Evolving Enterprise & Contact Centers with WebRTC

Edited by Dean Bubley of Disruptive Analysis

Table of Contents

.

Introduction to WebRTC

Web-based Real-time Communications (WebRTC) describes a set of standards and approaches used to embed real-time elements (voice, video or data) into websites and applications. While this has been possible for some years by other means, the complexity and cost involved limited it to a narrow range of specialized communications developers and service providers.

WebRTC helps "democratize" voice and video, by simplifying implementation, reducing technology licencing costs, and stimulating a broad ecosystem of enabling vendors and open-source elements. Both major companies and start-ups are involved – well-known names such as Oracle, Google, Cisco, Avaya, Amazon, Intel, Facebook, AT&T, Twilio and Telefonica are already providing WebRTC products and services.

Set up in 2011 by W3C (World Wide Web Consortium) and IETF (Internet Engineering Task Force), the WebRTC project enables browsers and other apps to act directly as communications clients, using JavaScript and a standard set of underlying connection and transport protocols. Importantly, it doesn't specify a way for such apps to *signal* each other or a server – this is left up to the developer to decide, depending on the use case intended.

Most of the early applications of WebRTC involved peer-to-peer browser communications, with a web-server used to set up the connection. The intention was to expose the API (Application Programming Interface) to millions of web and JavaScript developers, who would incorporate real-time voice and video into their sites. This goes beyond standalone communications (similar to Skype and other apps); the most interesting innovation occurs where WebRTC is *embedded* directly into applications for business processes, consumer/social or commerce sites, or entirely new formats for B2B/B2C/C2C interaction.

Although it was recognized that sometimes a server might be required rather than "pure" P2P (peer-to-peer) communications – for example for managing multi-party conferences – much of the initial vision was around creating entirely new services, rather than interconnection with existing systems.

However, that view changed rapidly during 2012-2013, when it became apparent than many valuable extra use cases could emerge from combining WebRTC with existing enterprise / contact center or telecom-operators' voice and video domains. This led to the emergence of "gateways" between WebRTC browsers/apps on one side, and systems such as contact centers, unified communications platforms, or telco networks on the other.

2014-2015 has seen further maturity and sophistication enter the WebRTC space. Better support of "non-browser" endpoints is a key feature – especially mobile apps for smartphones and tablets. WebRTC gateways have also been enhanced to support additional features for control, security and more robust user experience. Media server technology has improved to add capabilities such as video/audio-mixing, multipoint control, recording, analytics or media manipulation. General communications industry trends towards virtualization and "cloud" have also been reflected in the WebRTC world.

WebRTC is not without obstacles, however. It still not ubiquitous on all devices – in particular, Apple has yet to launch a browser which supports WebRTC. Microsoft is supporting a variant of WebRTC its new Edge browser, but the legacy base of Internet Explorer (with no native WebRTC support) is large (although 3rd-party plug-in solutions can add WebRTC to IE). A lengthy standards battle over video formats ended in a compromise position – but one which nobody is certain will actually be reflected in future products. Additional practical questions have been raised over performance, ease of development by non-communications specialist engineers, mobile native app support, and assorted other areas.

However, as in other areas of web and app development, assorted workarounds have emerged quickly – and WebRTC gateways seem to be a good location to solve many of the issues, without individual developers having to fix them on a case-by-case basis. Associated developer toolkits, cloud PaaS (Platform-as-a-Service) providers and various open-source projects have also helped broaden the options available for creating apps, reducing entry barriers, assisting with mobile use cases, supporting non-WebRTC browsers and blending in extra cloud-based functions to apps.

Further evolution is expected. Continued enhancements to the core WebRTC concepts are in the pipeline. A close sibling of WebRTC, called ORTC (Object Real-Time Communications), is being worked on and will be supported by Microsoft; it may evolve to become WebRTC 1.1 or 2.0 over time, although that is uncertain. Other semi-proprietary variants are around as well – to the extent that some people are now just referring to "RTC" as the general philosophy of adding voice/video to other applications.

According to Disruptive Analysis' forecasts of WebRTC support, there should be over 6 billion RTC-capable devices globally by 2019. This means that the technology will be accessible to the majority of PC/smartphone users in developed markets, either in users' normal browsers, or bundled fairly-easily into mobile apps. Existing corporate devices should mostly be WebRTC-ready – but perhaps more importantly, so should those of customers and B2B partners.
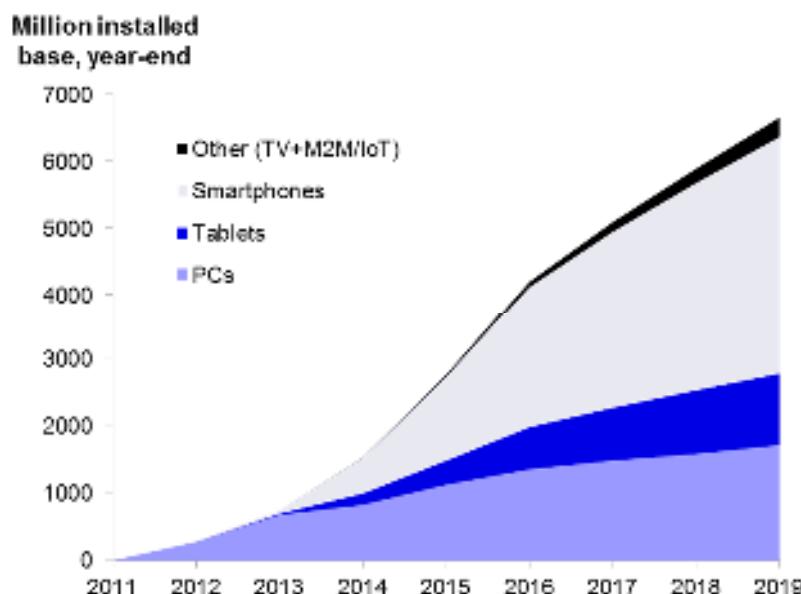

Figure 1: Mainstream adoption of WebRTC capability is a B2B/B2C catalyst. (source: Disruptive Analysis)

Use case landscape

There is no one single "WebRTC marketplace". Much like the rest of web technology used in enterprises and contact centers today, there are hundreds or thousands of highly individual use cases and applications.

However, it is possible to try to map the landscape and add some clarity to this diverse range of opportunities. At a very top level, WebRTC use cases can be divided between:

» Enterprise & Contact Centers (the focus of this document)
» Consumer Web & mobile/social
» Telecoms & TV/broadcast

Each of those domains has numerous sub-segments – and indeed they overlap strongly in many areas. We are also seeing early signs of other classes of use cases, such as IoT (Internet of Things), consumer electronics and M2M (Machine to Machine) devices. The wider WebRTC industry also includes a broad set of vendors, platforms, tools and enabling components.
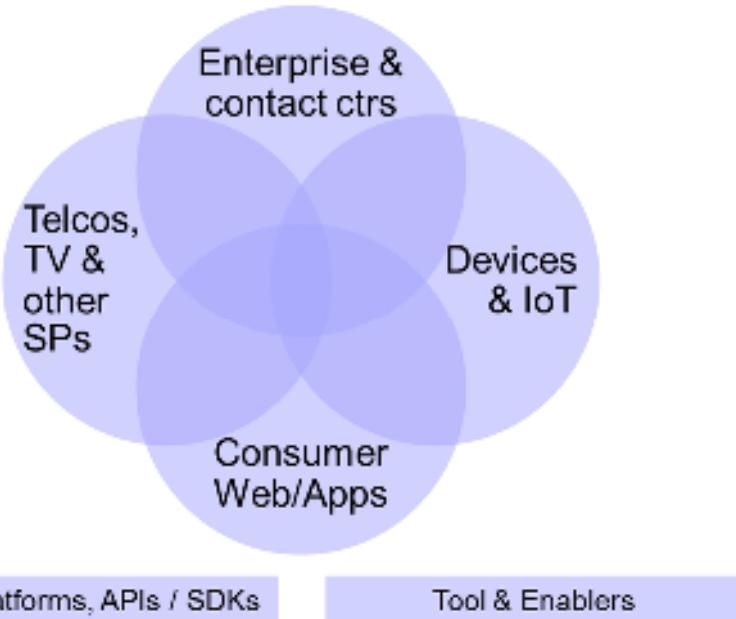
Figure 2: WebRTC use cases span a diverse range of markets. (source: Disruptive Analysis)

The enterprise and contact center markets are the leading sectors for WebRTC product launches and real-world adoption. Anecdotally, it also seems to have the highest level of awareness of the technology – most business communications professionals have at least heard of it and roughly understand its potential. The same is not true of many consumer web developers or telco executives.

Current enterprise-sector WebRTC focus is on:

» Contact centers and "multi-channel" customer interaction – both for agents and customers
» New cloud-based collaboration tools, often with timeline/activity stream UIs (User Interfaces) similar to social-networking
» Video-conferencing, both standalone and integrated with other collaboration and communication systems
» Customized vertical-industry solutions, especially for healthcare, finance, education, hospitality and e-commerce

In some cases, WebRTC is central to the various propositions, and in others it is more of an extension to, or integration with, an existing product. Often, its inclusion in a solution may not even be trumpeted loudly – WebRTC has already become "part of the fabric" of solutions such as Google Hangouts, Amazon Mayday and even Snapchat's video-sharing. Avaya and Google are also collaborating on WebRTC-powered contact center solutions.

Other use cases improve the functionality of existing solutions. For example, contact centers with remote agents can benefit from WebRTC solutions replacing VoIP-over-VPN architectures, which often suffer voice quality degradation and which are feature-limited. As well improved performance, this also provides a platform for future video support, plus tighter integration of the communication elements with the desktop agent's application itself.
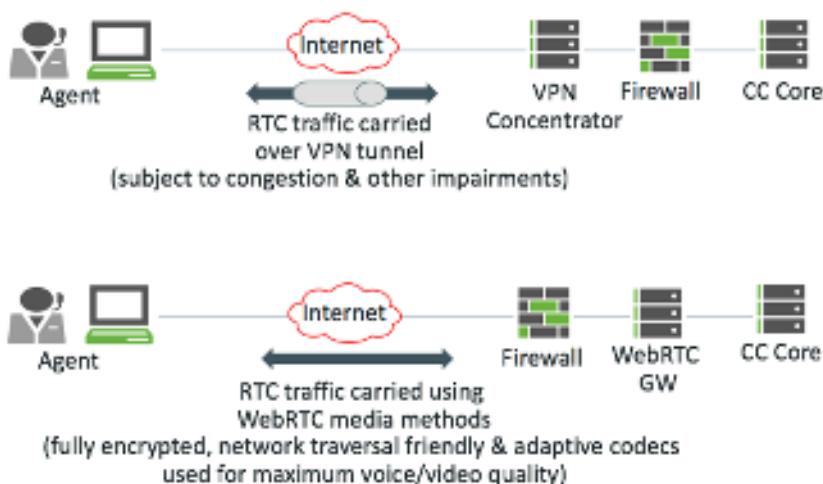

Figure 3: Comparing VPN vs. WebRTC approaches for agent connectivity (source: Oracle)

More transformatively, a range of new collaboration tools such as Cisco Spark and Unify Circuit are based around the technology at their core, while browser-based options have been added to developer platforms such as Twilio and AT&T's developer APIs. Educational software provider Blackboard liked WebRTC so much, it acquired a platform provider, Requestec.

There is generally a widespread sense of "inevitability" for WebRTC among larger enterprise / contact center users and developers – the questions now are "how much, how fast and which use cases first?" rather than "if?" There is also considerable momentum around both mobile-app and contextual communications use cases (see later in this document), aligned with WebRTC. Awareness of WebRTC or its benefits still seems lower among smaller businesses.

There is also a very strong alignment of WebRTC with the more general move to cloud-based solutions for UC, collaboration and contact centers – it makes it easier for the vendors to develop, as well as easier for the end-users

to deploy. WebRTC also continues to enjoy a growing profile at many corporate-focused enterprise & contact center communications & collaboration events (e.g. Enterprise Connect in Orlando and UCExpo in London).

While the current market emphasis is oriented towards customer care (and especially Amazon Mayday clones), it is now broadening to more general interest in WebRTC for collaboration/UC-type products, or "disunified" cloud communication platforms. The pace of change is also notable – with further acceleration and diversification of "productized" WebRTC expected to occur during the remainder of 2015 and through 2016.

Rollout of WebRTC-embedded vertical applications *inside* enterprises is mostly in pilot / early-adopter phases – we are only just starting to see voice/video integrated into employee-centric workflows. Re-engineering mainstream enterprise-type applications to incorporate embedded communications will take a while longer to materialize, although sectors like healthcare, hospitality and horizontals such as field-force management are leading the way. As elsewhere however, this situation is moving fast – with evolutions measured more in months than in years.

The longer-term trend perhaps fits with an emerging theme of "dis-unified communications", a controversial concept advanced by research firm Disruptive Analysis[1]. In future, traditional UC/PBX systems will not be the single "anchor" for all business communications, even with external applications connecting via dedicated APIs or custom integrations. End-users will seek out the "right tool for the job", whether or not their IT departments officially sanction the diversity. They will adopt extra communications capabilities either obtained separately, or embedded as secondary functions directly into diverse software applications.

In addition, many communications systems have external APIs and developer programs, so it is highly likely that other companies will offer WebRTC integration, even if the core offering is slower to evolve. This is already seen with Microsoft Lync/Skype for Business, for example.

A number of verticals are at the forefront of building WebRTC enterprise-based solutions, either as replacements for traditional contact center and UC/conferencing, or as deeply-integrated parts of workflows.

Among the most prominent are:

» Finance – which itself has numerous sub-categories, including banking/cards, insurance broking and claims and so on.
» Healthcare and medicine – again, sub-divided into separate niches such as telemedicine, online consultations, general-purpose videoconferencing, medical research and trials, and various others
» Education – both pedagogical tools and as broader/cheaper general videoconferencing systems for distributed university research teams.
» Retail / e-commerce is starting to emerge more strongly, especially relating to contact center and mobile app integration.
» Hospitality – hotels, resorts and other guest-oriented businesses are looking at using WebRTC in kiosks, mobile "assistant" apps, website inquiries and other contexts.

Use case Example 1: Sector-specific Collaboration

The world of enterprise communications systems is changing. Phone-centric IP-PBXs and presence-driven "unified communications" is slowly morphing into formats that look more like social media services, with timelines and activity streams. Others are more messaging-led.

The metaphors of discussion threads, shared documents, "likes" and ad-hoc groups and events fit particularly well with collaboration and fluid, small, cross-functional teams. With the general business focus on "agility", managers

1 http://disruptivewireless.blogspot.co.uk/2015/01/dis-unified-communications-webrtc.html

are putting emphasis on the ability to create new linkages between individuals and across silos, and rather less on standalone phone calls. Instead, various instances of communications – videoconferences, group calls, voice-messaging, recordings, hypervoice analysis and more – can slot into the collaboration timeline.

A call initiated between the members in a given discussion thread brings its own "context". The reason for speaking is clear – it relates to a topic that is already being discussed.

Yet while some of this new breed of collaboration products and cloud-based services work very well as generic "horizontal" tools, various industries or departmental groups may have specific needs. For this reason, we are likely to see customized collaboration apps, either for particular industries (e.g. finance, healthcare) or particular functions (e.g. IT management and software development). Examples of sector-specific functions might include:

» "Ethical firewalls" in finance, limiting interaction between people in equities and investment-banking teams for regulatory reasons
» Encryption and compliance requirements in healthcare
» Desire for real time screen-sharing, for example collaborative work on software coding
» Integration with particular existing intranet functions such as expenses, or industry-specific training
» Ability to include external participants such as customers, auditors, partners in a secure and well-managed fashion
» Alignment with particular management/oversight processes or assorted legal requirements (e.g. client confidentiality)
» Names and identities that are different to the normal corporate phone-number / employee database arrangement.

The net result is that we will see the emergence of a diverse set of niche collaboration tools – which could well have particular use cases for WebRTC directly embedded into their specific design. These may even be used alongside a traditional horizontal UC&C system, but just for different types and styles of interaction. Medical applications are perhaps the most advanced here already, for example giving doctors or nurses the ability to share patient details (i.e. context) along with voice/video sessions, or use additional data such as location (e.g. "Get me the closest cardiologist to this bed immediately!").

Use case Example 2: Hotel / Resort App

Many hotel chains have mobile apps, typically used by their loyalty-club members to manage their accounts. At present, relatively few "infrequent" guests use them. These could be hugely extended in scope and capability, improving customer satisfaction, acting as a platform for add-value services, and reducing costs in managing hotel properties or the chain as a whole.

While many hotel-app opportunities may come from other features beyond communications, there are several aspects that could exploit WebRTC:

» In-app contextual "speak to reservations agent", which would pass through details of customer name, account number, hotel/city of interest, dates etc., without starting the process from the beginning on the call. Could also automate routing to staff with appropriate language or even personal knowledge of that site/region.
» Video-concierge service for premium members, for sales or customer-service, or even a personal welcome by a hotel general manager on arrival. Could be extended to work while guest is outside the hotel.
» Replacement or enhancement of hotel desk-phones. Clever use of UI, notifications etc. could improve interactions such as ordering room-service, wake-up calls, and also bring upsell opportunities ("Speak to us about our tours").

» Two-way video communications for calls to hotel maintenance or services ("*This is where the shower is leaking*") improving time-to-fix and reducing wasted maintenance efforts.

» Provision of a temporary "guest web page" URL, so family members or work contacts could use www.hotelchain.com/GuestMrXYZ rather than an expensive overseas phone-call or the need to ask the front desk for a room number.

» Additional features / services for children, convention visitors, event organizers etc.

The interesting thing here is that the app could blend voice or video interaction with central hotel-group services (e.g. main contact center), with localized functions (front desk or housekeeping).

Where the user either doesn't want to download an app, or prefers to use a PC, many of these functions could also be delivered via a browser and web-page. Indeed, as well as the B2C customer app, the same platform could also support communications between staff, including mobile terminals used by maintenance workers, gardeners etc.

This concept can also be expanded for broader resort-type applications – theme parks, cruise-ships, or perhaps even whole towns, where a tourist and convention bureau supports the app.

Use case Example 3: Fleet Management

An increasing number of trucks and other company-owned vehicles incorporate telematics devices, often used for providing route information for the driver, and perhaps also collecting telemetry data, such as location, operating hours or engine parameters. These are increasingly based on cellular M2M-style connections.

Such systems could also be enhanced with real-time communications capabilities, embedded directly into the same device or console:

» Direct communications between driver and the fleet company's dispatchers. This could intelligently switch between audio-only and video, depending on whether the vehicle is moving or has the engine powered. It would remove the need for a separate corporate-sponsored phone for the driver, or a BYOD (Bring Your Own Device) implementation.

» External calls between driver and the warehouse, or to the delivery address, via a gateway at the HQ site. This could also record calls.

» Potential integration with one or more CCTV cameras in the vehicle (maybe one in the cab, and one in the trailer), with connections triggered by alarms or other situations.

» Possible driver-to-driver communications, as a form of more-modern CB (Citizens' Band) radio overlaid onto the mapping function. There may also be a requirement to interconnect different fleets – perhaps a system for vehicles at an airport, bridging between airlines' baggage vehicles and the site operator's own buses or fuel-trucks.

» In a smart-city scenario, it may be feasible to connect into 3rd-party systems such as traffic cameras, parking-space monitors or toll/congestion-charge agents.

» Maintenance functions, perhaps in conjunction with a separate smartphone, or wearable device, so the driver can use video to highlight problems or damage on the vehicle to engineers, or a breakdown service.

» Future evolution paths to self-driving or remotely-driven vehicles, requiring low-latency connections and new forms of telemetry

Rather than standalone "calls", many of these functions might be directly integrated into the firm's supply-chain management system, or a specific fleet-management application.

Variants of this type of system could also apply for emergency-services communications (many police vehicles already have laptops / browsers), or for public transport use cases with buses or taxis.

Frameworks for assessing ROI (Return on Investment)

Overview

This section considers ways to justify investment in WebRTC infrastructure, especially using commercial WebRTC gateways. Each use case will differ – the following sections outline some general principles, and give some examples and input data as a starting-point. Separate sections look at the cost/return drivers for contact centers, and UC & conferencing/collaboration.

Early WebRTC deployments have often been justified on the basis of experimentation or improved "cutting-edge" brand-image. But there is now an increasing desire for more quantifiable metrics, especially for large-scale deployments, where upfront costs may be significant.

The "easiest" ROI calculations for WebRTC are where it is used to replace existing CapEx / OpEx items, but at lower cost. Where WebRTC is used as an extension to an existing platform, the numbers are harder – and where WebRTC-based functions are intended to drive loyalty or incremental revenues, harder still.

Very different ROI calculations and considerations will apply to PaaS-based implementations, or those using open-source software – and also to implementations of WebRTC embedded into specific vertical or line-of-business applications.

WebRTC in contact centers

Several contact center use cases of WebRTC have relatively easy-to-scrutinize ROI for WebRTC gateway implementations. It is being used to reduce specific cost components such as:

» **Evolving agent desktop environments with WebRTC.** A traditional TDM or SIP-associated agent desktop with hard phone, PC, apps and ACD (Automated Call Distributor) "seat" can cost around $1,500 in terms of CapEx. By using WebRTC-based agent desktop solutions with existing contact center back-end platforms (or cloud-based alternatives) linked to a Chromebook/Chrome browser device and interface can bring this down to perhaps just $200-500.

» **Reducing inbound toll-free call costs**: Accepting customer calls via inbound toll-free services incurs costs to a business of maybe $0.04 - $0.07 per minute in the US. Taking an average $0.30 per call and a replacement of 20% of inbound phone calls with web/in-app communications, meaningful payback could potentially be achieved in a year for a 1000-seat call center.

» **Replace inbound phone trunks:** Although these are also being replaced with SIP trunks, WebRTC allows use of generic Internet connections, and gives more scope for advanced contextual/embedded communications.

» **Lower dial-in charges or VPN costs for home workers (see below):** Agents located remotely (e.g. home-workers) can use a browser interface, rather than PSTN dial-in incurring toll charges and trunk lines.

» **Faster time-to-resolution:** Better use of contextual data (e.g. name / web page / browsing history) can reduce call times and avoid information being repeated. Taking a cost per work-minute of $1.00, saving 5% of average call time could yield as much as $3M in implied cost-savings for a 500-person call center annually.

» **Lower maintenance for agent software:** shift to browser reduces costs for dedicated agent apps/clients.

» **IVR replacement.** In the longer term, in-context WebRTC in sites or apps could remove the need for phone menus, by driving calls directly from specific URLs or menus.

Sample ROI/payback-period calculation for remote agents using WebRTC	
1000 agents (with 50% incurring toll charges @ $0.0015/min)	$100K/yr.
DS-3 line for sufficient inbound PSTN capacity for 1000 agents	$230K/yr.
Costs of PSTN access for 1000 remote agents	**$330K/yr.**
Example cost of comparable WebRTC solution	**$260K upfront + $30k/yr.**

More complex calculations apply when looking at reduced time-to-resolution, improved satisfaction leading to better sales or retention, or the ability to up/cross-sell. For example, better upfront contextual information (e.g. customer's name and product/question) can reduce the time spent on each call. The majority of a contact center's costs are driven by personnel, so saving perhaps $1.00/minute with shorter and more-effective calls has an obvious payback.

Most WebRTC implementations in contact centers will also run in parallel to other "multi-channel" investments such as self-service, mobile apps, text chat and so on, making it hard to unpick the specific contribution of each. A typical use case might focus on self-service, but give an easier route to "upgrade" to an in-app call if needed. One survey suggested that 72% of customers prefer web self-service, but only 52% find what they need.

Estimating ROI is very hard where there are multiple unknowns: for example, user-behavior shifts, reduced online shopping-cart abandonment[2] or long term revenue uplifts from better customer engagement and loyalty. Convenient, contextual virtual support could reduce transaction abandonment by several percentage points, with major sales uplifts.

Adding video to previous voice-only contact centers brings extra costs, but largely unpredictable returns – thus an incremental pilot approach is recommended. Only a few % of a bank's customers may use a "video concierge" service initially, but the halo-effect of a modern way of interacting could show increased valuation in the *brand* rather than just direct revenues. Cloud/PaaS-based implementations will also have a different cost profile which is outside the scope of this document.
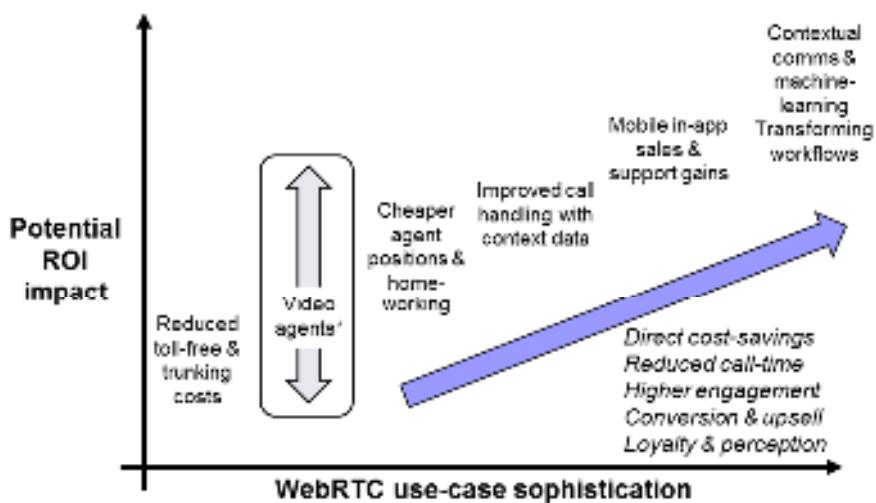

Figure 4: Roadmap for future WebRTC contact center & customer relationship management ROI (source: Disruptive Analysis)

---

2 Between 60-80% of online purchases are not completed. See: http://baymard.com/lists/cart-abandonment-rate

WebRTC in Unified Communications & Collaboration (UC&C)

The ROI of WebRTC implementations in UC&C is very context-dependent, owing to the variety of existing communications systems - especially comparing them to existing systems and business practices. The cost-profile of an "old-school" enterprise still using a traditional PBX, trunk-lines and audio-conferencing is already very different to a modern company primarily using Slack or Cisco Spark for messaging-centered communications.

The main sources of ROI for WebRTC in the UC&C space are likely to be:

» Removal of hard desk-phones for those still using them, by replacement with browser or UC clients (although this is also true of VoIP softphones)
» Reduced costs of supporting remote offices, home-workers or travelling employees with VPNs, dial-in connections etc.
» Cost-savings on 3rd-party conferencing or webinar services, and any associated dial-ins/800 lines and trunks, by bringing the capability in-house. (This is also true of other non-WebRTC VoIP implementations).
» Cheaper videoconferencing end-points – using PCs/browsers, or mobile devices, instead of dedicated terminals
» Personal web-portals reducing need for attendants, or reducing time wasted in email etc. arranging call times and appointments
» Replacing a % of standalone UC / conferencing use and spend with app-embedded alternative channels (e.g. built-into SAP or Salesforce or equivalents)
» Higher utilization rates of existing video bridge investments
» Start to consolidate multiple UC / PBX systems by bridging them all into a single web/client UC interface. This can then allow investment/maintenance to be capped on older platforms and a transition across to a single new system

For collaboration, reduced costs of using 3rd-party audio/video-conferencing platforms may be possible, especially in replacing international toll-free dial-in / dial-out. The global market for conferencing services is in excess of $7B annually (plus metered call charges), and while not all is targetable by WebRTC solutions, a sizeable fraction could be mitigated by "personal portal pages". This could also permit many calls to be upgraded to add video and screen-sharing, without incurring extra costs.

WebRTC can also permit companies to run webinars on their in-house UC platforms, rather than an external provider's. Removing the need for external "guests" to download plug-ins can reduce support costs for both parties, as well as reducing delays, frustration and missed meetings. Although pricing models are opaque (e.g. based on per-organizer annual subscriptions), a rough cost of $1000 to organize a commercial webinar for 200-500 attendees, with a good interface for registration and attendees, plus recording, is a reasonable benchmark.

There may also be much greater savings (as much as $1-3 per minute) if travelling employees can use local WiFi Internet connections, rather than roaming phone calls – although care in application-design needs to be taken, to avoid incurring roaming data fees instead. Some international travelers rack up charges of $500+ per month on calls.

However, the main ROI source for WebRTC in UC&C will likely come from other business process changes and workflow/productivity improvements - for example, reduced numbers of site journeys by engineers or sales representatives, by using video more effectively. An extra urban site visit might cost $200-500 per "truck roll", while for remote facilities like oil drilling sites, downtime can cost as much as $10-20,000 per day. If WebRTC-powered remote maintenance applications (e.g. with "see what I see" video) can mitigate major incidents and delays, then payback periods could be very swift.

Conclusion: Investment Direction

ROI calculation is a useful tool to analyze upfront investments for well-defined applications such as call centers. Implementations involving WebRTC gateways or cloud PaaS platforms can be compared directly to costs of agents' IT systems, or per-minute/month costs of calls and connectivity.

However, ROI is a much less-useful metric for more complex use cases, either involving possible revenue uplift, customer loyalty, or where WebRTC is just one part of a major overall process change. For these cases, companies should look to pilots and prototypes to model their own situation, and then consider large-scale implementations if they prove successful.

In the long-term future, WebRTC and analytical/contextual systems could completely reinvent certain workflows, leading to "exponential" growth and huge ROIs from new business models, rather than incremental improvements to current processes. Consider how the use of mobile apps, with messaging and notifications has helped the emergence of the "sharing economy" such as Uber and Airbnb. An "internal startup" perhaps funded with <$1M, could create completely new forms of voice- and video-enabled services.

Another source of positive return could be incremental market share wins, or higher margins, from quicker time-to-market spurred by better collaboration among designers, product managers and supply-chain participants. These are clearly company-specific.

For B2C companies such as retailers, transport or e-commerce vendors, small WebRTC-enhanced pilot solutions are inexpensive, yielding company-specific insights that help quantify returns more realistically. These can then be expanded to a wider audience of customers, or for a greater range of functions and device platforms. Knowing the percentage of users clicking the "speak to a sales advisor" button in an app, a firm can extrapolate and aim to justify investment in full-blown in-house gateways or custom developments, targeting a broader set of users and interaction points. A similar trend has been seen with live-chat functions in websites, which are now becoming an important channel, especially for certain demographic groups.

Analogies for WebRTC deployment

The idea of embedding two-way voice and video communications directly into applications and workflows is still relatively unusual. We are more used to "standalone" communications – picking up the phone to discuss a delayed shipment, emailing about a proposal, maybe a pre-planned videoconference for a team brainstorm.

Yet there are a number of developments of "integrated" or "embedded" interaction or communications which we now take for granted – but which were themselves very rare only 5 or 10 years ago.

For example:

» **Integrated chat / IM (Instant Messaging)**: Many applications and websites feature text-chat, either in a "silo" or inter-connected with other systems. Tech-support, collaboration and conferencing tools, e-commerce websites, financial trading systems and many more have the ability to message fellow users, or an "agent".

» **Video/audio streaming:** Ten years ago, watching one-way video, or listening to a conference recording (or music) required a separate "player" application on our PCs. But since the advent of YouTube and its peers, we access embedded content everywhere in the consumer and business web, both live and recorded. It might be videos embedded in a compliance-training course for a trader, a web-based interface to CCTV cameras, or a recording of the quarterly results announcement. They are all now "natural" parts of our interaction with software and the web, appearing in context.

» **JavaScript:** While the original thrusts of "Web 2.0", around 2004-7 were about user-generated content and a shift from static to dynamic web-pages, the key underlying technology was JavaScript. This essentially turned web-pages into "web apps" in the browser, with data fetched/updated in the background. This was – in essence – real time interaction. Today we expect web forms to self-update, menus to pull down automatically, and many other examples of "live" pages. Yet 10 years ago, most of these were a novelty.

Such features rarely appear as a "big bang" in the web. Users grow accustomed to a slow growth in use cases and instances over time, until they just become normal day-to-day. A similar set of incrementally-improving features apply to mobile apps, catalyzed by advances in versions of iOS and Android, or 3rd-party APIs. In similar, fashion, we should expect WebRTC and embedded voice/video to permeate our normal online experiences, as well as helping create new ones.

Going beyond "calls"

Many the early use cases for WebRTC have been "extensions" of existing communications platforms: using a browser attached to a UC system instead of a softphone, basic click-to-call buttons on a website linking to a call-center, bridging videoconferencing to a desktop PC browser and so on. Other examples have mostly been standalone calling services or apps – straightforward "talking heads" video chat, for example.

While all of these are indeed valuable and convenient, with benefits such as faster/cheaper development, or a reduction in the need for clunky browser plug-ins, they are not really that *novel*.

The true promise of WebRTC in the enterprise arises when we go beyond the familiar standalone "phone calls" or a "conferences". There are three sub-categories here:

» Application-embedded communications
» Novel formats for interaction
» Contextual communications

Application-embedded WebRTC

Many of the most interesting future use cases for enterprise WebRTC are not just browser-centric updates of generic conferencing or UC/collaboration systems. Ultimately, WebRTC's main beneficiaries will be software developers *outside* the traditional communications domain.

Either through direct integration into their own applications/sites, or by use of an intermediate cloud platform, they will finally be able to create what were termed "communications-enabled business processes" (CEBP) a few years ago. At the time, the promise was great – but the practicalities of actually creating such features were insurmountable for many. In particular, adding voice/video required specialist skills, incurred extra licensing costs, and were often a poor fit with mobile- and cloud-based delivery models that were becoming more important. Where the applications relied on traditional telephony, they often incurred extra costs and fractured user-experiences, flipping between the main software client and some form of "dialer" or softphone.

WebRTC – usually in tandem with deployment of cloud/browser versions of applications – mitigates most of these pain points.

In a fashion, embedded communications already exist for some enterprises – CRM, contact centers and customer interaction are often positioned by PBX/UC vendors as "applications" sitting on top of their core switching and call-control platforms.

But really, these are still "communications-primary" applications. Beyond UC or call centers, there is a huge range of other business applications that could benefit from *secondary* real-time communications features.

These could be "horizontal" software used in all industries: salesforce systems, ERP (Enterprise Resource Planning), supply-chain management, HR (Human Resources), even finance applications. Or they could be "vertical" applications used for very specific sectors – hospital management software, financial trading desk systems, process-control for chemical manufacture and 100s of others – including government and even emergency-services command and control.

There are possible roles for various forms of voice, video or real time data *integrated into the fabric of the app itself*. In other words, not just a "phone call" or even a "video-conference", but perhaps a different user-interaction model, or media-processing, as described in the section below. In some cases, it may make sense to interconnect with existing communications systems, while in others the apps may remain isolated.

Any of these app-integrated communications capabilities *might* be provided from an underlying UC platform and linked to traditional phone numbers, but equally they may well be offered separately – especially if they are built into cloud-hosted services, or will never need to connect to other users in "the outside world".

This concept can be challenging to understand, so a few examples are useful:

» An HR specialist could video-interview a candidate for a job *inside the employment section of the company's website*. The voice and video could be recorded and added to the employee's records if they were subsequently hired.

» A financial trader could use a dedicated terminal with the built-in ability to add domain-specialist analysts, or compliance officials to a call – perhaps while co-browsing the real-time streaming market prices. Encryption and recording could apply throughout.

» A doctor's surgery could have secure, encrypted "private practice" portals for video consultations and access to receptionists.

» Hospitals could link patients' bedside TVs or tablets to medical professionals around the hospital, or family members at home. Out-patients might have video links to allow nurses to remotely-supervise medication being taken, or blood-pressure measurements being made.

» Security systems could be upgraded from basic CCTV, to mechanisms with challenge/response two-way video.

» Remote field-staff in oil exploration and mining companies could use video and streaming telemetry data to allow HQ experts to diagnose problems, rather than sending additional people to remote areas, incurring travel and shut-down costs.

» Police or firefighters could have continual video-recording from cameras on jackets or helmets, plus additional "non-telephony" audio applications such as detecting gunfire, as well as medical vital-signs monitoring, or even video transmitted from a handheld robot or drone.

While none of these concepts – or thousands of others – specifically *need* WebRTC, its lower cost points, growing range of enabling platforms and wide developer base should make them much more viable than is currently the case.

Readers should note that "application-embedded" is also quite a broad category, as it includes both websites and more traditional non-cloud "owned" applications, although they may have a browser front-end. There will also be some applications wrapped up in non-conventional computing devices – there is an overlap of WebRTC with IoT

(Internet of Things) developments, especially where a given object has screen / camera / microphone included, such as a kiosk, or a virtual-reality headset.

Novel communications formats

A "call" is not the only way to describe and set-up a voice or video session. It implies one person dialing or initiating a connection to another, often unannounced, and then having a two-way conversation for a certain duration of time. Conferencing is the other major format we are familiar with, where multiple users connect via a "bridge", usually at a pre-scheduled time.

But WebRTC allows developers and enterprises to de-construct the building-blocks of traditional communications and re-build them in different ways that better suit a given application.

For example, we could get:

» One-way video, plus two-way audio (as seen in Amazon's Mayday service)
» Push-to-talk or walkie-talkie mode
» Prioritization based on speaker importance
» One-way voice messaging that then escalates to a two-way call
» Always-on video or voice "windows" between locations
» Video-presence, where a still image is sent between users, perhaps once a minute
» 3D video or stereo sound
» Manipulated / processed streams (imagine an "audio Instagram" altering voices, or a user-alterable level of background noise)
» Real-time data or telemetry alongside the audio/video (e.g. pulse-rate, or battery level)
» Presenters' microphone mode
» Video surveillance with additional two-way capabilities when needed
» "Whisper mode", where an additional audio stream is injected for one listener
» Ultra-low latency multi-way audio, for music or singing, or a band "jamming" remotely

None of these fits well with the traditional "phone call" model, or even conferencing as it is usually understood. And while WebRTC may not be the only way to create such applications, it should be a significant catalyst for their adoption. All that said, some years may elapse before many developers go beyond "the call" in designing user-interactions for communications, as we are so used to the familiar format.

The Future: Contextual Communications

The concept of "contextual communications" is rapidly gaining prominence in the enterprise / contact center market. WebRTC is taking a starring role in enabling it. While some of the possibilities discussed below are very futuristic, there are already real-world examples being deployed, especially in the contact center market, with Oracle's Service Cloud already supporting a number of the suggested use cases.

There are three common ways of defining "context":

» Communications features embedded *in* a context, i.e. an app, website or device, where the ability to call or send messages is a secondary feature. This is essentially the same as the "embedded" section, above.
» Communications-enabled applications and business processes, where voice/video interaction is *fundamentally integral* to a given app or workflow, rather than being a secondary feature.

» Communications *using* contextual data (e.g. from the browser "state", or external sensors) in order to perform a task better. This is where much of the current excitement lies, and is discussed more deeply below.

Basic examples of contextual data use have been mentioned for WebRTC for a while – for example, linking a voice/video call from a web page to data from browser cookies, or coupling them to text in partly-filled forms. These could help inform a customer-service agent (or a friend or colleague) to understand *why* you're calling. For example, if you hit the "click to call" button halfway through booking a flight online to query something, the agent shouldn't need to ask again which route you are interested in – they should have that contextual data passed along with the call setup.

This type of contextually-aware interface also enables a multi-channel interaction, helping escalate a user from self-care, to chat or interactive voice/video support at an appropriate point in their journey. The company can also decide which users merit the more resource-heavy options (e.g. video calls) based on identity/log-in, shopping-cart $ amount, or other ways of estimating customer value.
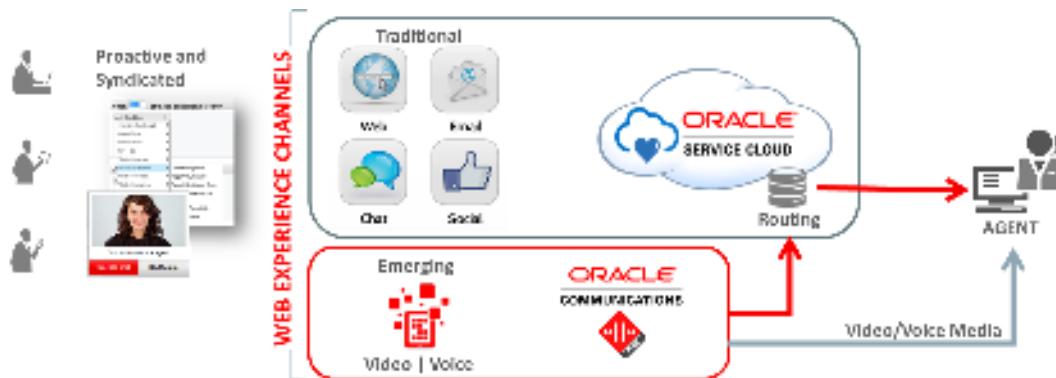
Figure 5: Example of Multi-channel interaction enabled by WebRTC. The Oracle Service Cloud provides the context and intelligence to connect customers to agents through web, email, chat and social channels. The Oracle Communications WebRTC Session Controller (WSC) gives organizations the option to add video and voice connections to enrich existing customer interactions. Routing to agents is still provided by the Oracle Service Cloud, while the WSC provides the video and voice media session control to agent desktops. (source: Oracle)

A major European insurance company is developing a mobile app for claims-management. By embedding WebRTC calls directly into the app, rather than having a "call" button trigger the native phone dialer, the app is able to pass details to the agent (e.g. car insurance vs. home insurance, policy number etc.), and use the app's user identity to help with call-routing. This can improve the efficiency of the call, and remove the need for the user to repeat basic information over the phone. In the future, the app may also include support of real-time data being pushed or pulled – perhaps a photo of a damaged vehicle, or a claim reference number to be sent, without the need for an SMS or email.

But there is a broader meaning of "context". It can involve using sensors to blend in "real world" status, as well as big-data style analytics and machine-learning to add in background information and process likely reasons/outcomes.

Thus a communications session could be enriched by input from sensors showing location, movement, background noise, acoustics, other people around you, temperature, power/battery, heart-rate, biometrics, facial expressions and so on. A conferencing application might be able to determine if you are in a room, or in the street – and apply a

"mute" function automatically. Video communication may be unsuitable if you seem to be in a vehicle, for example, and the application could react to that.

Real-time analytics could take this much further – for example, using voice-print to determine identity. Some contact-centers can already determine if callers sound angry or confused – and either change the agent's script, or perhaps have them transferred to a supervisor who can handle conflicts better. Database-driven insights might show that a caller has been browsing for a product repeatedly, but now appears to have made a purchase decision.

There can also be an archival and review function here – by using speech analytics and "hypervoice" concepts applied to recorded calls, searching for particular comments or themes in previous conversations will become easier. A collaboration tool with "learning" abilities might highlight people outside of a known team, who could bring a particular skill, contact or experience set – and send them an invitation to join a conference call. We may be able to get auto-generated subtitles for people with a poor network connection or an impenetrable accent.

A long-term future might even see a personal "intelligent avatar" knowing what we want to do, why we want to do it – and then blend a whole range of contextual drivers (apps, online activity, sensors, analytics, personal knowledge of your behavior and preferences etc.) to help you have a more productive, healthier life, blending in communications at its core. Think of it as Siri crossed with Sci-Fi style artificial intelligences, helping you both proactively and reactively.

However, in the more immediate term, enterprise / contact center applications will likely benefit first from embedding "virtual context" into the voice/video session:

» Where the user is "located" in the app or website
» What previous actions or pages have been visited
» Any preferences and configurations
» What device is being used
» Correlations with the CRM (Customer Relationship Management) record
» Information from social-network profiles (e.g. mutual connections on LinkedIn)

WebRTC will be pivotal here - driving the integration of voice and video with virtual context, especially web-page interactions or actions within a specific app. All these data points will allow the application to be shaped dynamically, so that both the audio/video experience is optimized, but *also* the outcome desired (higher sales completion, faster/happier contact-center resolution, better collaboration or process productivity).

We will also soon see basic examples of sensors/IoT helping to add value to existing real time applications by adding more contextual data, such as background noise and motion or orientation (is the phone flat on a table, being used as a speakerphone?).
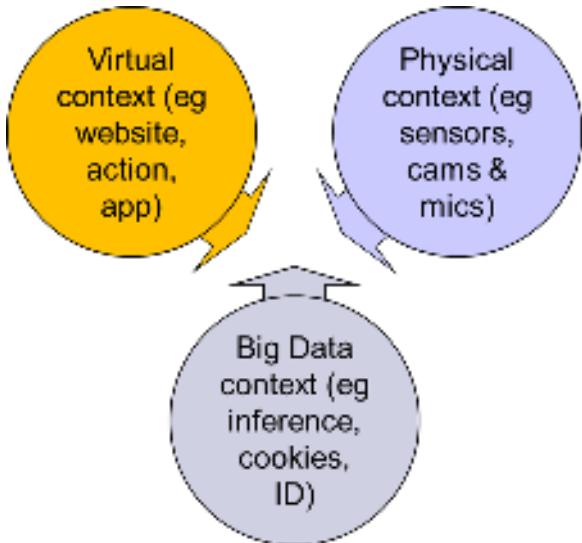
Figure 6: Three types of context (source: Disruptive Analysis)

Later we will see incorporation of communications capability in wearables, or video/voice biometrics into applications requiring extra security.

Will this mean wide adoption of WebRTC in IoT devices? Possibly, but most won't be browser-based, and may not even have screens or cameras. Whether the WebRTC codecs, media engines other protocols are necessary or optimal will be very dependent on specific use cases.

Overall contextual communications could be one of the key trends for enterprise and contact center voice and video over the next decade, although it is still very early days. WebRTC is a critical component and enabler, but it is important to keep an eye on its convergence with the physical world of sensors, wearables/IoT and the cloud-analytics domain.

Key recommendations for companies considering WebRTC

WebRTC is already making an impact for businesses, both for internal enterprise collaboration, and external-facing contact centers. Cost-savings are generated in terms of agent efficiency and telephony costs, while significant productivity and brand-image gains are becoming apparent as well. More generally, the speed and ease of creating new communications experiences with WebRTC is bringing in a new breed of innovators, as well as helping internal developers realize their projects more effectively.

Yet this is just the tip of the iceberg: future applications of app-embedded voice and video should enhance existing business processes, and enable new forms of employee and customer interaction. Blending real time communications with big-data will allow "intent" and "purpose" to help drive applications' behavior to give "magically personalized" customer service, or streamline workers' communications with each other. WebRTC will be a core component, along with cloud-based applications and analytical engines.

All that said, the industry is still at comparatively early stages. Device and browser support is not yet universal, while standards are still changing rapidly. Nevertheless, companies should start evaluating WebRTC immediately – we are at a comparable stage to the Web itself, around the year 1998. Starting with simple and controllable

implementations is possible – with the aim of rapid prototyping, testing, and iteration. The nature of WebRTC as a web-integrated technology allows it to be easily piloted on a subset of customers, or even internal "friendly trial" users.

Enterprises should also be conscious of the sheer diversity of WebRTC use cases. Consider *multiple* applications across the business, from contact centers to line-of-business functions, to mobile apps. All could potentially benefit from voice/video, in the same way that corporations use websites for many different functions and purposes.

Sources of advice and support for WebRTC abound. As well as vendors such as Oracle and its peers, there are thriving online communities around WebRTC, both technical and application-oriented. As well as dedicated conferences, the technology is now a cornerstone of most general enterprise collaboration/communication events, as well as those focusing on the future of the web. Verticals such as healthcare, finance and others also have specialist information on WebRTC uses available.

One last aspect to be conscious of is the growing presence of embedded voice and video capabilities in apps and websites used day-to-day. It is quite likely that most readers of this paper will have already encountered a WebRTC-powered application.