WebRTC, the day after

What's next for conversational services?

Emmanuel Bertin, Sébastien Cubaud, Stéphane Tuffin, Stéphane Cazeaux
Orange Labs
France
emmanuel.bertin@orange.com

Noël Crespi, Victoria Beltran
Institut Telecom
France
noel.crespi@it-sudparis.eu

***Abstract*—The WebRTC technology is gaining more and more attention in the Telco world. After having detailed this technology and its stakes, we remind in this paper some limits of the current architectures for conversational services, and we expose how webRTC could be used by Telcos. Two scenarios are possible: Extending the IMS footprint and introducing a new control plane.**

Keywords—WebRTC, HTML5, IMS

I. INTRODUCTION

Web Real Time Communication (WebRTC) [5] is an HTML5 [6] technology driven by web companies, commonly named Over-The-Tops (OTTs). It aims to offer Web developers native browser tools to insert real-time media stream exchange services easily into their web pages, including audio and video communication services, live video sharing and screen sharing. WebRTC also enables browser-to-browser data exchange, as for example for instant messaging and file transfer. In practical terms, for the "Web" user, this means it is no longer necessary for him to download, install and manually configure an application or to use some proprietary plug-in in the browser. All multimedia functions are natively integrated by the browser makers: codecs, management of exchanged streams, development APIs for web developers, etc. Simple knowledge of HTML, JavaScript and basic languages for web development will thus be enough to develop and integrate a real-time communication service into a web page.

We analyze WebRTC and the related technologies as a step further within a fundamental shift, as underlined by members of the IETF Internet Architecture Board in [11]: All IP-based services are progressively migrating at the edge from dedicated client/server protocols to Javascript APIs relying on the universal and service agnostic HTTP protocol. Internet Message Access Protocol v.4 (IMAP4) and Post Office Protocol v.3 (POP3) have for example been largely replaced by Webmail on most markets [11]. With WebRTC, the same may progressively happen to the de facto protocol for controlling multimedia sessions, that is, Session Initiation Protocol (SIP).

In this article, we study how this new technology might impact communication services and renew the architectures of telecom services. We detail the key stakes of WebRTC in section II, highlighting the differences with current telecom technologies. Section III reminds some issues with current solutions and Section IV draws some possible architectural scenarios. Finally, some conclusive remarks are presented in Section V.

II. WEBRTC

As all modern telecom architectures, WebRTC is based on the fundamental separation between the media path and the signaling path. The data flows containing the media are transmitted directly between browsers in a Peer-to-Peer fashion, without any predefined intermediate - natural intermediates in the network topology such as Network Address Translations (NATs) or firewalls are not considered here. The signaling path implies obviously intermediate servers but their protocols and architectures are left out of the WebRTC scope.

This Section discusses technical details of both data and signaling paths as well as the standardization efforts, the major differences with Telco solutions, and the current implementations.

A. Standardisation and ecosystem

WebRTC is currently being designed by three standardization bodies:

- Internet Engineering Task Force (IETF) [1] is defining the media plane, the overall WebRTC architecture and the security framework;
- World Wide Web Consortium (W3C) [2] is defining the HTML5 JavaScript APIs for browsers;
- 3rd Generation Partnership Project (3GPP) [3] is integrating WebRTC access into the IP Multimedia Subsystem (IMS) [4] infrastructures (work started in November 2012 from the initiative of Telco stakeholders).

The push for IETF and W3C standardization of this technology is being led by the OTTs, such as Google (primarily), Mozilla, Skype-Microsoft, and now also by telecom equipment manufacturers such as Cisco, Ericsson, Oracle and to a lesser extent Alcatel-Lucent. Microsoft is currently working on its own development interfaces for WebRTC (CU-RTC-Web). Despite the rejection of its proposals to the IETF, Microsoft announced in mid-January 2013 that it is continuing its work and has a lab-version of an

978-1-4799-0980-3/13/$31.00 ©2013 IEEE

Internet Explorer plug-in [8] using these proprietary APIs. Apple, on its part, is remaining silent, excepting that they express their position in favor of H.264 as video mandatory-to-implement codec.

B. Media Plane

The WebRTC media plane (audio/video/data) differs from the media plane used by Telcos for their Voice over IP / Rich Communication Services (VoIP/RCS) deployments. OPUS and G.711 were chosen as mandatory-to-implement audio codecs with the main aim of choosing royalty-free codecs. This is the reason why Telco codecs (e.g., AMR or AMR-WB for mobiles) were rejected although they are widely deployed. G.711 was chosen for interoperability with legacy networks and OPUS, which derives from the Skype codec and has been very recently standardized by the IETF, aims to provide superior quality. However, it has recently emerged that OPUS is still covered by patents. Moreover, the possibility of introducing the Telco codecs AMR, AMR-WB (and G722) as "recommended" additional codecs has not yet been dismissed. Support for these codecs in WebRTC implementations would mean less transcoding costs in interoperability situations.

The same rationale applies to the choice between video codecs to be embedded in browsers. The decision is still pending on whether to use the Google VP8 codec [9] or the H.264 codec [10]. The proposal to use VP8 is based on the same argument than OPUS for audio. There is also the same risk linked to the possible existence of patents, amplified by the non-standardization of VP8. The difference between video and audio here is, on the one hand, the higher cost of video transcoding and, on the other, the fact that there are fewer existing Telco services and infrastructures – however all using the H.264 codec.

It is worth noting that the WebRTC standards allow for any codec to be negotiated if the browser implementation supports it. The obstacles to Telco codec support in browsers may therefore be tackled with solutions that reduce the license cost for browser makers in the future. Technical solutions (e.g., use by the browser of codecs already implemented on the devices) and commercial solutions are also possible.

The WebRTC media plane is designed to avoid, as far as possible, the need to relay browser-to-browser media streams to intermediaries. On the contrary, steering media through one or more Session Border Controller (SBC) is the norm in Telco VoIP/RCS deployments. Such systematic media relay approach makes it possible for Telcos to manage security, make lawful interceptions undetectable, facilitate the crossing of most NATs/Firewalls and help provide potential solutions to certain engineering problems (e.g., IPv4/IPv6 interoperability and management of IPv4 address spaces to be recovered). On the WebRTC media plane, security is managed by encryption, while the crossing of most NATs/Firewalls is directly managed by browsers using the Interactive Connectivity Establishment (ICE) principles, without inserting a media relay (i.e., a TURN server) in most cases.

Encryption handling is another difference between WebRTC and Telco media planes. With WebRTC, media encryption is compulsory whereas it is hardly ever used by Telcos for VoIP and RCS. Regarding to the media encryption, the key point is the choice of the method to exchange the master encryption key, since it is required by Secure Real-time Transport Protocol (SRTP) media encryption. OTTs are pressing to use the Datagram Transport Layer Security (DTLS)/Real-time Transport Protocol (RTP) method where the master key is exchanged within an end-to-end DTLS secured channel (i.e., the SRTP master key is not communicated in clear text) while Telco VoIP/RCS encryptions are based on the Secure Real-time Transport Protocol (SRTP)/Session Description Protocol Security Description (SDES) method where the SRTP master key is simply exchanged in clear text over SIP signaling. Which alternative for key exchange will be used for standardization has not yet been decided. For browser-to-browser communications that are subject to lawful intercept requirements, there is a significant difference between the two approaches. Implementing lawful intercept with DTLS/SRTP would force the Web service provider to deploy a media relay decrypting and re-encrypting any communication. SRTP/SDES would however allow the Web service provider to duplicate and decrypt only the streams for the intercepted targets, which is usually a small subset of the user base.

The last difference is that the WebRTC media plane incorporates an exchange of information on the quality of the network. This creates more intricate options for adapting the media coding to best-effort network conditions than in the case of the Telco VoIP media plane whose underlying principle is a managed network quality.

We can finally conclude that the WebRTC media plane is not designed for interoperability with traditional telecommunication networks, nor to take into account the regulatory obligations imposed on operators – which might however be imposed to every service provider in the future.

C. Signaling Plane

It should be noted that, apart from the principle of codec negotiation (i.e., SDP offer/response), call set-up signaling does not appear in the work being carried out at the IETF and the W3C. This is a deliberate choice made by the OTTs. In fact, two individuals connected to the same website can use different browsers. It is therefore a minimum requirement that these browsers are able to exchange media streams, and hence the work carried out by the IETF. In addition, call signaling has to be implemented in JavaScript that is downloaded from the service's web site. The service provider therefore develops both the client script and the server, thereby undertaking the signaling path. As this path is not standard, the caller and the callee need to be connected via the same Web service provider. This is consistent with the non-cooperative nature of the prevailing OTT approach: Web communication services remain in their proprietary "bubbles". This could potentially be alleviated by the emergence of an identity provider role distinct from that of the communication service providers [13]. Commercial and technical partnerships between some web companies might also lead to a certain level of interoperability.

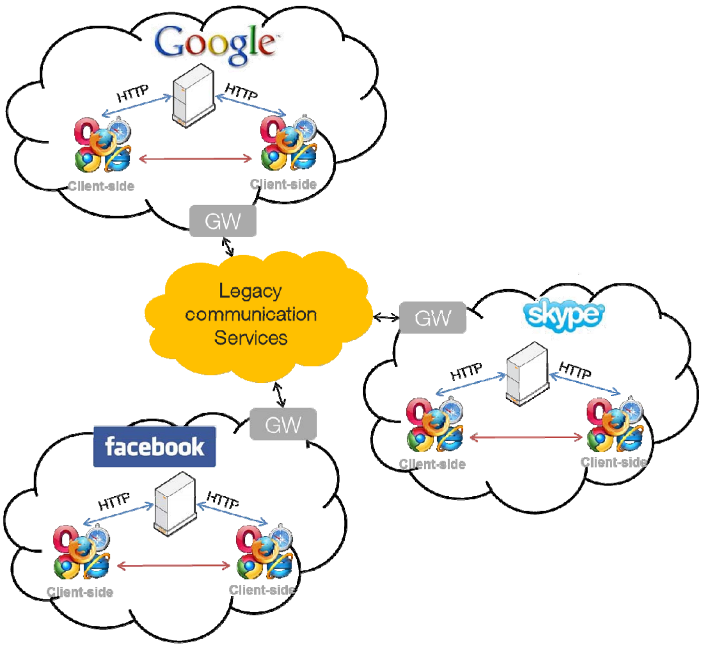

Fig.1 Each WebRTC service of each OTT remains isolated

It should also be noted that the use of proprietary signaling facilitates the development of innovative services by removing the need for standardization at the interface between the user device and the network platforms.

D. Current implementations

For Desktop versions, Google's Chrome browser – now the number one browser in the world – already incorporates WebRTC, as does Firefox. The date when Internet Explorer will support WebRTC with its CU RTC Web variant, is more uncertain (IE11 preview doesn't feature WebRTC) while WebRTC support in Opera is on its way. On Android, beta versions of Chrome and Firefox are currently available.

It is necessary for these browsers to be compatible with one another in order to achieve complete transparency regarding the browser used, and therefore to prevent restricting the end user of a service to a particular browser. Google and Mozilla have already demonstrated the interoperability of Chrome and Firefox respectively.

At short-term, the key challenges for WebRTC will certainly be its penetration within the devices and its impact on performance, including battery. We believe that at midterm (typically in 2-5 years) WebRTC market penetration will be achieved to a great extent due to the overall market interest in HTML 5 (including all the major web companies). Performance will be solved by the end devices' increasing computation power, possible access to device-hardware accelerators, and the limited number of browsers vendors.

III. WHAT ARE THE CURRENT ISSUES IN CONVERSATIONAL SERVICES?

WebRTC technology is one of the disruptive factors of the ecosystem of conversational services, along with business and usage transformations. The ecosystem of conversational services is indeed fast-changing, from business, technological and usage points of view. As far as Telcos are concerned, the current technical solutions for these services rely mainly on the IP Multimedia Subsystem (IMS) standard. In this Section, we detail some of the inherent limits of this model that may provide us with requirements to make use of WebRTC for the Telco services.

It is important to first highlight that the current IMS model is inherently based on a complete control of the whole technical environment (user device, service provider, service developer, network provider) by a few actors. But the new ecosystem has made it unrealistic for an IMS provider to control the whole targeted devices and the service development dynamism. The IMS presents indeed some intrinsic constraints, as detailed below.

Firstly, methodology-wise: the IMS was designed (re-using the traditional methodology from the PSTN) by starting from the service requirements, and then defining the functional entities, the procedures for data exchange between those entities, and finally the protocols that allow such exchanges. This approach presents two major drawbacks. First, this implies that the core IMS architecture is designed to fulfill service requirements that were defined by telephony experts roughly 10 years ago. Thus, the IMS is not fully adapted to the current competitive market. Second, the non-functional properties of the architecture (i.e., availability, security, extensibility, scalability, etc.) that largely depend on the protocol stack, are seen of secondary importance. This might for instance hamper the use of a single core IMS for different services, especially if the services require various QoS, or levels of availability and/or security, for example between a low-cost residential service and a five9 enterprise telephony service.

Secondly, architecture-wise, the IMS has chosen to define non-atomic generic functional entities, which have often stateful implementations. In the current implementations, most of these entities are on the signaling path, even when only a subset of their functions is needed to offer a service and that for all the duration of the service. This induces obviously high platform costs. Moreover, these entities should often evolve to enable new services to be offered, implying new deployments. Moreover, these architectural choices imply a tight coupling between functional entities all along the signaling path: user devices, P-CSCF, S-CSCF, HSS, AS, etc. – unlike the modern software architecture paradigm of loose coupling between software modules. In terms of implementation, this leads to a huge matrix of interoperability end-to-end testing that added months of delay before deploying any new element. In addition, the registration procedure allowing this tight coupling implies high hardware costs (to handle the load of SIP register messages). It implies also a complicated handling of multi-device (as even an occasional device must be provisioned and registered – this is especially heavy for SIM-free devices).

Thirdly, protocol-wise: the choice of protocols hampered a few non-functional properties of the IMS architecture, despite multiple enhancements of some of these protocols such as SIP and Diameter. Beyond protocol complexity that has consequences on the operational cost (e.g., end-to-end interoperability testing), the most impacted non-functional property is certainly the extensibility. SIP entities act as both client and server, creating grips between the evolution of the devices and the core infrastructure. The separation between service-related information and protocol-standardized elements within SIP messages is unclear. Moreover, SIP does not provide a graphical rendering framework, which holds service providers back from having a complete control of its services on heterogeneous devices (it should be here noted that major OTTs are also device or mobile OS makers, enabling them to fully control the value chain – while Telcos are not able to tightly integrate IMS services on smartphones). Finally, these chosen protocols are not the one currently massively used by the mainstream IT, preventing leverage with existing frameworks or middleware.

Fourthly, implementation-wise: the model followed by operators mainly consists in buying solutions from vendors in a fully-dependent way. Operators have chosen customized, potentially non-interoperable vendor solutions, while standards propose more and more technical options due to trade-offs between stakeholders with different interests... Such dependency has multiple drawbacks (in spite of many well known advantages such as industrial products; externalization of development risks and sharing of the development costs by many customer companies). Firstly, the costs of the platforms are important – either CAPEX or OPEX depending on the vendors' strategies. Secondly, the evolution of features has to be integrated in the vendors' roadmap – not necessarily being compliant with expected services or market demands. It affects therefore the ability of the IMS to generate direct revenues by reducing costs and developing rapidly attractive services while it limits the ability to generate indirect revenue by providing differentiated services to subscribers. The innovation cycle is indeed slow, and it requires many end-to-end integration tests.

Fifthly, cost-wise: the huge initial investments needed on each IMS platform (CAPEX and even more OPEX) as well as the licensing policy of IMS vendors (per user cost) obstruct a quick amortization. This is an issue to compete on a more and more cost-driven market. Moreover, the current standardization processes require time (in the order of magnitude of years) and consensus within the ecosystem, preventing rapid deployments of services.

These reasons explain why the IMS demands a complete control of the ecosystem (i.e., user device, service provider, service developer and network providers) and why it seems not fully ideal within the current heterogeneous ecosystem of the conversational services. Still, from an operator point of view, the IMS is today the only technical solution to deploy enhanced telephony services over IP. Moreover, at short-term, the IMS presents the undeniable advantage to permit coexistence with circuit-switched technology.

IV. ARCHITECTURE SCENARIOS

The promises held by WebRTC to provide the technology for standard, multi-device, ubiquitous access to distributed services have been previously presented. It should enable end users to experience seamlessly their multi-media services on different devices at low costs (as long as an adequate IP network is available). This section describes the network architectures over which operators could deploy WebRTC technology as well as the associated business opportunities.

WebRTC technology can be used in different strategic positioning by an operator. Two main business opportunities can be outlined as far as conversational services are concerned: on one hand, WebRTC can be used to extend the footprints of existing core service infrastructures (typically IMS networks) and, on the other hand, it can be used as a cutting-edge technology for a new core service infrastructure based on the Web. The following subsections analyze both alternatives in more detail.

A. *Extending the IMS Footprint*

WebRTC allows web browsers to communicate with each other through a variety of real-time means such as voice, video, conferencing, messaging, or gaming. All this is performed without the need to handle plugins or third party software. This new world of real-time possibilities over the Web can be seen as damaging to existing Telco business models. However, network operators are positioning themselves to benefit from this coming communication revolution. With WebRTC, Web developers can build real-time services more easily and rapidly and these services can be deployed more universally across different devices. This will reinvent communication services, increasing the value of mobile services. Instead of limiting these services to the Web, a bridge between the telecom and the Web will enrich the panorama of communication services. Network operators should therefore move fast to embrace this new technology for upgrading WebRTC-enabled services with Telco features of value (e.g., QoS, transport and call control). It should succeed in boosting revenues via extending the current conversational services on browser-enabled devices as well as via developing B2B opportunities with actors on the Web.

Merging IMS with the Internet world could expand the ecosystem of Telco operators. Indeed, both worlds can grow up synergistically through the interoperability of WebRTC with IMS. On the one hand, WebRTC gives Telcos the opportunity to extend their services to Web-connected devices and allows subscribers to access and use their IMS services from different devices (e.g., PCs, tablets, TVs, etc.) over multiple access technologies (i.e., fixed and wireless networks). On the other hand, Web applications can be enriched with operator network services and web developers can benefit from developing applications without special knowledge on the complexity of telecom networks. Furthermore, guest IMS subscriber support can allow WebRTC-enabled communication between IMS subscribers and non-IMS users.

The industry is aware of the potential of WebRTC-IMS interoperability and is working on this direction. Ericsson has already developed solutions to let WebRTC-enabled browsers connect easily with IMS-based communication networks. The application Messagenet Talk enables WebRTC to work with Android devices and provides interoperability with PSTN. Mavenir's WebRTC Gateway provides interworking between IMS and WebRTC, allowing mobile operators to extend their services beyond the Internet. Moreover, in a recent 3GPP SA Plenary, a work item for WebRTC access to IMS has been approved [3]. Its objectives are to specify service requirements for:

1) The ability for WebRTC clients to access IMS,
2) How IMS clients communicate with WebRTC clients connected to IMS,
3) The ability to realize any IMS services to the WebRTC client,
4) Access to IMS client capabilities for WebRTC clients connected to IMS,
5) The ability to support applicable IMS access types (e.g., LTE) for WebRTC clients connected to IMS,
6) The ability for an IMS service provider to offer IMS services to users (with or without IMS credentials) interacting with a 3rd party website that is using the WebRTC client.

From the technical point of view, interworking between a WebRTC-enabled browser and an IMS device requires translations at both the signaling and the media plane. Interoperability is concerned to the three key components of WebRTC: SIP signaling, ICE, and Media coding. The components of a WebRTC-IMS gateway are detailed in Figure 2.

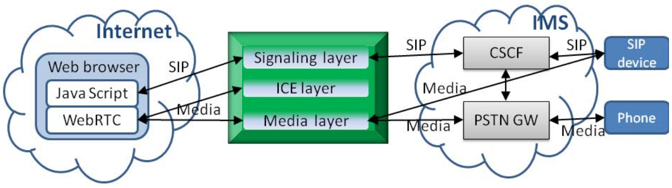

Fig.2 WebRTC-IMS interworking

While IMS devices use standard SIP signaling, WebRTC potentially relies on proprietary solutions for signaling. Nevertheless, WebRTC can use standard SIP over Websocket and, in this case, no signaling translation would be necessary. WebRTC makes mandatory the use of ICE but many IMS devices do not support this feature. Regarding media, it is necessary to deal with WebRTC ciphering and multiplexing. OPUS and G711 are the only mandatory to implement audio codecs. G711 is not optimized for being used on mobile access and, since the support of AMR and WB-AMR by Browser makers is uncertain, audio transcoding between Opus and mobile Telco codecs might be required. The situation is even more confused regarding video codecs as the choice between VP8 and H264 has not yet been made at the IETF. Not retaining H264 as a mandatory to implement codec in WebRTC would be a serious issue as H264 is the Telco video codec and video transcoding costs could be a show stopper. Lastly, to cover IMS services such as RCS and Conferencing, the interworking with non-RTP media such as MSRP and BFCP should be addressed.

B. Toward a new Control Plane

A new control plane making use of WebRTC is today still not designed. However, this section presents the objective, and underlying requirements for such a new control plane based on the Web paradigm (i.e., WebRTC and service-oriented computing). Some requirements considered subjectively mandatory in the current context and others as nice to have.

The mandatory objective of providing innovative and differentiated services in order to boost direct and indirect revenues require both rapidity and flexibility in service composition and in the evolution of the architecture. This implies requirements of extensibility, modularity and indirection or high level of abstraction (e.g. using names instead of addresses). These properties are particularly fundamental at the boundaries between different roles (e.g. between business partners, between services and networks). Providing or using existing tools to facilitate the creation of basic or more complex services and their composition is also key to gain success in this area. It is also important to note that the lifecycle of services differs from the one of the technologies enabling their implementation, as well as from cycle of the software or hardware products hosting them. It is therefore essential for the architecture to have a complete independence with the underlying technology (e.g. protocols) as well as the underlying software base (e.g. OS).

The mandatory securization of the services forces the architecture to have security services of authentication, access control, confidentiality and anti-DoS (Denial Of Service).

The nice-to-have objective of reducing the associated CAPEX cost of a service leads essentially to try to reduce hardware costs and software costs of the infrastructure (i.e. platforms and networks). The former is possible as soon as a clear separation between the service and its hosting infrastructure is possible whereas the latter can be achieved by using open-source components and internalizing the development – yet to be assessed on a case-by-case basis.

Reducing the OPEX cost is other nice-to-have objective that leads mainly to try to reduce the cost of the required manpower to manage the services. This can be achieved by imposing a high degree of availability to the architecture and by mutualizing as much as possible the management of the technical infrastructure (i.e. the same technical infrastructure products are used by many services). The former requirement can be addressed by the distributed ability to replicate and self-organize (e.g. self-configuring, self-repairing), whereas the latter by again a separation between the service and its hosting infrastructure allowing different people to manage the services, the required network facilities, and the hosting platforms.

Another nice-to-have objective is service independence. With major threats on the direct revenue of conversational services, we believe that an operator can no more afford to exploit one service infrastructure per type of service (e.g. conversational, audiovisual, M2M…). The HTTP protocol coupled with browsers offers this opportunity [12].

A chance is that these requirements are not new and not surprisingly, software architecture already had to address in specific context (e.g. enterprise IS) most of the same constraints: after a few decades of technology evolution, Service-Orientated Computing (SOC) emerged as an adequate solution to provide a technology and OS-independent framework of re-usable, technology-agnostic and networked distributed middleware components providing services (or functions in architectural terms). SOC [14] is a computing paradigm that utilizes services as the basic constructs to support the development of rapid, low-cost and easy composable distributed applications, even in heterogeneous environments. The major innovation of SOC is the move from the object oriented paradigm to a service oriented one, i.e. from stateful to stateless [14]. This architectural style owns most of the required properties looked for a Telco service control architecture: extensibility, modularity, high level of abstraction, independence toward technology, OS and hardware, security via potentially a service infrastructure bus (e.g. confidentiality, integrity) and basic services such as authentication, anti-DoS or authorization functions. A possible macro-level architecture relying on these principles is presented in Figure 3.

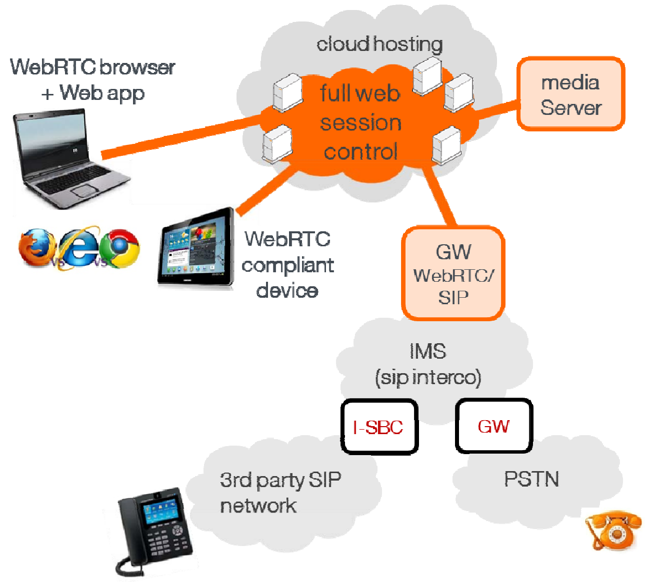

Fig.3 Principles of a full web control plane

Moreover, the use of a cloud computing environment can provide benefits at various levels of abstraction: firstly, an Infrastructure as a Service (IaaS) can foster the required extensibility property via its inherent elasticity, optimize the independence between the infrastructure and the services via the use of virtualization and a potential access network independence and fulfill the robustness property of the architecture via the ability to distribute and replicate network and computing resources. It is a natural infrastructure base for SOC, completing and emphasizing existing properties. Secondly, a Platform as a Service (PaaS) can provide the necessary tool to distribute the required components of service control architecture within a cloud environment. Finally, a Software as a Service offer can be seen as a business opportunity to open some functions of service control architecture to third-parties (e.g. MVNO). Aside from these general properties of cloud computing offers, two additional properties are essential for a Telco service control architecture: the ability for the cloud to offer QoS guarantees (e.g. latency) for inter-component communications and the ability to ensure a high degree of security within the environment (i.e. server and network).

V. Conclusive Remarks

The renewed conversational services ecosystem brings new challenges that require a breakthrough in the way operators manages their service control infrastructure. We presented two scenarios, one based on the webRTC access to the IMS, and one focused on the renewal of the Telco service infrastructures. In both cases, a part of the service logic will rely on the javascript code loaded in the browsers. This enables a greater flexibility to implement attractive services. Long-lasting standardization activities are no more required. New service can rely only on javascript programming language to implement the service logic (a technology widely taught in educational courses or in online tutorials).

Concerning more specifically the second scenario, the evolutions lead by IT actors have laid down the foundations of suitable solutions. These evolutions encompass the interface between the user and the service control architecture using HTML 5, the core service control components using SOC principles and a hosting of these components in a cloud computing environment. Additionally, these technologies offer the advantage of fitting multiservice and multi-device purposes. Other technologies (e.g. autonomics) might be also beneficial to address the aforementioned challenges, even if they are likely to be mature in a longer term only.

Acknowledgment

The authors kindly thank colleagues who helped them to achieve this article, especially Frederic Pujol, Didier Neveux and Jean-Michel Portugal.

References

[1] IETF Real-Time Communication in WEB-browsers Working Group, http://tools.ietf.org/wg/rtcweb/charters.

[2] W3C Web Real-Time Communications Working Group, http://www.w3.org/2011/04/webrtc/..

[3] 3GPP Active Work Program for WebRTC access to IMS, http://www.3gpp.org/ftp/Specs/html-info/FeatureOrStudyItemFile-580062.htm..

[4] G. Camarillo and M. A. Garcia-Martin, “The 3G IP multimedia subsystem (IMS): merging the Internet and the cellular worlds,” Wiley, 2007. .

[5] WebRTC website, http://www.webrtc.org/.

[6] W3C HTML5 Specification, http://www.w3.org/TR/html51/.

[7] Customizable, Ubiquitous Real Time Communication over the Web (CU-RTC-Web) draft, http://html5labs.interoperabilitybridges.com/cu-rtc-web/cu-rtc-web.htm.

[8] CU-RTC-Web User Guide, http://www.html5labs.com/prototypes/cu-rtc-web/cu-rtc-web/documentation

[9] J. Bankoski, J.Koleszar, L. Quillio, J. Salonen, P. Wilkins, Y. Xu, “VP8 Data Format and Decoding Guide”, IETF RFC 6386, November 2011.

[10] “Advanced video coding for generic audiovisual services,” Int.Telecommun. Union-Telecommun. (ITU-T) and Int. StandardsOrg./Int. Electrotech. Comm. (ISO/IEC) JTC 1, Recommendation H.264 and ISO/IEC 14 496-10 (MPEG-4) AVC, 2003

[11] Bernard Aboba, Jon Peterson, Henning Schulrzinne, and Hannes Tschofenig, “The Future of Web Applications or How to Move into the Post Standardization Area”, IETF, Oct. 2010

[12] Lucian Popa, Ali Ghodsi, and Ion Stoica. 2010. HTTP as the narrow waist of the future internet. In Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (Hotnets-IX). ACM, New York, NY, USA

[13] Mozilla BrowserID, https://login.persona.org/

[14] M.P. Papazoglou and D. Georgakopoulos, “Service-Oriented Computing”, Communications of the ACM, Oct. 2003