Report type:	**Research Note TSSG-2012**
Report title:	**WebRTC Research**
Author:	**Brendan O'Farrell**
email:	**bofarrell@tssg.org**
Group:	**Telecommunications Software and Systems Group**
Address:	**WIT, West Campus, Carriganore, Waterford, Ireland**
Revdate:	**July 08, 2012**
Revnumber:	**N/A**
Docdate:	**July 03, 2012**
Description:	**WebRTC research**
Legal:	**(C) Waterford Institute of Technology**
Encoding:	**iso-8859-1**

GLOSSARY

RTC:	Real time Communication.
JSEP:	JavaScript session establishment protocol.
API:	Application programming interface
ICE:	Interactive connectivity establishment
STUN:	Simple transversal of UDP through NAT
NAT:	Network address translator.
UDP:	User datagram protocol.
TURN:	Transversal using relays around NAT.
VP8:	Video codec
DTLS:	Datagram transport layer security.
STRP:	Secure real-time transport protocol.
SIP:	Session initiation protocol.
SLA:	Service level agreement.
RTP:	Real-time transport protocol.
IETF:	Internet engineering task force
XMPP:	Extensible messaging and presence protocol
VoIP:	Voice over internet protocol
W3C:	World Wide Web consortium
URL:	Uniform resource locator
HTTP:	Hypertext transfer protocol
AEC:	Acoustic echo cancellation
AGC:	Automatic gain control
iLBC:	Internet low bitrate codec
iSAC:	Internet speech audio codec
RFC:	Request for comment
CELT:	Constrained energy lapped transform
CBR:	Constant bitrate
VBR:	Variable bitrate
PLC:	Packet loss concealment
HTML:	Hypertext mark-up language
MPEG:	Moving pictures expert group
TCP:	Transmission control protocol
XML:	Extensible mark-up language
VAD:	Voice activity detection
OTT:	Over the top
STC:	Society for technical communications
DTMF:	Dual tone multi frequency
ITU:	International telecommunications union
PSTN:	Public switched telephone network
GIPS:	Global investments performance standards
VP:	Video protocol
PBX:	Private branch exchange
MPLS:	Multiprotocol label switching
LTE:	Long term evolution

LAN:	Local area network
HD:	High definition
SDP:	Session description protocol

Table of Contents

INTRODUCTION

The primary aim of this document is to gain an understanding of WebRTC (Real Time Communication). How and why is it being developed? The potential benefits of real time communication and the effects it will have on the trillion dollar telecommunications industry. It will look at the organisations developing WebRTC and those which will set the standards for developers and users alike. The document will investigate the required components of WebRTC and how and when it will be deployed as standard. The document will also review the commercial opportunities and take a look and the future possibilities of WebRTC.

As WebRTC is still in the process of having standards and protocols ratified, this document will require regular updates.

WHAT IS WEBRTC (REAL TIME COMMUNICATIONS)

"WebRTC is a free, open project that enables web browsers with Real Time Communications (RTC) capabilities via simple JavaScript APIs. The WebRTC components have been optimized to best serve this purpose." WebRTC [http://webrtc.org]

WebRTC allows real time peer to peer audio visual communication via a HJTML5 compliant browser. Not all browsers have WebRTC capability at present. At this time of writing both Google Chrome and Opera 12 have it available to test, Firefox will have it the 3rd-4th quarter 2012 as will Internet Explorer. All four browsers are aiming to have WebRTC in their browsers by the end of 2012. There are no plugins required for WebRTC to work and no expensive pieces of hardware either. Just a WebRTC enabled browser, a camera (which is often quite standard on all new laptops), a mic/headset or mic/speakers and you real time communication available to you.

Having WebRTC integrated in a HTML5 enabled browser means you can now make real time audio video calls to any other WebRTC enabled device including such devices as tablets, smart phones, e-readers etc. The quality of these calls are only limited by the quality of the hardware, the audio and video codecs are open sourced and of a very high quality giving the client an excellent audio visual experience.

With IEFT having setting the standards for protocol and signalling, and W3C having setting the standard for the API's for app developers, this means millions of Java Script developers can now deliver and define web based communication like never before. No longer will it be the domain of the small number of SIP developers and VOIP system resellers.

WebRTC has the potential for real change in how we communicate, it has the chance to revolutionise the communications industry. It will however require that all invested parties comply with the standards laid down by W3C and IEFT, whether this comes to bear only time will tell.

KEY FEATURES OF WEBRTC

The 3 key features of WebRTC

- Media streams (getUser media): This is used to gain access to the users' camera and microphone. It can also be used to design WebRTC applications.

- Peer connection: This is the engine required to make high quality audio visual calls on the web.

- Data channels: The specification for data channels has yet to be ratified. It will be used for such things as, file transfer, games, co-browsing, shared whiteboard, shared document editing and more.

https://dvcs.w3.org/hg/audio/raw-file/tip/webaudio/webrtc-integration.html[Media Integration Coding examples]

MEDIA STREAMS

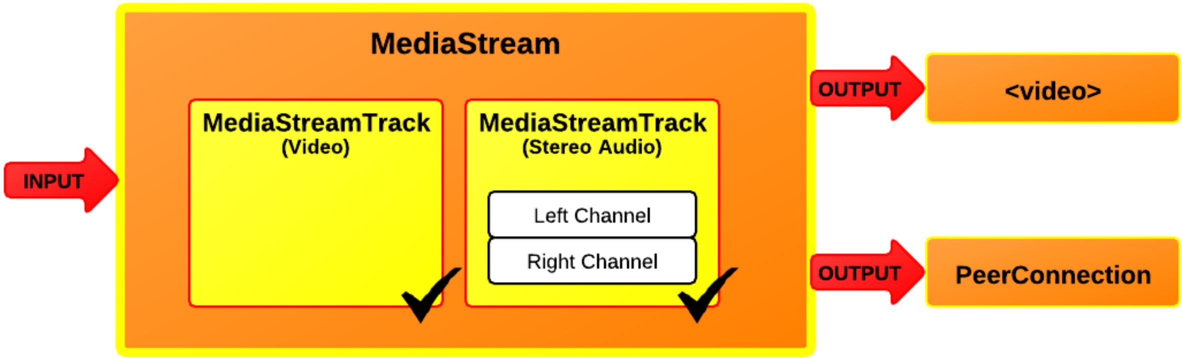
A media stream represents a media source, containing one or more synchronized media stream tracks. It can be converted to an object URL and passed to a <video> element. You use the getUser media API to get a media stream for the webcam and microphone. The user is then prompted to allow their consent, an example of this can be seen in Opera 12 where a text box drops down and requests permission from the user to allow use of their camera.

Some examples of using the getUser media. The photo booth app allows you take a photograph of the video image and add different effects to it. This app is great fun for the consumer but commercially it can be used for a multitude of reasons. You could be looking to buy a car, the calle can be showing you the car live using his smart phone and you can take snapshots of say the engine number or some damage that might require a quote to repair. This is only the tip of the iceberg.

Another example is face recognition. We could do away with passwords for online banking, social media sites, online shopping or any interaction which requires a password to gain access to an account. These are only two examples of using getUser media; we can expect to see the number of these apps multiply daily.

MEDIA STREAM DIAGRAM

http://dev.w3.org/2011/webrtc/editor/getusermedia.html



PEER CONNECTION

This is the engine behind making high quality audio/visual calls on the web. A peer connection allows us to the take the media stream and sends it across the web peer to peer. The actual code that implements PeerConnection is now a part of https://developers.google.com/talk/libjingle/ [libjingle]. While PeerConnection has no session protocol, and no XMPP/Jingle is required, we've reused many useful components from the libjingle package. http://www.webrtc.org/reference/api-description[WebRTC]

Peer connection is the API for establishing audio video calls (sessions).

Some built in features:

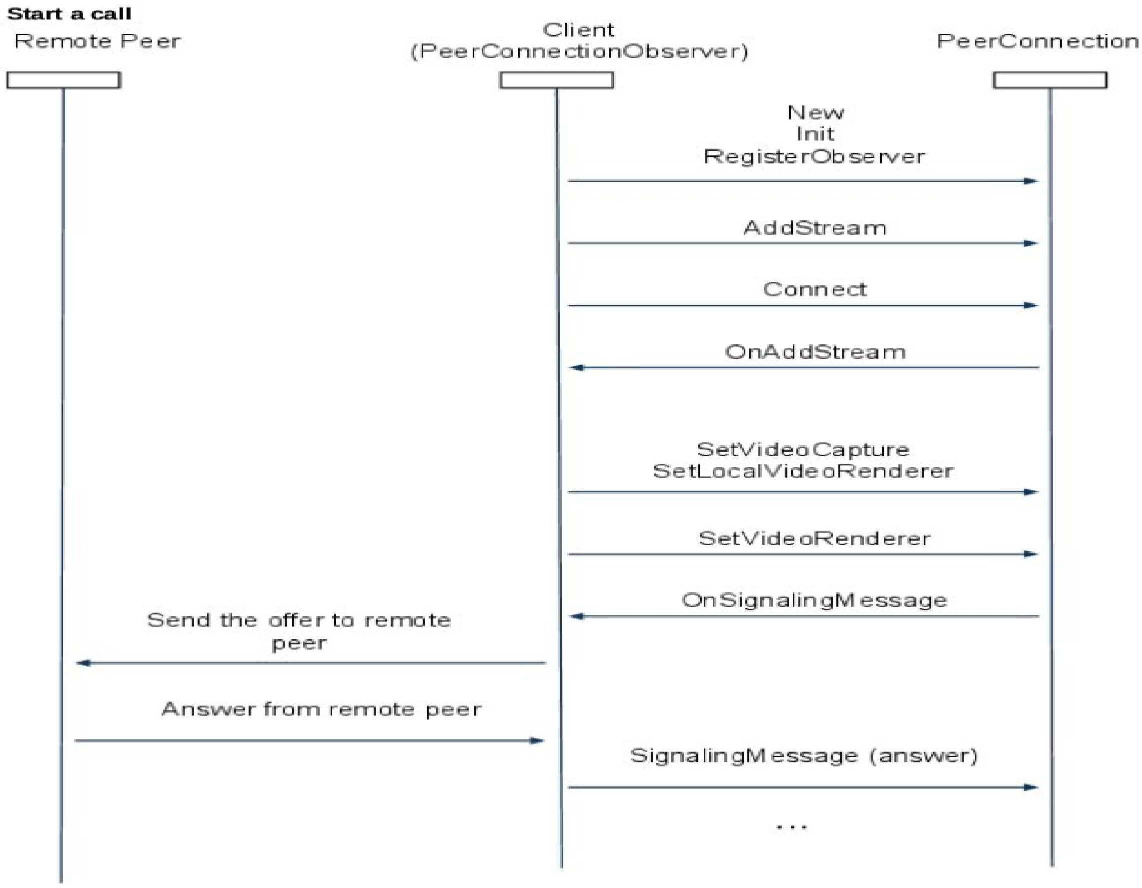
- It establishes peer to peer links
- Manages all the various audio and video codecs
- Sets the encryption
- Tunes the audio video streams to make better use of the bandwidth

PEER CONNECTION CALLING SEQUENCES

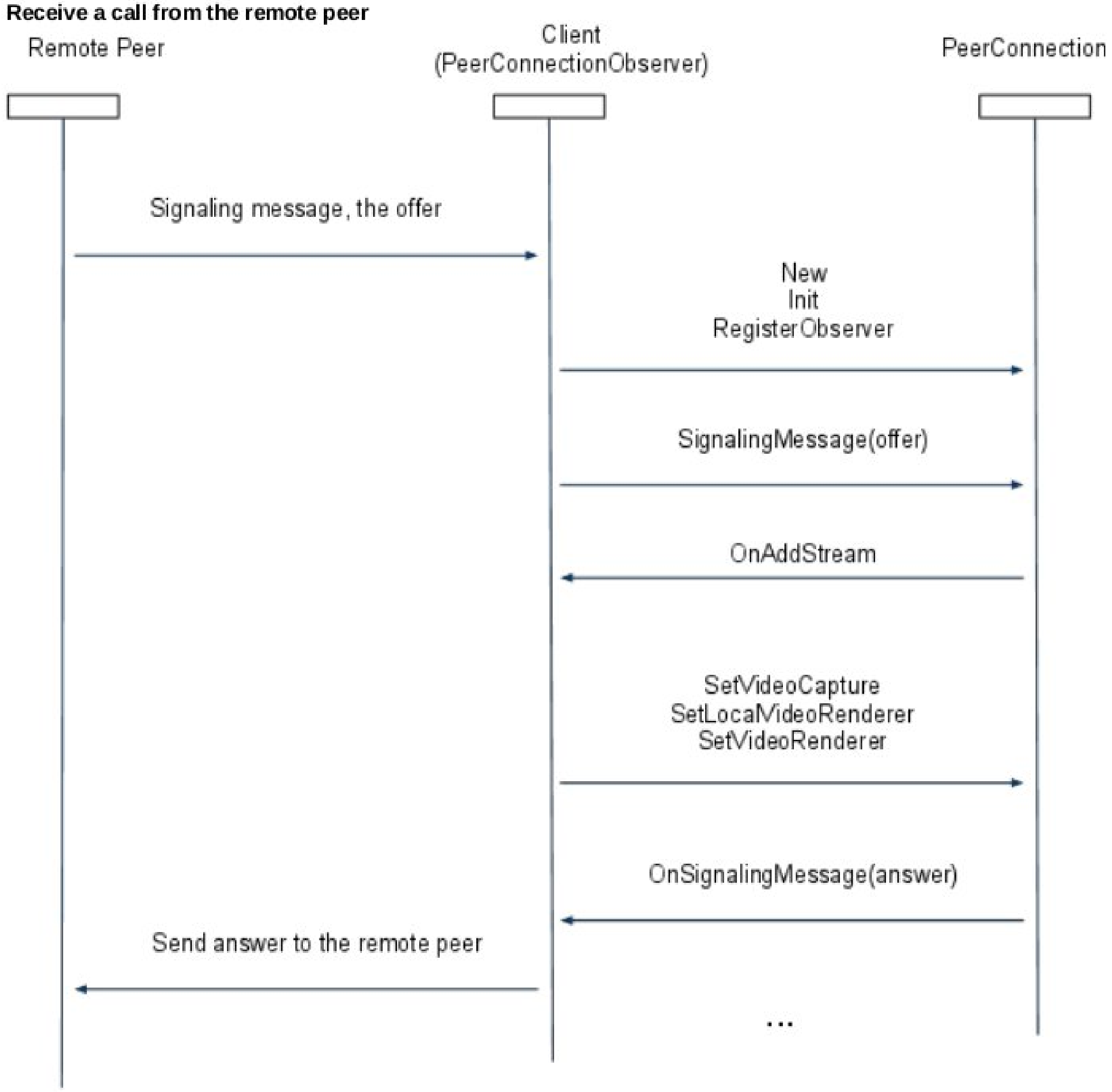
ALL FOLLOWING SEQUENCE DIAGRAMS ARE AQUIRED FROM WEBRTC.ORG

http://www.webrtc.org/reference/api-description[WebRTC API description]

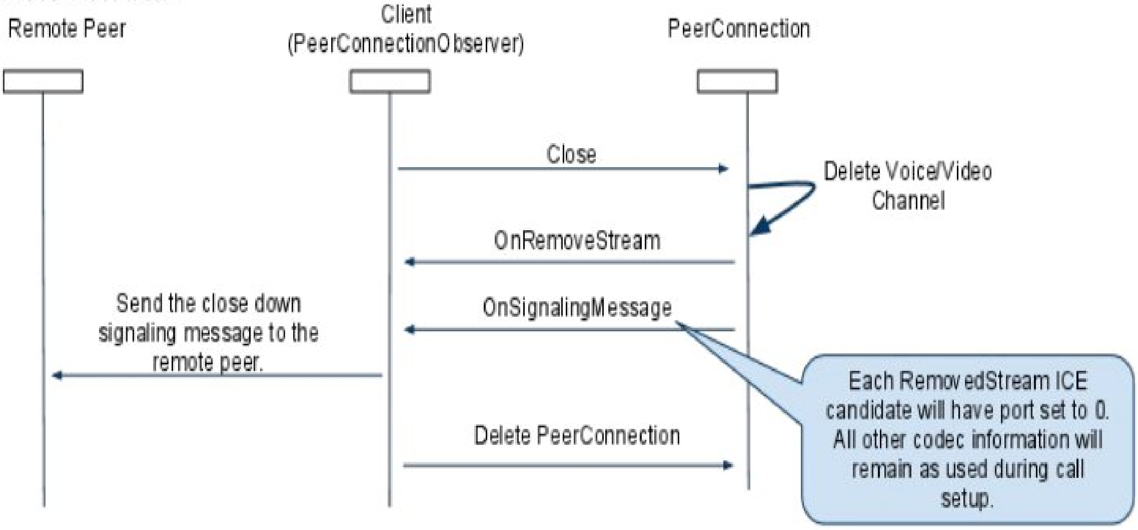
Calling sequences



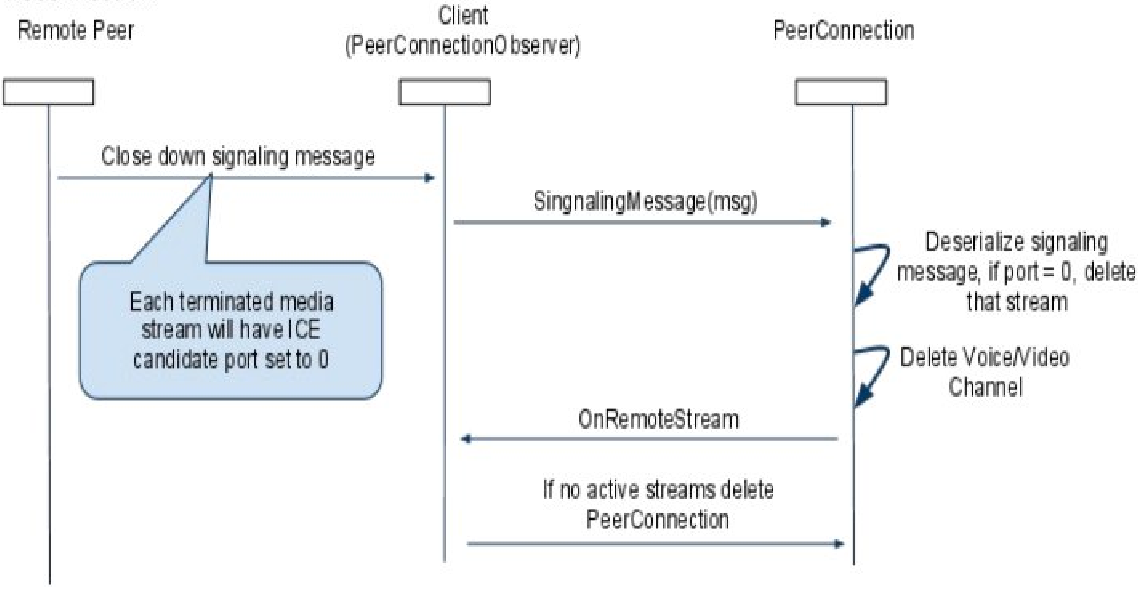
Receive a call from the remote peer



Disconnect a call



Remote Peer initiates the disconnection



DATA CHANNELS

A data channel is a peer to peer exchange of arbitrary application data. It has low latency, high message rate/throughput and optional unreliable semantics.

Key features of data channels:

- Leverages peer connection session setup
- Multiple simultaneous channels, with prioritization
- Reliable and unreliable delivery semantics
- Built in security (DTLS)
- Congestion control
- Can be used with or without audio and video
- Similar API to websockets

A sample of real world use cases:

- In gaming if you need to send data about say positions, directions. It is more efficient to send them over a peer to peer connection than over HTTP
- Real time text. An example of this is sending code or a process to an engineer who is out on site
- File transfer, no more having to drive to your accounts with boxes of paper
- Remote desktop applications
- Decentralized networks, you can communicate on a private encrypted channel.

At present the specification for data channels has yet to ratified, here are some initial proposals and an interim report.
http://dev.w3.org/2011/webrtc/editor/webrtc.html#datachannel[Data channels WebRTC]http://lists.w3.org/Archives/Public/public-webrtc/2012Jun/att-0063/W3_Interim_June_2012_Data_Channel.pdf[Interim report on data channels from Randell Jesup/IETF]

THE COMPONENTS OF WebRTC

There are 3 components to WebRTC

- Audio
- Video
- Network

AUDIO COMPONENTS OF WebRTC

Audio

The WebRTC project offers a complete stack for voice communications. It includes not only the necessary codecs, but other components crucial for a great experience. This includes software based acoustic echo cancellation (AEC), automatic gain control (AGC), noise reduction, noise suppression and hardware access and control across multiple platforms. There are audio codec standards set by the WebRTC working group charter. They are the iLBC and the iSAC audio codecs, both developed by Global IP Solutions. Global IP Solutions was purchased by Google in 2010.Since then Google have provided the audio codecs royalty free.
http://www.webrtc.org/ilbc-freeware/ilbc-extra-documentation[webrtc.org]

There is a third codec which many have been clamouring to be accepted into WebRTC, and that is the Opus codec. On the 2nd of July 2012 The IESG approved the following document: Definition of the Opus Audio Codec (draft-ietf-codec-opus-16.txt) as Proposed Standard. http://datatracker.ietf.org/doc/draft-ietf-codec-opus/ [Opus draft document]

The iSAC audio codec

iSAC is a robust, bandwidth adaptive, wideband and super-wideband voice codec developed by Global IP Solutions used in much Voice over IP VoIP and streaming audio applications. iSAC is used by industry leaders in hundreds of millions of VoIP endpoints. This codec is included as part of the WebRTC project.https://sites.google.com/site/webrtc/faq#TOC-What-are-the-parameters-of-iSAC-[webrtc.org]

Features

- The sampling frequency is 16kHz (wideband) or 32 kHz (super wideband).
- Adaptive and variable bit rate is 10 kbit/s to 52 kbit/s
- Adaptive packet size is 30 to 60ms.
- Complexity comparable to G.722.2 at comparable bit-rates and algorithmic delay of frame size plus 3ms.

The iLBC audio codec

iLBC is a free narrowband voice codec that was developed by Global IP Solutions used in many Voice over IP VoIP and streaming audio applications. In 2004, the final IETF RFC versions of the iLBC codec spec and the iLBC RTP Profile draft became available. This codec is included as part of the WebRTC project.https://sites.google.com/site/webrtc/faq#TOC-What-are-the-parameters-of-iSAC-[webrtc.org]

Features

- Bitrate 13.33 kbps (399 bits, packetized in 50 bytes) for the frame size of 30 ms and 15.2 kbps (303 bits, packetized in 38 bytes) for the frame size of 20 ms
- Basic quality higher then G.729A, high robustness to packet loss
- Computational complexity in a range of G.729A

OPUS audio codec

“The Opus codec is designed for interactive speech and audio transmission over the Internet. It is designed by the IETF Codec Working Group and incorporates technology from Skype's SILK codec and Xiph.Org's CELT codec.

The Opus codec is designed to handle a wide range of interactive audio applications, including Voice over IP, video conferencing, in-game chat, and even remote live music performances. It can scale from low bit-rate narrowband speech to very high quality stereo music” .http://www.opus-codec.org/[Opus-codec.org]

Features:

- Bit-rates from 6 kb/s to 510 kb/s
- Sampling rates from 8 to 48 kHz
- Frame sizes from 2.5 ms to 60 ms
- Support for both constant bit-rate (CBR) and variable bit-rate (VBR)
- Audio bandwidth from narrowband to full-band
- Support for speech and music
- Support for mono and stereo
- Support for up to 255 channels (multi stream frames)

- Dynamically adjustable bitrate, audio bandwidth, and frame size
- Good loss robustness and packet loss concealment (PLC)
- Floating point and fixed-point implementation

You can read the full specification in the http://tools.ietf.org/html/draft-ietf-codec-opus-16[latest Internet Draft].

VIDEO COMPONENTS OF WebRTC

"The VP8 codec is the codec of choice for the WebRTC project, introduced in 2010 as part of the WebM Project. It includes components to conceal packet loss, clean up noisy images as well as capture and playback capabilities across multiple platforms. WebM is an audio-video format designed to provide royalty-free, open video compression for use with HTML5 video. The project's development is sponsored by Google."
Inc.http://www.webmproject.org/[WebMproject]

The Vp8 video codec

VP8 is a highly efficient video compression technology that was developed by On2 Technologies. Google acquired On2 in February 2010. It is the video codec included in the WebRTC project https://sites.google.com/site/webrtc/faq#TOC-What-are-the-parameters-of-iSAC-[webrtc.org] This is the first time a video codec that has been open sourced compares favourably with the industry standard H.264. In July 2010 The mpeg- tech group carried out a comparison between H.264 and Vp8.http://lists.mpegif.org/pipermail/mp4-tech/2010-July/009388.html[Mpeg-4 Tech group] Every year, the MP4-Tech experts group compare every h.264 implementation in order to track performance and quality improvements. The Graphics and http://www.compression.ru/video/codec_comparison/h264_2010/vp8_vs_h264.html[Media Lab of Moscow State University] published a new, deep study of the performance of VP8, x264 and XviD implementations.

It's unusual that Mpeg-4 tech group would test a codec other than h.264 but they did with vp8 and they prove that results are respectable in many areas.

In HDTV for example, VP8 performed similar to x264 (considered the best implementation of h.264 by previous comparisons) but with 5-20% lower encoding speed. Comments from VP8 developers say that "old comparisons results have an inherent bias against VP8 because input sequences were previously encoded using another codec before being applied to VP8".

These results can improve very quickly with optimizations, and the Russian lab hasn't yet tested implementations of VP8 other than the one provided by Google developers. Ronald Bultje, David Conrad, and Jason Garret-Glaser, x264 developers, are now creating a native VP8 video codec implementation for the open source FFmpeg project.This is the most in depth study done to date, all othe studies that compare the two codec have been specific and not as wide ranging.

The way seems widely open for the WebM Project to provide a truly free, high quality codec for the world .http://www.osnews.com/story/23525/Deep_Analysis_of_the_VP8_Codec_by_H_264_Experts[OS News] Although at the time of writing the VP8 codec still has to be ratified. THe VP8 encoder and decoder are available from The WebMProject. http://www.webmproject.org/tools/vp8-sdk/group__vp8__encoder.html[VP8 Encoder] http://www.webmproject.org/tools/vp8-sdk/group__vp8__decoder.html[VP8 Decoder]

NETWORKING COMPONENTS OF WebRTC

“Dynamic jitter buffers and error concealment techniques are included for audio and video that help mitigate the effects of packet loss and unreliable networks. Also included are components for establishing a Peer to Peer connection using ICE / STUN / Turn / RTP-over-TCP and support for proxies. This technology comes in part from the Libjingle project”. http://www.webrtc.org/faq#TOC-Network[WebRTC.org]

Libjingle

Libjingle is a collection of open-source C++ code and sample applications that enables you to build a peer-to-peer application. The code handles creating a network connection (through NAT and firewall devices, relay servers, and proxies), negotiating session details (codecs, formats, etc.), and exchanging data. It also provides helper tasks such as parsing XML, and handling network proxies.

Features of Libjingle:

- A multi-user voice chat application
- A multi-user video conferencing application
- A multi-user live music streaming application
- A peer-to-peer file sharing application

Libjingle is available on Google Code for both Windows and UNIX/Linux operating systems. The source code is provided as part of our on-going commitment to promoting consumer choice and interoperability in Internet-based real-time-communications. This code is made available under a Berkeley-style license, which means you are free to incorporate it into commercial and non-commercial software and distribute it.
https://developers.google.com/talk/libjingle/[Libjingle project].

ICE/STUN/TURN

ICE and STUN are standardized methods for establishing a peer-to-peer connection on the internet, even if the two end points are behind private network addresses (NAT). At present Google's current stack deviates from the official standard. They are working to rectify this. Google will also support TURN servers to allow connections through tougher firewalls, where relaying and encapsulation are needed. Exactly what type of TURN will be supported has yet to be defined.

WHO ARE THE COMPANIES BEHIND WEBRTC

There are number of companies involved in the WebRTC project in its present state. They are Google, Mozilla, and Opera on the browser side with W3C and IEFT on the standards side. Only recently Microsoft have also shown an interest in WebRTC and they are currently recruiting developers to work on a project combining Skype with WebRTC, so development could occur earlier than expected for Internet Explorer browser users.

GOOGLE

Google have been to the forefront of this project. They want to develop a standard based real time media engine that will be available in all browsers. In order to drive the development of real time communication Google have released nearly $70 million worth of open source code to developers. This open source audio and video codecs came about through the acquisition by Google of two companies in particular Global IP solutions and On2 Technologies.

In early 2010, Google purchased On2, a technology company that developed the VP series of codecs, with the latest codec being VP8. On2 has always positioned its codecs as a patent free replacement to the H.26x series of codecs, which were standardized, patented and widely used in the communications and broadcast industry. Google then went about opening On2's technologies to the world and open sourced VP8 under the name of the WebM project. The idea was to replace H.264 for web videos and by that, reduce patent costs for everyone, including Google themselves.

Also in 2010 acquired Google acquired another technology company, Global IP Solutions (GIPS), a company known for their media frameworks – a piece of technology that makes developing VoIP and video calling applications easier. At the time, GIPS had the largest market share in VoIP, which caused much of the telecommunications industry to search for alternative solutions as a valuable revenue source in SLA was disappearing. As with On2, Google took GIPS' technology and open sourced it. This time they threw out all voice and video codecs that had patent owners and added an additional layer – a JavaScript API as an integration layer to web browsers.

Why did Google seemingly dismiss the profitable patented codecs of both GIPS and On2? They wanted to have bidirectional media processing and coding technologies available in all browsers. By releasing the audio visual codecs royalty free they increased and set the pace for developing real time communication in all browsers. Google then proceeded to push it as a standard at the IETF and W3C. The IETF organisation is responsible for the protocol and signalling standards. W3C set the standards for the API's.

Google host a forum page on the latest WebRTC developments at https://plus.google.com/113817074606039822053/posts[Google + WebRTC]

MOZILLA/FIREFOX AND WEBRTC

Mozilla attended IETF 83 in Paris 2012, and showed a demonstration of a simple video call between two Browser ID-authenticated parties in a prototype build of Firefox with WebRTC support.

Mozilla have also been experimenting with integrating social features in the browser from their Persona social media product and combining it with WebRTC to establish a video call between two users who are signed in using Browser The Social API add-on, once installed, provides a sidebar where web content from the social service provider is rendered. http://hacks.mozilla.org/2012/04/webrtc-efforts-underway-at-mozilla/ [Mozilla/demo]

In early 2012 Mozilla had stated they were going to adapt the VP8 video code. At present some confusion reigns. Mozilla conscious of the fact that most smart phones and tablets run the H.264 codec, have started looking at using that codec and offering the VP8 codec a separate alternative.

Mozilla are aiming to have real time communication available in Firefox last quarter 2012.

OPERA

Opera released its new version of its web browser Opera 12 on June 14 last. The new version includes support for WebRTC. Although Opera less than 3% of the browser market they are becoming much in the mobile phone and tablet market.

MICROSOFT AND WEBRTC

At the IEFT meeting 83 in Paris Microsoft indicated interest in supporting WebRTC in their IE browser most probably in order to support Skype, as Skype are already using the VP8 video codec, which looks likely to be the chosen standard video codec for WebRTC. WebRTC was not expected in the IE browser until sometime in 2013. However in June of this year Microsoft posted a job on their website looking for developers to develop WebRTC in Internet Explorer. With this development it now looks as if Microsoft will join the other browser companies in launching WebRTC late 2012.

http://arstechnica.com/information-technology/2012/06/opera-12-arrives-with-webcam-apis-and-experimental-webgl-support/[Arstechnica.com]

THE ORGANISATIONS SETTING THE STANDARDS FOR WEBRTC

The organisations which are responsible for the WebRTC standards are IETF which is responsible for protocol and signalling and W3C which will look after the standards for API's for all app developers.

WEB REAL-TIME COMMUNICATIONS WORKING GROUP CHARTER (2011)

"The mission of the Web Real-Time Communications Working Group, part of the Ubiquitous Web Applications Activity, is to define client-side APIs to enable Real-Time Communications in Web browsers.
These APIs should enable building applications that can be run inside a browser, requiring no extra downloads or plugins, that allow communication between parties using audio, video and supplementary real-time communication, without having to use intervening servers (unless needed for firewall traversal, or for providing intermediary services)

The working groups scope is enabling real-time communications between Web browsers require the following client-side technologies to be available:

API functions to explore device capabilities, e.g. camera, microphone, speakers (currently in scope for the Device APIs & Policy Working Group)

API functions to capture media from local devices (camera and microphone) (currently in scope for the Device APIs & Policy Working Group)

API functions for encoding and other processing of those media streams,

API functions for establishing direct peer-to-peer connections, including firewall/NAT traversal

API functions for decoding and processing (including echo cancelling, stream synchronization and a number of other functions) of those streams at the incoming end,

Delivery to the user of those media streams via local screens and audio output devices (partially covered with HTML5) ". http://www.w3.org/2011/04/webrtc-charter.html [W3C]

IETF (Internet Engineering Task Force)

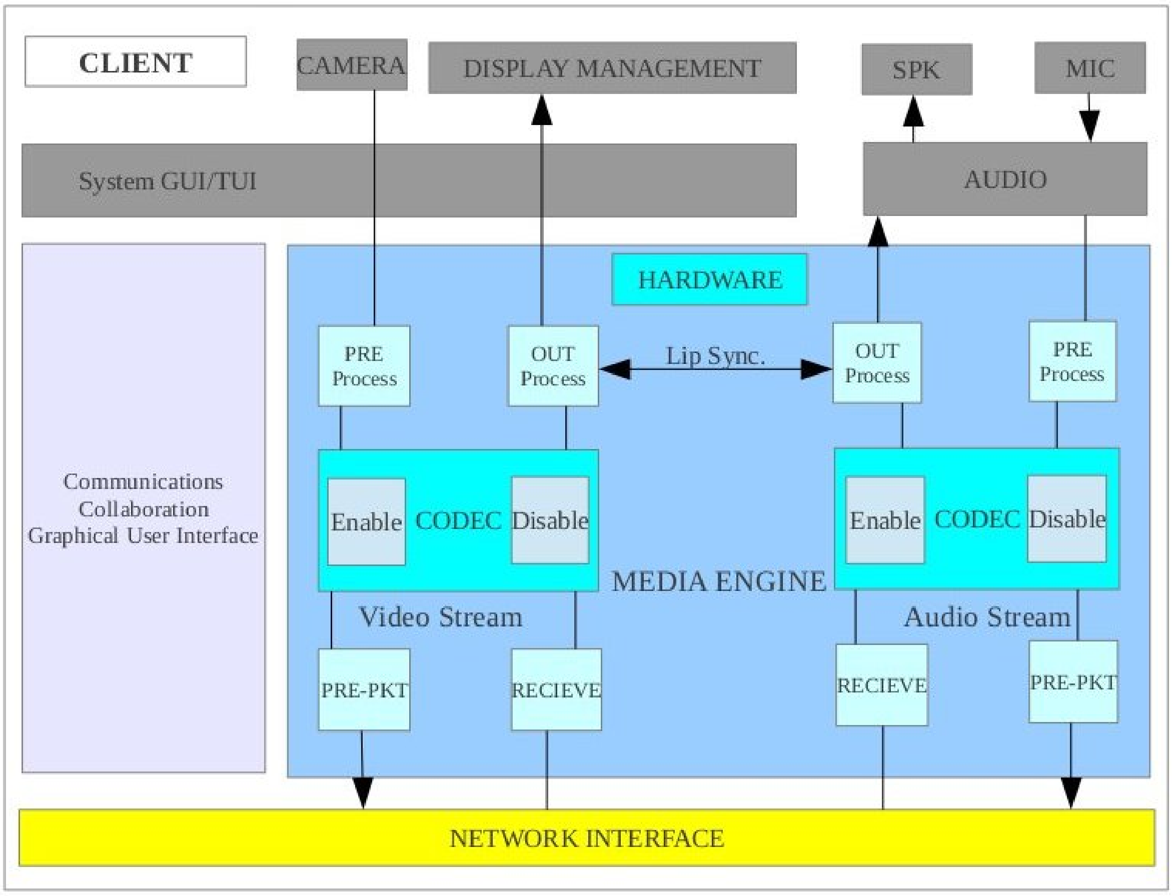
"The goal of the IETF is to make the Internet work better." http://www.ietf.org/[IETF] IETF is made up of working groups and informal discussion groups. When a group has completed their task they are disbanded. Each group has its own charter; this tells them what is required of them and how to proceed. IETF is open to anyone who wishes to participate, aside from meetings; discussions are often held via mailing lists. Entry fees cost $650 per person to attend the meetings. IETF is not open to companies; you join on an individual basis. There is no voting procedure in IETF; it operates on a consensus basis.

The IETF RTCWEB WG was formed in April 2011 and is currently generating RFCs in moving to a standard. The IETF is charged with setting all the protocols and signalling involved in WebRTC. To View all drafts please click on link. http://tools.ietf.org/html/draft-ietf-rtcweb-rtp-usage-01[IETF and WebRTC]

TECHNOLOGY

There are a number of elements are required for a client with real time communication, a media engine, an interface and a framework. See figure 1.

REAL TIME COMMUNICATION INTERNAL DIAGRAM



The large grey area houses the communications /collaboration GUI, this is the visual interface. The visual interface can be a hard interface such as a tablet screen, phone keypad, pc or any other device. The media engine is housed in the blue area and its main function is that it manages the real time transmission and receipt of a video/audio stream, the rest of the diagram contains the framework.

LIST OF MEDIA ENGINE FUNCTIONS

Here are the set of functions which enable the media engine to deliver and receive high quality sound and vision.

Audio

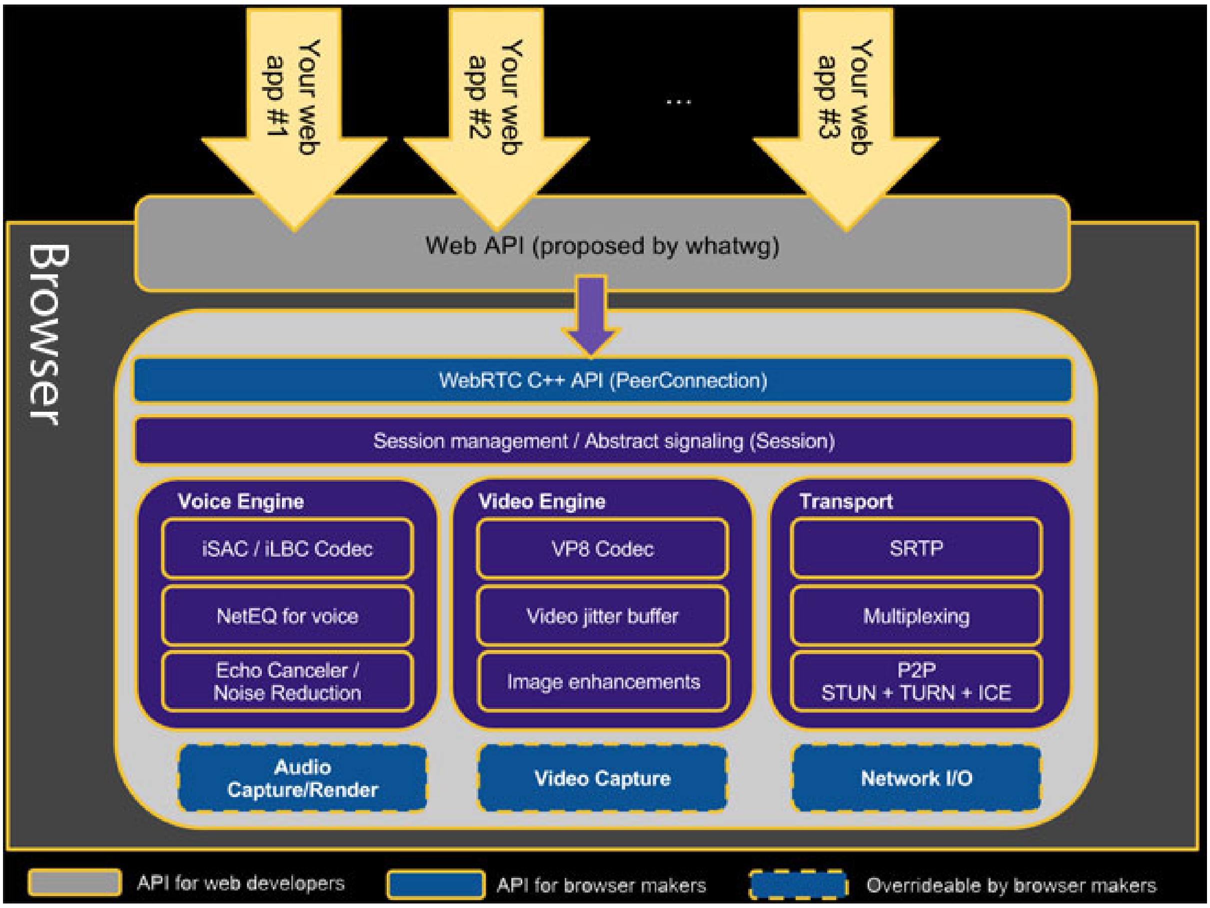
- Setup and control the hardware
- RTP, compression, encryption, statistics, etc.
- Produce low-latency audio from microphone
- Conceal loss, de-jitter and play audio from the network
- Cancel echo, VAD, reduce noise, etc.
- Manage codecs
-

Video

- Render video, capture camera input
- Video processing (blue screen, gamma, etc.)
- Conceal loss, de-jitter and play video from the network
- Cancel echo, VAD, reduce noise, etc.
- Manage codecs
- Bandwidth Management

The main aim of WebRTC is to combine the media engine and a set of standard APIs with the result being a browser capable of real time communication. In figure 2 you can see the.

INTERNAL WORKINGS OF A MEDIA ENGINE (Diagram).



The WebRTC Media Engine uses both a set of standard components, including codecs to minimize the issues of two WebRTC end points communicating, It also includes a set of standard APIs so a server that the browser connects to can control the WebRTC Media Engine in the client. Beyond the basic media functions, WebRTC includes an API set that enables the controlling server software to cause a direct connection between two WebRTC devices without any other external signalling. By merely telling two WebRTC devices to communicate, the server can initiate an IP based voice or video communications. http://www.pkeconsulting.com/pkewebrtc.pdf[PKE Consulting]

ESTABLISHING A CONNECTION

Client 1: caller	SERVER	Client 2: callee
	1: Provide signalling chanel	
2: Connect to server		2: Connect to server
3: Set up peer connection		
4:Send offer to server		
	5: Distribute signalling to other client	
		6: Accept signalling offer
		7: Set up peer connection
		8: Send answer to server
	9: Distribute signalling to other client	
10: Accept signalling offer		
11: Send ok to server		
	12: Distribute signalling to other client	
		13: Direct streaming established

CODE SAMPLES

http://www.webrtc.org/reference/native-apis [WebRTC native API's]

http://www.webrtc.org/reference/api-description [PeerConnection native API’s]

https://code.google.com/p/libjingle/source/browse/#svn%2Ftrunk%2Ftalk%2Fexamples%2Fpeerconnection%2Fclient [Sample client application]

THE IMPLICATIONS AND BENEFITS OF WEBRTC

- You will organise your phone calls (audio visual) using your preferred social media application or your office suite.
- The PSTN line is only used if one caller/calle is not connected to the web, or if the internet signal is too low.
- Your company's website now becomes its call centre.
- Your weblog becomes your personal communications assistant.
- You don’t require an operator anymore; you just need to be connected to the internet.
- Contacting anyone or anybody becomes a one touch, simple process.
- The ability to communicate becomes embedded in your email, your webpage, your TV, games device, apps and much more.
- Your phone number is no longer necessary anymore. It's from past technology.
- All OTT communication can be used anywhere or on any smart device
- Any web developer can create new communication services for their clients.
- Advertising enters a whole new world of opportunities.
- Call a friend using a social media API. From the chat/online window you can click on a friend and start a video call with them in the browser.

- Video conferencing / sharing files from inside an app. The user calls someone from inside an app and is able to share files through the peer to peer connection as they are talking via the data channel.

- Games that integrate video, audio and data sharing. A game where you see the video and hear the audio of your opponent and play the game live using WebRTC data channels.

- Apps that can draw down your contacts from say your LinkedIn account; you can instantly have an audio visual meeting without leaving your office.

- Call centres where you can have a full interactive conversation with sound and video. For example a Sky service call, they can not only talk you through the process, they can watch you do it and give you guidance

- http://www.nojitter.com/post/240001981/the-rise-of-webrtc-the-rules-of-communications-are-about-to-be-rewritten?pgno=3[Alan Quayle]

To see case studies used by IETF please click on link provided.
https://datatracker.ietf.org/doc/draft-ietf-rtcweb-use-cases-and-requirements/?include_text=1[WebRTC caes studies and requirements from IETF]

WEBRTC AND THE CURRENT TELECOMMUNICATIONS INDUSTRY

We will look at VoIP, video conferencing and how they will have adapt and change with the advent of WebRTC. It will also look at the most successful OTT audio visual communication company Skype.

WEBRTC DISRUPTIVE TECHNOLOGY FOR ENTERPRISE COMMUNICATIONS

This is an article from Chris Vitek of STC. STC is one of the largest self-governing bodies in the US telecoms industry. Chris is a very well respected board member of STC and an extremely successful business man. This article takes a very objective look at WebRTC and its future in the telecoms industry.

"The ubiquitous adoption of true peer-to-peer voice and video communications is upon us. It will be free, easy-to-use and compatible with SIP via SDP. WebRTC has been embraced by the IETF and the W3C just as SIP was 16 years ago. The ITU may have to sit this one out for a while since the architecture does not lend itself to centralized control. Which is the precise issues that contact centres and enterprises will need to address if they want to compete for the modern, smart-phone or tablet wielding consumer.

To help put WebRTC in perspective I have chosen to use some common use-cases for the technology.

Web-page initiated communications. With traditional tools there is a need to download an application that will support live communications. Multiple clicks and loading a new application on your PC is not the stuff of convenience, not to mention the enterprise security issues that come into play. HTML 5 has gone a long way to minimize the number of clicks, but it still requires the user to download the application. WebRTC is different. With WebRTC the voice/video application will be embedded in the browser (called via Java Script) so no download is necessary. The impact on the user is that a voice/video communications session can be initiated with a single click. Fast, easy-to-use and convenient and it will work the same way from a smart-phone, tablet or computer, except for Apple (more about Apple later). The simplicity of this approach will drive adoption by consumers. Imagine a LinkedIn hyperlink that connects you live with your business contacts. No more 10 digit dialling, no more phone directories and it works the same way to-and-from all of your devices. Now imagine an e-mail shipping confirmation that has embedded links for the customer service contact centre. If the package does not arrive on time the customer simply clicks the hyperlink and connects with a representative to get information about the status.

The use cases are compelling for both personal and enterprise use; however, the enterprise implementation of this technology can only be described as disruptive. Traditional communication are routed through a public switched telephone network (PSTN) using rout selection based on a multi-digit, DTMF dialling plan. WebRTC communications are not. They are routed via TPC/IP and DNS. Traditional trunks are not necessary nor are traditional switches. There will be a period of years where enterprises need to support traditional and WebRTC communications. This will be a time of great transition. Should the enterprise continue to support a traditional infrastructure? Should they explore SIP switches? Are the current crop of SIP switches compatible with WebRTC? Does the web team at any given enterprise understand the engineering issues surrounding WebRTC? The result of answering any of these questions wrong could cripple a B-2-C company in a matter of months.

Consider the office products business. 12 years ago 99% of their orders came through telephone conversations of fax. Today, 70% of orders come through the web. If an order goes bad, then the customer has to send an e-mail or talk to customer service. The latter case requires that the customer picks up the phone and dial a number. Then, the customers have to identify themselves, the order and the item. This all takes time. In the WebRTC model, the confirmation e-mail can have embedded links next to each item. These links can offer options to e-mail or call customer service about the item. If a call is initiated, then information about the call (item SKU, order number, customer’s identifier) arrives at the contact centre before the call arrives. Routing to the right agent and CTI are triggered by the associated data and the customers can solve their problems more quickly. These communications will be supported by SDP compliant, peer-to-peer signalling; however, they will not be handled by a telephony carrier, they will communicate directly with the IP enterprise infrastructure. Basically, virtual SIP-like sessions established on the enterprise

Internet infrastructure. Network security folks will have a significant hand in architecting these solutions. Where to place and how to configure the SBCs, WebRTC gateways, SDP protocol conversion, firewalls, proxies, web services, user agents and/or SIP switch (ACD) will all come into play. Further, there will be a need rethink processes in terms of this new communications infrastructure. Some of the most complex processes that customer service operations support today will become automated in ways that we never imagined.

WebRTC is available today on Google's Chrome browser (Alpha) and Opera Mobile. Mozilla/Firefox has an active development project underway. Microsoft has publicly supported the effort, but they have a lot on their plate with the roll out of Win 8 later this year. Voxeo has an SDK available. Apple offers a competing solution in the form of Face Time.

Skype took nine years to acquire 700 million users and it requires a downloadable application. Without the need for a download, WebRTC has the potential to triple the number of voice end-points in the public network within two to three years. At the same time it will create an order of magnitude increase in video end-points. Ubiquitous availability of WebRTC will drive adoption at a much faster rate than Skype. Contact centres and web developers will need to collaborate more than ever to leverage this new opportunity."

http://www.stcconsultants.org/forums/posts.asp?group=&topic=417632&DGPCrPg=1&hhSearchTerms=&#Post417632[Chris Vitek]

VoIP A BRIEF HISTORY

There are three main components to VoIP:

- Network
- Signalling
- Media

Network

The network gets the data from one point to the other. It has its own characteristics which then affect what types of signalling and media are used. In the case of the internet, we can generally speak about these characteristics:

- No guaranteed quality of service: data sent might not be received
- No guaranteed latency: data sent may arrive with different delay characteristics
- Heterogenic in nature: different parts of the network behave differently (Wi-Fi, LAN, Ethernet, MPLS, LTE, etc.)
- Asymmetric: NAT and firewall devices may restrict reaching certain addresses

Signalling

- Signalling is the addressing part of VoIP. It includes
- Registration and discovery – how do I tell the world how to reach me?
- Dialling – how do I dial, receive calls, drop them, etc.
- Capabilities – how do participants in the call understand what each side is capable of?
- Supplementary services – the usual suspects: hold, mute, transfer, forward, park, conferencing …

The above must be done in the same "language" between all participants – all should know how to communicate in SIP, H.323, and XMPP.

You can look at signalling for another angle, and that is non-functional features it needs to provide:

* Networking – IPv4, IPv6, UDP, TCP, etc.
* Security – things like privacy, authentication, etc.
* Connectivity – ability to connect endpoints no matter where they are. This focuses mainly on NAT traversal issues
* The non-functional features also fit in with requirements for the media.

Media

Media is what we're here for. The moment enough knowledge exists between the participants of the call, media kicks in and starts to flow between the participants to get us our call.

Media is usually thought of as the voice and video codecs along with their transport mechanism. In recent years, media got wrapped by vendors into components called media engines or media frameworks. These took care of the codecs and their transport. As everything else, these vendors tried (and are trying) to move up the food chain, and for them this means adding some signalling features into these media engines. It works great most of the times, as it eases the integration and the development efforts of application vendors.

WebRTC and VoIP

WebRTC is a media engine that target browsers. It offers a Java Script API that is being standardized, so it will be available to all web developers eventually.

WebRTC also include a bit of signalling into it – the non-functional NAT traversal mechanisms: STUN, TURN and ICE. Luckily, these 3 mechanisms are also the ones defined for NAT traversal use in both SIP and XMPP.http://bloggeek.me/voip-signaling-101/[Bloggeek]

When Google purchased GIPS and open sourced their codes, the VoIP industry had to look for alternatives. GIPS were the largest voice engine company around and almost every VoIP developer had some dealings with them.

With Google having open sourced the codes, it meant there was no real way you as a customer can get an SLA from Google for maintaining and improving the GIPS voice engine. And it is an issue, as a lot of GIPS customers used custom built voice engine packages – ones that are specific to their clients' needs.

The VoIP can still continue but it will have to adapt. It will have to include the audio codecs of WebRTC which will be awkward but not too difficult. They can take WebRTC and embed it into their own products – web based or not. Gateways will need to develop to connect the world of WebRTC and that of SIP and PSTN. The two worlds can live together once the industry adapts and evolves.

	Classic VoIP	WebRTC
Signalling	SIP or H.323 in most cases	Undefined
Media transport	RTP/RTCP	RTP/RTCP
Security	SRTP in SIP,H.235 in H.323	SRTP
NAT traversal	STUN/TURN/ICE in SIP,H.450.x in H.323	STUN/TURN/ICE
Video codecs	H.263, H.264	VP8
Voice codecs	G.7xx series of codecs, and then some	G.711, iLBC, iSAC

Chart by http://bloggeek.me/webrtc/ [Tsahi Levent-Levi]

SKYPE

Skype was launched as a start-up in 2003 and to date it has 100 million plus active users. Microsoft purchased Skype in 2011 for an astonishing $8.5 billion, which is basically around $1.000 per user.

Presently Skype are running an ad campaign called, "It's time to Skype". One of the more effective ads is titled "When did it become okay to text Mum Happy Birthday?" Skype believe that face to face communication is being lost in this digital age and it needs to be brought back.

While Skype may view the advent of texting and instant linkage through social media as a threat to its market share in communication, it may have a bigger threat on the horizon, WebRTC. Skype may just become obsolete or will it? Microsoft state they did not purchase Skype for what it is but what it will become. Three things have occurred recently which lend credence to that statement. One that Skype have started to adapt the VP8 video codec, at present this is the codec of choice for WebRTC. Secondly they have been actively trying to recruit developers to develop an app which will incorporate WebRTC and Skype. The app will allow communication between their legacy customers and WebRTC. The third move was partnering with Facebook, to offer Skype calling through Facebook.

These three issues indicate that Skype is heading towards a pure web based service, but more importantly it means that Skype are not afraid to change to fit the current or future market. And at present the biggest change of all is WebRTC.

Skype can and will adopt WebRTC when the time comes. They will do the necessary changes in their network architecture to fit WebRTC right into their business plan and continue to grow.

SOME NEW INNOVATIVE APP'S

- Asterisk/Digium. They are integrating WebRTC into their existing open source PBX which means that with a software update to their PBX, smartphones, tablets, laptops and desktops can become endpoints for all their PBX features and services. http://www.youtube.com/watch?v=8tBYyub1oC0[Digium discusses WebRTC and Asterisk] http://www.digium.com/en/[Digium]

- Bistri. Video chat which allows you to have video effects and take photographs/screenshots and share them on your social media site. Bistri runs in the browser, so there's no need to install additional software or plugins.http://bistri.com/[Bistri]

- Phonbooth. Similar to Bistri in all deatils

- frisB (Free global calling) provides free global calling between any web browser and any phone (or web browser) with no downloads. http://www.frisb.com/[frisB]

- Tenhands Is a desktop HD video service, it's free and built for business needs. They're in the process of adding WebRTC to their service. .https://www.tenhands.net/Home.htm[Tenhands]

- Vox.io is a simple service that lets you make voice and video calls straight from your browser.http://vox.io/[Vox.io]

- Twinlife. Similar to Vox.io but designed for the older generation. Simplicity

- Utribo (Software as a Service): "Connect" by Utribo is a service that enables subscribers to receive calls made in a web browser to their computer, phone, or PBX. Based on WebRTC, "Connect" provides voice and video calling capabilities.http://www.utribo.com/our-service/[Utribo]

- Voxeo Labs (Open source enabler for WebRTC services): Voxeo's Phono is a jQuery plug-in that turns any Web browser into a multichannel communications platform, capable of placing and receiving VoIP telephone calls from the browser, as well as handling real-time chat communications. jQuery is a cross-browser JavaScript library, so developers can do more and code less. Phono is a client-side solution and requires zero server-side logic on the part of a developer; all communication is handled by the Voxeo Cloud. Using the Phono plug-in, applications such as Your Second Phone have been created and are available in the Chrome Web Store.http://voxeolabs.com/[Voxeo Labs]

- Zingaya ("Call" button for websites) enables voice calls through any computer from a webpage. Visitors can click that button and the call is forwarded to the website operator's preferred land-line or mobile phone. All that is required is a website; all the visitors need is a browser and microphone.http://zingaya.com/[Zingaya] http://www.nojitter.com/post/240001981/the-rise-of-webrtc-the-rules-of-communications-are-about-to-be-rewritten[Alan Quayle]

CONCLUSION

Up until now you could do anything in your browser except for real time video calling, of course you could use flash but flash is not available on all devices or platforms. WebRTC gives you real communication and will take flash out of the equation.

Allowing the audio and video codecs to be open sourced under a very lenient open source license has made it very attractive to place WebRTC into commercial products. With Google, Microsoft, Opera and Mozilla backing WebRTC it would seem that all is well in the browser world. But until all the standards have been set by W3C and IETF, and adapted by all interested parties we won't celebrate the dawn of a new communication phase.

Aside from all the political posturing WebRTC is here to stay. It will bring great innovation, vast commercial opportunities and most importantly real time communication.

REFERENCES

http://www.webrtc.org/
https://www.ietf.org/meeting/83/index.html
http://browserid.org/
https://mozillalabs.com/blog/2012/03/experimenting-with-social-features-in-firefox/
http://mozilla.org/persona/
https://github.com/mozilla/socialapi-dev
http://www.voipusersconference.org/2011/webrtc1/
http://voxeolabs.com/
http://phono.com/
http://tools.google.com/dlpage/chromesxs
http://phono.com/webrtc
http://sites.google.com/site/webrtc/home
http://www.slashgear.com/google-webrtc-browser-voicevideo-chat-released-to-take-on-skype-and-facetime-01156079/
http://thenextweb.com/google/2011/06/01/google-releases-developer-preview-of-webrtc-its-open-real-time-voice-and-video-platform/
https://github.com/mozilla/mozilla-demo-social-service
https://plus.google.com/106186763111547737548/posts
https://developer.mozilla.org/en/JavaScript
http://hacks.mozilla.org/2012/04/webrtc-efforts-underway-at-mozilla/
http://files.labs.ericsson.net/apis/web-real-time-communication/videochat-example.zip
http://bloggeek.me/webrtc/
http://bloggeek.me/webrtc-series/
http://www.youtube.com/watch?v=9dpzUQmtumk
http://bloggeek.me/category/voip/
http://www.electronista.com/articles/11/08/03/skype.using.vp8.even.with.microsoft.patent.gripes/
http://www.webpronews.com/skype-welcomes-you-back-to-humanity-2012-04
http://bloggeek.me/webrtc-kills-enterprise-video-conferencing/
https://groups.google.com/forum/
http://tools.ietf.org/html/draft-ietf-rtcweb-jsep-00
http://www.pcmag.com/article2/0,2817,2332339,00.asp
http://www.w3counter.com/globalstats.php
https://www.ietf.org
http://datatracker.ietf.org/wg/rtcweb/
http://www.w3.org/2007/uwa/Activity.html
http://www.w3.org/2009/dap/
http://www.w3.org/2011/04/webrtc-charter.html
http://www.nojitter.com/post/240001981/the-rise-of-webrtc-the-rules-of-communications-are-about-to-be-rewritten
http://zingaya.com/
http://vox.io/
http://voxeolabs.com/
http://www.utribo.com/our-service/

http://www.utribo.com/our-service/
https://www.tenhands.net/Home.htm
http://www.frisb.com/
http://bistri.com/
http://www.digium.com/en/
https://plus.google.com/113817074606039822053/posts
http://tools.ietf.org/html/draft-ietf-rtcweb-rtp-usage-01
http://www.nojitter.com/post/240001981/the-rise-of-webrtc-the-rules-of-communications-are-about-to-be-rewritten?pgno=3
http://www.theverge.com/2012/6/27/3120273/skype-for-browsers-webrtc-job-listing-rumor
http://arstechnica.com/information-technology/2012/06/opera-12-arrives-with-webcam-apis-and-experimental-webgl-support/
http://www.opus-codec.org/
http://www.pkeconsulting.com/pkewebrtc.pdf
https://developers.google.com/talk/libjingle/
http://www.w3.org/2011/04/webrtc-charter.html
https://datatracker.ietf.org/doc/draft-ietf-rtcweb-use-cases-and-requirements/?include_text=1
http://www.stcconsultants.org/forums/posts.asp?group=&topic=417632&DGPCrPg=1&hhSearchTerms=&#Post417632
http://www.webrtc.org/reference/api-description
http://dev.w3.org/2011/webrtc/editor/webrtc.html#datachannel
http://lists.w3.org/Archives/Public/public-webrtc/2012Jun/att-0063/W3_Interim_June_2012_Data_Channel.pdf
http://techcrunch.com/tag/webrtc/
http://tech.dailybuzz.com/
http://gigaom.com/2012/06/26/skype-webrtc-web-client/
http://vline.com/
http://arstechnica.com/business/2012/01/hands-on-building-an-html5-photo-booth-with-chromes-new-webcam-api/
http://www.youtube.com/watch?v=E8C8ouiXHHk&feature=youtube_gdata_player
http://francisshanahan.com/index.php/2011/stream-a-webcam-using-javascript-nodejs-android-opera-mobile-web-sockets-and-html5
http://blog.gingertech.net/2012/06/04/video-conferencing-in-html5-webrtc-via-web-sockets/

ACKNOWLEDGEMENTS

David Kirwan

Eamonn Power

Miquel Ponce De Leon

Faraz Ur Rehman Sharif

Cathal O Connor