Presentations for WebRTC at TPAC 2012

Harald Alvestrand

Chrome Implementation Status

Chrome M24 implementation status

- RTCPeerConnection committed, no flag
  - PeerConnection00 behind a flag
- Audio (G.711, Speex), Video (VP8)
  - Working on OPUS
- RFC-Compliant ICE, with backwards compat
  - TURN is "nearly there"

Bad news

- No DataChannel
- No DTLS-SRTP ("nearly there")
- No DTMF (waiting on you guys)

Stats API

Model, Status, Issues

Model (so far)

- Entity - Attribute model
- Unstructured entity list - snapshotted
- Entity has "local" and optional "remote" part
  - Timestamp attaches to "local" or "remote"
- Attributes are name-value pairs
- Names are a controlled namespace (IANA)
- Values are primitive types (int, string...)

Implementation status

- API added to WebKit
  - M24: Apps can call it, but get no callback
  - Shortcut: All values are strings
- Patches in flight for getting the first stat out of Chrome - target canary after M24
- Demo app written to learn how to use API

Open Issues

- What objects should we represent?
  - Starting: draft-alvestrand-rtcweb-stats-registry
  - Contains only "completely obvious things"
  - Need use cases to drive expansion
- How do we represent relationships?
  - Suggestion: Name objects - attrs have name values
  - Requires a getByName operation on a stats return
  - Enumerating all names required for debugging
  - Meaningful or random names?
- More types of attributes?
  - Multivalued attributes or .n notation for attr-names?
  - Structured attributes?
- Security issues?

Removing Things

MediaStreams and MediaStreamTracks

What do I mean by "remove"?

- Object at index N gets removed
  - Array no longer has a reference to it
  - Other references may exist
- Object at index N+1 is now known as "object at index N"
  - "Shift down"

Why remove things?

- Signal "I'm done with this object"
  - but we already have close() and friends?
- Allow Garbage Collection
  - but there aren't many of these things?
- Simpler Iteration over "all living objects"
  - including "are there any tracks of this type"
  - but it's only another line of code...
- If you remove at end, you can add a new object "in place" of the old one (mostly people think about this at index#0)

Why not remove things?

- Identifying an object to others: Stream N
  - Does that reference need to be stable?
  - Is that reference known outside your JS context?
- If the index is in protocol:
  - Renegotiation needed (it is, anyway)
  - Before renegotiation is done, references to OTHER streams (N+1) are ambiguous
- If we want to collect some info for all streams/tracks, even the closed ones:
  - Need a reference to the removed object
  - We now have 2 lists of objects to walk

Proposed solution

- Do not remove.
- object.close() transitions objects to a passive, minimally resource-consuming state.
- Indexes never change.
- For living objects, iterate while skipping state == closed.
- For stats, iterate over all objects.

DTMF API

Please, let's finish

DTMF API requirements

- Send DTMF according to RFC XXXX
- Support UIs that want to give feedback to the user on what's sent when
- Don't make it easy for users to harm themselves (such as tones in the feedback)

Present API issues

- 1 year ago:

```
interface AudioMediaStreamTrack : MediaStreamTrack {
    readonly attribute boolean canInsertDTMF;
    void insertDTMF (DOMString tones, optional long duration);
};
```

Issues:

- Makes track implementation complex for functionality only used in conjunction with PeerConnection
- No feedback channel for when the tones actually play out

Proposal

Two new functions on RTCPeerConnection

- pc.canSendDTMF(MediaStreamTrack)
- pc.sendDTMF(MediaStreamTrack outTrack, tones, duration, optional callback)

Callback triggers when tones start and end, and takes the character being played as parameter at start, or empty string at end. User can do whatever he likes about that.