WebRTC

Deltas between WhatWG and Cisco/ Mozilla Proposals

Cullen Jennings/Cary Bran/Tim Terriberry

What's the Difference?

The proposals are relatively close and have been updated with feedback from the mailing list.

Compared 7/21 versions of:

- https://github.com/mozilla/rainbow/wiki/RTC_API_Proposal
- http://www.whatwg.org/specs/web-apps/current-work/webrtc.html

The slides highlight deltas between the drafts interpretation of the:

- Stream API
- PeerConnection API
- Other Stuff

The Cisco/Mozilla W3C WebRTC submission will be updated based on feedback and submitted as a candidate working document

Stream API

Stream API

MediaStream Interface

```
[Constructor(in MediaStream parentStream)]
interface MediaStream {
    readonly attribute DOMString label;
    readonly attribute double currentTime;

    MediaStreamTrack[] tracks;
    MediaStreamRecorder record(in JSONHints hints);

    const unsigned short LIVE = 1;
    const unsigned short BLOCKED = 2;
    const unsigned short ENDED = 3;
    readonly attribute unsigned short readyState;

    attribute Function onended;
    attribute Function onReadyStateChange;
    ProcessedMediaStream createProcessor(in optional Worker);
};
MediaStream implements EventTarget;
```

Copy constructor for Forking

Returns time played since creating the stream

Hints to the recorder regarding the type of data contained by the MediaStreamTrack

Blocked state, stream can be reanimated from blocked state

onended is a subset of onReadyStateChange

Omitted

Purple == Cisco/Mozilla
Red == What WG

Stream API

MediaStreamTrack Interface

```
interface MediaStreamTrack {

    readonly attribute MediaStream stream;
    //IANA MIME media type
    readonly attribute DOMString kind;
    readonly attribute DOMString label;
    attribute boolean enabled;

    attribute Function onTypeChanged;
    attribute DOMString[] supportedTypes;
    attribute JSONHints hints;

    readonly attribute MediaBuffer buffer;

    // Extension for Audio API
    readonly attribute double volume;
    //Gain to be applied
    void setVolume(in double volume,
                   in double optional startTime,
                   in double optional duration);
}
```

Reference to containing Stream

Hints and event mechanisms regarding the type of data contained by the MediaStreamTrack

Access to underlying media data

Audio API Gain controls`

Purple == Cisco/Mozilla
Red == What WG

Stream API

MediaBuffer Interface

```
interface MediaBuffer {
    readonly attribute MediaStreamTrack track;
    // IANA MIME media type
    readonly attribute DOMString kind;

    // Codec specific
    // Example - may return the next Ogg packet in the stream
    attribute int sequenceNr;
    Object getBufferData(args);
};
```

The MediaBuffer interface allows web applications to access the MediaTrack's underlying media data.

JSONHints

```
JSONHints {
   "audioType": "spoken | music", "videoType": "fast | slow", "videoSize":
   "height x width", "videoFPS": "fps_n / fps_d", ... TBD ...
  };
```

JSONHints are to the passed into the MediaStreamTrack (from the programmer) to describe the kind of data the MediaStreamTrack is carrying

Purple == Cisco/Mozilla
Red == What WG

Stream API

MediaStreamRecorder Interface

```
interface MediaStreamRecorder {
    readonly attribute MediaStream stream;

    void getRecordedData(in Function onsuccess, in Function onerror);
    void getRecordedData(in BlobCallback? callback);

    void stop();
};

function onsuccess(DOMString type, DOMFile file);
function onerror(DOMString error);

interface BlobCallback {
    void handleEvent(in Blob blob);
};

partial interface URL {
    static DOMString createObjectURL(in MediaStream stream);
};
```

Reference to containing Stream

What WG has sample code that shows it being called, but not specified in the interface

WhatWG and Cisco/Mozilla diverge here, WhatWG is more explicit with BlobCallBack interface.

Cisco/Mozilla has error handling

onsuccess "type" parameter is the a string as defined in RFC842.

Records to a file

Blob is a file containing data in a format supported by the user agent for use in audio and video elements.
Records to a file.

Purple == Cisco/Mozilla
Red == What WG

Creates a Blob URL for the passed in MediaStream

Stream API

NavigatorUserMedia Interface

```
interface NavigatorMedia {

void getUserMedia(in JSONHints hints, in Function onsuccess,
                  in optional Function onerror);

void getUserMedia(in DOMString options,
                  in NavigatorUserMediaSuccessCallback? successCallback,
                  in optional NavigatorUserMediaErrorCallback? errorCallback);
};
Navigator implements NavigatorMedia;

const unsigned short PERMISSION_DENIED = 1;
const unsigned short RESOURCE_BUSY = 2;
const unsigned short RESOURCE_UNAVAILABLE = 3;

function onsuccess(MediaStream stream);
function onerror(unsigned short errorCode);

[Callback=FunctionOnly, NoInterfaceObject]
interface NavigatorUserMediaSuccessCallback {
    void handleEvent(in LocalMediaStream stream);
};
[NoInterfaceObject] interface NavigatorUserMediaError {
    const unsigned short PERMISSION_DENIED = 1;
    readonly attribute unsigned short code;
};
[Callback=FunctionOnly, NoInterfaceObject]
interface NavigatorUserMediaErrorCallback {
    void handleEvent(in NavigatorUserMediaError error);
};
```

WhatWG and Cisco/Mozilla are pretty close

WhatWG is more explicit with callback interfaces; Cisco/Mozilla uses the

Vary on callback implementation style and robustness of error codes but essentially do the same thing

Purple == Cisco/Mozilla
Red == What WG

Stream API

MediaStream HTML Element Additions

```
partial interface HTMLMediaElement {
    attribute MediaStream stream;
};

partial interface HTMLCanvasElement {
    attribute MediaStream stream;
};
```

Both proposals propose video and audio element capabilities. Cisco/Mozilla is explicit with defining a partial interface

Canvas Element Support

Purple == Cisco/Mozilla
Red == What WG

Stream API Summary

Rough list: Is forking the way proposed in the whatwg spec make sense?

PeerConnection API

Stream API

PeerConnection Interface

```
constructor PeerConnection(DOMString config, Function sendSignalingMessage, optional DOMString negotiationServerURN)
interface PeerConnection {
     void processSignalingMessage(DOMString msg);

     const unsigned short NEW = 1;
     const unsigned short NEGOTIATING = 1;
     const unsigned short LISTENING = 2;
     const unsigned short OPENING = 3;
     const unsigned short ACTIVE = 4;
     const unsigned short CLOSED = 5;
     readonly attribute unsigned short readyState;

     void addLocalStream(in MediaStream stream);
     void removeLocalStream(in MediaStream stream);
     readonly attribute MediaStream[] localStreams;
     readonly attribute MediaStream[] remoteStreams;

     void open(in DOMString addr);
     void listen();
     void accept(in DOMString addr);
     void send(in DOMString text);
     void close();

     attribute Function onconnecting;
     attribute Function onopen;
     attribute Function onMessage;
     attribute Function onIncoming;
     attribute Function onRemoteStreamAdded;
     attribute Function onaddstream;
     attribute Function onRemoteStreamRemoved;
     attribute Function onremovestream;

     attribute Function onReadyStateChange;
}
PeerConnection implements EventTarget;

[Callback=FunctionOnly, NoInterfaceObject]
interface SignalingCallback {
     void handleEvent(in DOMString message, in PeerConnection source);
};
```

States are 0-based in What WG doc

Additional states on from both

optional codec negotiation server URN

Naming convention different same functionality

Additional functions for opening connections, listening for connections, and accepting an incoming connection requests

Additional events for connecting and opening a stream

Naming convention different same functionality

Signaling callback interface

Purple == Cisco/Mozilla
Red == What WG

PeerConnection API Summary

Other differences

Cisco/Mozilla want a SINGLE PeerConnection object that can handle multiple incoming connections.

.....

Other Stuff

Other Stuff

List the other stuff that is not in the API here

SDP?

Use cases ?