Introduction to WebRTC - 2015

The primary goal of this paper is to examine the status of enabling browsers to deliver real-time media (voice and video) as part of their basic capabilities. Within this paper, that capability is referred to as WebRTC, a simple name, first coined by Google. Within the industry there are different names for this, but in this paper the term WebRTC is intended to be a generic description of standards based real time enabled browsers. This paper uses public material from Google as well as the standards efforts to define what WebRTC is, some of the potential impacts, and the remaining issues that need to be closed to bring this technology to fruition.

Overview

WebRTC is an effort, started by Google, to build a standard based real time Media Engine into all of the available browsers. Since 2002 Global IP Solutions (GIPS - formerly Global IP Sound)) wrote object-code for the likes of Nortel (Avaya), Webex (Cisco), Yahoo and IBM to support their PC-based telephony applications. In 2010 Google purchased GIPS for $85M. In 2011, using the technology it acquired in the acquisition of Global IP Solutions (formerly Global IP Sound), Google has created an open source version of the WebRTC Media Engine and implemented it into Chrome. With WebRTC in a browser, a web services application can now instruct the browser to make a real time voice or video connection to another WebRTC device or to a WebRTC media server using RTP. With a HTML5 and WebRTC enabled Browser, a soft client is now just HTML pages from the server as the visual interface with WebRTC APIs and Media Engine to define the communications path. With the signaling and protocol standards coming from the IETF and the APIs for app developers from W3C, now communications can be defined and delivered by millions of Java Script developers, not just by a small number of SIP developers and VoIP systems vendors. The first WebRTC enabled browsers, Chrome and Mozilla, will come out later this year, in fact, the current Chrome browser has WebRTC hidden behind a flag, but the capability is there for testing and trials.

What this means for communications over the next five years could be merely interesting or transformational. Obviously for contact centers the impact will be significant. With the majority of contact center interactions being preceded by a web site visit, the change will be immediate. If each web page has a real-time communications object, the flow from the web information structure into the contact center work flows can be tightly integrated. This will quickly lead to convergence of the contact center with the web site teams in many organizations.

The impact for UC and general communications and collaboration may be much more significant, but longer term. For existing vendors, the ability to easily support soft clients on a range of devices will enhance their ability to deal with the exploding BYOD revolution. For example, within the Android environment, there are over 50 distinct versions/products to deal with. With WebRTC, a vendor should be able to build a small number of versions of HTML linked to device screen size and use the common WebRTC for media. The task of supporting a range of devices is reduced by at least an order of magnitude.

However, this raises a very interesting question; if I want to communicate with you and your "system" supports direct guest access using WebRTC, then merely pointing my device browser at your server and, assuming you are available and want to interact with me, presto...we are interacting. I am not using a client from my server, but one from yours through HTML and WebRTC. Similarly, a "web conference" can now be hosted by just sending the URL of the hosting server and having the attendees join. The value of us all being on the "same" system may be dramatically reduced.

Potentially, WebRTC and HTML5 could enable the same transformation for real time that the original browser did for information. In 1990, the challenge was to have servers interact to move information between individuals. Email (and somewhat IM) is the last remnant of that era. After the browser emerged, the interactions process changed to the end user pointing the browser at the server where the information or application resides. Similarly, WebRTC may enable me to point my browser at your server and interact with you without having system level federation. The result of this change, combined with search to find the places to point at, created the Internet revolution that has changed industries, societies, and politics. Can WebRTC and voice/video enabled browsers be the genesis of yet another transformation? With a certificate from LinkedIn indicating we are connected, can I now communicate with you on your server with a simple browser?

WebRTC Technology and Standards

Every real time communications client requires three elements; a framework, a visual user interface, and a Media Engine. These three components are shown in Figure 1. The white box labeled Control and Apps is the visual interface, the blue box is the media engine, and the rest is the framework. In a typical hard client such as an IP phone, the framework consists of the processing chips and the OS. In a soft client, the framework is the device/OS the client is running in. The visual interface can be a hard interface such as a phone key pad or a screen presentation in a PC or other device. The function of the Media Engine is to manage

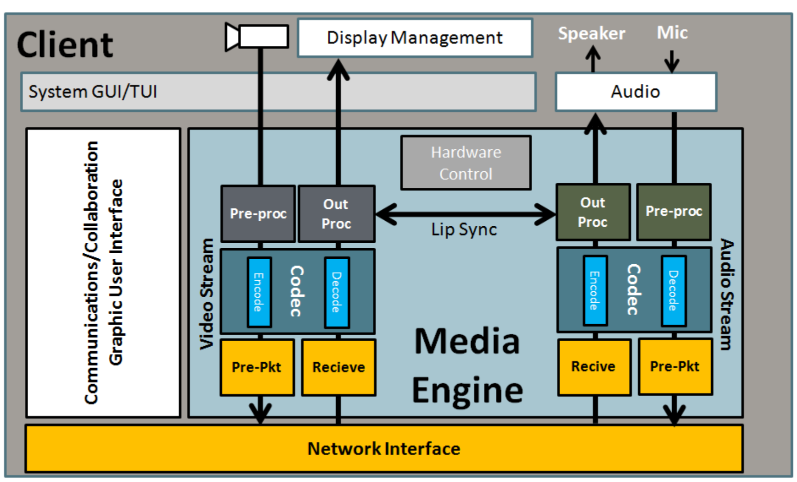

Figure 1 Media Engine Components

the real-time transmission and receipt of a video/audio stream.

The Media Engine includes a set of functions that deliver quality voice and video:

- 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 WebRTC effort is to take the Media Engine, combine it with a set of standard APIs and create browser based solution to real-time communications. In Figure 2 the WebRTC implementation of a Media Engine as part of a browser is shown. 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 signaling. By merely telling two WebRTC devices to communicate, the server can initiate a IP based voice or video communications.

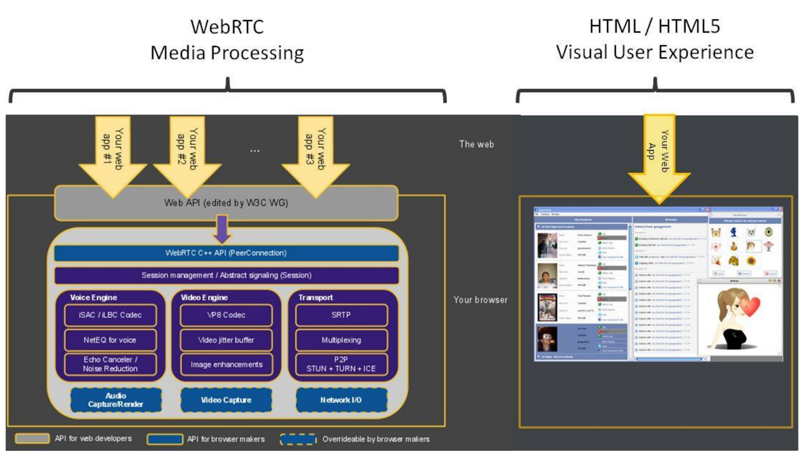

Figure 2 WebRTC and HYML5 in a Browser

WebRTC is being standardized in two bodies; the protocols and interoperability are being driven in the IETF and the APIs for web development are being driven in the W3C standards body. The IETF RTCWEB WG was formed after a BOF at IETF 80 in April of 2011, and is actively generating RFCs in moving to a standard. In W3C the WEBRTC WG was created in May of 2011. The W3C WEBRTC WG is developing high level APIs and device control (microphone, camera, network) as well considering a Peer Connection

API proposal originally proposed in WHATWG: http://dev.w3.org/2011/webrtc/editor/webrtc.html. While standards activities are sometimes volatile, there are multiple browser implementations today and expected standard implementations in late 2015. The most significant issue to be resolved is which codecs will be mandatory in the standard. This is exacerbated, especially in video, by the royalty nature of some codecs and others being proprietary to a specific vendor. There are currently multiple plug-ins for IE and Safari. Microsoft has committed to support for ORTC, that is being integrate into the standards. And if Microsoft and Apple get on board, the advent of browser enabled communications is almost upon us.

Peer to Peer

Often, WebRTC is referred to as Peer to Peer communications. This should not be confused with Browser to Browser communications. While WebRTC can be delivered in a browser, it can also be in any other end point device. As many new endpoints are in fact becoming browsers, the capability to use WebRTC in a variety of devices will be significant. For example, WebRTC could be in a television, car, a toaster, or even your clock radio. With many new televisions incorporating significant processing power and cameras, the ability to use WebRTC for home telepresence is in the near future.

In addition to a plethora of potential end points, a peer can also be a value add point. For example, a Media Server could be a peer, or a gateway to the PSTN. This capability to incorporate peer services in the media stream will enable advanced capabilities far beyond simple point to point connections.

Triangles and Trapezoids

Having browsers with real-time capability will open a new set of real-time applications. While it is not possible to anticipate all the potential new applications, there are some examples that can be foreseen. It is important to think of this as more than a simple PC technology. As more and more devices such as smartphones and tablets have WebRTC enabled browser capability and the 4th generation wireless networks enable continual use, this may become the core of all device communications. In fact, there is no requirement in the WebRTC standard that the device actually have a browser.

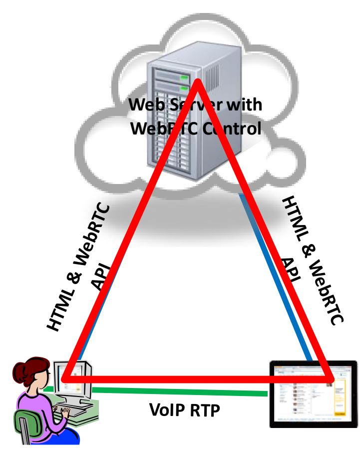

Figure 3 Basic Web Server with WebRTC Control

The first and most obvious application is to enable two browser devices to open a communications path. This is shown in Figure 3. Here a web site has caused the WebRTC client in both a desktop PC and a tablet to connect to each other with a direct RTP/IP connection. In this case the control is coming from the web server, but the media path is being delivered peer to peer. This enables the web site owner to enable communications on the site without requiring the site or hosting to actually provide or manage the bandwidth between the devices. This level of application is appropriate to add communications to any site that could identify two parties that might want to interact. For example, LinkedIn could use this to enable its subscribers to open a voice channel. Or any site that was based around individuals relating to other individuals could quickly and easily add real-time communications. If the seller of an item was connected to eBay, a connect icon could appear on the item page, and by simply clicking it a potential buyer could be talking to the seller. The key point is that enabling this capability is a small number of lines of Java Script code in a web server. Now the server defines the user experience through the HTML5 browser, versus a client that the user came with. This enables the user experience to be customized, in the same way a web page provides a customized experience based on the information and purpose. For example, a user interface for an overlay to a social networking site might look very different from one that is a sales site. The experience is being pushed from the server, instead of communications being pulled by the client as in traditional telephony.

In the WebRTC community, any communications where both peers are controlled from a single server (or server system) is referred to as a triangle. This is differentiated from a trapezoid, which follows current telecom systems where there are two servers.

If vendors decide to deliver communications services to end points using WebRTC, then they can deliver direct connections between them on different vendors servers using the WebRTC standards. This opens the possibility of interconnected systems using WebRTC as the standard. As shown in Figure 4, the WebRTC standard does not define the mechanism for communications between the servers, merely the control of the client and the communications between clients (or media servers). So two different vendors could have their UC control systems interoperate using SIP or any other means and then each tell their respective end point device to connect to the IP address of the end user node on the other system using WebRTC. As the end point devices are merely connecting to each other, this implementation may enable easy inter-enterprise communications. Because each device is just following the API instructions from the server, the fact they are actually talking to two separate servers does not preclude them from now connecting to each other directly.

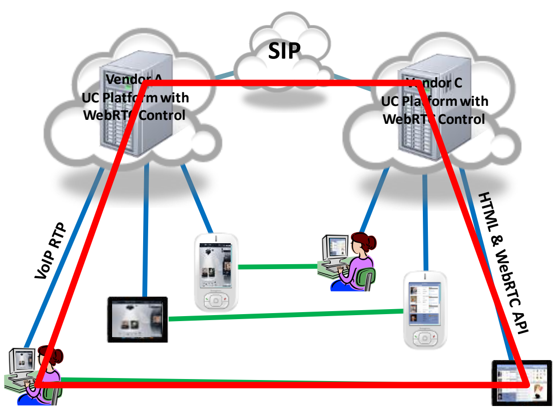

Figure 4 Inter-system Communications with WebRTC

Media Servers and Gateways

By adding a Media Server capable of conferencing and other services, WebRTC can now be used to provide conferencing, as shown in Figure 5. The Media Server function to deliver this can be co-located with the control web server or can be in a totally separate location. This opens a new set of applications possibilities. For example, organizations that work with groups can use this to facilitate meetings in a controlled fashion. Whether a business, government or non-profit, the ability to rapidly set up communicating groups will be much easier. In fact, creating a simple web conferencing site with HTML5 and WebRTC would be fairly easy. And, as there are no requirements for downloading clients or belonging to a group, simply sending a URL to all of the participants is sufficient to kick off a meeting. The addition of a Media Server opens the door for added value services such as recording and meeting indexing. Again, the point is that adding this to an existing web server is easy, though the cost structure of the media server will require business models that defray that cost.

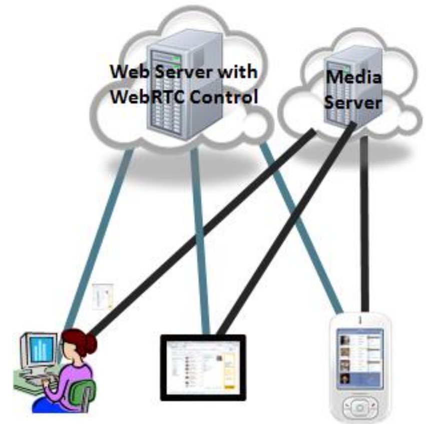

Figure 5 WebRTC with a Media Server

SIP Integration

WebRTC can be integrated with SIP in two ways. There are provisions in the standards that should allow two devices, one controlled by a ISP server and one by a WebRTC based web server to actually interact with direct peer to peer in the media channel. This is shown in Figure 6.

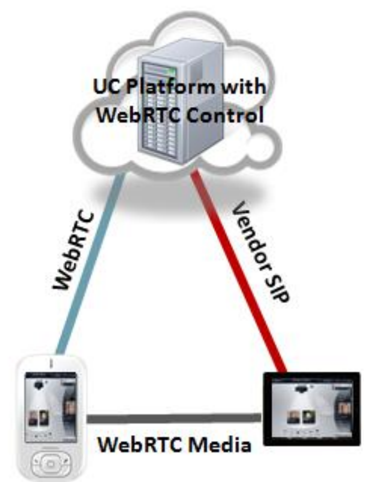

Figure 7 WebRTC and SIP Clients with Direct VoIP Connections

The alternative is to use some form of media server between the two devices. This could be integrated as a conferencing platform, an SBC or a pure media server. While an intermediate tether point has great value, one critical issue is maintaining common codecs end to end, so as to avoid transcoding that degrades audio/video quality as well as potentially significantly increasing latency. In this case the media server is acting as a peer to the WebRTC web server and terminating the WebRTC connection and then mapping into a similarly terminated

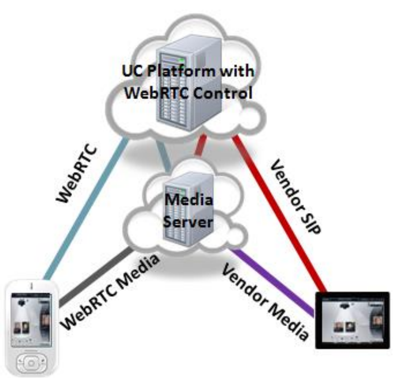

Figure 6 WebRTC and SIP Clients with Media Gateway

SIP session. Similarly, this is how most PSTN/TDM gateways will be built.

WebRTC in the Enterprise

The enterprise has numerous uses for WebRTC for general solutions that are not vertically specific. WebRTC enables BYOD (Bring Your Own Device), next generation customer interaction through web integration, and opens the potential for direct WebRTC communications between employees and the outside world. Because WebRTC creates an experience that comes from the host web server, the experience the enterprise provides can be unique and tailored to deliver advantage. Legacy Contact Center Integrated with WebRTC through media servers

WebRTC and BYOD (Bring Your Own Device)

WebRTC can be used to integrate open BYOD devices into a more traditional Unified Communications architecture. For vendors like Avaya, Microsoft, Siemens, Shoretel, and Cisco, the use of WebRTC is a logical way to integrate another set of devices. However, unless the vendor chooses to make their SIP operated devices to have a direct capability to integrate to the WebRTC devices as shown in Figure 6, this may require a gateway. In Figure 6, the device on the right is using SIP control with the server and the device on the left is using WebRTC. By assuring that the proprietary SIP device can receive a WebRTC connection, this would enable integration with open WebRTC devices. This implementation presents an easy way to accommodate both the explosion in BYOD devices as well as a mechanism to create an easy "guest" capability. In this guest mechanism, anyone with a WebRTC browser device could "come" to the corporate UC Platform and use that device to receive integrated UC communications and collaboration with the users inside the company. One potential ramification of this is the elimination of the need for what has been described as "federation". If anyone can go to anyone else's system and get essentially the same level of capability as the direct participants, the need to federate between systems may go away. If I need to collaborate with Bob at XYZ Company, I just point my browser at the Guest URL on his system and now we are collaborating with all of the tools his system can provide.

However, integrating between WebRTC and SIP at the endpoints is not the only way to deliver this. It can also be done through a "gateway", probably implemented as a media server that both devices can connect to. Figure 8 shows this design. In this architecture, the WebRTC client is talking to the media server using WebRTC signaling, protocols, and codecs. Similarly the SIP client is using SIP signaling, protocols, and codecs based on the vendor choices. The media server is providing any required translation of the codec streams as well as providing a port level interface that will connect to each client. As shown, the media streams may now be different due to the choice of codecs that were made for the client. Acme Packet demonstrated a WebRTC to SIP gateway implemented in an SBC at the WebRTC Conference and Expo in November, 2012.

The combination of HTML5 and WebRTC opens the world of BYOD (Bring Your Own Device) in a powerful new way. With HTML5 and WebRTC, any compliant device can become a highly integrated end point without running a local application and without local data storage. By implementing a true cloud model where the only data on the device is that which is to be displayed, sensitive data is not exposed outside of the enterprise except when it is an authorized use. This is a solution to the huge issue of maintaining privacy and compliance for data. By only sending the data to the device that will actually be displayed and using the built in HTML5 and WebRTC technologies, a new generation of highly secure implementation are possible. With the emerging 4G networking technologies, the performance and feel of these "applications" will be equivalent or better than current "local apps". In fact, the vast majority of meaningful apps in today's smartphone and tablet world are just local presentations of cloud/server data or information. And, of course, a communications applications only has value if you actually have a network to communicate over, a single phone connected to nothing is one hand clapping....

Redefining Customer Interaction

Virtually every company today has a web site. Some are sophisticated, representing the way the company does business, while others are simple and basic. Regardless of the level of web presence, for many customers and prospects, the web site is where the interaction with the company starts. When the customer or prospect concludes they cannot complete their needs on the web site, they move to the phone and the call center. However, this transition generally loses the context that was developed on the web site. In fact, 70-80% of contact center interactions in western business is proceeded by a web site visit.

WebRTC enables the customer interaction to come directly from the web pages and drive how that interaction is handled through the business logic of the web site. The benefits of this are two-fold: the transition from a web to a real-time experience is potentially more seamless with the web actions defining the skills required to meet the customer need and also to provide new information about the success or failure points of the web site interactions. By monitoring and analyzing which web site actions and events led to the highest requirements for additional agent interaction, the process of web business and better outcomes can be enhanced. By integrating the agent driven interaction into the web environment, it is now possible to define where the issues are with customers completing their needs through the web. With an average web interaction costing a small fraction of an agent interaction, this optimization can return huge benefits. As shown in Figure 8, each agent/customer pair is, in fact, a WebRTC triangle, without a service provider or other entity.

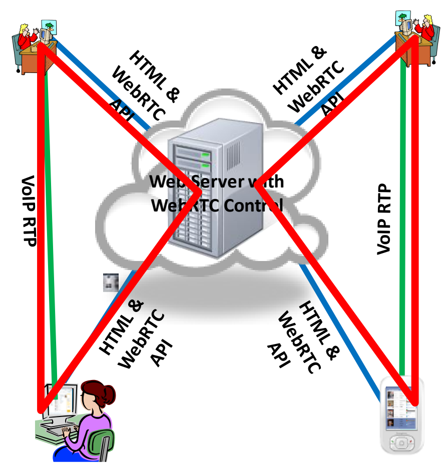

Figure 8 Agent Integration with WebRTC

For organizations with agents, the use of WebRTC enables a new paradigm where a direct connection is started by clicking on an object on any web page. As the simple connection logic is now built into the web server, this can drive the agent selection, but the WebRTC is used to connect the users browser to the agent.   One key value is the capability for the agent to be on any WebRTC device. While this type of system could be integrated using SIP devices on the agent side, using WebRTC enables similar device independence. This would make incorporating home agents much easier as their device type would not be important to how they interacted with the control system. As all of the communications services are now integrated into the web server, as the customer navigates this can be used to alter the inter-human experience.   The lack of high costs in this model may develop new broker models for many activities. A web site could represent thousands of micro-consultants, and using WebRTC connect them to an individual needing a service.

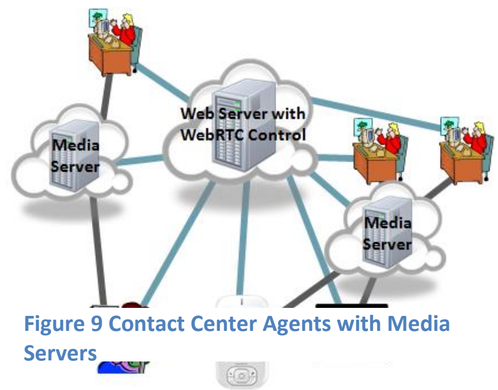

Figure 9 Contact Center Agents with Media Servers

This integration between the web side and the contact center side can be accomplished in two paths: a completely new system, with the web site at the core, may be the right path for some businesses. However, for many businesses that have an extensive investment in telephony based contact centers, integrating WebRTC into those environments may be the right decision.

Of course, adding in a media server into the mix as shown in Figure 9 enables functions like call recording, but it also enables other capabilities such as IVR, moderated interactions, speech recognition based tools, etc. As shown, there can be multiple media servers, either as platforms or as a cloud service. The media servers can be mixed, both in type (premise or cloud) and in network/geographical location. This implementation can have different and new capabilities. For example, when looking to buy something, the review site could become a conference with a knowledgeable agent. When going to another page the agent/conference could change. In this case the "conference" is tied to a page with a moderator linked to the same page. As the WebRTC connection is defined by the active web page, a model where the end point moved from one conference to another as the web pages changed would be easy to implement. By having an agent actively monitor certain pages and making WebRTC "on" and live for those pages, an open interaction with all current page "readers" could be deployed.   This concept of context and state related to a specific web page is an interesting capability that WebRTC enables.

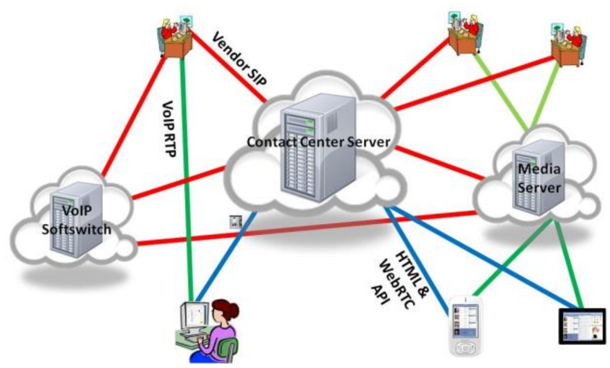

Figure 10 Integrating WebRTC into Existing SIP Contact Centers

Of course, this can all be done as an extension to an existing contact centers shown in figure 10. All of the existing contact center vendors have discussed the potential of using WebRTC as way for customers to come into the contact center. In fact, most have some form of click to call, whether implemented through the PSTN or even through flash technology. With WebRTC, this capability will become ubiquitous and the choice for future integration will be based on whether to extend the legacy contact center into the web world or begin anew customer interaction methodology based on and integrated to the company web site.

The Enterprise Portal

One potential significant application of WebRTC in the enterprise is an enterprise portal that enables external access to individuals through WebRTC. The concept of an enterprise portal is a web site that allows external access using WebRTC. An example would work like this:

On the "Contact Us" web page there would be a link "Web Interaction" or "Browser Communications". That link would link to a URL/web server that is the enterprise portal.

Upon arriving at the enterprise portal, the visitor is asked to enter the name of the person they want to interact with. This is to avoid the use of the portal as a way to "walk the org chart". Alternatively the company may just decide to publish a directory. This may be more common in SMB than in large enterprises. By entering a name for an employee (or something close based on search corrections), the user is offered "Is this the person you wanted?" with the actual name or names. Where there are multiple people with a similar name (John Smith), titles or departments can be used to enable the visitor to choose which one is right.

After selecting the employee, the visitor is taken to the employees "access page" The access page can have presence and availability, potentially tuned to who the visitor is based on cookies or other certificates such as LinkedIn or Facebook. For most visitors, the page would offer an opportunity to interact by entering the visitor's name and a short note why an interaction is needed. This page can be customized to the enterprise or the user, by asking for specifics, for example it could have a radio button for custom or client so the visitor could indicate their relationship to the employee.

This request can then be sent to the employee, enabling her to decide if it is important now. But it also can become a form of instant chat. If the employee is in a meeting, she might type a response saying, "come back at two". This would be displayed to the visitor. An extended chat is also possible, without registration or a common server other than the portal. If the employee wants to interact, she can push the interact button and the visitor is connected through WebRTC. On the employee side this could be done with WebRTC or through the proprietary implementation of the enterprise vendor, using SIP for example.

After the interaction starts, the user experience of the visitor will be defined by the portal, resulting in a common experience between the employee and the visitor that reflect the company and its implementations. Through the power of HTML5, the experience can reflect a powerful experience.

The key point of the enterprise portal is that each visit by a visitor to the web page is a unique experience, just as a visit to Company A's web page does not have any direct interaction with a visit to Company B. In this way, a significant part, if not the majority of external interaction may rapidly move away from traditional phone numbers to the portal. In fact, as employees become familiar with the capability to manage their interrupts through the portal and as companies add contextual services to their offering, it is reasonable to believe that many employees may have a voice mail message stating, "Hi, I do not accept voice calls by phone or listen to my voice messages, if you want to speak to me, please go to my Personal Portal page at www.abc/contact/bobsmith."

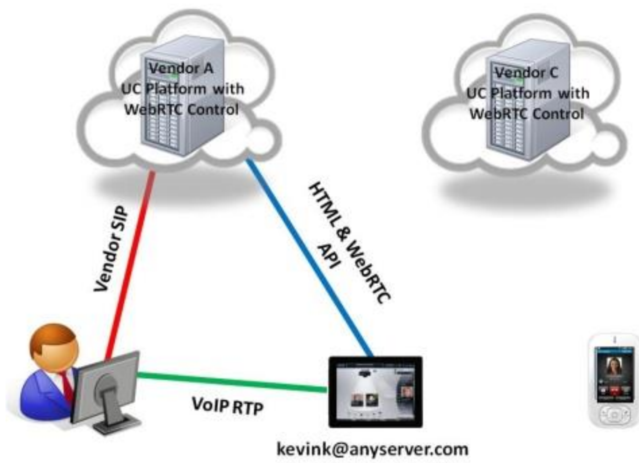

Figure 12 Connection to Vendor A Portal

Examining how a portals might work is a sequential "federation" model without the complexity of making diverse systems interact and match their separate visions and implementations. For example, a sequence of talking to Avaya, Cisco and Google might go like this. First a connection is made through the web server to "kevink@anycompany" (I am using the email shorthand for clarity, in fact it would be a URL). As this is an Avaya system, the visitor gets the complete Avaya user experience, for example the Flare version today. This is shown in Figure 11, Next the visitor goes to the company using Cisco to talk to johnc@companyserver.com as shown in Figure 12. Now the experience is defined by the Cisco system and would reflect the Jabber experience and tools base on the WebEx family of collaboration. Finally, by going to larryp@giantweb.com. In this case the experience might be defined by Google circles and new web tools. In each case the event is totally independent. while there may be links throughout the web to communications addresses representing users, companies,

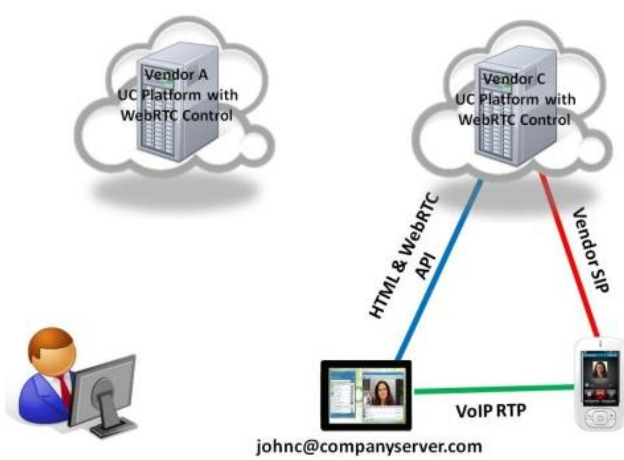

Figure 11 Connection to Vendor C Portal

services and more, each becomes an independent event, with an experience that is optimized toe the goal of the real-time communications for that site.

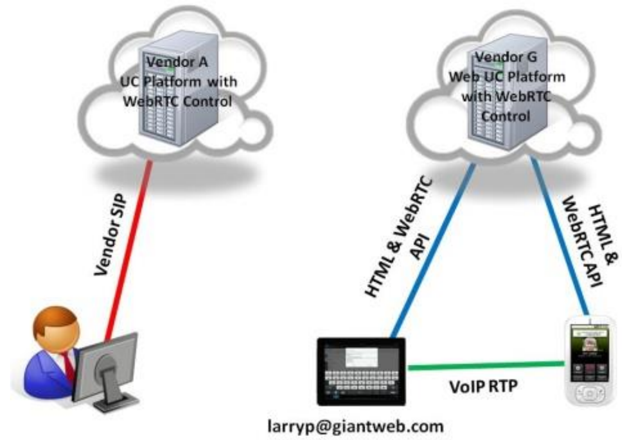

Figure 13 Connection to Vendor G Portal

WebRTC and Telcos/Mobile Operators

Telco Service Providers and Mobile Operators see WebRTC as both a threat and a potential way to create new services. While WebRTC enables new Over the Top (OTT) services, similar to how Skype introduced "free' VoIP, it also has the potential of extending their services in new way.

The potential impact of OTT through WebRTC is inherently obvious. Today we find people for phone calls through phone numbers and their administration by service providers. With the advent of WebRTC, rich communications experiences are possible without having a server negotiate on your behalf. As these services become available, it is entirely possible that the concept of buying a phone number may go the way of AOL.

However, WebRTC also enables service providers to offer new rich services that leverage their networks and other capabilities. By combining WebRTC with new networking like 4G and VoLTE, telcos and mobile operators may have the formula to create rich new services themselves. With the new device capabilities these services can extend far beyond their current capabilities. Much as the television delivery companies (cable, telco and satellite) have begun to deliver their content through non-tradition web/internet delivery, the telcos and mobile operators can leverage WebRTC to extend their service beyond their captive devices. For example, the upcoming VoLTE (Voice over LTE) and RCS (rich Communications Suite) promises to enables virtually all Smartphones to interact with each other through service tightly integrated to the user experience in the device (think of how FaceTime works in Apple, but across all devices). As shown in Figure X, this service can be extended to non VoLTE/RCS devices, such as a television or a WiFi tablet. It also can be used to deliver a quality experience for people wanting to interact with the VoLTE/RCS subscriber, using techniques similar to the enterprise portal. In this case the phone number may still be the primary identity, augmented by other services.

Thinnking outside the Telecom Box

Each of these potential implementations uses the WebRTC open design and multi-device availability to potentially re-think how communications is managed. While these examples represent relatively linear extrapolations of today's communications systems, the much more interesting ones may be based on adding real-time media to systems in ways that have not been thought of before. For example, WebRTC could be used to deliver large multi-party audio (or video) in gaming applications using HTML5 as the interface. Alternatively, as the WebRTC Media Engine will be in the device, even if there is a separate application running, that application can use the WebRTC to do the real-time audio. So a multi-player game as shown in Figure 14 could use WebRTC as the audio delivery path. In addition, with face recognition software and capture, the avatars in the game could now have actual user faces overlaid.

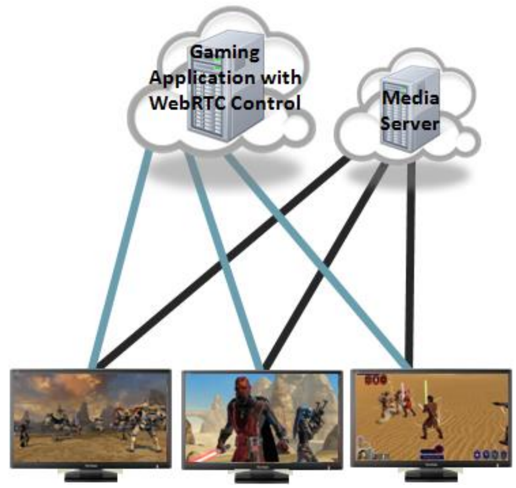

Figure 14 Gaming Using WebRTC for real-time audio and video

This opens the door to standardization of audio environments and server delivery. While we can think of this as a multi-party game, the same could apply to many new applications like social networks. The groups on LinkedIn could now become virtual conferences with active users interacting. By adding spatial environments and spatial audio, virtual worlds could be accommodated on any device.

Finally, adding real-time media to any applications will be possible. While the opportunities to enhance such a site as eBay are obvious (seller vide conferences at given times where the seller demonstrates the product for sale),or for traditional social sites such as Facebook or Pinterest, the reality is that it will open the door to totally new sites. Imagine the Barney (purple children's figure) site where children can now interact with each other and with a computerized Barney by clicking the "Barney Walkie-Talkie" that invokes WebRTC. Just as the original web created not only discontinuities in existing businesses, but whole new business models, WebRTC will do the same for real time services and applications.

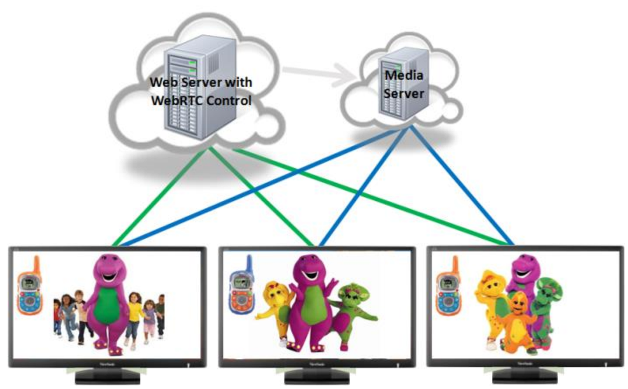

Figure 15 Barney Site Chat with walkie-talkie icons

Other examples of applications are already emerging. One company that focuses on-line music training and tutoring is planning to use WebRTC to enhance the remote experience, just a preliminary pre-cursor to using WebRTC in a plethora of distance learning applications. Another application, called Joyride was shown in early 2013 by Vobi. In Joyride, your friends can suggest music for you to listen to while you are driving and then join you on e audio conference in with you in your car. In this case, your presence in your car is the context to enable a new class of applications.

We will no doubt see explosive new applications over the next five years. Just as the web has created billions of dollars of new businesses while disrupting existing businesses, WebRTC promises to change the world in powerful new ways.

WebRTC Benefits

The benefits of WebRTC can be expressed in two ways, from either the web server or the user point of views. Together these two benefits define the way WebRTC will impact the communications environment.

WebRTC enables any web server to deliver a unique real time communications experience, with simplicity and reliability, without dependence on service providers or other services.

WebRTC enables users to participate in a communications experience as delivered by any web site without downloads, registration or general cost.

In order to better understand these benefits and how they relate to the impact of WebRTC, it is important to understand the meaning of them. First, the elements of the server side benefit.

- " ***WebRTC enables any web server to deliver a unique real time communications experience***" The critical point is that with WebRTC any website can become a control and delivery point for real time communications and that the web site, through HTML/HTML5 controls the experience that the user receives. With tens of millions of operating web sites, this opens up communications in a major way. This is fundamentally different than today for most communications experiences where the near end control defines the experience, regardless of the destination. Further, that experience does not have to reflect traditional telecommunications values. For example, an image of a walkie talkie could represent the initiation of communications, rather than a more traditional telephone. And real time can now be added to any web site. Iin another example, if I were to tag an image or other point using Pinterist, that pin could now have an easy representation that I am willing to talk about that interest point. The initiation of communications is now through Pinterist, not through a phone call.

- ***"... simplicity and reliability***," WebRTC delivers simplicity, and through that reliability and availability in two ways. First, by putting the elements of communications into the browser in an open standard it eliminates the complexity of developing separate soft clients for each device. With many devices, operating systems, and even in Android skins, the challenge for anyone deploying a platform today is that complexity of support. Every time an OS changes, the client must be re-certified. With WebRTC that challenge moves into the browser realm, where the number of touch points is dramatically lower and there is an eco-system of interoperability that has been established. The second critical element of simplicity is that the WebRTC client is stateless and uses stimulus input through the graphic side of the browser to the server to initiate state change. When we started to deliver VoIP in the mid 90s, the model for real time on the IP infrastructure was H323 operating between PCs. In this case the end point was stateful (it understood its own state) and capable of local state changes. As we developed the initial VoIP systems, this option was considered and rejected as introducing huge complexity and unreliability. The telephony system had been developed in a model where the end points were not independent units, but rather presentation level interfaces of the core. All of the initial developers of VoIP followed this model, not the H323 model. Unistim from Nortel, Skinny from Cisco, and "H323" from Avaya all did the same things, inputs at the device were sent to the server and the server instructed the device what to do. The state of the devices was maintained on the server. As SIP developed, the concept of an intelligent independent end point was driven into the overall architecture. In SIP, each end point is both stateful and capable of self state change. This has led to increased complexity and the general lack of interoperability that exists in SIP today. WebRTC is a return to a stateless implementation where the stimulus input is through the visual browser interface and the WebRTC media engine is under the control of the web server. This dramatically simplifies the implementation.
- "... ***without dependence on service providers or other services***." In the pre- WebRTC world, the ability to engage in communications is dependent on one of three elements: participation in the PSTN, membership in a separate IP based community, or separate applications with clients. IN the case of the PSTN, membership through a service provider is required, and the PSTN only delivers the most rudimentary of LCD (Least Common Denominator)services: an assurance that each telephone number is a unique representation of a physical location, a mobile device associated with a user, or some service, and voice communications that cannot exceed the G711 standards, can only be degraded. In the case of communities like Skype and Lync Federation, communications requires buying a product or subscription. finally, services such as WebEx and Go-to-meeting require that the participants download clients for the experience. In all of these cases, the requirement for a third party to be involved in the communications system limits the scale and diversity. With WebRTC, there is no such requirement. Each web site is essentially its own "service provider", without a requirement of any relationship to a party outside of itself and the user it is enabling to communicate.

From the user perspective, the benefits are similar.

- "WebRTC ***enables users to participate in a communications experience as delivered by any web site***" with WebRTC a user can go to any web site and immediately have that web site deliver a communications experience that is unique to that web site. Instead of having communications options defined by a few service providers and applications, now the user can literally choose a web site and have the experience be unique to that site. In this way communications is no longer a separate event, but part of the overall experience of visiting that site.
- ***"... without downloads***" Not having to download a client or plug-in for each communications experience is an obvious benefit, especially when the emerging number of devices, both private and public are considered. With televisions, cars, appliance, kiosks all becoming web enabled, having a client or plug-in is virtually impossible to maintain. For each service a separate plug-in would mean one for each site. Most of us have pages of identities for each of the sites we visit, imagine if each of those sites offered their own communications client? With WebRTC, whether a site is visited daily or once in a lifetime, the user does not need to undertake any separate activity to enable real time communications.
- ***"... registration or general cost***." WebRTC is not a service or a vendor, it is a standard. As such, when web sites are WebRTC enabled, any user can participate in the communications experience at that site without separate registrations or cost. The user is now free to immediately use real time from any web site without having to join a group like Skype or have Lync for federation. The gatekeepers of communications are moved from a position of control and mandatory tolls to optional when required.

WebRTC as a Game Changer or Disruption

To understand how technology will impact the future, it is critical to think in the context of the past. As George Santayana said, " Those who cannot learn from history are doomed to repeat it". The history to which I refer is the original introduction of the World Wide Web (www or web for short) and the original HTML browser. When these technologies were introduced in 1991-3, there were many who said that they were not interesting or transformational. In fact, the quote "I can do that with AOL" was oft heard at that time. What was not clear was the impact of an explosion of users and web sites and the emerging relationship of open movement between them. The pre- web/browser world was a world of dedicated server connections and defined services. With the browser and the web, user were freed to go directly to where the information they wanted resided.

WebRTC creates the same transformation in communications that the web and browser created for information. No longer will users be limited by their server to who and how they communicate, but rather freed to use the web paradigm to connect to any server that can define a communications experience.

To really understand if a technology is disruptive, three areas must be considered:

- Core Technology - The industry the technology is actually in - the segment base
- Delivery - Industries or services that deliver the technology to customers
- General - Adjacent or other industries that are impacted by the change

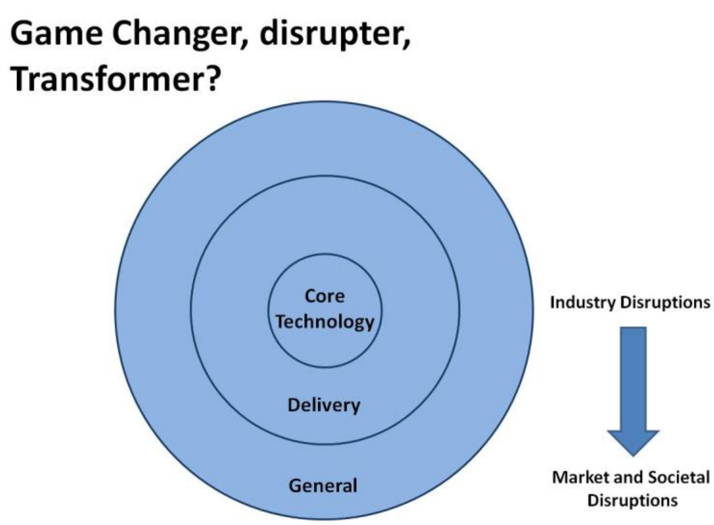

Figure 16 Impact of Transformation

If we look at the original web/browser, it is clear that all three were impacted in major ways. The Core IT industry changed and introduced players that used the web/browser to develop new products and displace existing players, however, it was not totally a disruption. Microsoft stayed a player with the browser and eventually Bing, IBM is still there, and others have continued. In the services segment the change were more dramatic, for example AOL disappeared and Yahoo emerged. Finally, the web/browser impact many other industries. From classified ads that were replaced by eBay and craigslist to changing the model for banking and insurance, from retail to even entertainment, virtually every industry has been changed in dramatic fashion. Finally, entire new multi-billion dollar business segments have been created, from search to social , dating to sex. We would definitely say that the web/browser was disruptive.

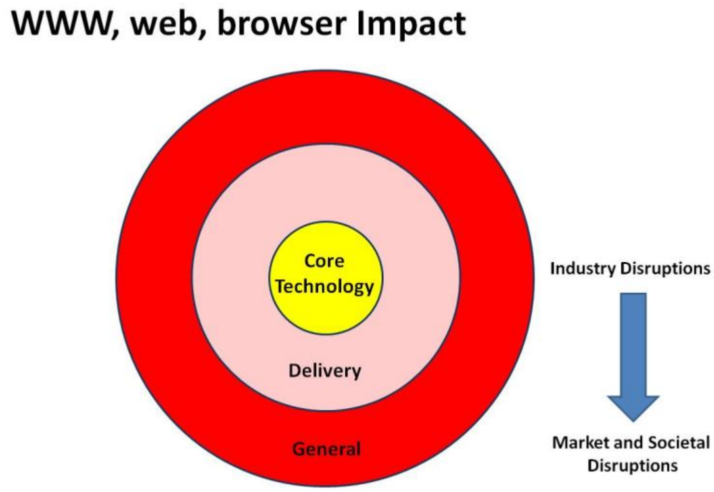

Figure 17 Impact of the WWW and Browsers

On the other hand, VoIP has been much less disruptive and more evolutionary. While it had a major impact in the center in the telecom world, with the emergence of Cisco and others, it did not cause a complete change. In fact, the technology was adopted by the incumbents. In the service segment the impact was even less. While some channel partners changed, generally, the service providers and channels for telecom remained the same. While Skype and SIP trunking have impacted the business, they have not caused landscape shift that are dramatic.

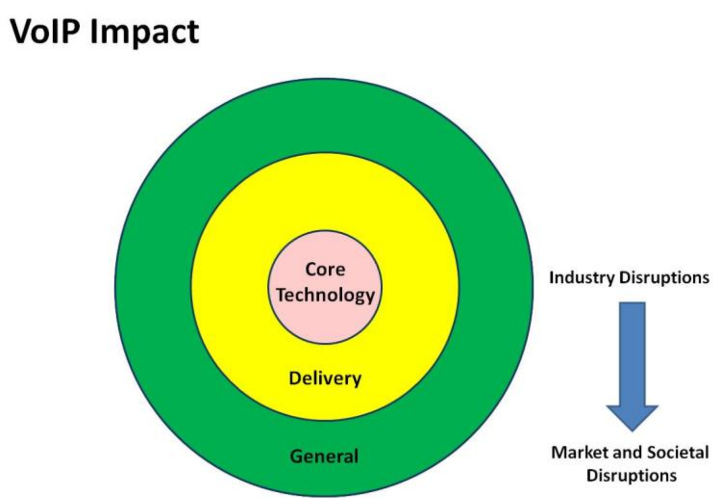

Figure 18 Impact of VoIP

Finally, the impact of VoIP on the customer business has been purely evolutionary. It has not enabled major shifts in other industries nor caused new industries outside of telecom to be created.

So, for a technology to be truly disruptive, it must create major changes outside of the core industry.

Evaluating WebRTC on this scale can lead us to understand the potential for disruption.

Telecom Equipment Industry - in the core industry, WebRTC is a potential major change element. It allows new players in the conferencing and web conferencing spaces to enter quickly. It pushes the web site and contact center parts of organizations together, opening the way for market share gains through that change. It allows could vendors to change the landscape. However, it will be adopted by the existing vendors for BYOD, guest portals, and contact center, assuring that they will not be totally displaced.

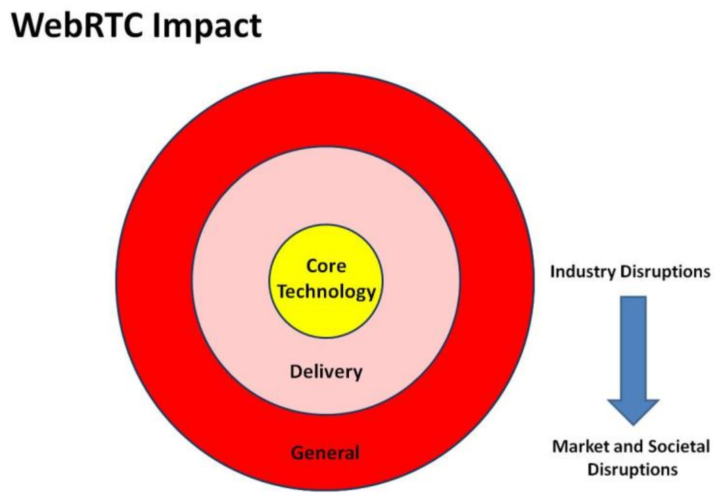

Figure 19 Impact of WebRTC

The Service provider Industry - WebRTC could be a much larger disrupter in the services space. For traditional telecom service providers, the ease of new entrants building OTT and site dependent communications could reduce the need for their services. Enterprises adopting WebRTC for customer care and guest portals may have a dramatically reduced need for PSTN trunk access. Large industries like conferencing may be dramatically changed. In the channel space, the need for web integration may drive major changes

Changing the Business - For actual end customer companies, WebRTC is probably more of a disrupter than for the others. It makes real-time communications something that is available anywhere, anytime for business process and customers. By implementing WebRTC properly, companies can get significant advantage. In health care, WebRTC enables new models to interact with home bound patient using video without plug-ins and using readily available devices. It can reduce doctor hospital visits during the day, increasing efficiency, and it can expedite billing management. In retail it can create new in-store experiences as user have WebRTC enabled in the physical store for personal assistants and staff interaction. In fact, in virtually any industry, a clear and compelling use of WebRTC for change can be readily created. And the potential goes further, as WebRTC devices become available, new business models will be created.

With this analysis, it is clear that WebRTC has the potential to be a true disrupter.

Long Term Impact and Barriers

As WebRTC becomes available across the browser systems and the web developers become familiar with how to use this new capability, we will see new applications and services that cannot be thought of without considering the specific use case or value. Much as the developers of the original web browsers could not anticipate how browsers could be used to build eBay or Facebook, this new technology has the potential to change the telecommunications landscape, and maybe the business world as well.

As with the original web, with WebRTC there no longer needs to be a high-level system interface to accommodate communications between individuals. If you know the URL of *MY* communications control server, by merely pointing your browser at that address you can now have high value interactions with me. Obviously security and reputation are required to control this, but those things have been resolved for the information web, they should be resolvable for the real-time web. The result in the enterprise space is obvious. By having a direct connection to your system, the need for my system and your system to federate goes away. While there may be some arguments around the value of continual presence, this can be rapidly replaced by instant availability. If my contact base had the URL of the WebRTC connect point for all of my contacts, one click would open up the browser and connect to a web page where my certificate would get me the availability of my contact for immediate communications.

This concept that WebRTC may eliminate the value of inter-server communications is understandable in the enterprise. However, the reality is that most of the non-transport functions of carriers/service providers today provide similar end user connections through connecting servers. As the PSTN and its derivatives in wireless require that you be represented by a carrier to interact with other parties and their carriers has created a system where server interconnection is a requirement and a cost. As Skype demonstrated, alternatives that do not require this are rapidly adopted. In the ultimate WebRTC world, the need for complex inter-relationships between the service providers may disappear, just as it has on the web today. As I move from one web site to another, there is no coordination or relationship. In this world, you no longer depend on the carrier to connect you to someone, you just go to their URL and click to connect with real time media.

There are a few issues that must be resolved before WebRTC can be widely adopted. If the API allows a server to turn on the microphone and/or camera on a users device and send that media somewhere without the user's knowledge, then there are major security concerns. While this potential exists today on devices like the iPad that do not have an active "camera on" indicator, having an open interface that enables a server to turn on media services is a major new security issue. Having either hardware, OS, or browser security features that assure that the user knows when this happens and/or authorizes it will be critical for adoption.

Another potential glitch in the WebRTC world is the support of both Microsoft and Apple. Microsoft has committed to ORTC, a derivative that is on the standards path for 2016 and Google and Mozilla have both committed to ORTC support. Apple has been in the standards, but has made no commitment at this point. If Apple decides to take the same path with WebRTC that they did with Flash, their ability to limit it's functional use in iPhones and iPads would reduce both adoption and potential new business models. The ability to have WebRTC capabilities delivered through the app store would become critical and Apple could block that as being duplicative of existing Apple capabilities such as FaceTime. While open source plug-ins for the Safari browser and for Internet Explorer would make WebRTC available to those user communities, not including it in the standard browser release will reduce acceptance and use. Over time, browser options without WebRTC support may become less desirable as sites begin to leverage the capability. The browser community needs to come together, as they have in HTML5, to make WebRTC an open reality. The other issue has been codec convergence with Google focused to VP8 and Microsoft/Cisco focused to H.264 for integration to existing UC and video systems. This appears to be resolving as Google has committed to H.264 support in Chrome (it is already in Firefox) AND Microsoft is openly looking at VP9 versus H.265 for a next generation codec. Assuming this comes together in 2015, dramatic changes will occur and new opportunities will follow.

While there are some challenges, ultimately the value of open browser based real-time communications will drive the industry to overcome them. The result has been a surge of WebRTC apps. Amazon mayday, Facebook Messenger, and a number of customer support apps form organizations like Amex have already been deployed using WebRTC. I strongly encourage communications vendors, end user organizations, service providers and web developers to keep an eye on WebRTC, lest you be overwhelmed by the changes it may bring on. For more information, webrtcworld.com is a great source.

