Second Screen for HbbTV – Automatic application launch and app-to-app communication enabling novel TV programme related second-screen scenarios

Christoph Ziegler
Platforms for Broadcast Services
Institut für Rundfunktechnik
Munich, Germany
Email: ziegler@irt.de

*Abstract***—The article presents an open standards-based framework that enables bi-directional communication between web applications for TV and second-screen devices. It also provides mechanisms for the automatic launch of applications from the TV on the second screen. Due to its compliance to the HbbTV standard, the framework allows broadcasters to create novel interactive applications with direct programme relation potentially targeting millions of already deployed devices on the market.**

I. Introduction

Using a smartphone or tablet while watching TV has become a common habit today. Different studies investigated the parallel usage of secondary screens while watching TV. Böhm et. al. [1] found that 77% of the German internet users surf the web while watching TV. A study by Google [2] came to similar results for the U.S. TV audience. According to both inquiries, the most popular online activities on the secondary device are mailing, search, online shopping and social networking. What both analyses also spotted is that most interaction on the secondary device is not related to the TV programme. However, the Google study sees TV as a "main catalyst for search". Rycak [3] could show a temporal correlation between the broadcast of different TV shows and the page views of with regards to content related Wikipedia articles. Research conducted by Red Bee Media Ltd. [4] showed that 52% of the British smartphone owners have used their secondary device in order to get additional information on a programme.

The complementary use of secondary screens holds a substantial commercial relevance. This is indicated by the fact that "44% of the respondents use their secondary screen to find out more about brands or advertising" and "40% would be willing to receive offers or promotions on their smart devices based on products featured on TV". What the study also brings up is that only 1 in 5 interviewees have used a dedicated application for complementary second-screen usage, although 78% see the secondary screen as an appropriate tool to get engaged with a TV show [4]. This shows the great, yet untapped, potential for the programme-related companion applications.

II. The second-screen situation

A. Confusion about the term

An increasing number of services are available that promise what is often called a *second-screen* experience. In fact there is no common ground on what "second screen" actually means. This manifests, for example, in the difference between the explanations the English and the German Wikipedia [5] articles provide for second screen. The German article describes it to be the usage of a secondary device in order to get *additional information on the running TV programme* – in [1] a similar explanation can be found. Whereas the English entry [6] does not restrict the context to a broadcast TV programme. It describes it to be "[. . .] an additional electronic device (e.g. tablet, smartphone) that allows a content consumer to *interact with the content* they are consuming [. . .]" while this may be TV shows, movies, music, or video games. Steglich et. al. [7] propose a definition that emphasises the *contextual link* between primary and secondary screen that is used to access *supplementary information and services*.

This reveals that the term itself does not imply a certain feature set for a second-screen labelled application. Section II-B provides an overview on applications available today that offer their users additional value from the accompanying usage of their second-screen device.

B. Companion applications on the market and their features

A number of services are available today that fall under the scope of at least one of the definitions discussed above. One category consists of applications that offer users additional information on the TV programme. This includes applications of programme guide publishers like TV Guide and TV today that provide information on the programme content and the scheduling of different channels. Another category of applications extends the possibility to get additional information with the facility to get involved with other viewers e.g. to discuss about a programme. This is also referred to as *social TV* applications. Examples are GetGlue, Umami, ConnecTV, miso, IntoNow, Yap TV, Zeebox, Couchfunk, Zapitano, McCheckin. Furthermore, there are services that help users to find content of their interest – linear or non-linear, e.g boxfish, nextGuide, Fayve or Tweek TV. Another category of services offered by broadcasters allows viewers to get involved with a TV show. One example is the 4now application. It provides viewers of

978-1-4799-1412-8/13/$31.00 ©2013 IEEE

the British Channel 4 with the possibility to participate in polls or quizzes around a show. The German ARD group offered a web application accompanying the crime thriller Tatort which allowed the users to interactively investigate on who is the murderer.

Beside these groups there are services that do not relate to the content displayed on the TV screen but rather to the device. Some TV manufacturers provide applications for mobile devices that allow the users to control the native device functionalities of their TV like volume up/down, channel up/down or to share media files across the devices. Examples are Samsung Smart TV, Philips Net TV or LG Smart TV). Furthermore there are game consoles that provide applications for mobile devices offering the functionality to interact with the game on the TV screen (e.g. XBox SmartGlass, PlayStation Official App).

C. Common functional requirements and available solutions

All services described above require a certain link between the first and the second screen. This link may be contextual or physical. Contextual means that there is a semantic relation between the content on both screens, e.g. when additional content to a TV programme is displayed on the second screen. The contextual link can be established manually or automatically. The manual link is established by the user who can for example "check-in" for a certain TV show, or navigate to the appropriate menu item in the companion application. The automatic link requires the application to automatically determine what content is played back on the TV. This can be accomplished on the basis of audio fingerprinting [8] or with the help of a physical link.

A physical link provides a communication channel which the devices can use to exchange information. To establish a physical link, the devices need to support *discovery* and *connection* mechanisms. Discovery describes the process of achieving mutual awareness. Connection is the state of being constantly associated to one another. Steglich et. al. [7] provided an overview of prevalent technologies for discovery and connection, i.e. UPnP/DLNA, Bonjour, NSD, WebIntents, WebActivity, AllShare and AirPlay. In order to exchange information devices must have the ability to *communicate*. Steglich et. al. list WebSockets, WebRTC, WebIntents, WebActivities, UPnP/DLNA and AllShare as candidate technologies to implement communication between devices.

D. Enabling TV programme-related companion applications

A major limitation of the approaches presented in section II-C is that they are either subject to vendor-specific restrictions or that they are only supported by a limited set of devices. A "one-fits-all" solution, that is desirable for a broad uptake of offered services, can currently not be built upon these technologies.

The HbbTV standard [9], [10] defines a browser profile for connected TV devices. This includes facilities to link internet content to broadcast content. Thus, it provides broadcasters with a standardised platform to offer TV-tailored interactive web-applications that go along with the TV programme to the end-users. The deployment of connected TVs that support the HbbTV standard increases. Goldmedia found that in 2013 there are 2.5 million households in Germany with at least one HbbTV device that is connected to the internet and that there will be 13.3 million in 2016 [11].

Section II-B illustrated services that shows different possibilities to foster an engagement of the viewers with the TV programme. Furthermore applications are mentioned that permit the users to employ a communication path between devices. But there is no service available that is related to the running TV programme and allows interaction across the devices at the same time.

In 2011 Schnepf et. el. demonstrated a HbbTV application that is able to communicate with a web-application running on a mobile device [12]. A restriction of the solution is that the inter-device connection can only be used by that particular service. Whenever a user switches to another broadcast channel or application the communication path breaks down. However, what their work showed is the combination of the potential for programme relation with the help of the features of HbbTV applications and the capability to communicate between the devices. Below a system will be presented that generalises this approach by developing an infrastructure that makes these capabilities available across services.

III. SECOND-SCREEN FRAMEWORK

The Institut für Rundfunktechnik developed a second-screen framework that handles the mechanisms for discovery and connection of a HbbTV and a second-screen device's browsers plus the communication between applications running in these browsers. Furthermore, it allows the launch of applications on the second-screen triggered by the HbbTV application. Connections are persistent and can be used by any application affiliated to the same instance of the framework. The framework exposes its features to the applications via a JavaScript API. Below, the functioning of the system is described.

A. Discovery, connection, app-launch, communication

Figure 1 shows the components of the framework plus the client parts of an HbbTV application and its accompanying second-screen application as well as the relations between the components. The framework components are distinguished between those running on the server side and those running on the client side. Furthermore, the interfaces between the framework and the client application are highlighted. In the following, the role of the framework components for the discovery, connection and communication will be explained.

1) Connection management: The connection management handles the processes for the discovery, the connection and the automatic launch of applications on the secondary screen. It consists of a server component, an HbbTV client component and the application launcher component. The server component controls and manages the connections between devices and coordinates the permanent storage of information on associated devices in a data base (connection persistence in figure 1). The HbbTV client component is a JavaScript library which is loaded by the HbbTV application. The library provides adapters for the communication with the server. It provides API calls for the initialisation of the processes for establishing connections and launching applications plus the monitoring

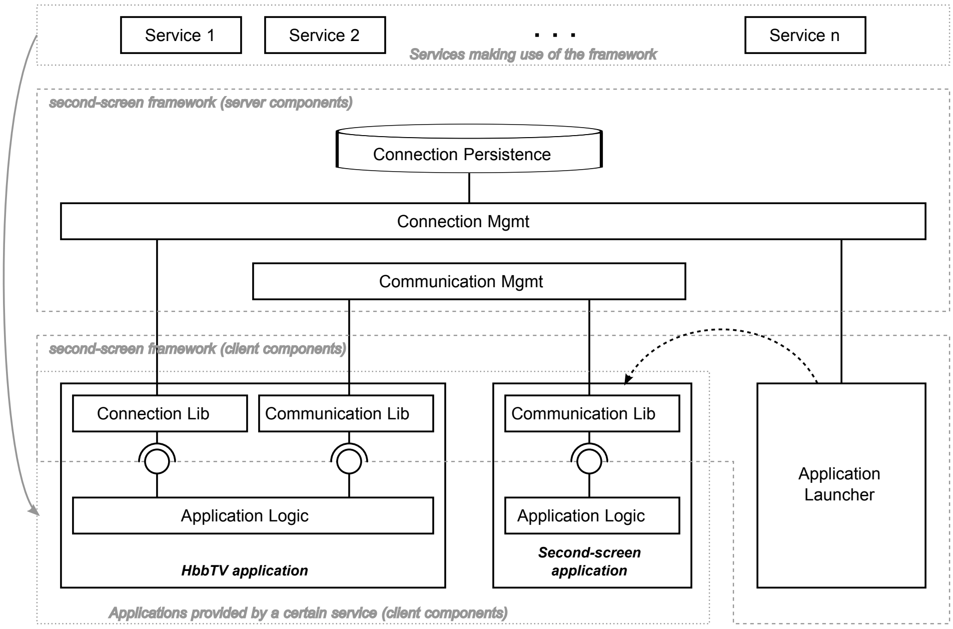
Service 1
Service 2
. . .
Service n
Services making use of the framework
second-screen framework (server components)
Connection Persistence
Connection Mgmt
Communication Mgmt
second-screen framework (client components)
Connection Lib
Communication Lib
Application Logic
HbbTV application
Communication Lib
Application Logic
Second-screen application
Application Launcher
Applications provided by a certain service (client components)

Fig. 1. Component diagram of the second-screen framework.

of the connection status. The application launcher component handles the launching of applications.

When the discovery is initiated, an application is loaded from the framework server to the HbbTV device. The GUI of the application displays a QR code (cf. [13]) to the user and the instruction to scan the QR code with an appropriate scanner app. The QR code contains a URL that refers to the application launcher on the framework server. Furthermore the URL contains a random but unique identifier (ID) as a parameter. The ID is generated by the server component of the connection management and identifies the HbbTV device. By scanning the QR code the URL is loaded by the browser of the second screen device – the application launcher appears to the user. A dialogue asks the user to confirm the connection process.

If the user acknowledges the connection, the second-screen device also obtains an ID from the framework. Moreover, each device now stores its ID in a persistent browser cookie. The IDs of both devices also get stored in the connection persistence data base as a pair of IDs. Both devices now become and remain associable.

After the successful connection process, the browser on the HbbTV device leaves the connection management application and gets automatically redirected to the application that originally initiated the connection process. The HbbTV application can now send a URL to the paired device. When the launcher application receives the URL, the request of the appropriate resource is executed and the companion application gets launched in the browser of the second screen.

2) Communication management: The communication management handles the message exchange between connected devices. It consists of a server and a client component. The server component provides a message queue. The client component is a JavaScript library that is loaded by the HbbTV application and the companion application. The library provides adapters for the communication with the message queue.

Via the JavaScript API, applications can send messages to connected devices and can subscribe for messages from them. Messages are exchanged via the server.

B. Technologies and patterns

1) Discovery – the QR code solution: The device discovery is solved by the means of a QR code. The approach is not new and was also used by Schnepf [12]. For the QR code generation the XZing library [14] is used and is integrated into the business logic of the connection management.

In contrast to other solutions for device discovery, the proposed discover process does not run automatically and requires user interaction. On the other hand it does not demand any particular network set-up. This has the advantage that application providers do not need to make any further assumptions on the device capabilities of the end-users other than that the devices have an internet connection and have a web-browser installed. Both requirements are fulfilled by HbbTV devices and the second-screen devices that are in the focus of the envisioned scenarios.

2) Application launcher: The application launcher component is designed to start web applications in the browser of the second-screen device. In the current version of the framework, this component is realised as a web-application itself. The application launcher of a connected device registers for messages containing URLs. Once a URL is available the appropriate application gets loaded in a new browser tab.

This solution faces the issue that the browsers of many mobile devices pause the execution of JavaScript in hidden tabs. As a consequence, the application launcher stops waiting for messages once an application has been launched. When a

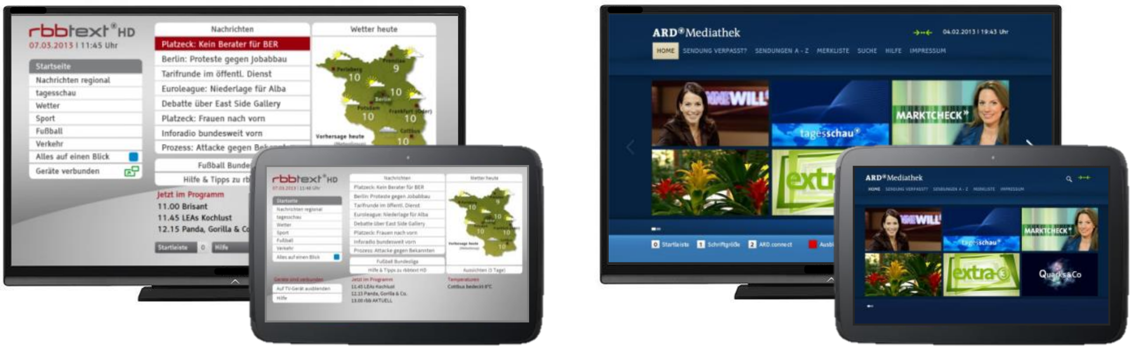

Fig. 2. rbbtext (left) and ARD-Mediathek (right) HbbTV applications with their accompanying second-screen applications

new application is to be launched the user needs to return to the launcher application.

For further optimisation, two options are envisaged. One is the integration of the application launcher functionality into a library that is loaded by the companion applications. New applications could than be launched by a running companion application. On the other hand the ability to start new applications would be restricted to those that integrate the library. Whenever the user leaves the context of the companion application the connection breaks down.

The other option is the integration of the launcher functionalities into a native mobile application. This could either start applications in the default browser of the device or embed a browser view into its own UI in order to provide the runtime for the companion applications. A drawback of this approach is the higher effort for the development and maintenance of the component for different mobile platforms, e.g. Android, iOS, Windows 8 etc.

3) Message exchange: The technologies that come into consideration for the implementation of the communication between connected devices introduced in section II-C do not apply to the system since they are not compliant with the HbbTV standard.

Hence the message exchange between connected devices is realised with the long-polling technique (cf. [15]). When a device subscribes to messages of a connected device, the client component of the communication management sends an HTTP request to the server component of the communication management. The server holds the request in case there is no message or sends the message in the HTTP response to the subscriber.

The long-polling approach reduces the number of requests over time compared to a periodical polling where clients would need to submit requests at an interval of 0.5 or 1s. Thus it significantly reduces the server load. On the other hand, it has to be taken into account, the hold of HTTP requests leads to a large amount of open HTTP connections. This requires an appropriate configuration of the server.

4) Cross-domain issues: All HbbTV and companion applications that use the framework shall be able to use the same inter-device connection for communication and application launch. In order to fulfil this requirement, the framework needs to be hosted as an independent instance on another domain as the affiliated services. This situation requires special handling of requests to the third domain which will be exemplified in the following.

Asynchronous requests from a web application to a server are usually handled by means of the XMLHttpRequest object [16]. The XMLHttpRequest is subject to the same-origin-policy [17] which intends to prevent third-party applications from an unauthorised usage of the JavaScript code. The HbbTV and companion applications are loaded from another domain than the JavaScript code that handles the communication with the framework server. Hence, the use of the XMLHttpRequest object is not applicable.

W3C specified a mechanism to allow cross-origin request [18]. The method called cross-origin-resource sharing is not part of the current HbbTV specification. Thus, interoperability cannot be guaranteed and method is not an option for the framework implementation.

To overcome these restrictions, the adapters for the communication to the server components were implemented on the basis of the JSON-P pattern [19]. The JSON-P pattern has two side effects. First, only GET-requests can be made to the server which limits the size per request. Large messages have to be split and submitted in several requests. Second, JSON-P bypasses the same-origin policy which lowers the barrier for a misappropriate use of the framework. This is prevented by an additional authorisation method.

C. Prototypes

At the time of the editorial deadline for this paper, the framework was prototypically integrated into two existing HbbTV services of the ARD network – others are planned. The existing prototypes are presented in the following.

1) rbbtext: The rbbtext (cf. figure 2) is an HbbTV application of the German public broadcaster rbb. It builds upon the classical teletext (cf. [20]) and intents to use the facilities of the HbbTV standard for the navigation through and presentation of content and information in order to improve the user-experience of the service. It provides the same information as the classical teletext and is connected to the same content management system. In contrary to the teletext users can navigate directly to a menu item by the help of the arrow keys on the remote-control. Furthermore the improved graphical presentation of content with the help of HTML unleashes the potential of today's high-resolution TV displays.

The second-screen enabled version of the rbbtext HbbTV application contains a new menu item that offers the user the option to connect a device. By selecting the button the user initiates the discovery process (cf. section III-A). Once the companion application is launched, the user can make use of a synchronised navigation – what happens on the one screen happens on the other screen as well. Furthermore the user can hide the application on the TV screen from the second screen. This allows the user to continue watching the broadcast TV show while browsing through the range of information of rbbtext on the second screen. Another use-case is the launch of URLs to content on the second-screen that's beyond the scope of HbbTV. The editors of rbbtext can include links to additional information in the HbbTV application that can not be displayed by the HbbTV device but by the second-screen device.

2) ARD-Mediathek: The ARD-Mediathek (cf. figure 2) is the video-on-demand portal of the ARD group. In the second-screen enabled version of the ARD-Mediathek users can initiate the connection process (cf. section III-A) by pressing the numerical button 2 on the remote-control. As soon as the appropriate companion application is launched on the second-screen device, users can browse through the range of video clips of the ARD-Mediathek on the second screen. They can use the full text search on the second-screen which can be cumbersome with the numerical key set of the classical remote. Once the users discover a clip, they can select the clip to be played back on the TV screen. Moreover, they can remotely operate the video player controls like play-pause, fast forward and fast rewind options of the video player in the HbbTV application. The users can also maintain lists of videos for subsequent consumption.

IV. Conclusion

A system has been developed that allows the creation of second-screen applications that avail themselves from a communication channel that enables interaction with the content across the screens. The connection, once established, can be used by multiple services which allows a convenient usage for the end-user. Since the solution is web-based, it is not restricted to run on a certain operation system which reduces required development efforts.

Prototypical integrations into two existing HbbTV applications were demonstrated. The prototypes foremost exemplify the quality of the second screen as an input device. However, their functionalities do not exhaust the set of potential scenarios that can be realised by means of the framework. The facilities to link HbbTV applications to the TV programme and to launch applications on the second-screen provide an infrastructure for the creation of meaningful programme-related companion applications. This can increase attractiveness of broadcast programme and trigger user engagement.

Acknowledgment

The author likes to thank Klaus Merkel and Michael Probst from IRT who made leading contribution to the concept of the second-screen framework.

Further thanks to rbb Innovationsprojekte and ARD Online for the integration of the framework into their services. Thanks to their prototypes, we were able to present the idea and benefit of the second screen framework in an appealing way. Furthermore we gained valuable insights into the usability of our API.

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 284534 (FI-CONTENT) and n° 603662 (FI-CONTENT 2).

References

[1] K. Böhm, R. Esser, and C. Mensch, *Die Zukunft der Consumer Electronics - 2012*. Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V. (BITKOM) and Deloitte, 2012, ch. Der Second Screen als Enabler für neue Anwendungsszenarien, pp. 30–34.

[2] Google. (2012, 08) The new multi-screen world: Understanding cross-platform consumer behavior. [Online]. Available: http://www.google.com/think/research-studies/the-new-multi-screen-world-study.html

[3] M. Rycak. (2012, 11) Wikipedia-Zugriffszahlen bestätigen Second-Screen-Trend. [Online]. Available: http://www.martinrycak.de/wikipedia-zugriffszahlen-bestatigen-second-screen-trend/

[4] Red Bee Media Ltd. (2012, 10) Broadcast industry not capitalising on rise of the second screen. [Online]. Available: http://www.redbeemedia.com/sites/all/files/downloads/second_screen_research.pdf

[5] Wikipedia. Second screen. [Online]. Available: http://de.wikipedia.org/wiki/Second_Screen

[6] Wikipedia. Second screen. [Online]. Available: http://en.wikipedia.org/wiki/Second_screen

[7] S. Steglich, M. Lasak, M. Tritschler, and L. Bassbouss, "Tutorial 2: Multiscreen app development," Presentation at 3rd FOKUS Media Web Symbosium, March 2013.

[8] P. Cano, E. Batlle, E. Gmez, L. de Gomes, and M. Bonnet, *Audio Fingerprinting: Concepts And Applications*. Springer Berlin Heidelberg, 2005, ch. 17, pp. 233–245.

[9] *ETSI TS 102 796, v1.1.1, Hybrid Broadcast Broadband TV*, European Telecommunications Standards Institute Std.

[10] K. Merkel, "Hybrid broadcast broadband TV, the new way to a comprehensive TV experience," in *14th ITG Conference on Electronic Media Technology (CEMT)*, 2011.

[11] K. Goldhammer and M. Birkel. (2012, 04) Potenziale von Smart TV für lokale Fernsehsender. Goldmedia GmbH Strategy Consulting. [Online]. Available: http://www.blm.de/apps/documentbase/data/pdf1/120425_Goldmedia_Smart_TV_Lokal-TV_final.pdf

[12] U. Schnepf and J. Hof, "Social-TV und HbbTV – Die Zukunft des Fernsehens ist sozial," *FKT – die Fachzeitschrift fur Fernsehen Film und Elektronische Medien*, vol. 65, pp. 559–561, 2011.

[13] *ISO/IEC 18004 – Information technology – Automatic identification and data capture techniques – Bar code symbology – QR Code*, International Organization for Standardization/ International Electrotechnical Commission Std.

[14] Google. (2013, 05) XZing (Zebra Crossing) – Multi-format 1D/2D barcode image processing library with clients for Android, Java. [Online]. Available: http://code.google.com/p/zxing/

[15] Wikipedia. Push technology. [Online]. Available: http://en.wikipedia.org/wiki/Push_technology#long-polling

[16] World Wide Web Consortium, W3C. (2012, 12) XMLHttpRequest. [Online]. Available: http://www.w3.org/TR/XMLHttpRequest/

[17] World Wide Web Consortium (W3C). Same-origin policy. [Online]. Available: http://www.w3.org/Security/wiki/Same_Origin_Policy

[18] World Wide Web Consortium, W3C. (2013, 01) Cross-origin resource sharing. [Online]. Available: http://www.w3.org/TR/cors/

[19] S. Stefanov, *JavaScript Patterns*. O'Reilly, 2010.

[20] Wikipedia. Teletext. [Online]. Available: http://en.wikipedia.org/wiki/Teletext