Interactive and Truly Toll Free Customer Communication

FRAFOS GmbH

FRAFOS GmbH | Windscheidstr. 18 Ahoi 10627 Berlin Germany | info@frafos.com | www.frafos.com

1. The New Shopping Experience

Juliana has just received her brand new Jeans from her favourite online retail shop. After opening the package she is disappointed that the jeans she has long waited for have a defect. She logs into her account and clicks the “Call Me” button, which connects her to the call centre of the online retail shop. Already as part of the call establishment, the information about Juliana and her latest orders are transferred to the call centre and the lady behind the phone has a good idea about who is calling and what might be the matter even before Juliana starts complaining about the defect. In order to have a better view of the issue the call centre worker starts a video call with Juliana and includes the Customer Complaints Manager into the call. After determining that the jeans were indeed defect the call centre worker promises Juliana a refund and starts the process.

2. Reduce Cost and Increase Customer Satisfaction

Current technologies for interactive communication with customers are currently mainly limited to phone calls or chat.

Toll free calling numbers while being free for the customer are expensive for the online retail shop. Further, having to navigate between the web site and the phone is inconvenient for the user.

Interactive chat based communication is more cost effective but less customer friendly. While such services are often combined with a callback service, this callback service usually requires the customer to reveal her phone number to the online retail shop which is not appreciated by all customers due to privacy reasons and incurs costs on the online retail shop.

FRAFOS is offering online retail shops a more interactive and cost effective customer communication channel. Based on state of the art WebRTC technology, the FRAFOS platform integrates “Call Me” button in the web pages of online shops. Customers wishing to contact the online shop can from the web page directly communicate with the shop’s call centre without using a phone or specialized plugins. This has the following advantages for the online shop:

- **Reduced costs:** Toll free numbers are toll free only for the customers and not the online shop. Actually, toll free minutes are more expensive than normal calls. By establishing the communication over the Internet, customer calls become free for both shop and customers.

- **Improved user experience:** A customer wishing to communicate with an online shop does not have to take out his phone, type in the call centre number and navigate between the browser and the phone. By integrating the telephony service into the browser, the customer can remain on the web page and can discuss any products or issues she might be having without having to deal with both the phone and browser.
- **Faster customer processing time:** Already as part of the call establishment, information about the web page the customer is looking at and the customer profile can be sent to the call centre worker. Thereby, the call centre worker can get information about the possible issue and the caller much faster and hence reduce the processing time that is usually spent asking the customer to indicate her name, order identity and other information needed to identify the issue.
- **Innovative customer interaction channels:** Beside audio and text as the primary customer interaction channels, deploying the FRAFOS platform enables online shops to consider more innovative channels like video chat or conference calls. Inviting a specialist to the customer call in order to have a more detailed answer or starting a video conference call during which the customer can show what is wrong or being taught how to solve a problem can be part of the call centre solution.

3. FRAFOS Interactive Customer Communication

The FRAFOS solution is based on WebRTC technology. WebRTC integrates real time communication, e. g., audio and video, into the browser without the need for specialized plugins.

The FRAFOS "Call Me" button is a customizable application that is easily integrated into any web site. The online shop can determine the look of the button as well as the destination to be called.

By clicking the "Call Me" button, the customer can start audio and video calls to the online shop call centre. After establishing a WebRTC call with the FRAFOS WebRTC gateway, the FRAFOS WebRTC gateway will then establish a VoIP call to the shop's call centre using the widely used Session Initiation Protocol (SIP). Beside establishing a call, information about the customer and the web page from which the call was initiated is transmitted. This information can then be provided to the shop's call centre using open interfaces such as XML RPC.

Figure 1 Web based customer interaction

The FRAFOS solution provides the following advantages:

- **Non-disruptive integration:** Even though the solution is based on WebRTC, the call centre itself does not have to support WebRTC. The FRAFOS WebRTC gateway translates between WebRTC and SIP, which is the VoIP technology most likely supported by the used call centre. Hence, no need for updating the call centre.
- **Customizable:** FRAFOS customers have the freedom to customize the look and feel of the "Call Me" button so as to smoothly fit into the shop's design.

- **Cloud ready:** The FRAFOS components are provided as a software solution that can be used on dedicated hardware, installed in a hosted system or run over a cloud technology such as Amazon or openstack.
- **Extendable:** The FRAFOS solution already supports audio conferencing and will soon support video and chat communication. These features are controlled through open interfaces. So, once the online shop decides to extend its customer communication channels with conferencing or video then all that would be needed is to use the proper interfaces and no changes to the infrastructure would be needed.
- **Scalable:** The FRAFOS solution can scale from a few calls up to thousands of concurrent calls using the same software. This enables our customers to grow without the worry of having to start with an over dimensioned system.

4. 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.

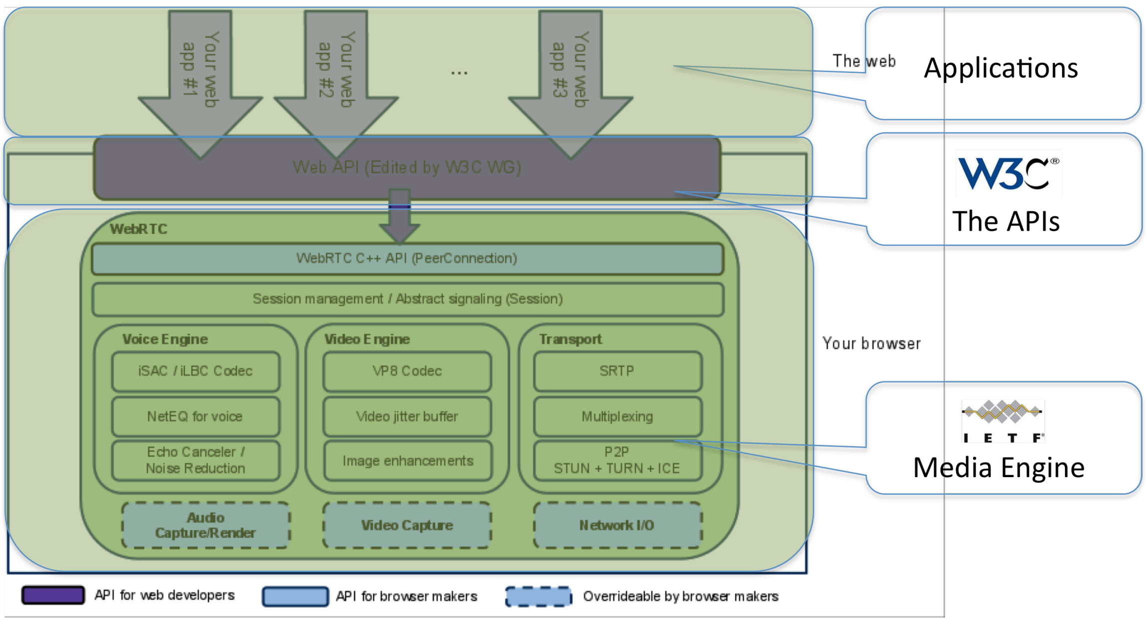

Figure 2 WebRTC framework

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 2, 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

---

[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.

provided at a web server, see Figure 3. A user wishing to use this application downloads the 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.

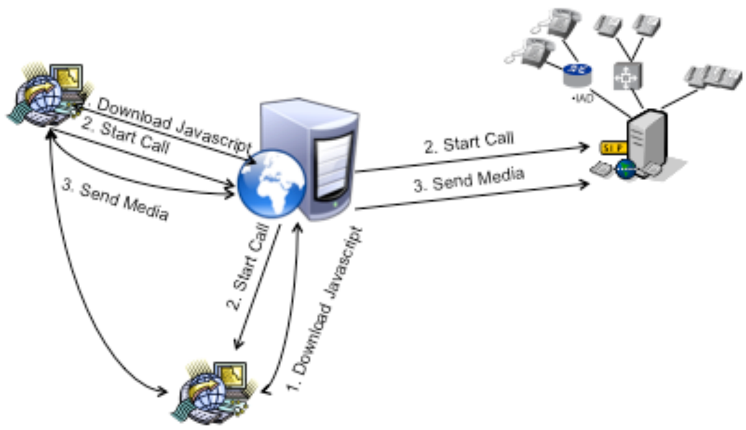

Figure 3: High level WEBRTC flow

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.

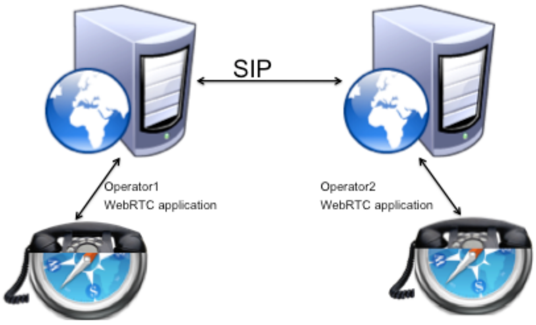

Figure 4: WebRTC Trapazoid

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 4, 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

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 and 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 is providing Session Border Controllers and WebRTC servers, deployable on virtual machines, hardware and in the clouds. The technology provides for safe and robust interconnection between SIP devices, PBXs, PSTN gateways and WebRTC. The product line is a serial industry award winner and has collected the 2015 WebRTC Product of the Year, 2014 Internet Telephony Product of the Year and 2014 Red Herring Winner awards.