WebRTC for the Enterprise

FRAFOS GmbH

1. Introduction

While first designed as the interface to display information provided by web servers, web browsers are now used as the access to social networks, the interface to online games and for exchanging emails and messages as well as for streaming audio and video content. Thereby web browsers have become the main access interface to the Internet and have actually become synonymous with the Internet itself for a large portion of the Internet users.

Up until recently the communication capabilities of web applications were limited to either text-based communication such as messaging or email or non-real-time audio and video, e.g., streaming. The combination of real time services such as a voice call or a video conference with a web application was only possible using either a separate application or proprietary plug-ins that lack open specifications, and interoperability and are often limited to certain platforms.

In order to add real time capabilities to commercial browsers in a standardized manner and move from proprietary solutions, the major standardization groups responsible for the advancement of the Internet protocols and applications have launched the HTML5 and real-time web (WebRTC) initiatives to complement web applications with real time media features.

In this paper we first provide a brief introduction to the WebRTC technology and then explain it usage in the enterprise environment and how the ABC WebRTC gateway can be deployed in an enterprise environment.

2. Introduction to WebRTC

Current approaches for supporting real time communication in web applications are based on either using a separate application or a plug-in such as a Flash plug-in. Using a separate application would mean leaving the browser and launching a new application. Thereby there can be no real integration of the content presented by the browser and the real time content. Solutions based on plug-ins provide tighter integration between the real-time content and the provider's web pages. However, plug-ins such as Flash are proprietary and do not work in all environments. In particular Flash does not work over IOS used for iPhones for example. Another issue with the Flash technology is its centralized model. A Flash plug-in that was downloaded from domain X can only communicate with a server in domain X. This means that an application provider that is offering a number of applications in the form of Flash plug-ins will have to deal with all the signalling and media traffic generated by the plug-in. This restriction was introduced so as to prevent a malicious application from sending traffic to some destination and hence attacking that destination.

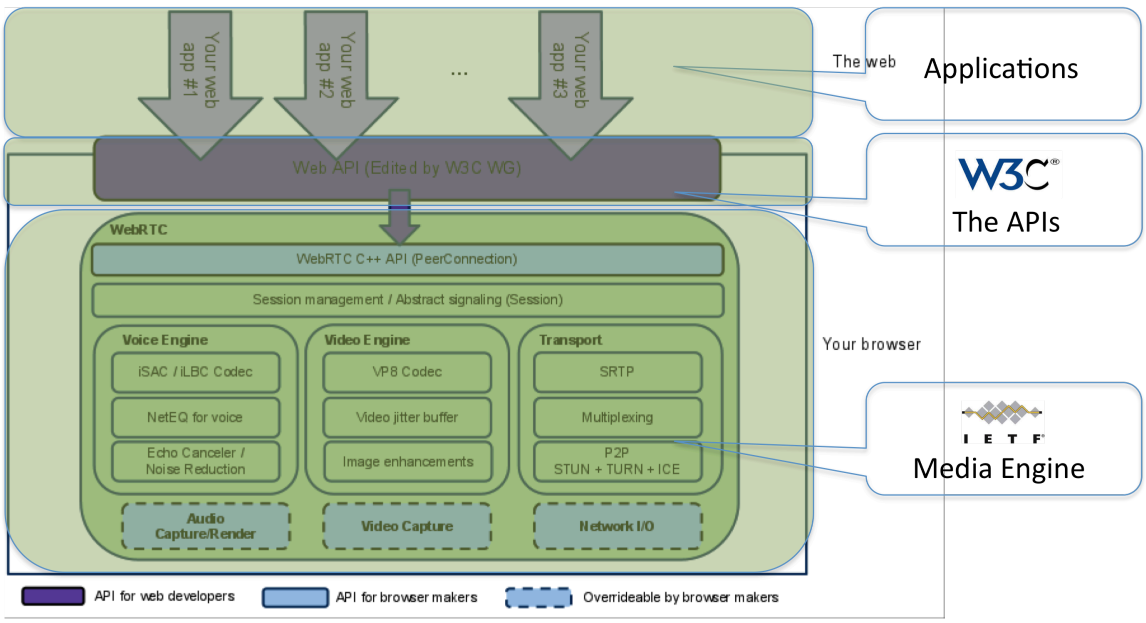

Figure 1 WebRTC framework

New working groups have been created in W3C and IETF standardization groups aiming at defining elements of real-time communication in the browser[1,2,3].

Based on the WebRTC framework proposed by the IETF and W3C the vendors of browsers are extending their browsers to support the sending and reception of audio and video. The specified WebRTC framework, see Figure 1, is based on the following main parts:

- Browser API: To provide application developers with the ability to send and receive audio and video streams directly from a browser, browsers must be enhanced with capabilities for controlling the local audio and video devices at the computing device at which the browser is running. These capabilities are exposed to application developers through a well-defined application programming interface (API).
- Web application: The typical mode of running a web application is for the user to download a Javascript from a web server. This script runs then locally at the user's system but interacts with the web server for executing the application logic. The web server can instruct the Javascript to conduct certain actions and the script can send feedback information to the web server.
- Web server: The server provides the Javascripts for the users and executes the application logic.

An application developed in Javascript would then use the browser API to capture camera and microphone data from the host computer and send it to some receiver. In order to avoid the restriction of a centralized model that is used with the Flash technology, the WebRTC framework indicates that a browser can send data to a host other than the one from which the application was downloaded if that host consents to receiving the data. This is only done, however, after receiving consent from the callee.

With such a framework a web telephony application is developed as a Javascript that is provided at a web server, see Figure 2. A user wishing to use this application downloads the

[1] Web real-time communications working group charter. W3C. Dec.2010. http://www.w3.org/2010/12/webrtc-charter.html

[2] RTC-Web IETF working charter proposal. Mar.2011, http://rtc web.alvestrand.com/ietf-activity

[3] J. Rosenberg et al. "An architectural framework for browser based real-time communications. IETF Internet draft. "work in progress", Feb.2011.

script. When making a call the Javascript then informs the web server about the call destination and the web server contacts the final destination. Once the callee has answered, the web server forwards the response of the callee to Javascript running at the caller's system. The Javascript now instructs the browser to use the local audio and video devices to exchange audio and video content with the callee.

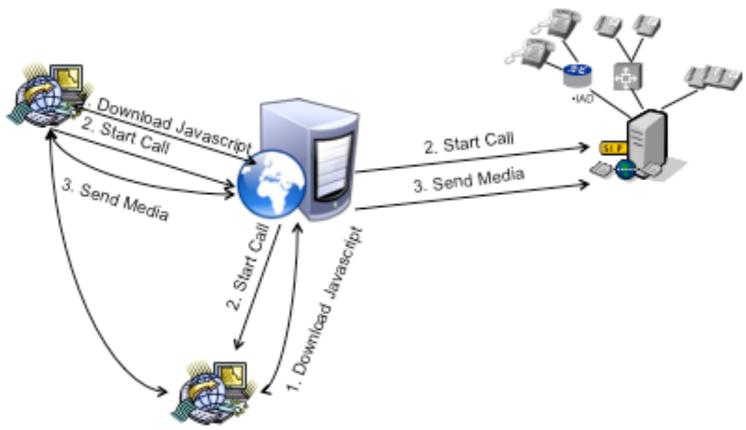

Figure 2: High level WEBRTC flow

In order to ensure that the type of applications that can benefit from the integration of real-time services with the browser is only limited by the imagination of the developers, the WebRTC framework is only defining the API to be provided by the browser as well minimal security requirements needed to avoid the misuse of WebRTC applications for initiating denial of service attacks.

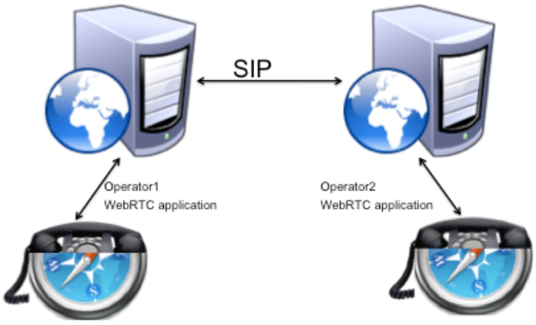

Figure 3: WebRTC Trapazoid

To enable browsers using different application providers to communicate with each other (e.g. a user logged in to Facebook wants to call someone that is logged in to linkedin) a so called RTC trapezoid , see Figure 3, can be used. In this case the two providers use a widely used VoIP signalling protocol in between such as the Session Initiation Protocol[4] to federate between them. However, each of their respective browser-based clients signals to its server using proprietary application protocols built on top of HTTP and Websockets.

[4] J. Rosenberg, et al., "SIP: Session Initiation protocol", IETF RFC 3261, June 2002

3. WebRTC in the Enterprise

Most enterprises have moved or are planning to move their telephony system and call center services to a VoIP based solution.

A telephony system in the enterprise usually serves one or more of the following scenarios:

- Onsite communication: The ability of the employees of the enterprise to call each other.
- Offsite communication: The ability of the employees in of the enterprise to call to the rest of the world.
- Remote Employee communication: Enterprises need to enable employees that either work from home, in a smaller branch or are on the road to use the enterprise telephony system for communicating to other employees or for offsite communication. This is often achieved by providing the remote employee a VoIP application that is connected to the enterprise through a VPN in order to ensure the security of the communication.
- Customer communication: Customers can call the call center of an enterprise and communicate with sales and support employees. In order to provide for a toll-free calling number, enterprises need to buy this rather expensive service from a telephony service provider.

By deploying WebRTC in the enterprise environment, enterprises will not only reduce the costs of their telephony services but also easily introduce video and chat in a simple manner:

- Cost savings:
  - By offering WebRTC services instead of a toll-free number for their call center and support services, enterprises can significantly reduce the costs paid for the toll-free services.
  - Remote workers can connect to the enterprise telephony service over any IP connection without having to provide the remote worker with any special devices, VPN or VoIP software application

- Ease of use: Services such as conferencing, messaging and connection to calendar and address books are just some of the common features offered by modern VoIP telephony solutions for the enterprise. These services are usually not available for remote employees and customers connecting to the enterprise through a telephone. WebRTC removes this barrier and can significantly contribute towards a more customer friendly communication.
    - Rich communication: Telephony services are restricted to voice communication. WebRTC applications integrate seamlessly video and messaging. This can be of benefit for both remote workers and customers.
    - Uninterrupted communication: A customer looking for a service often queries the enterprise description and contact information through a browser. Then the customer will have to leave the browser and use the phone to call the enterprise. With WebRTC the customer remains on the enterprises' web page and can discuss any products or issues she might be having without having to deal with both the phone and browser.
- Security: All communication between WebRTC applications is encrypted. While current VoIP solutions offer the possibility of using encryption this is not often used. Even if used then in the enterprise scenario, security is often only available for remote users. Customers contacting the enterprise can only rely on the security mechanisms provided by the public telephony system, which is poor at best. WebRTC technology is actually the first communication system that already integrates state of the art encryption standards as an intrinsic part of the system and not as an add-on.

4. Deploying the ABC WebRTC Gateway in the Enterprise

Remote workers are usually connected to the enterprise telephony system over a VPN or similar. Customers call the enterprise either over a VoIP call or directly over a PSTN connection.

Usually, in order to protect the telephony system of the enterprise, a session border controller (SBC) is located on the border of the enterprise.

Figure 4 presents a high level overview of the migration path from a SIP/PSTN based solution to one that incorporates WebRTC as well.

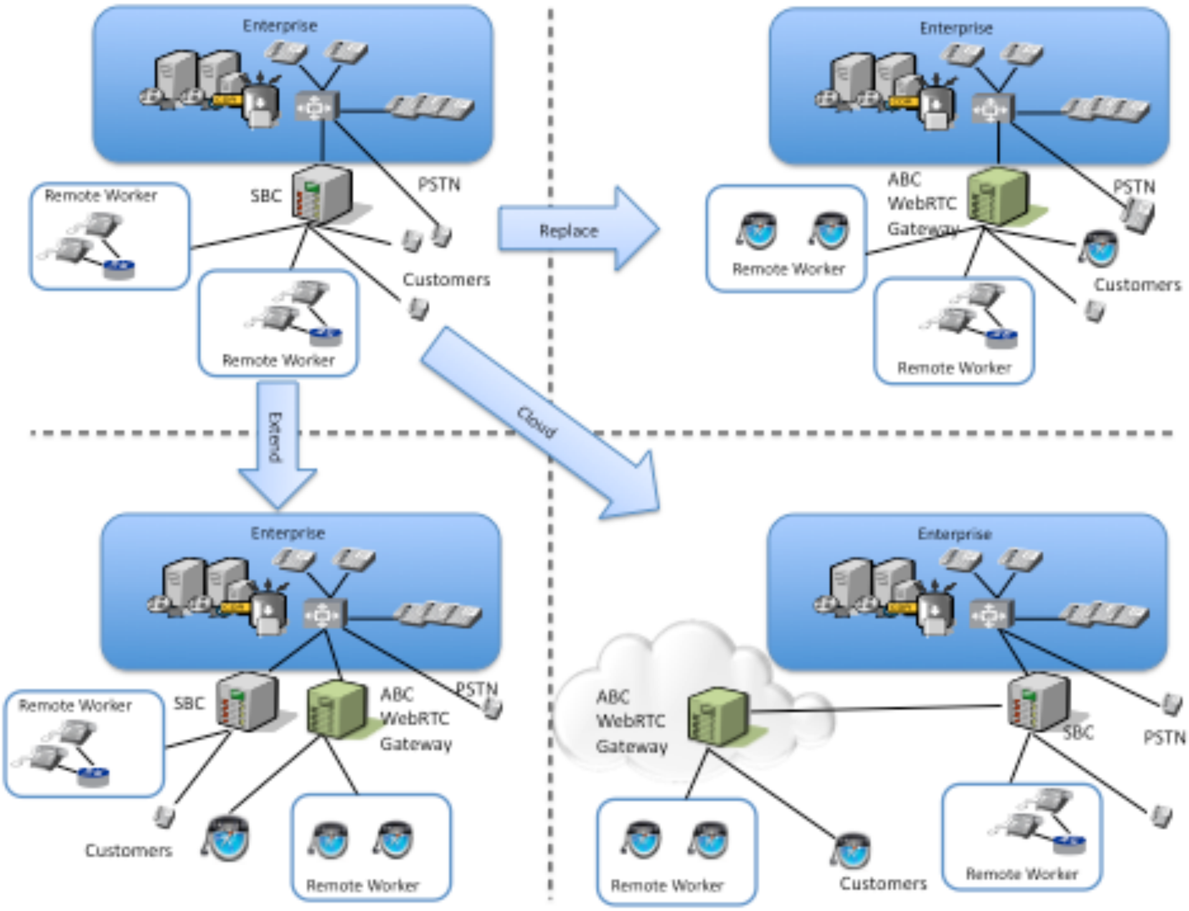

Figure 4: Migration to WebRTC

As a starting point we assume an enterprise structure of a very general nature consisting of the VoIP solution used at the enterprise, namely a PBX, call center or something similar. Remote workers and small branches of the enterprise are often connected to the central enterprise infrastructure through an SBC. The same or another SBC also separate the enterprise infrastructure from the public Internet. Calls to public numbers or from customers are routed through the SBC. Direct PSTN connections are usually handled by the infrastructure directly.

One can in general distinguish three migration paths:

- **Replacement:** In this case the SBC used by the enterprise is replaced by the ABC platform. The ABC solution provides SBC as well as WebRTC gateway functionality. The ABC platform would in this case process both SIP and WebRTC services.
- **Extension:** In addition to the used SBC, the enterprise would deploy the ABC WebRTC gateway in parallel. VoIP calls to and from remote workers and customers would be processed by the already deployed SBC. Calls using WebRTC would be handled by the ABC WebRTC gateway, which would then forward them to the enterprise VoIP servers. The ABC WebRTC gateway can be installed on a dedicated device or run as a virtual machine on an already available hardware.
- **Cloud:** This is probably the least intrusive migration path. Instead of deploying the WebRTC gateway directly as part of the enterprise infrastructure, the ABC WebRTC gateway is deployed on a cloud service. WebRTC calls to the enterprise are routed to the WebRTC gateway in the cloud. The ABC WebRTC gateway would then route the translated calls to the enterprise. From the point of view of the enterprise, the ABC WebRTC gateway would be similar to a remote worker or a public caller. So unless some additional security mechanisms are introduced such as VPN the enterprise might want to route the calls from the ABC WebRTC gateway through the already deployed SBC.

5. Technical Specifications

Supported Platforms Linux	**High Availability** Active/Hot Standby redundancy model
WebRTC Features Javascript SIP over WebSocket NAT traversal using ICE, TURN, STUN JsSIP support	**QoS Control** Bandwidth limitation and management Call admission control per peering partner/trunk
Media Services Routing audio codec including G.711 and OPUS. Routing of video codec including VP8 Dynamic jitter control NAT/NAPT on media RTP inactivity monitoring Codec filtering	**Call Routing** Call blocking and filtering Embedded routing engine Load balancing Peer monitoring and availability detection Alternative routing on failure Table based routing for LCA
Media Applications Call recording Announcement services Software based transcoding (G711u/a, G726, OPUS, iLBC, L16, G722, Speex; on request: G729a, G729a/b, AMR)	**SIP** Registration pass-through Registration caching and offload SIP header manipulation SIP Back2Back UA
Management Capabilities GUI based configuration and monitoring Secure embedded web-based GUI SSH access SNMP V2 status and logs Local logging of alarms, events and statistics REST and XML RPC based open interfaces	**Protocol Support** UDP, TCP WebSocket Translation between transport protocols Per source/destination transport layer mediation SNMP, NTP, SSHDNS RTP, RTCP, SRTP TLS, DTLS, SDES
Virtualization Amazon cloud Virtualization software OVM, KVM ..	**Hardware** Hardware independent

6. About FRAFOS

FRAFOS GmbH is a manufacturer of VoIP solutions with offices in Berlin and Prague. FRAFOS was incorporated as privately held company in May 2010, in Berlin, Germany.

The history of FRAFOS team and technology goes back to the late nineties. As researchers at the prestigious German public R&D institute Fraunhofer FOKUS, the FRAFOS founders were the among the first to work the SIP and RTP standards and to develop open source solutions that paved the way for the VoIP revolution.

FRAFOS offers SIP and WebRTC session management and security solutions of the latest generation that come either as a standalone solution or as a cloud ready implementation. The flagship product of FRAFOS, the ABC platform, offers open interfaces and built in multimedia applications such as recording and announcements. The ABC solution enables service providers to simplify their service infrastructure and prepares them for future challenges.