Open Source project [9], which is used in Philips TV displays. This framework also enables applications to create graphics on the TV with the DirectFB graphical interface [10]. This is similar to our solution, however our solution is easier to implement in the mobile application, since it uses HTML to describe the content that should be displayed. Our solution is also more portable because it is not limited to a specific television manufacturer. Any television (or other device) can be used as a 2nd screen device, regardless of the manufacturer, as long as it provides the possibility to browse Internet with web browser. Because there are some differences between different web browsers they must be taken into account when presenting web pages on the 2nd screen from the mobile application.

3.3 Comparison with AirPlay

AirPlay [11] is a proprietary technology from Apple that enables streaming of video, audio and photos between Apple's devices and software over local wireless network. The audio-streaming part of the AirPlay protocol stack is available to be implemented also by other companies. AirPlay is similar to DLNA in the concept of streaming multimedia content between devices, except that it is limited to only Apple's devices. In addition it also provides 2nd screen functionality, which is not present in DLNA. AirPlay Mirroring enables any content that is displayed on the device (iPhone, iPad, Mac) to be duplicated on the Apple TV. Applications can also create arbitrary graphics on the Apple TV, which is similar to the jointSPACE project. But as with jointSPACE also with AirPlay the downside is portability since only Apple devices can use this technology. The benefit of our presented solution is also in easier implementation of 2nd screen functionality in mobile application.

3.4 Comparison with Anymote

Google provides it's own Anymote Protocol [13, 14] that applications on a mobile device use to communicate with Google TV. Anymote Protocol is a messaging protocol based on a client-server model and enables applications to act either as an independent remote TV controller or they can add more functionality to a Google TV application. This protocol is not used for streaming any kind of multimedia content between devices. Anymote service implementation on Google TV only supports one-way communication. However it is possible to obtain the Google TV device IP address during the device discovery phase of the pairing process used by Anymote, and use it to setup a custom two-way communication between the remote application and Google TV. Anymote Protocol can be relatively easy implemented in the Android mobile applications but official libraries for other platforms don't exist. Similar to the AirPlay, Anymote Protocol is only used to connect to Google TV, which is a huge restriction.

4 Conclusions and further work

In this research paper we have presented a platform for providing 2nd screen functionality to mobile applications. We have described the architecture and technical insight of the platform and compared the platform to some of the most known related technologies: DLNA, AirPlay, Anymote and solutions from TV manufacturers. A pilot application has also
been developed on top of the presented platform in order to test and proof the viability of the solution. 

Future work includes upgrading the platform from the pilot stage to the production stage. The 2nd screen functionality will be integrated into additional applications. Users will evaluate these applications and we expect to receive positive and useful feedback about possible improvements and new ideas, which would also reflect on the 2nd screen platform itself.

Acknowledgement

The operation that led to this paper is partially financed by the Slovenian research Agency, grant No. P2-0246.

References