

# How might webRTC be used?

# “The proposals”

Version history: v0.10d DARFT

Date: 2 August 2013

Author: Istvan Lajtos

Contents

[How might webRTC be used? 1](#_Toc363216614)

[“The proposals” 1](#_Toc363216615)

[1. Introduction 2](#_Toc363216616)

[2. Technical summary 2](#_Toc363216617)

[3. Operators landscape 3](#_Toc363216618)

[4 Network Equipment Manufacturers landscape 5](#_Toc363216626)

[5 webRTC SWOT analysis 9](#_Toc363216635)

[6 The value propositions 9](#_Toc363216636)

[7. Use case scenarios 11](#_Toc363216639)

[8. Options for operators leveraging webRTC technology 14](#_Toc363216649)

[9. webRTC Weaknesses and Related Opportunities for Operators 15](#_Toc363216652)

# Introduction

This document is to describe the advantages and highlight the benefits of webRTC (Real Time Communications) technology. WebRTC (Web Real-Time Communications) is a technology for a joint initiative between W3C and IETF aimed at achieving native browser support for real-time communications over the web.

In practical terms, WebRTC provides the ability to stream media (voice, video, images and text) between two browsers and exposes this functionality via a set of simple web APIs so that developers can quickly and easily build these capabilities into new online services.

The current landscape is quite fragmented and there have been a few trials and demos have been implemented by key industry players. Mozilla and Google have developed the Internet browser solution supporting webRTC in Chrome and Firefox. Traditional equipment manufacturers like Ericsson, Huawei etc. have developed a WCG (Web Communication Gateway) product for bridging the IP and Telco worlds together.

# Technical summary

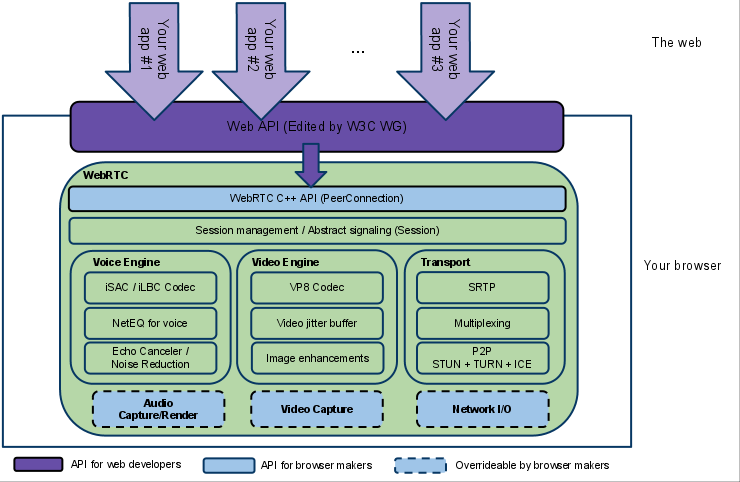
webRTC is currently being defined by W3C and IETF standardisation bodies, which enables Internet browsers on devices with real time communication options via Javascript APIs. W3C are developing the APIs specifically for the real-time audio, video and/or data transfer, whereas IETF is developing the signalling aspects of webRTC technology.

* the browser APIs developed by W3C for webRTC in brief are:
  1. getUserMedia - it enables audio and video stream or both local devices
  2. MediaStream - it enables peer to peer media streaming between devices
  3. DataChannel - it enables connecting remote peers using NAT, ICE, STUN

Real –time communication is not a new capability (many websites already support voice & video chat) but with WebRTC it will be much easier to implement and not rely on the user having to download and install browser plugins such as Flash hence lowering the barriers to adoption.

The main advantage of this technology is as follows:

1. it does not require any plugin for apps executed in the browser (JavaScript and HTML5)
2. there is no need to for any native apps is required to support real time communication

 source: [www.webrtc.org](http://www.webrtc.org)

1. webRTC architecture

webRTC features implementation depends on browser manufacturers and its vary from vendor to vendor. The current snapshot highlights the Internet browser features landscape and timelines as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vendors** | **Desktop Browsers** | **W3C APIs** | | |
| **DataChannel** | **PeerConnection** | **GetUserMedia** |
| Mozilla |  | *Nightly builds* | *Nightly build* | *Nightly builds* |
| Google |  | *Needs development* | *Support* | *Support* |
| Opera |  | *2014* | *2H ‘13* | *Support* |
| Samsung |  | *unknown* | *Unknown* | *Support* |
| Ericsson |  | *Support* | *Support* | *Support* |
| Microsoft |  | *2014/15* | *2014/15* | *2014/15* |
| Apple |  | *2014/15* | *2014/15* | *2014/15* |

1. Browser landscape and supported features status

This means the current browser implementations are fragmented and it depends on the vendor to control and implement features. (For example Microsoft Internet Explorer and Apple Safari) This means more fragmentation, which is going to be the challenge for device manufacturers to select the right browser implementations on mobile platforms.

Microsoft has also developed their version called CU-RTCweb solution which is similar but mainly differs from the W3C & IETF defined standards. There is no intention in this paper to provide details Microsoft’s implementation proposal. According to the latest updates, Microsoft is not planning to develop their own proprietary solution forward which could cause even more fragmentation in mobile development.

Mobile OS vendors not always own the browser implementation. Device vendors could also face issues with proprietary browser implementations like webkit on Android which is more likely to be replaced by Chrome or Firefox browser implementations.

# Operators landscape

There are several mobile operators started investigating the opportunities of webRTC and have been experimenting a working solutions in their lab environment. Other mobile operators are working with start-up organisations like Voxeo, Twillio and Tokbox. More details are provided below with operators consent and using information available on the public domain.

## AT&T

AT&T have already developed solutions in webRTC space and published their proposals at OMA conference was held in July 2012. AT&T have been heavily involved in W3C standardisation of webRTC and also exposed some early API solutions only to their developers. The API implementation is basic and provides capabilities towards IMS/telephony for developers.

References: <https://att.io/docs/webrtc>

## Vodafone

Vodafone activities are mainly around W3C /IETF standardisation activities and cooperating with Twilio and Voxeo in the webRTC space. Vodafone has been experimenting webRTC in their test bed. Expressed concerns about the issues surrounded of webRTC around security, firewalls and unstable browser implementations etc.

References: <http://www.slideshare.net/deanb/web-rtc-presentation>

## Orange

Orange is currently working on exploring opportunities through exploiting their own developer Javascript SIP stack. Their solution of webRTC proposal was leveraging Websockets through Mobicents server. There are no concrete plans for implementations at this stage presented.

References: <http://lanyrd.com/2012/mobicents-summit-2012/sxfqp/>

## Telefonica

Telefonica is also very active in the webRTC space. Telefonica Digital which is their group arm acquired Tokbox in October 2012 for evolving their 2012 summer launched TuMe IM services with voice and launch TuGo (video and VoiP services). Telefonica is one of the pioneers of Mozilla Firefox OS and developed full HTML5 based touchscreen smartphone propositions for Latin American market. The Firefox OS will be the first natively enabled webRTC smartphone which will revolutionise future communications.

TuGo is not using WebRTC directly as it is now implemented as a native application on each mobile platform. Anyway, they see the potential of webRTC for making portable apps based on the browser, and indeed they see interesting the availability of WebRTC libraries/components for native applications to ease the portability between different OS platforms.

Mozilla is integrating webRTC in the last version of the Firefox browser (21.0+). It is expected to have a porting for Firefox OS, but it has not been scheduled yet for short term version.

Finally, Tokbox released as Open Source the OpenTok for WebRTC  Platform (<http://tokbox.com/opentok/webrtc/docs/index.html> )  This kit allows to develop video chat applications using OpenTok under a freemium business model (pay per minute packages of group chat, first block for free) and different commercial support levels. This webRTC is on production level but with some limitations compared to the Adobe Flash Player version.

References: <http://www.slideshare.net/deanb/web-rtc-presentation>

## Deutsche Telekom

DT has presented their strategy in the 2012 webRTC conference which was confirming that they have a solution running in their test bed.

The current status is DT is currently exploring and evaluating more opportunities and understands the business model for leveraging this technology.

More likely, that webRTC is considered in DT core products to enhance the capability of RCS for example.

DT has seen webRTC and disruptive technology which would help operators to leverage real time communications by providing the connectivity to customers.

It could also spare network resources and provide a unique solution to specific B2B2C use cases.

References: <http://www.slideshare.net/deanb/web-rtc-presentation>

## KT

KT is in the process of conducting preliminary study to review various services models and technology feasibility. Once the strategic positioning is defined and the technology solution is stabilized, we would consider commercial service based on the feasibility study results.

- KT has conducted preliminary study on WebRTC and it is quite at an early stage to consider it as commercial solution

- The reasons are as such: lack of browser coverage, minimal device access, pushes capability not supported

- It needs to be improved to meet the current demands of market users’ level of expected

QoS via WebRTC

- As Web RTC will continuously be developing and KT will monitor the service environment and will take necessary action when the market becomes more ready

References: *to be provided*

## SKT

SKT is active in exploring opportunities of webRTC being an advanced RCS operator in Korea. SKT has been testing webRTC in test lab and active members of webRTC W3C standardisation body.

References: *to be provided*

* 1. China Mobile

In Q3 2012, China Mobile (with Alcatel-Lucent) proposed a WebRTC related work item (‘RTCWeb client access to IMS’) in 3GPP SA1.

In Q4 2012, China Mobile developed a WebRTC demo to implement interoperate between WebRTC and IMS/RCS/PSTN base on CMCC G3 Communication Service, including browser to browser and Browser to CS/PSTN . (Note: G3 Communication Service is IMS service based on China Mobile’s SNS, over 40million users, website: http://g3.weibo.10086.cn/))

In Q2 2013, China Mobile released a new version of Fetion IM clients for Windows, iOS and android, which integrate WebRTC and support video call. The new version of Fetion web IM which support WebRTC is in the development plan. (Note: Fetion is China Mobile ‘s IM service, over 300 million users,website: <http://feixin.10086.cn/>)

Currently, China Mobile mainly focus on research and development on interoperate between WebRTC and IMS/RCS/PSTN/Fetion IM , Open API, Billing, Qos , WebRTC based value-added service.

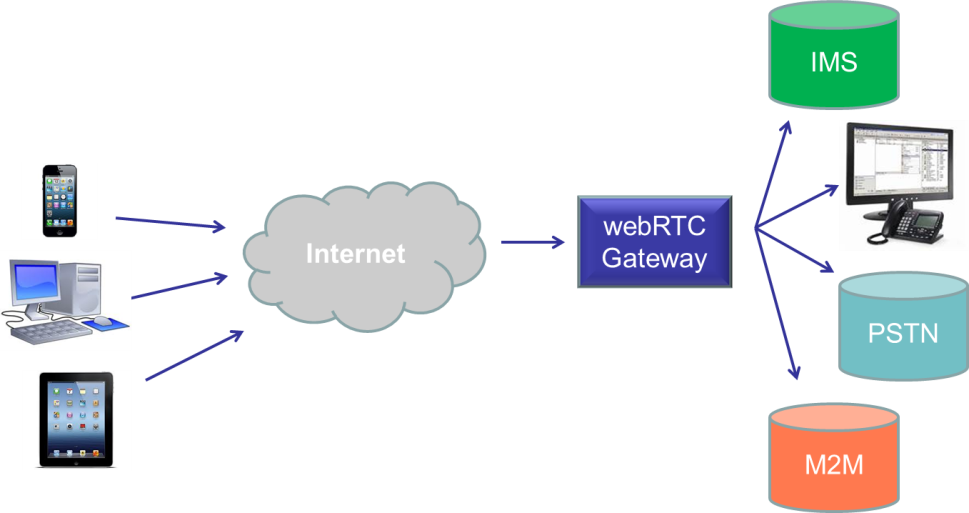
References: *directly provided by China Mobile*

# Network Equipment Manufacturers landscape

The following vendors’ have developed a gateway solution leveraging webRTC. It worth mentioning the 2 main different implementation, Browser to Browser communications versus Browser to Gateway communications

## 4.1 Browser to Gateway Communication (telco integration)

The main telecom equipment manufacturers and a number of small vendors are also developing their webRTC gateway solutions. The purpose of their gateway implementation is to offer connectivity between telecoms IMS and Internet areas. The majority of network vendors take this approach as described below.



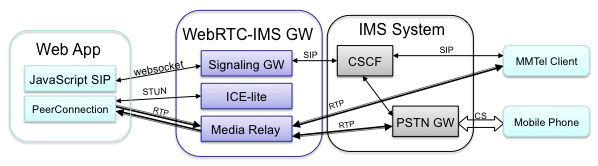
1. Browser to gateway communication architecture

This is the basic principle of browser to gateway communication. Many vendors have implemented IMS communication using the gateway as a SIP proxy. Vendors usually support media transcoding functionality in the gateway for different media types for audio and video traffic. Many of them also provide security features, support of ICE/TURN /STUN for transit traffic through firewalls and lawful intercept possibility.

The current vendors’ landscape looks like this:

### 4.1.1 Ericsson

Ericsson has developed a WCG (Web Communication Gateway) which is bridging the IMS (telco) and Internet worlds. It is mainly focusing on the browser to gateway combination path. The high level architecture design example for integrating to IMS (MMTel client) for making a fixed line call (PSTN) from a web application has been published on their blog site as shown on below picture.



Source: http://labs.ericsson.com/blog/webrtc-interworking-with-traditional-telephony-services

<http://www.ericsson.com/res/site_JP/press/2013/doc/mwc_2013_highlights.pdf>

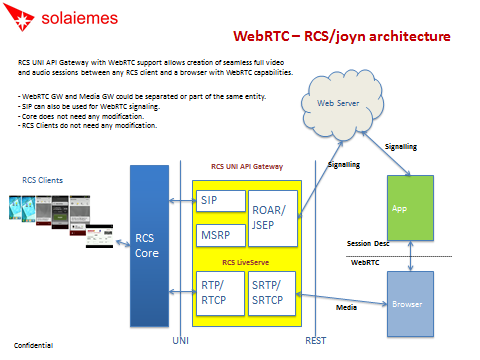
1. Ericsson architecture for webRTC interworking with PSTN

In this architecture Ericsson is providing connectivity and exposing RESTful APIs for developers to access real-time communication services with IMS integration. The Web App is leveraging browser Javascript APIs in the browser which then runs SIP over Websockets.

Ericsson is pretty active in IETF standardisation activities where the signalling protocol is being discussed in more detail.

### 4.1.2 Solaiemes

Solaiemes is one of the small vendors based in Spain, in the IMS space who developed a solution for webRTC and RCS interoperability. The rationale behind their implementation is to provide the solution for operators interconnecting telco world with web world. Their gateway high level architecture has

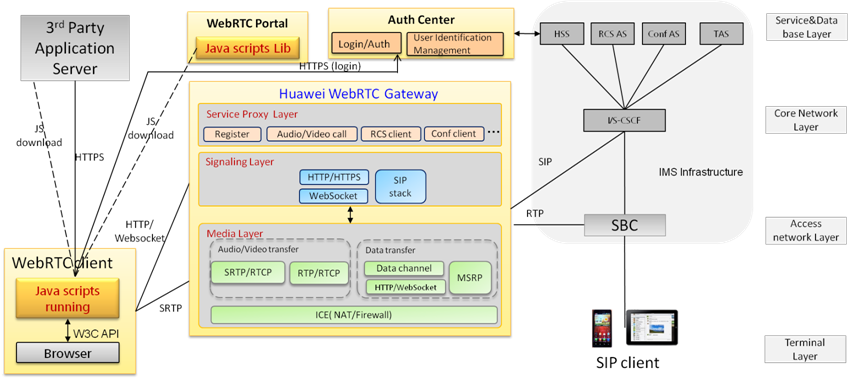


Reference: <http://www.slideshare.net/jaquayle/webrtc-workshop-slides-from-solaiemes>

<http://blog.solaiemes.com/2012/06/how-webrtc-will-empower-rcs-ejoyn-demo.html>

1. Solaiemes technical implementation architecture

### 4.1.3 Huawei

The implementation of Huawei webRTC gateway architecture is enclosed below.

1. Huawei gateway implementation

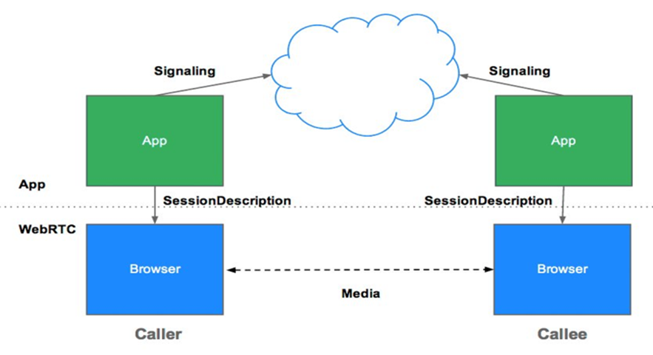
Reference: <http://www.huawei.com/en/about-huawei/newsroom/press-release/hw-259848-webrtc.htm>

Huawei has announced their first webRTC gateway solution which bridges the web and telco worlds. The implementation is integrating operators IMS core on one side and exposing Javascript APIs on the other side for web developers.

## Browser to Browser communication

### 4.3.1 Google Chrome

Google has started webRTC, by acquiring a company called GIPS (Global IP Solutions) in 2010. They made the solution free of charge for developer to use. Their intention was to create a real-time communication on the Web accessible for anybody without a restriction of having a dedicated client or plugin solution leveraging browser resources. (<http://webrtc.org>)



1. Java script Session Establishment (JSPEP) protocol architecture.

When the signalling process has been successful then data streaming can be started between two peers, Caller and Callee in this example above. Google has proven the browser to browser communication using Google HTTP server between Chrome and Firefox. Google has a strategy to introduce webRTC to the lately launched services called Google Hangouts. When users are registered/logged in to the service (using a web-server) could be calling each other’s, whilst they are online.

There are challenges with browser to browser communication as both users have to online, connected to the same server. The users need to know each other’s contact details (email, userID, number etc.) to be able to communicate.

### 4.3.2 Firefox

Mozilla are providing the nightly builds and supporting all the three main APIs for browser communications. Mozilla is active in W3C standardisation activities and released Firefox desktop and Android mobile browsers (version 21.0a1 on 16th May) which supports auto update and full features of webRTC. Later plans in 2013 will include Firefox OS based browser support of webRTC.

Reference: <https://blog.mozilla.org/futurereleases/tag/webrtc/>

### 4.3.3 Opera

Opera is providing some W3C API functionality but not the full suite.

Other browser vendors are also planning to implement webRTC in the browser; however W3C APIs implementation is not confirmed. The other challenge is the mobile browser webRTC support as the roadmap is different compared to the desktop version of the same browser solution.

# 5 webRTC SWOT analysis

The below summary contains the strength/weaknesses/opportunities and threats of webRTC for mobile operators. It depends on the use cases; this summary below is describing the generic technology features

|  |  |
| --- | --- |
| **STRENGTHS** | * allow operators for leveraging internet resources for communication to extend their RCS services on the Internet * it does not require plugin or additional client software for the in-browser communication * it provides simple web APIs for developers (web developers could use real-time communication in webapps) |
| **OPPORTUNITIES** | * consumer services could be developed for peer to peer real-time communication using browser resources (numerous use cases have been demod by gateway providers) * enterprise service es (enhancements) using webRTC   technology (numerous use cases have been developed)   * Operators have generally appetite to discuss this more * Operators could work together with OTT players to provide signalling for webRTC * Operators could leverage webRTC for internal use for sales tracking, offering customer service agents to access CRM in a seamless way etc. |
| **WEAKNESSES** | * There is no killer app exist for webRTC * lack of lawful interception / standards * codec issues (VP8 free codec versus H.264 licenced) * security issues * unstable browser implementation * browser features are driven by different vendors (different strategies and different level of API support) * slow mobile penetration strategy (dependent on browser / OS vendors) * lack of addressing and ringing people * both parties are willing to communicate real-time have to be connected to the same webserver * Standards are not finalised (being worked on) |
| **THREATS** | * webRTC could be leveraged without the use of RCS services * OTT players could offer real-time communication browser to browser services without operators involvement |

# 6 The value propositions

## 6.1 Enterprise space

The B2B solutions are driven by the more mature webRTC desktop implementations compared with mobile ones.

webRTC is more understood by the enterprise side where it offers cheaper and more flexible communication systems. It also offers a better interaction with customers and improved interaction with external partners.

webRTC could be in principle be running on any Internet connected device and it doesn’t even require a conventional browser it to run. (For example webview)

There has been many services accepted by the corporate world like numerous VoIP services where integrating web based communications services is more feasible

With that in mind, there is a lot of interest from equipment manufacturers like Siemens, Avaya etc. or cloud based API players like Twilio or Voxeo to provide webRTC based solutions for these enterprise customers.

Operators OTT play

(it could be considered deploying webRTC services without utilizing IMS core)

1. Operator could provide service extensions through web/soft phones leveraging their RCS services (RCS-webRTC interoperability) – IMS extension to the web
2. Operator specific OTT service deployment (examples are like Telefonica TuMe and Tugo solutions)
3. Operators specific cloud hosting for purely to their own customers (SaaS solutions with 3rd party solution provider)
4. Operators could extend their customer services by adding full video / audio interaction for customers (access to device info real-time etc.)
5. Operators could also offer a second screen OTT play like TV or content proposition.
6. Offering services for OTT provider for addressing, ringing people and offering address book integration

## 6.2 Consumer space

There is no killer app exists for webRTC yet but could play an interesting role in consumer space for enabling very innovative new services like:

* 1. Plugin-less browser to browser communication without client application
  2. Offering communication and calling options for medium sized social networking sites like LinkedIn for example for video based job interviews etc.
  3. Online learning / education (courses, tuition etc.)
  4. Many opportunities for the next generation unified enterprise communication “consumerazation” such as multiparty video conference
  5. Online gaming (using the smartphone as a remote control)
  6. Language exchange websites (online translation services)
  7. Mobile e-health (such as providing solutions for fitness, exercising etc.)

This is an unpredictable space in terms of user experience, monetization and legal aspects. More likely services will appear following the fermium model such as Yammer, Twitter or Facebook leveraging webRTC.

((more commercial options are welcome…..))

# 7. Use case scenarios

There is several use case scenarios could be considered for webRTC. In terms of segmentation, there are B2B and B2C use cases can be described:

## 7.1 B2B use cases

### 7.1.1 Click to call

In this use case the customer could access Customer Services by selecting a single click in the browser and placing the call when searching online.

1.) Customer is assembling furniture purchased in Ikea

2.) Then the Customer realises there is a screw missing from the

package

3.) Customers then picks up an RCS smartphone and makes a call

to Ikea customer services

4.) Ikea Customer Services agent receives the call and answers it in

the browser and asks customer about the issue regarding

furniture assembly

6.) Then the agent asks the customer to turn on video

stream during the phone call to check what screw s is missing

7.) The agent then can see on his/her browser the video stream

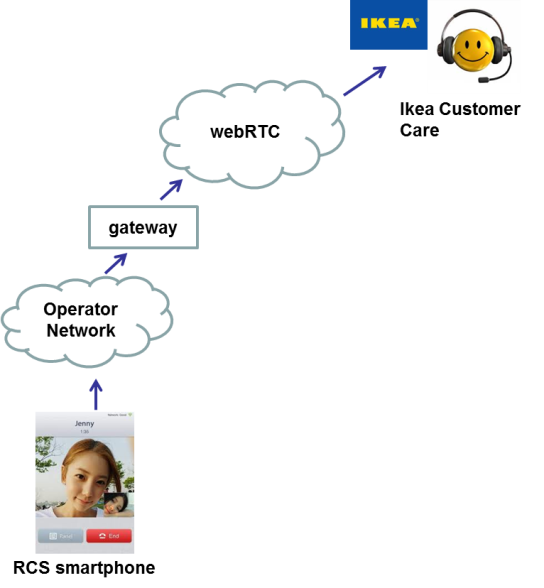
from the customer and can checks the missing screw type

8.) The agent places an order with the missing screw and

asks customer for any other help

9.) The agent also offers customer to send to the customer’s

phone a fully assembled 360 degrees image of the furniture



### 7.1.2 Multiparty conference calls

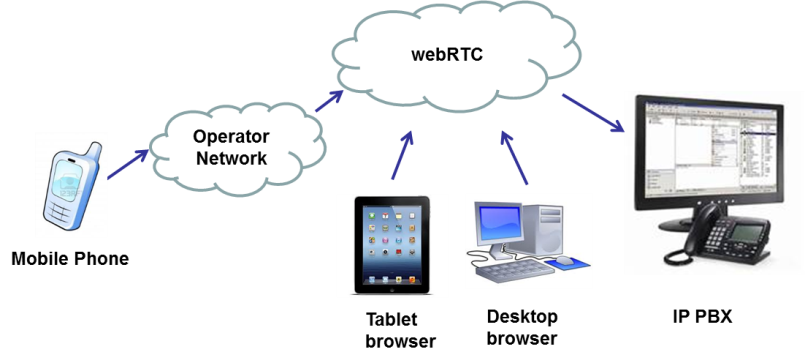
This is the next generation webex type multiparty conference call scenario when connected parties don’t have to use plugins and additional piece of software for multiparty conferencing.

### 

1. Customers are connecting a central conference service without downloading a client/plugin
2. Audio and video streams can be shared between all parties
3. RCS smartphones supported by operators could share video streams with any device connected to the same conference platform leveraging webRTC integration to operator services

### 7.1.3 webRTC to IP PBX call

In this scenario customers can make calls to IP-PBX to use webRTC as the interface between the two technologies.



1.) Customer is calling from a mobile an extension in IP-PBX

2.) The call is routed from the operator’s network through

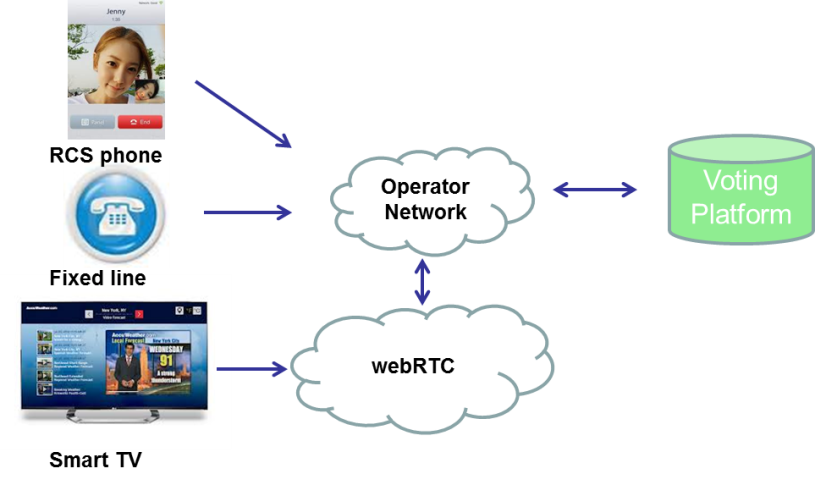
webRTC to IP-PBX exchange

3.) Operator picks up the call and puts through the calling party

to the called party’s extension

### Surveys and voting

In this use case scenario customer can engage with online voting/survey system using any connected device leveraging webRTC technology. It could be RCS enabled smartphone, connected TV and / or a landline.



1.) Customer is watching a television program where voting is

offered

2.) When the voting is opened, customer has a wide range of

access to various devices to place a vote

3.) Customer can use the interactive service on a smartTV

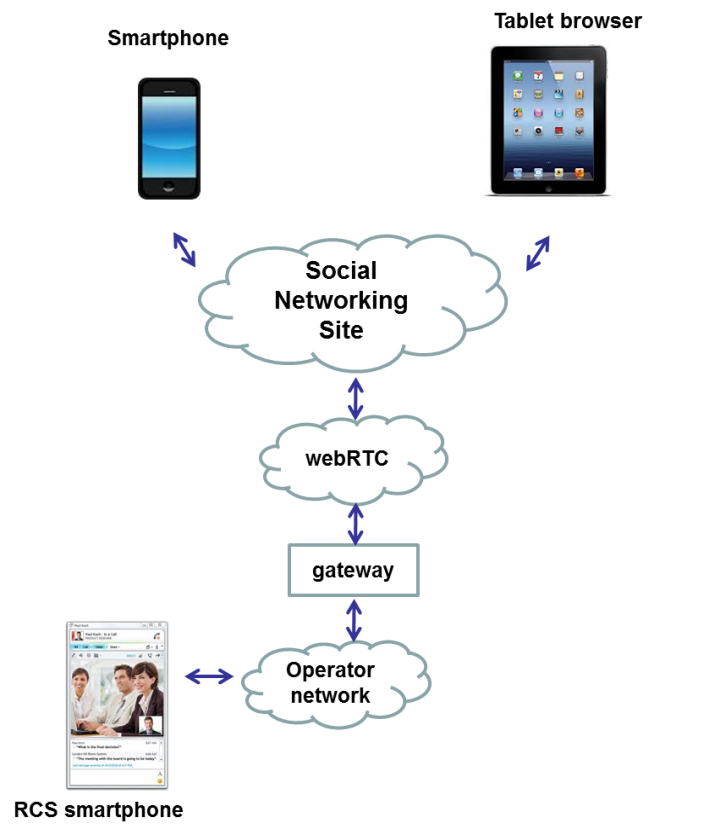
to place a vote online (as an example)

## B2C use cases

Consumer type use case scenarios are probably less developed at this time. These examples are some high level proposal about how webRTC technology can be deployed in consumer space.

### 7.2.1 Social networking breakout

In this use case scenario customers are using the social networking protocol to communicate (chat, file share and making audio/video calls) can involve a third party who is not registered at social networking site to an interactive conversation (sharing audio/video streams)



1.) Customer A is using a smartphone and registered at

Social Networking Site

2.) Customer B is using a tablet registered at the same

Social Networking site

3.) Customer A and Customer B is chatting and sharing

photos etc. (using the social networking protocol)

4.) Customer B mentions about their common friend who

they have not met for years and they would like to invite

to join a call. Customer C, is that person who is not

registered at Social Networking Site.

5.) Customer A has the contact details and calls Customer C

on the social networking (like Facebook Messenger)

6.) The social networking site has access to Customer C

contact details through web API

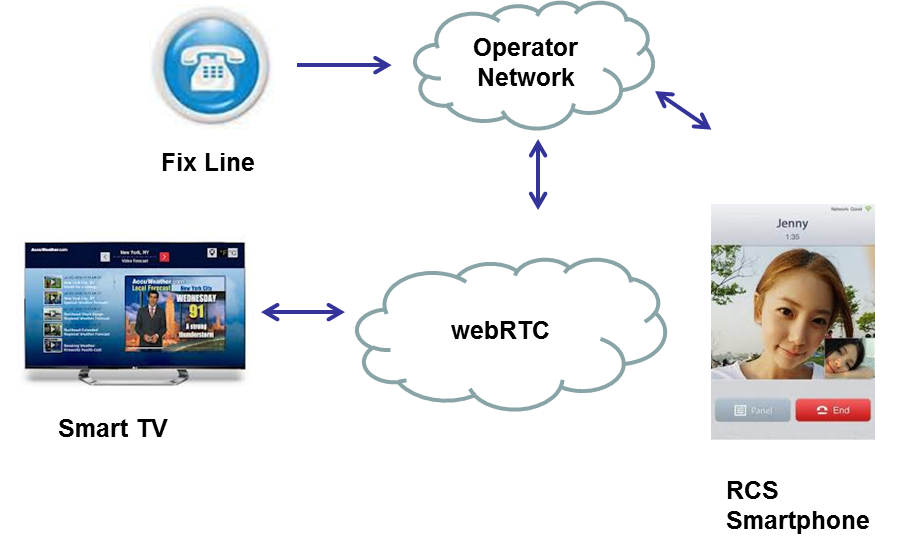
6.) Customer C answers the call and using RCS smartphone

turns on video stream during the call and shares live

video stream with Customer A and B

### 7.2.2 PSTN call to webRTC

In this use case scenario customer who has no access to RCS smartphone can connect with thos customer who has and using their connected smartTV could share video/audio and streaming/file sharing features leveraging webRTC.



1. Customer is making a phone call from fix line to a friend who is using a smartphone
2. Customer who is called shares the news about their new-born baby
3. Then asks the calling customer to turn on the connected TV to share some videos and photos ut the about their new-born baby (customer calling from the landline has got no access to a smartphone)
4. Called customer then shares the videos / photos

# 8. Options for operators leveraging webRTC technology

Operators can leverage webRTC is several ways. Operators who have integrated IMS in their core network can benefit as well operators who don’t have IMS infrastructure.

## 8.1 Operators having IMS / RCS based services

There are several options available for those operators who have integrated IMS/RCS services in their core networks. These options are available but not limited to:

1. Consider deploying webRTC gateway solutions to bridge IMS and IP communications
2. Provide Quality of Service for webRTC (offering guaranteed service quality solutions for business customers)
3. Harnessing a wider pool of web developers by exposing straight forward Javascript APIs
4. Combine RCS services to provide operator identity and leverage MSISDN for addressing
5. Provide a wide range of RCS services through 3rd party software which prefers webRTC technology on multiple devices

## 8.2 Operators having no IMS / RCS based services

There are some options available for those operators who have no integrated IMS/RCS services in their core networks. These options are available but not limited to:

1. Implement own Internet Player (OTT) like services. This is a typical implementation

proposal for no-frills operators\*

1. Develop own webRTC only communication services for business customers

In these options listed above operators could provide the enhancement for webRTC making it a fully featured real-time communication service like addressing, adding quality of service, enterprise integration, address book integration etc.

CONCLUSION FROM PSMC Shanghai June ’13

So the areas were proposed to add / cover in the white paper is:

1. Embrace a cross operator opportunity leveraging webRTC (providing **authentication / identity** which are not part of the standard!)
2. Operator opportunities for adding signalling framework to webRTC (to overcome addressing / ringing issues)
3. Demand an operator specific codec (set of codecs) for audio /video and push for standardisation for 3GPP
4. Explore more monetization options of webRTC for operators

# 9. webRTC Weaknesses and Related Opportunities for Operators

Several elements shown in  [Figure 3 More Detailed Architecture](#page8) are either missing, or incompletely specified in WebRTC standards. While enabling a key objective of the WebRTC specification (simplicity, by focusing purely on peer connection and user media access requirements), the lack or limited specification of these elements represents key weaknesses of WebRTC standards as a means to implement an end-to-end web-based communication service.

The following sections describe several of these weaknesses, and related opportunities for addressing those weaknesses in deployment or further standardization.

## 9.1 Codecs

With WebRTC, browsers effectively become media engines capable to encode, decode and stream media. As with all the real-time protocols, media has to be negotiated between endpoints before a session can be established. There is a clear desire within the W3C and IETF working groups to specify Mandatory to Implement (MTI) codecs for audio and video. One of the key principles outlined by some of the browser vendors is that codecs have to be royalty-free. Unfortunately there is a disagreement on what codecs should be adopted as MTI and in particular there is no expected agreement on this subject for video codecs.

### 9.1.1 Opportunity:

By defining a mechanism to gain access to the natively installed codecs on devices, and subsequently facilitate media negotiation with users of standards-based Operator services (e.g. VoLTE, RCS), we can ensure a higher quality connection (e.g. AMR-WB for voice) and better interoperability (e.g. higher probability of having compatible video codecs), resulting in improved user Quality of Experience (QoE).

## 9.2 Trust, Identity and security

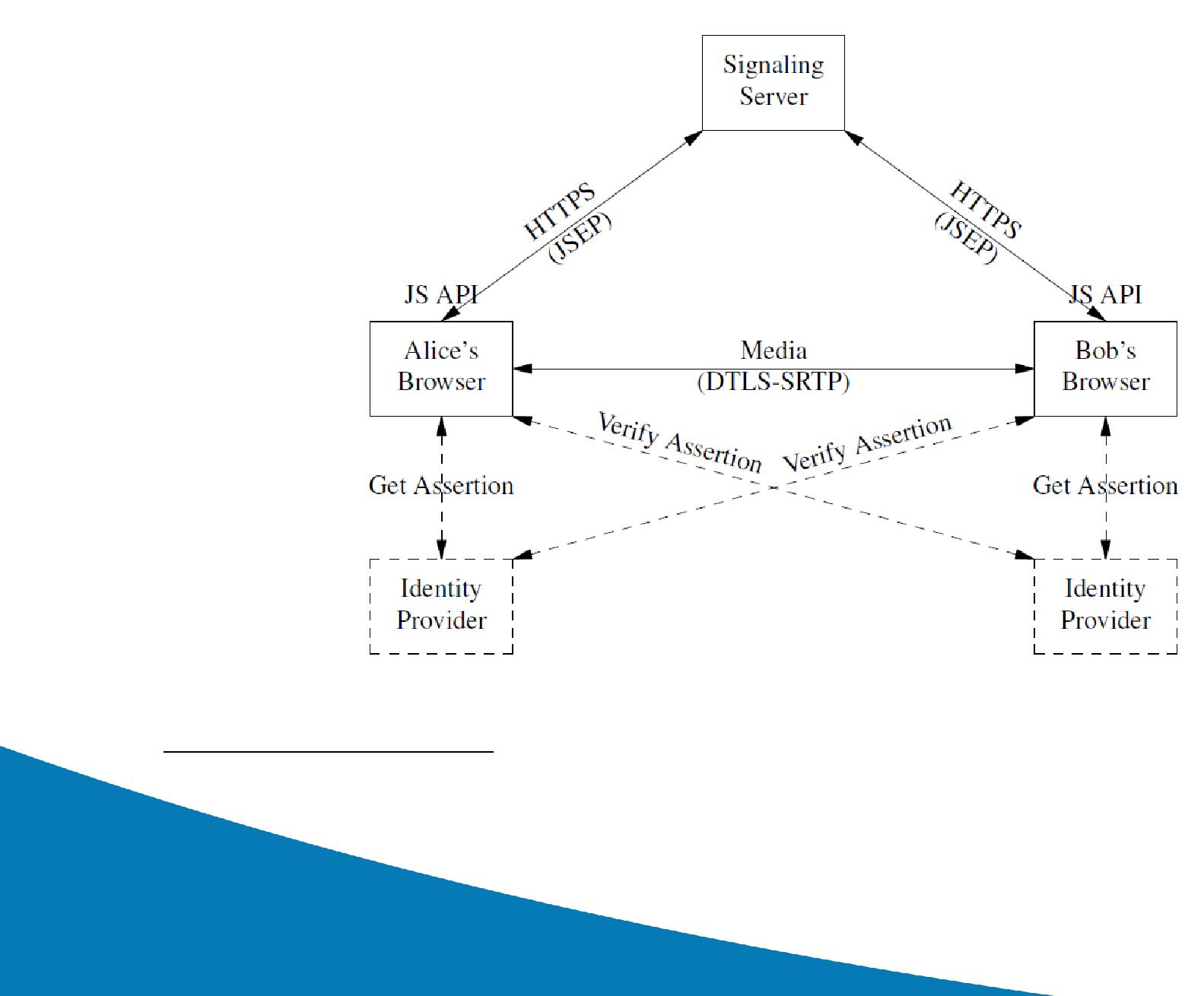
The WebRTC specification recommends a rather new approach towards media security. The IETF draft  [Security Considerations for RTC-We](http://tools.ietf.org/html/draft-ietf-rtcweb-security-04)b5 captures the nature of threats associated with the new functionality enabled in the browser through the WebRTC APIs.

Key security principles identified in the context of WebRTC are:

* Browsers are the Trusted Computing Base, i.e. provides the set of all software components that are critical to security
* Identity is checked by the browser
* An authenticated user is not necessarily trusted by the other party
* Calls should be encrypted between WebRTC clients

As shown in  [Figure 4 RTCWeb Security Architecture,](#page10) the current proposal in the IETF draft  [RTCWEB Security Architectur](http://tools.ietf.org/html/draft-ietf-rtcweb-security-arch-06)e6 is to enable a generic identity validation service provided through the browser (the browser as trust anchor):

* To allow identity assertion from a *user-selected* Identity Provider
* To support verification and possibly other identity services

***Figure 4 RTCWeb Security Architecture***

### 9.2.1 Opportunity:

Operators can establish themselves as WebRTC identity providers, thereby allowing subscribers to use their phone number as a trusted identity, at least of the subscriber to which the number is assigned.

## 9.3 Signalling API

The WebRTC specification does not mandate a standard signaling protocol for WebRTC. Instead, the API relies on an IETF draft, for the JSEP ( JavaScript Session Establishment Protocol7).

JSEP’s stated goals are to:

• Allow easy translation to common signalling protocols and architectures

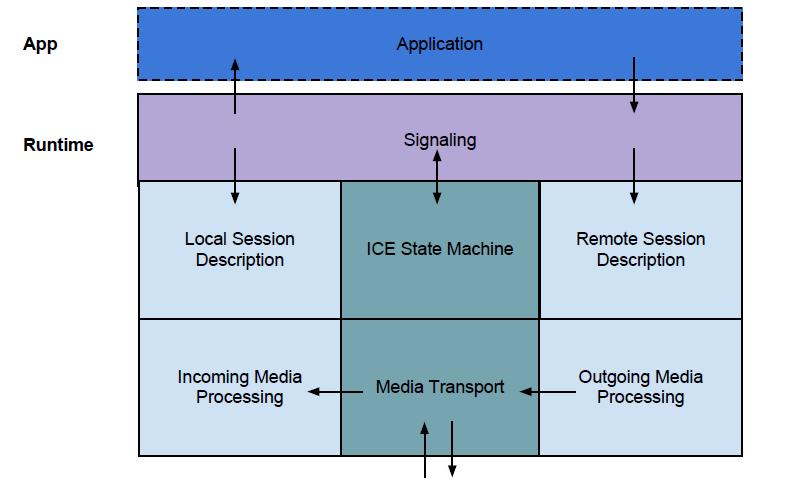
• Support early transport negotiation, using the Session Description Protocol (SDP)

• Allow local description (defining what capabilities the app intends to use for the session) to be changed by app

• Change session parameters at any time

• Allow direct manipulation of session state

Figure 5 JSEP Stack and SDP/Media Flow



JSEP describes how to set and exchange:

* Local format description
  + What I want/am going to do
* Remote format description
  + What the other side wants/is going to do
* Local/remote transport info
  + Where is media going to go, and how

Architecture principles for JSEP are:

* Signalling and transport are separated
* Signalling state moved into application code
* Media controlled via local and remote session descriptions (SDP blobs)

How the SDP blob is transported, and other call signalling needs are left to the application; only the results of the signalling matter for the APIs.

Thus while JSEP defines how essential elements of SDP and media flow work in WebRTC, it does not address how the SDP elements are obtained by apps.

In other communication service standards, (e.g. IMS/SIP), SDP is delivered via the signalling protocol (SIP). For WebRTC however, various approaches to signalling are possible, including for example RESTful APIs and SIP/WebSockets. While the diversity of these signalling protocol approaches reflects the reality of diverse vendor approaches in early WebRTC implementations, it does potentially complicate the developer experience for WebRTC-enabled apps.

### 9.3.1 Opportunity:

By abstracting signalling protocols via JavaScript libraries for developer convenience and API consistency, increased adoption by developers can occur, while enabling early implementations to support diverse signalling protocols.

## 9.4 Best effort communications

WebRTC is a web technology and expected to be used over the public internet as well as in managed networks. In order for WebRTC to be attractive for business use cases and specifically in the enterprise, there is value in being able to prioritize or guarantee the rate of real-time data transport.

### 9.4.1 Opportunity:

By defining a standard network API for Quality of Service (QoS) allowing the application to request priority for flows, prioritized or guaranteed data rates can be offered.