Experiences from implementing PeerConnection and related APIs

@ Ericsson Labs

Background

- First implementation of the device element during spring 2010
- No peer-to-peer communication spec at the time
- A WebSocket-based, server-relayed solution, ‘MediaStreamTransceiver’ was implemented during summer 2010
- Working ConnectionPeer (i.e. the previous spec) implementation in late 2010
- The current PeerConnection implementation is constantly being improved and updated as the spec changes

Target platform

- Ubuntu 11.04
- Epiphany web browser
- WebKitGTK+
- GStreamer
- Libnice

WebKitGTK+

- WebKit port to the GTK+ GUI toolkit used by GNOME
- Using GLib and GObject
- Graphics provided by Cairo, HTTP by libsoup, **media playback by GStreamer**

GStreamer (1)

- Pipeline-based open source media framework using GObject
- Pipeline elements provided by plugins
- A great number of plugins for different codecs, transports, etc.
- Fairly easy to create new plugins from existing code from other projects
- Simple to work with and provides great functionality, we really like GStreamer

GStreamer (2)

- Since WebKitGTK+ already uses GStreamer for media playback, we could hook into some of the existing infrastructure
- For example, the existing sink element for rendering video in the video element
- Thanks to GLib, WebKitGTK+ and GStreamer share the same main event loop

Libnice

- An open source ICE library based on GLib with GStreamer elements
- Simple to use with GStreamer
- Since we have only run libnice to libnice we have now idea how interoperable it is

PeerConnection and MediaStream

- Experimental implementation made in WebKitGTK+
  - Includes getUserMedia
- Focus has been on the API and connection part
- The flexibility that the APIs (and the environment) provide is different from how things are traditionally done – this is great!
- The APIs can in fact be implemented with reasonable effort

Examples of feedback

- Each MediaStream must be identifiable (label)
- Use SDP per MediaStream to avoid glaring
- Misc. stuff on enable/disable components (aka tracks)

Ericsson Labs blog posts

- A series of blog posts have been made on Ericsson Labs during our implementation work that contain more details of our experiences
- http://tinyurl.com/beyond-html5

Ericsson Labs release

- Since May 20 this year we have made our modified WebKitGTK+ library publicly available on Ericsson Labs: https://labs.ericsson.com/apis/web-real-time-communication/downloads
  - Latest release from July 20
  - Easy installable through Ubuntu's package management system
  - Google Group for developer support
- We also have a movie :-) : http://www.youtube.com/watch?v=cx4rZH7fLp