On the Technical Challenges for Enabling Multiscreen Media Consumption Applications via Hybrid Web Technology

HEIKO DESRUELLE [1], *MARTIN LASAK* [2], *FRANK GIELEN* [1]
[1] *Department* of Information Technology
Ghent University - iMinds
BELGIUM
{heiko.desruelle,frank.gielen}@intec.ugent.be
[2] *Fraunhofer Institute for Open Communication Systems*
GERMANY

Abstract

Personal computing has evolved towards mobile and embedded devices. This trend enables users to access their media content via a variety of devices, ranging from tablets, to smartphones, smart television sets, and recently even in-vehicle infotainment systems. Platform-independent and multi-screen media consumption highlight some of the new innovative applications opportunities for mobile computing. Nevertheless this enormous market potential, the innovation is heavily constrained by a number of technological challenges. In this paper, we discuss the main issues for seamless multi-screen media consumption applications. Moreover, we present the webinos platform as a candidate solution for multi-device ubiquitous applications, based on open and standardized Web technology.

Keywords: multi-screen applications, rich internet applications, ubiquitous web, HTML5

1 Introduction

The Internet is drastically changing the way people consume their multimedia content. The diversity of Internet-connected devices is rapidly increasing from traditional desktop and laptop PCs, to more ubiquitous devices such as tablets, smartphones, home entertainment systems, and even in-vehicle infotainment (IVI) setups. Ubiquitous computing should enable new and innovative media consumption applications. In this connected eco-system, personal content should be available at any time, anywhere, and from virtually any device. Moreover, multimedia applications should focus on providing a rich ubiquitous user experience, by enabling multi-device scenarios for remote control and multi-screen playback. Existing approaches, however, are often confined to proprietary technology, a limited set of supported operating systems, only local area network (LAN) support, etc. [1].

In this paper we elaborate on the technological challenges for enabling seamless multi-screen media consumption applications and propose a generic solution based on open and standardized technology. The remainder of this paper is structured as follows. Section 2 elaborates on the current technical challenges for creating seamless multi-screen media applications. Section 3 discusses the Webinos platform as a generic solution for creating multi-device media consumption applications. Section 4 evaluates the proposed solution via a prototype implementation. Section 5 covers a comparison with related work. Finally, our conclusion is drawn in Section 6.

2 Multi-screen Application Challenges

Today, application developers are facing various technical challenges when designing and implementing software for media consumption over multiple screens.

- **Compatibility:** The rise of mobile devices has heavily fragmented the application runtime eco-system. Developers need to take into account a wide variety of available hardware platforms and operating systems. E.g., many devices only support a specific set of programming languages, media codecs, proprietary communication protocols, etc. In order to reach a viable market share, developers often need to create and maintain a number of different implementations of the same application. This challenge needs addressing, as it only increases development costs and the applications' time-to-market.
- **Scalability:** Multiscreen application scenarios typically involve multiple devices accessing the same application. All participating devices will need to share a mutual state, allowing an action or event on one device to seamlessly propagate to the other devices (e.g., update their view). In result, scaling the number of participating devices quickly becomes a challenge due to the increasing synchronization communication.
- **Performance:** Media consumption applications are highly sensitive to user interaction delays, as users tend to quickly notice hiccups (e.g., delayed media playback when after clicking a control button). For multiscreen media consumption applications, this constraint turns out to be even more of a challenge, as events on one device will need to propagate to one or more remote devices, whilst still meeting a hard time constraint.

3 The Webinos Platform

The Webinos platform described in this section aims to generically enable multiscreen applications over a wide variety of devices (i.e., PC, mobile, home entertainment,

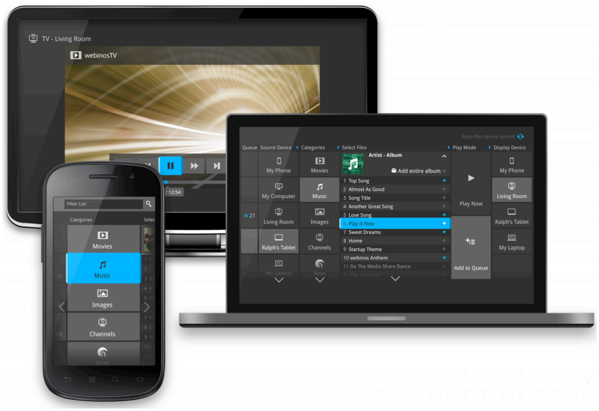

Figure 1: Multiscreen media discovery and playback

and IVI). Moreover, this platform aims to be scalable and performance efficient in terms of multi-device usage [2]. Webinos does so by leveraging the Web and its standardized technology as a generic application platform.

To ensure this broad range of device compatibility, Webinos applications are written as Web applications based on HTML (structure), CSS (styling), and JavaScript (logic). In result, applications can be accessed from virtually any Web-enabled device's browser. The browser communicates with the actual Webinos platform over a HTML5 WebSocket interface. This bidirectional communication interface enables Webinos applications to access native APIs, or to engage communication with remote devices. We refer the interested reader to [2] for a more elaborate discussion of the Webinos approach and the platform's internal structure.

4 Prototype Implementation

A prototype of the Webinos platform was implemented. The platform is built on top of Node.js [1], a server-side JavaScript runtime for Google's V8 engine. The Webinos platform's code is released as an open source project and available online. Moreover, a proof-of-concept multi-screen media consumption application was designed and implemented. The application enables the remote discovery and playback of various media types (i.e., audio, video, and pictures). With this application, media playback can be started and controlled from any personal device, regardless of the physical device on which the content is actually stored (see Figure 1).

In order to evaluate Webinos' ability to adequately address the challenges raised in Section 2, a Living Lab study was organized. A home-like setup was created for evaluating the application in a controlled environment (see Figure 2). Twenty external test users were selected and invited based on their persona matching. The test setup included two Android tablets, one HD television set, and two laptop computers. A set of multimedia files (images, video, music) was distributed across each of these devices Test participants were asked to arbitrarily discover remote content and start it's playback.

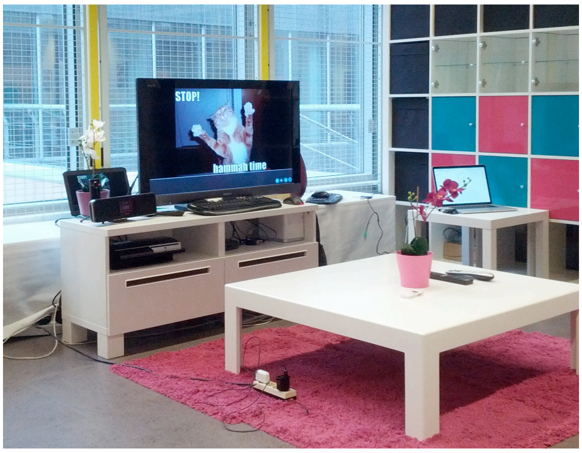

Figure 2: Living Labs application evaluator

5 Related Work

Various mobile application frameworks and operating systems aim to simultaneously combine features of multiple devices. Nevertheless, these solutions are often tightly bound to a vendor-specific set of devices, or proprietary platforms (e.g., the proprietary connected TV platforms, which enable second screen applications via smartphone devices). As a counter, the Munin framework and Gibraltar platform aim to broaden this scope with a more flexible peer-to-peer design for distributed mobile applications over the Internet [3].

6 Conclusion

In this paper, we presented the main technical challenges that developers are facing when creating multimedia applications for multiscreen consumption. Moreover, we present the webinos platform as a generic enabler for ubiquitous and multiscreen applications. Future work includes a detailed analysis of the conducted Living Labs evaluator.

Acknowledgments

The research leading to these results has received funding from the European Union's Seventh Framework Programme under grant agreement number 257103 (Webinos project).

References

[1] Desruelle, H. et al., Multi-device application middleware: leveraging the ubiquity of the web with webinos, *Journal of Supercomputing*, Vol. 66, 2013, pp. 4-20.

[2] Desruelle, H. and Gielen, F., Architectural Modifiability Considerations for Designing a Multi-device Web Application Platform, *Procedia Computer Science*, Vol. 19, 2013, pp. 895-900.

[3] Lyle, J. et al., On the Design and Development of webinos: A Distributed Mobile Application Middleware, Distributed Applications and Interoperable Systems, 2012, pp. 140-147.

[1] Node.js, http://nodejs.org/