Requirements for a comprehensive second-screen experience

Christoph Ziegler
Platforms for Broadcast Services
Institut fur Rundfunktechnik (IRT), Munich

Hybrid TVs allow broadcasters to provide their viewers with interactive applications that go along with the programmed content. New possibilities evolve with the availability of second screens that can be linked to these applications. Our contribution will present different scenarios that illustrate how this affects the user experience and foreshadow new possibilities for businesses. We will discuss key requirements for realising such scenarios. Moreover we will point out problems developers face implementing these to run in today's browsers. The following requirements we will be addressed:

Device discovery:
is the process of making devices aware of each other. It is crucial for establishing a link between devices in order to share information. There are a number of proposals on how to solve automatic discovery like UPnP/DLNA or Bonjour. There are also attempts to make discovery mechanisms available in browsers like NSD or Web Intents. However, the existing approaches are either subject to vendor-specific restrictions or they are only supported by a limited set of devices.

Cross-service inter-device connection:
A connection between devices is needed to be able to assign information that is to be sent from one device to the right target. Connections are established after the discovery process. Connections should be between devices, they should not be specific for applications of certain services. During one evening of watching TV users switch channels and might use a plenty of applications. Having to execute the discovery mechanism in order to use a companion application on the second screen on every channel change seems cumbersome and should be avoided.

Remote application launch:
We are interested in mechanisms that allow an application starting a certain application on a connected device. A hybrid TV application of a certain service could use this feature to launch a companion application on the second screen. By this means the connected second screen can always be in contextual sync with the TV content. Remote launch of applications from second screens on connected TVs can be interesting to allow users pushing content they discovered on the second screen to the TV. The DIAL protocol describes a way to realise app launch on the TV from the second screen. However, there is no solution at the moment that makes the possibility for bi-directional remote application launch available to browsers on TVs and second screens.

App-to-app communication:
is needed to exchange information between an application running on TV devices and second-screen devices that are connected to them. App-to-app communication is useful for a plenty of things e.g. to remotely control the TV application from the second-screen or to push additional on a running TV show information from the TV to the second screen. For browser based applications app-to-app communication requires a server in the middle which causes high operating costs for application providers. Support of efficient technologies for bi-directional communication like WebSockets is limited on today's hybrid TV browsers.

We gained experience in this field during the development of a framework that aims ease the integration of these above mentioned features features into hybrid TV services. We plan to demonstrate this system and a couple of currently available applications that make use of it at the workshop. This will allow us to create a lively picture of what might be possible when the screens converge.

By giving a colourful overview about key scenarios of TV accompanying second-screen usage our contribution could provide the right setting for other contributions going more into depth on how the respective requirements could be technically solved and how the availability of these features to the end-users can be fostered by different standardisation efforts.