



Followup on the HNTF work

- **Is Web Intents the solution to our requirements ?**
- **If yes, are there modifications to ask for, and can we organize some sort of HNTF lobby ?**
- **If not, what can we do about it as a group ?**



HNTF Reqs

■ Discovery

- Interface HTML(5) with discovery
- Compatibility with existing standards (UPnP and Bonjour)

■ Exposing services provided by HTML



Web Intents

- **Initially described as the solution for discovery**
- **Does not do discovery, just passive registry**
 - Needs additions for discovery
 - Sony: extend UPnP to interface with Web Intents
 - Unmodified UPnP: requires a UPnP proxy for discovery
- **Is browser-locked**
 - Does not define a standard way to provide extensions
 - De facto leaves extensibility to the « good will » of browser vendors

Actual problem: Define services distributed across devices, communicating together

- « Multiscreen » apps are part of it
- Two or more HTML webapps find each other and cooperate on something
 - One app on tablet and one on TV (multiscreen)
 - One app on each smartphone in a multiuser (possibly serious) game
 - One app has to discover the other(s), i.e. the other has to **expose itself on the network**.
 - Apps have to communicate (exchange messages)



Requirements of these scenarios

From initial HNTF reqs

- 1. Webapp discovers services**
- 2. Compatible with unmodified UPnP**
- 3. Webapp can exchange message with discovered service**
- 4. Webapp can advertize itself as service provider**



Coverage of these requirements

■ Web Intents

- R1: not without external help, UPnP services have to be registered with the browser
- R2: not with Sony proposal
- R3, R4: no

■ Network Service Discovery (Opera/Cablelabs)

- R1, R2: Yes!
- R3, R4: No



Missing

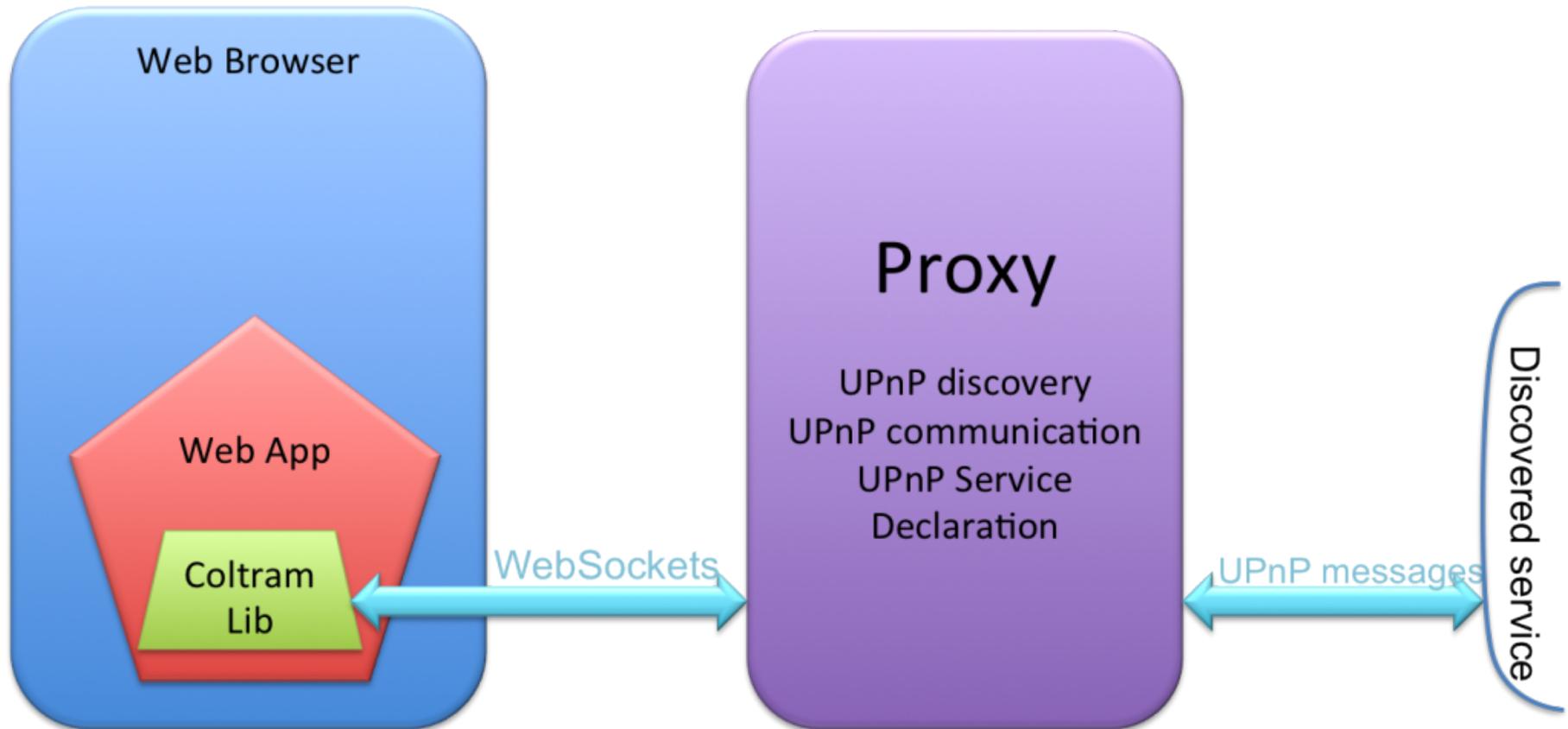
■ In Web Intents

- Interfacing with unmodified UPnP/Bonjour

■ In both:

- an abstraction of the messaging to allow a web application to send or receive actions without worrying about the details of the underlying messaging syntax
- API to expose a service implemented by this web application.

A solution for today





Features of the solution

- **Built from existing free software (PC* and Android)**
- **Very small library in browser**
 - Set of classes: Service, Action and ActionArgument, used for the representation of the exposed service and to call actions
 - **connect**: to connect to the Web Sockets server (the proxy), providing a **callBack** called when the connection is operational
 - **addServiceDiscoveredCallback**, **addServiceRemovedCallback** and functions to remove the callbacks
 - **Action.call**: calling an action from one of the discovered services
 - **exposeService**: passing a Service object and a serviceFunction function
- **With this architecture, it is easier to use already implemented messaging in the proxy**
- **Could use Network Service Discovery**



Back to my questions

- **Is Web Intents the solution to our requirements ?**
 - No
- **If not, what can we do about it as a group ?**
 - Push Network Service Discovery
 - Push for addition of Service Advertisement
- **If yes, are there modifications to ask for, and can we organize some sort of HNTF lobby ?**
 - We had momentum as Web&TV IG, let's build on that.



Discussion

- **Thank you for your attention**



Scenarii for the solution

- an HTML widget A discovering another HTML widget B on another machine (with different OS): B exposes itself on the network, A discovers B, A calls an action declared by B (changing the color of an HTML element in B).
- an HTML widget discovering and communicating with a Intel Tools test UPnP service: discovering the Network Light and turning it on and off
- an HTML widget discovering and communicating with an MPEG-U Widget (in GPAC), the HTML widget acting as a switch for a virtual light in the MPEG-U Widget
- an MPEG-U Widget (in GPAC) discovering and communicating with an HTML widget , the MPEG-U widget acting as a switch for a virtual light in the HTML Widget