Telecom ParisTech position paper

Jean-Claude Dufourd and Cyril Concolato

The team at Telecom ParisTech has two points of focus that make participation in the W3C Web and TV workshop important for us:

- discovery and second screen issues
- HTML5 video and MSE issues

Discovery and Second Screen

We have been involved in the Web and TV IG for years, in particular in the Home Network Task Force. HNTF Requirements are almost consensual, including web technologies and discovery and compatibility with existing standards such as UPnP and Bonjour.
First Web Intents looked like a possible solution, then Web Intents got recalled.
Then Network Service Discovery is a simple API for discovery with UPnP and Bonjour and DIAL. NSD lets the web app deal entirely with communication. The assumption is to use XHR or WebSockets and implement the communication in JS. If so, you need the IP of the service you want to communicate with: exposing this IP opens the door to all sorts of hacking and fingerprinting.
Because of the resulting pushback, NSD has essentially grinded to a halt.
Our suggestion to hide the IP/url and manage all communication with the discovered services was not accepted despite its merit for privacy protection.

Another reality is that browsers do not want UPnP for various reasons, including interoperability problems with TVs. So NSD is going nowhere very slowly.
Solutions in the field include: variants of DIAL, including in HbbTV (semi proprietary), QR-codes for toolboxes usable today, Java proxies using WebSockets…

Within COLTRAM, a 3-year research cooperation between Telecom ParisTech and Fraunhofer FOKUS, we have an implementation of NSD with extensions. COLTRAM is a platform for cooperating web apps on an heterogeneous network. Tasks are implemented as a constellation of web apps that are mapped onto available devices. Web apps can be moved from device to device while the task is running.
The implementation is usable with any modern browser, as it is a proxy (in Java) that the web page connects to with WebSocket. We tested it on Windows, MacOSX, Ubuntu, Android, and remotely with iOS and HbbTV browsers. We have chosen not to use the IP/URL of the services, and implement the communication with the service directly within our system: the web app does not know the IP of the service, and cannot connect directly to the device hosting the discovered service.
COLTRAM is currently based on NSD, but could be using other methods.

So what is the best route to create an ecosystem around second screen, and in general, around the browsers of all the devices in the home ?

1. Discover with UPnP and manage the communication so that IP are not revealed and communication is constrained to valid UPnP messages, or XHR proxied by the discovery module ?
2. Forget about UPnP and just make other browsers discoverable, just creating a subset of safe entities that can discover each other and communicate, network of cooperating web apps. But then Media Servers, Media Renderers (TVs), Printers, etc are out of the picture ?
3. Define safe entities (such as a display in webscreens) and design specific protocols for the discovery of those safe entities; such safe entities could include MediaRenderers, MediaServers...
4. Use an object sharing server, that everyone discovers and that proxies UPnP services

HTML5 video and MSE

Telecom ParisTech has been working for a long time on interactive videos on aspects related to: video file formats, such as MP4; to streaming of videos, in particular with the recent MPEG-DASH standard; or to standards for interactivity, such as the early MPEG-4 BIFS or W3C SVG. The recent developments of HTML 5 video and of the Media Source extensions open the way for new and interesting interactive video applications in the browsers. In this context, Telecom ParisTech proposes an open-source implementation of the DASH and MSE standardsin the context of SVG documents, as an evolution of its SVG 1.2 Tiny support. The interest of the team is in having tools to produce interactive video content, such as MP4Box, and lightweight scriptable video players, as opposed to full-fledged web browsers. In addition, Telecom ParisTech researches solutions to improve the existing systems and proposes new features and questions such as:

- low-latency interactive videos: how to better integrate WebRTC with other Web video technologies?
- streaming of vector graphics: how to provide graphically-rich and tightly synchronized overlays?
- multi-device synchronization: how to synchronize multiple browsers?
- multi-network synchronization: how to synchronize in the browsers streams coming from different sources?
- and use of synchronized metadata: how to deliver any streams back to the JavaScript application, in time for processing and synchronized with the video?