WebRTC and IMS: Parallel Universes on a Collision Course?

Stéphane Tuffin
Orange Labs Networks
Orange, France
stephane.tuffin@orange.com

Frédéric Martelli
Orange Labs Products & Services
Orange, France
frederic.martelli@orange.com

***Abstract*—With IMS and WebRTC being both recognized as key technologies for communication services and as technological investments are ongoing there is a need to understand what each technology can do that the other cannot. This paper provides an extensive analysis allowing for a side by side comparison of IMS and WebRTC taking into account the respective standards but also how the technologies are implemented and the ability of their respective ecosystems to drive further evolutions.**

Keywords—IMS; WebRTC; Real-time communications

I. Introduction

With VoLTE deployments spreading worldwide [1], IMS is recognized by the Telecommunication industry as a key technology for communication services. In the meanwhile, WebRTC, which is reportedly supported by 1.5bn devices EOY '14 with more than 4bn anticipated EOY '16 [2], is gaining significant traction from the Web industry. This paper aims at making clear what each technology can and cannot do.

Rather than limiting the scope of the paper to a strict analysis of standards (WebRTC is defined as a media protocol and an associated API while IMS is defined as an architecture for delivering communication services), the paper provides an analysis allowing for a side by side comparison (e.g. control planes used in combination with WebRTC are covered). Furthermore, it considers how the technologies are implemented and the ability of their respective ecosystems to drive further evolutions.

II. Different Business Goals Embedded in Each Technology

The IMS technology has been developed for a service provider federation, the telephone companies in collaboration with technology providers commercializing off-the-shelf components. In a service provider federation, each provider delivers a common service to its own subscribers. Interconnection of service providers allows the common service to benefit from a network effect. Standardization of IMS components allows multiple service providers deploying the same product to benefit from a wider amortization of development costs while allowing some competition (primarily on price and quality) between technology providers. The modular IMS architecture distinguishes the core components that should be reusable between different communication services (e.g. IMS Call Session Control Function) from service specific components (e.g. IMS Application Servers). Lastly, IMS is meant to allow IMS client providers, most notably mobile phone manufacturers, to embed the components giving access to the federated services independently from the service provider. Consequently, the user interface of an IMS service is implemented by mobile phone players rather than service providers.

In loose relation to the IMS technology, some APIs are proposed on devices and by some service providers to enable the integration of these federated communication services in third-party or service provider specific applications.

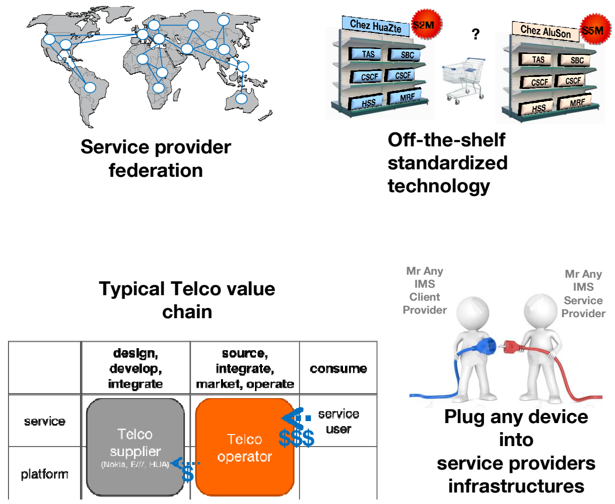

Fig. 1. IMS business goals

WebRTC has been developed to add real-time communication capabilities to the Web. WebRTC deals with two aspects of real-time interactive audio/video/data: media plane interoperability between endpoints and media plane programmability at the endpoints. WebRTC programmability allows software developers to establish/modify/terminate real-time interactive media flows and to connect these flows to input-output devices (mic, camera, earpiece, screen…). Web Browsers are the primary endpoints covered by the technology together with Mobile OS applications. The main WebRTC technology providers (i.e. WebRTC stacks providers) do not intend to directly monetize the WebRTC stacks (e.g. Web Browsers are not sold to their users). WebRTC is designed to allow a service provider to develop and deploy[1] its own code on the participating endpoints. Here the service provider is meant to be in direct control of its user interface and application logic.

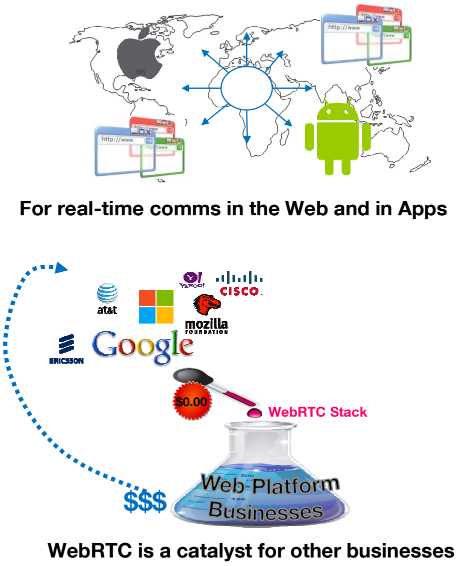

Fig. 2. WebRTC business goals

More gradual differences about what these technologies enable and are intended for are deriving from these business characteristics of IMS and WebRTC. This is the subject of the remainder of this paper.

III. INCREMENTAL VARIATIONS ON PUBLIC TELEPHONY VS. TAILORED COMMUNICATIONS

The IMS model works when the business perspectives of a communication service generate enough consensus to trigger investments among the interdependent players (telephone companies, technology providers and end-user device manufacturers). Initial IMS investments in fixed and mobile telephone companies are triggered by the migration of public circuit switched telephone networks to IP. In mobile networks, it is anticipated that VoLTE (i.e. Telephony over LTE) will generalize this movement at smartphone manufacturers. Incremental innovation would then only necessitate a fraction of the initial IMS investment at the three types of involved players. On the other hand, innovative features yielding no return on investment at one of the three types of players block the machinery.

[1] e.g. as a user browses the provider's Web site, as a user installs the provider's App from a Mobile AppStore

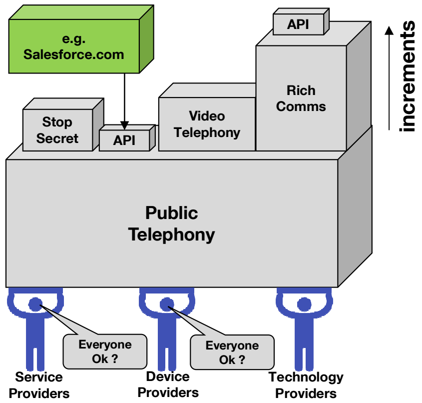

Fig. 3. Variations on public telephony

With WebRTC, companies investing in communication features[2] depend on the investments of WebRTC stack providers (for the quality and breadth of Browsers and Mobile OS support, for the evolution of media plane features). A key difference with IMS investment dynamic is that communication service aspects out of WebRTC's scope (e.g. user-experience design; context awareness embodiment; management of identities; session control logic;…) are locally decided at each investing company without requiring vast consensus between service providers, technology providers and device manufacturers.

Going along with this, the choice of the components complementing WebRTC stacks is left to each investing company. Depending on the targeted communication features, some will select a Telephony -oriented framework [3]-[5] while others will select significantly different components such as:

- Publish-Subscribe, Real-time DataSync [6] and "Room"-based frameworks [7]-[9] which better fit group communications or rendezvous based conferencing than Telephony-oriented frameworks (primarily meant for one-to-one, spontaneous call).
- Simple and programmable communication frameworks which better fit cases where communication is not a core product feature. This allows the service provider to focus on its core objectives (e.g. integration to core product features and own IT choices) more efficiently than Telephony-oriented frameworks carrying the complexities and specificities of the Telephony heritage and allowing limited freedom in IT choices (e.g. Cloud hosting; Middleware; Development language…).
- Pure session management frameworks unhindered with identity management features [10]. This can be a better fit than a Telephony-oriented framework (which embeds identity management) when there is the need to use an independent user authentication/authorization/profile management system (e.g. to leverage a social network; to integrate an enterprise directory/authentication system such

[2] Whether as standalone communication applications or as communication features embedded in other applications

as Microsoft Active Directory; to benefit from alternative authentication methods such as behavioral biometrics…).

- "NoBackend"[3] frameworks, which permit shortened software development time by focusing efforts and competencies on the client side while a Telephony-oriented framework would require both client and server developments.

In parallel, the media features commonly available in WebRTC implementations and rarely found in IMS implementations such as video, simulcasting[4], screen sharing and the data channel[5] illustrate the greater motivation of WebRTC technology providers to support media features going beyond classical telephone calls.

Conversely, the audio codecs and encryption mechanisms retained in WebRTC implementations illustrate how much the importance of public telephone network differs in WebRTC and IMS:

- Adaptive Multi Rate - Wide Band (AMR-WB) is generally unavailable in WebRTC implementations, which is detrimental to the interworking with public mobile telephone networks (increases its cost and deteriorates quality)
- WebRTC end-to-end encryption with keying material mandated to stay in the endpoints is detrimental to public telephone network requirements regarding lawful interception [11].

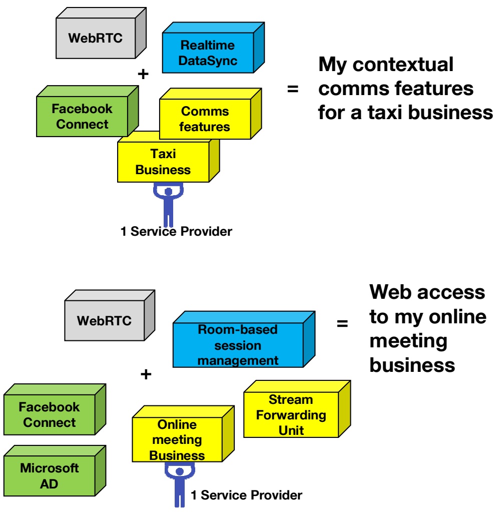

Fig. 4. Tailored communications

[3] http://nobackend.org/

[4] Simulcasting enables "Selective Forwarding Unit" based videoconferencing

[5] Supporting generic data on the media plane allows for a range of use-cases such as peer to peer CDN, game control, remote machine control, mesh networking and possibly fog computing

IV. Different Innovation Cycles, Different Speeds and Ways of Meeting User Expectations

With IMS there are two distinct types of user-facing innovation:

- Innovation based on broad consensus between service providers, device makers and technology providers (such as in Rich Communications Services, Video over LTE and VoWiFi);
- Local footprint innovations requiring support from few service providers with the condition that standard UNI and NNI are left untouched (as it is the case for some supplementary services and services that could be based on IMS-related APIs).

In the former case, standardization has to be the starting point of the innovation process. Then the innovation may start being implemented by device manufacturers, by technology providers, integrated[6] and deployed by each of the federated service providers. Ultimately, a new cycle can restart, fed by new ideas and deployment feedback. Such an innovation cycle is slow by definition and difficultly accommodates user-centered design as validity of design assumptions are slow to be tested.

In the latter case, the innovation cycles may be faster but innovations are of limited scope as UNI and NNI must be unaffected.

Beside user-facing innovations IMS may also have to evolve to reduce the overall cost of public telephony infrastructures, to support additional network access types (e.g. MSAN with POTS cards for PSTN renewal) and to adapt to overall technological waves (e.g. virtualization).

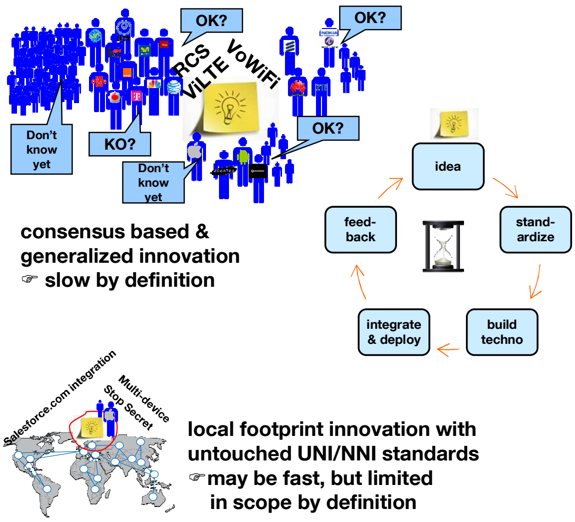

Fig. 5. IMS innovation cycles

[6] This is a significant burden due to the tight integration with connectivity networks (e.g. dedicated radio bearer and associated engineering) and due to the number of interworking scenarios (legacy devices, communication services and third-party service providers): significant IMS innovations require fine end-to-end integration and verification of impacts.

With WebRTC there are also two distinct types of innovations:

- Innovations using the available features in the WebRTC media stack
- Innovations in the WebRTC media stack

In the former case, there is no need for standardization. A single service provider can bring innovations to its whole user base. It can also apply continuous improvement process in a tight loop in order to improve its service. As WebRTC is not meant (and generally not used) for federating multiple service providers and as WebRTC is loosely coupled with connectivity networks, the end-to-end integration and verification are simplified in that regards. However with WebRTC, the integration and verification complexities would rather come from the fragmentation and the evolutions of device platforms supported by the service provider.

In the latter case, there is a need to have the key WebRTC stack providers supporting the innovation to maximize its value. It can be noted that several key WebRTC stack providers are sharing their code bases in open source (Google, Mozilla/Cisco and Ericsson but not Microsoft), probably in the aim of spreading the influence of their own strategical choices. This has two side effects on innovation speed. Firstly it tends to increase the speed of innovation propagation to other competing WebRTC stacks (i.e. there is a working code that could be taken as a reference or reused). Secondly, it allows other players with incidental interest (such as Tokbox, Jitsi, Temasys…) to request or patch incremental features and bug corrections. Finally, there are only three key WebRTC stack providers (Google, Mozilla/Cisco, Microsoft) with a potential fourth one (Apple/Ericsson): this low market fragmentation allows them to implement and evaluate media stack evolutions on significant market shares and standardize later once interest is demonstrated and when conflictual positions need to be reconciled.

One factor that can slow down innovation in the WebRTC stack is when there is an impact on the code developed by service providers on top of the WebRTC API (i.e. when compatibility cannot be maintained, even with shims).

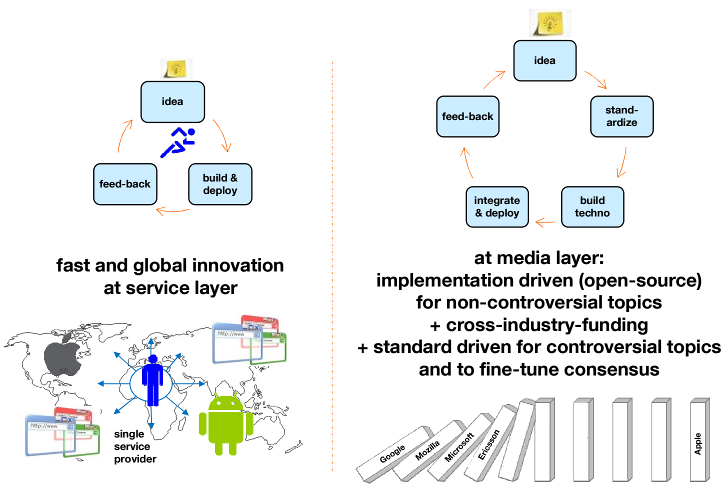

Fig. 6. WebRTC innovation cycles

V. Different postures about connectivity network integration

In the IMS deployment model, each telephone company integrates an IMS system with the network(s) it operates. This is especially the case for access networks in which specialized network processing are applied to the media flows established through IMS systems. These treatments are meant to mitigate the adverse effects of concurrent flows (bandwidth starvation, delay, jitter) by giving IMS voice a prioritized access to transmission scheduling and bandwidth. On LTE, they also allow for voice coverage enhancements (TTI bundling), optimized battery lifetime (DRX) and voice continuity during handovers with 2G/3G (SR-VCC) and Wi-Fi (Mobile IP). IP network interconnections (including roaming) also receive special care as the IMS system steers the flows through IP interconnections points, apart from Internet peering, whose capacity is managed according to distinct service level agreements between the interconnected telephone companies.

In side-cases, IMS is used over unmanaged IP connectivity access networks (i.e. networks not managed by the company owning the considered IMS system) as it is the case for the VoWiFi extension of VoLTE, Wi-Fi being used for coverage extension when there is no cellular coverage. However, the slow uptake of IMS features specifically aimed at unmanaged networks (such as restrictive firewall traversal support, adequate supervision features, application layer adaptation to variable network conditions, encapsulation overhead optimization…) highlights that it is not a core focus of the technology.

On the smartphone's side, the IMS voice stack is due to be well integrated, by the device maker, in the hardware (e.g. DSP-based voice processing) as Telephony is still a core product feature. This contributes to voice quality[7]. Such care is due to the importance of Telephony quality for selling smartphones and is not an intrinsic property of the IMS technology[8].

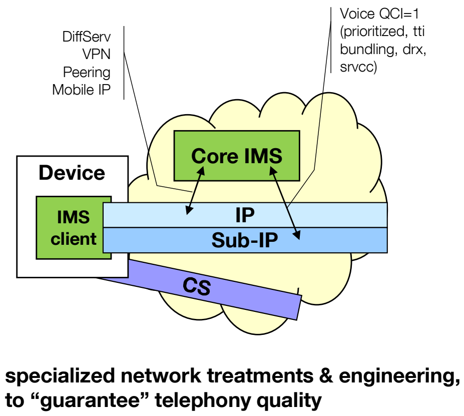

Fig. 7. Connectivity network integration in IMS

[7] Including optimized battery lifetime during voice calls
[8] Video quality also received such care (e.g. DSP based codecs) though not intrinsic to IMS

In the WebRTC deployment model, Internet connectivity is assumed. Despite that specialized network services are sporadically considered[9] [12],[13] in this technology there is no agreed practice for generalizing their use over the Internet. So, rather than assuming access to specialized network treatments, WebRTC deals with the impairments caused by concurrent traffic to real-time interactive media at the higher layers with:

- RTP congestion control techniques [14] (e.g. to adapt the frame rate, resolution, redundancy and retransmission according to observed bandwidth, latency and packet loss), addressing video first with known intents to cover audio later;
- Codec specific mechanisms (such as opus in-band FEC).

Simultaneously, congestion control of web transport protocols (e.g. WebRTC's congestion control, QUIC's congestion control) is moving from the inside of operating systems (kernel space) to software running on top of them (user space), and especially within Browser engines and Web stacks. This should trigger a variety of experimental algorithms and facilitate tuning. We expect to see further attempts in the following fields:

- Cross-layer congestion control where the transport layer leverages radio layer throughput estimation (as enabled by Android M Link Capacity Estimation API and as illustrated by the CQIC prototype [15],[16]);
- Cross-application congestion control, where the Web stack orchestrates how the network interfaces of a device are shared by simultaneous application flows.

Further dealing with QoE on top of unmanaged IP connectivity, WebRTC implements solutions for:

- Providing endpoint based KPIs (i.e. WebRTC's getStats API) to feed media quality analytics services;
- Supporting Internet access protected by restrictive firewalls (common in enterprises).

In the same aim, some additional features are still under development, such as methods for selecting the media path (among several possible ones) with the best performances for real-time interactive media and anchor-less mobility between IP access networks [17].

Lastly, given the increasing bargaining power of application providers with respect to Internet connectivity providers and given the increasing opacity of transport protocols toward the network (ciphering, multiplexing, QoE adaptation), attempts are being made (such as in SPUD [18]) to rethink how application providers and network providers could collaborate. In these attempts the two types of players trade visibility and adaptations to enhance their respective performances in a neutral way rather than in the scope of business agreements.

[9] Chrome allows to tag the media flows with specific DSCP values

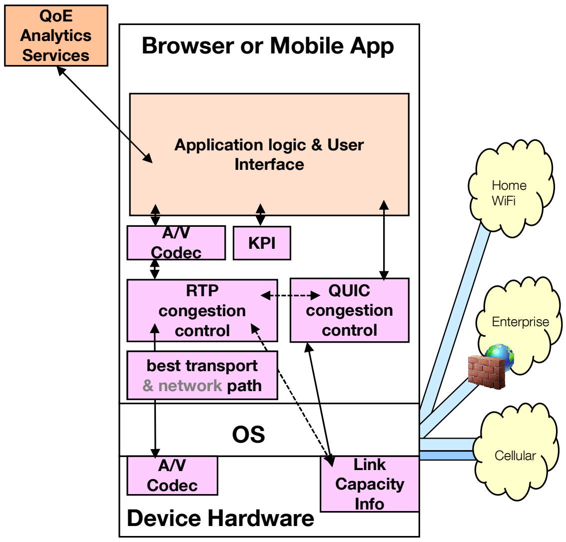

Fig. 8. Connectivity network integration in WebRTC

VI. DIFFERENT POSTURES REGARDING LAWS AND REGULATION

IMS is designed and deployed to be compatible with lawful interception and judicial requisition requirements while WebRTC is designed to work around pervasive monitoring. First, IMS systems supporting public telephone networks are deployed on the territory of jurisdiction of the served users[10] while systems (e.g. Web sites) used to establish WebRTC sessions can be located anywhere. Second, the end-to-access edge encryption model adopted in IMS deployments allows lawful authorities to intercept unencrypted media from IMS core networks while WebRTC mandates end-to-end DTLS-SRTP which makes media interception unpractical [11] (media is encrypted with private keys located in the endpoints).

Similarly, IMS-based telephone networks are designed to support emergency calls requirements[11], while such requirements are not imposed to systems used to establish WebRTC sessions. Those systems, in general, do not have reliable caller location information and cannot be trusted given the diversity of players. However, once available, the solution developed in the scope of EC's standardization mandate M.493 [19] should enable any electronic communication system providing an originating call service to numbers in a national telephone numbering plan to comply with the obligation of supporting emergency calls requirements.

[10] IMS roaming excluded from this statement as implementation orientations are not yet settled ("S8hr vs. LBO")

[11] Call routing to public safety answering point (PSAP) based on caller location; transmission of caller location to PSAP; call-back from PSAP; unauthenticated emergency calls …

VII. Combining the Two Universes ?

A. *WebRTC access to IMS*

While WebRTC and IMS differ in the points previously enumerated, WebRTC and IMS can interwork through a gateway function. For IMS, the expected benefits are to extend its reach to devices with no embedded IMS client and to give service providers more control on the client-side application, in better cost/quality conditions than with non-WebRTC downloadable applications. In addition, the concept of guest IMS subscriber can allow IMS communications with non-IMS users. The drawback of this IMS add-on is coming from the gateway adjunct which has cost and quality impacts given the differences in the two media planes.

B. *Interconnecting WebRTC to Telephony*

While WebRTC doesn't define a standard interconnection interface, it is not the case of Fixed and Mobile Public Telephone Services which have standard Network-Network Interfaces (NNI) standardized.

Whether WebRTC interconnection with Public Telephone Services might influence these NNI is questionable: as we have seen before WebRTC is not meant to primarily interwork with public Telephony.

C. *Transplanting pieces of WebRTC in IMS*

We have seen that features related to unmanaged networks were not a core feature of the IMS technology while we have seen in section V that WebRTC includes RTP congestion control technics, endpoint based KPI (i.e. WebRTC getStats), restrictive firewall traversal with some potential for cross-layer congestion control, cross-application congestion control and anchor-less mobility.

The uptake of VoWiFi questions the opportunity of integrating such features within the IMS media plane and thus of transplanting the corresponding pieces of WebRTC in IMS.

D. *WebRTC beyond communication silos*

From a pure technical standpoint, WebRTC can be used where communication services are distributed by a service provider federation (different administrative domains interconnected through an agreed NNI and common service features).

For specialized business to business communications requiring agile evolutions (e.g. contextual communication between the enterprises of a given supply chain) this might indeed prove useful. We have seen that the IMS approach of service federation tends to ossify rather than to provide agility: ripping the full benefits of WebRTC in such contexts calls for a more agile NNI approach than the IMS one.

Going beyond communication silos would also require the users (humans or machines) of one administrative domain to securely discover and identify users of another administrative domain. The uniform Identity Management framework embedded in IMS solves that problem. However, it is detrimental to impose a unique identity management framework when the administrative silos have varying identity management needs (e.g. various pre-existing identity management solutions; various trust level requirement on identity attributes).

The reThink project [20] both investigates the ProtoFly approach [21] as a potential solution to the NNI agility issue and it attempts to solve the identity management issue mentioned above with a novel approach [22].

VIII. Conclusions and Perspectives

Although both IMS and WebRTC are technologies for real-time interactive communications we have seen they are meant to opposite business goals:

- IMS is meant to support the distribution of communication services by a federation of service providers
- WebRTC is meant to support communication services designed under the control of a single service provider

Next to these opposite service delivery models, IMS and WebRTC are also following opposite paths on the technology enablement side:

- IMS is meant to create a market of standardized swappable off-the-shelf components
- Aside from the media stack, WebRTC leaves to the market the definition of the complementing components required to build a working service

This has significant consequences as the IMS approach encourages the mutualisation of communication services on the same components while WebRTC induces competing and purpose specific components: WebRTC adopters get a broader choice but have to establish strategies.

Regarding the application fields of IMS and WebRTC, the differences in investment rationale and in media plane features clearly indicate different orientations:

- IMS can do public telephone network services and is meant to allow for incremental innovations on top of exiting telephony networks
- WebRTC is not firstly meant to do public telephone network services but can apply to tailored application fields, including telephony-like services as long as PSTN breakout is not dominant. A faster support of non-telephony media features is witnessed in WebRTC compared to IMS

We have also seen there is a huge difference between WebRTC and IMS about communication service innovation: with a more efficient way of meeting user expectation in WebRTC due to more autonomy at the service providers and faster innovation cycles. Given the way technological developments are organized, WebRTC is possibly in position to receive more innovations than IMS at the media stack level.

The postures regarding media quality management are radically different: while in IMS, media quality management is focused at when the service is accessed from the service provider's network, in WebRTC it is independent from the connectivity network provider with a fundamental approach consisting in adapting the media to variable network conditions at device application level. This field is rather active in WebRTC with several approaches in their early stages. So even if there are already hints that a WebRTC-like approach can give better results for video [22], we are not yet in a position to anticipate the upcoming results in other fields (e.g. audio, selection of the best network for a real-time interactive flow, smart interactions with non-WebRTC flows). It is however clear that if media quality management at device application level brings significant gains it will be first covered by WebRTC and not by IMS.

Finally, the analyzed combinations of IMS and WebRTC exhibit some unexplored potential beyond the traditional interworking approaches, namely the transplant of pieces of WebRTC technology in IMS to improve unmanaged network handling. On the opposite, the analyzed combinations show that IMS do not provide suitable solutions to empower WebRTC beyond communication silos: here, novel approaches would be required.

Following up on this analysis of what IMS can do that WebRTC cannot and conversely, comes the question of the investments at Telcos. How to take the best of past and ongoing heavy IMS investments? How to best prepare the networks for being used with WebRTC-powered applications? How to invest in WebRTC-powered applications?

REFERENCES

[1] Global mobile Suppliers Association, “VoLTE deployments booming,” Retreived from http://www.gsacom.com/news/gsa_425, April 2015.

[2] Disruptive Analysis, “2015 Q1 update: WebRTC market status & forecasts report,” Retreived from http://disruptivewireless.blogspot.fr/p/blog-page_30.html.

[3] Project Clearwater, “About clearwater,” Retreived from http://www.projectclearwater.org/about-clearwater/.

[4] Freeswitch, “Freeswitch,” Retreived from https://freeswitch.org/.

[5] Digium, “Asterisk WebRTC support,” Retreived from https://freeswitch.org/.

[6] Phil Leggetter, “Choosing your realtime web app tech stack,” Retreived from http://www.leggetter.co.uk/2013/12/09/choosing-realtime-web-app-tech-stack.html, December 2013.

[7] Tokbox, “Opentok API architecture,” Retreived from https://tokbox.com/developer/concepts/architecture/.

[8] Temasys, “Getting started with WebRTC and SkylincJS,” Retreived from https://temasys.github.io/how-to/2014/08/08/Getting_started_with_WebRTC_and_SkylinkJS/.

[9] Bistri, “Javascript library reference,” Retreived from http://developers.bistri.com/webrtc-sdk/js-library-reference/.

[10] Rtc.io, “Rtc.io,” Retreived from https://rtc.io/.

[11] Ben Adida et al, “CALEA II: Risks of Wiretap Modifications to Endpoints,” Retreived from https://www.cdt.org/files/pdfs/CALEAII-techreport.pdf, May 2013

[12] E. Janczukowicz et al., “Specialized network services for WebRTC: TURN-based architecture proposal,” AWeS’15 Proceedings of the 1st Workshop on All-Web Real-Time Systems, Article No. 3.

[13] K. Haensge et al., “QoS-based WebRTC access to an EPS network infrastructure,” Intelligence in Next Generation Networks (ICIN), February 2015.

[14] IETF, “RTP media congestion avoidance techniques (rmcat),” Retreived from https://datatracker.ietf.org/wg/rmcat/documents/.

[15] Google & UC San Diego, “CQIC: Revisiting cross-layer congestion control for cellular networks,” Retreived from http://cseweb.ucsd.edu/~snoeren/papers/cqic-hotmobile15.pdf, February 2015.

[16] Google, “Sharing network state with application endpoints,” Retreived from https://www.iab.org/wp-content/IAB-uploads/2015/08/MaRNEW_1_paper_19.pdf, August 2015.

[17] Google & Vidyo, “Improvements to ICE Candidate Nomination,” Retreived from https://tools.ietf.org/html/draft-uberti-mmusic-nombis-00, March 2015.

[18] IETF, “Session Protocol for User Datagrams (spud),” Retreived from https://datatracker.ietf.org/wg/spud/documents/.

[19] European Commission, “Standardization mandate to the European Standards Organisations in support of the location enhanced emergency call service,” Retreived from http://www.etsi.org/images/files/ECMandates/m493.pdf, May 2011.

[20] reThink project [Online]. https://rethink-project.eu/

[21] reThink Project, “Framework Architecture Definition,” Retreived from https://bscw.rethink-project.eu/pub/bscw.cgi/d29495-2/*/*/DOI-D2.1.html (DOI 10.18153/RTH-645342-D2_1), July 2015

[22] reThink Project, “Management and Security features specifications,”, Retrieved from https://bscw.rethink-project.eu/bscw/bscw.cgi/d30341-3/*/*/*/DOI-D4.1.html (DOI 10.18153/RTH-645342-D4_1), September 2015

[23] Signals Ahead, “Behind the volte curtain, part 3 - benchmark study of ir.94 and multiple ott video chat applications,” Retreived from http://signalsresearch.com/issue/ir-94-video-telephony-benchmark-study/, March 2015