Images Source: Google Images

KAMAILIO WORLD
CONFERENCE & EXHIBITION
BERLIN - GERMANY, APRIL 16-17, 2103

# WebRTC and VoIP: bridging the gap

victor.pascual.avila@gmail.com @victorpascual http://es.linkedin.com/in/victorpascualavila

# Intro

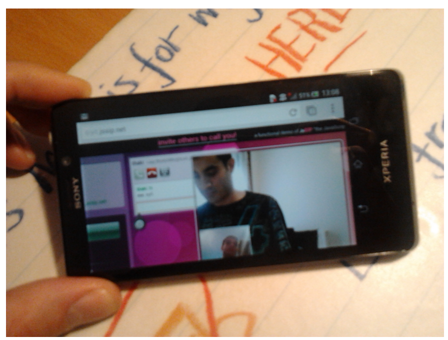
- What is WebRTC (Real Time Communications)?
  - A browser-embedded media engine (still being finalized)
  - Emerging and standard method of web-based RTC (W3C/IETF/3GPP)
  - Another type of access framework
- Why the hype?
  - Web: most dynamic, innovative place on planet
  - RTC has largely been absent
  - WebRTC delivers RTC to those that create the Web
- WebRTC is posed to grow rapidly and WebRTC will be an important access method in the future for Service Providers, contact centers, and enterprises
  - Number of developers: 1000s of SIP programmers vs. 100,000 of JavaScript programmers
  - Deployment: IP phones must be physically deployed or software installed (software updates may not be automated) vs. Browsers automatically updated to add support; client updates automatically pushed to browser
  - Several dozens of vendors with unique SIP implementations vs. Handful of browser vendors with unique WebRTC implementations Opensource code available
- Moving really fast, lots of early implementations, customer trials… BUT still a number of open issues
  - MTI video codec discussion, SDP O/A, strict-firewall/HTTP-middlebox traversal, Microsoft's CU-RTC-Web, Apple, etc

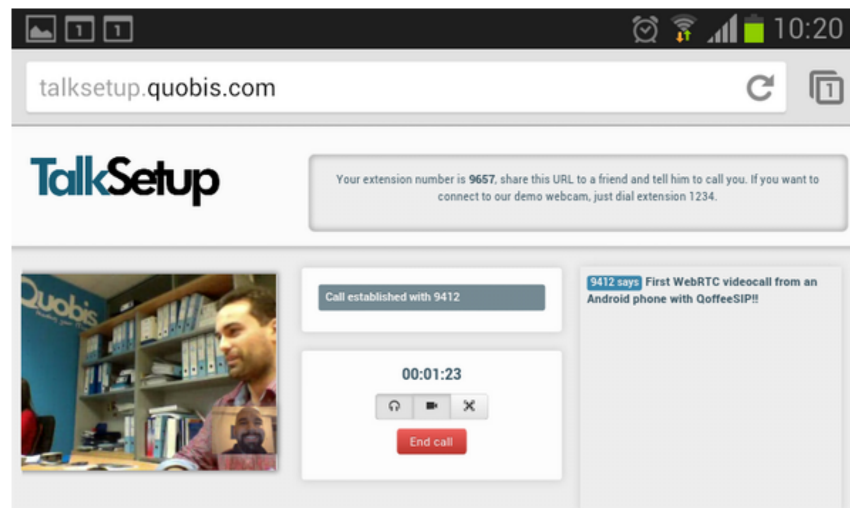
# Use Cases

- WebRTC enables innovative use cases on the Web
  - WebRTC It's not meant to be the new Web Telephony
- But interworking towards existing legacy is required
  - Extend existing SIP environments into the exciting new web domain
- Service Provider subscriber access via WebRTC methods
  - Browser-based RTC to complement SIP offerings or simply extend existing SIP-services over web
    - Voice service extension with web-phone
    - New Telco-OTT services (over-the-top)
    - RCS-e (rich communications services – enhanced)
    - Conferencing
    - Web-based comms provider PSTN break-out
- Enterprise UC without thick or thin client soft phones
  - Easier to maintain & break single UC vendor lock
- Contact centers embedding RTC into customer service web pages
  - Customer satisfaction & lower costs
- Not only Web browsers (e.g. Chrome, Firefox) but also native support via apps or OS (e.g. set-top boxes, FirefoxOS)

# Signaling Plane

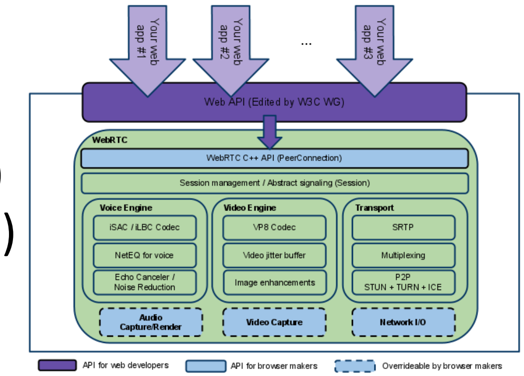
- WebRTC has no defined signaling method – Don't panic, it's not a bad thing!
  - JavaScript app downloaded from web server. Popular choices are:
- SIP over Websockets
  - Standard mechanism (draft-ietf-sipcore-sip-websocket) – soon to be RFC
  - Extend SIP directly into the browser by embedding a SIP stack directly into the webpage – typically based on JavaScript
  - WebSocket create a full-duplex channel right from the web browser
  - Popular examples are jsSIP, sip-js, QoffeeSIP, or sipML5






- Call Control API
  - propietary signaling scheme based on more traditional web tools and techniques
  - GSMA/OMA extending RCS-e "standard" API to include WebRTC support
- Other alternatives based on XMPP, JSON or foobar

# Media Plane




- A browser-embedded media engine
  - Audio codecs – G.711, Opus are MTI
  - Video codecs – H.264 vs. VP8 (MTI TBD - IPR discussion)
  - Media codecs are negotiated with SDP (for now at least)
  - Best-of-breed echo canceler
  - Video jitter buffer, image enhancer
  - Requires Secure RTP (SRTP) – DTLS
  - Requires Peer-2-peer NAT traversal tools (STUN, TURN, ICE) – trickle ICE
  - Multiplexing: RTPs & RTP+RTCP

- Yes, your favorite SIP client implementation is compatible with most of this. But, the vast majority of deployments
  - Use plain RTP
  - Do not support STUN/TURN/ICE
  - Do not support multiplexing (ok, not really an issue)
  - Use different codecs that might not be supported on the WebRTC side

# Option 1: Just ignore the gap


And expect  all deployed equipment to be upgraded soon

# Option 2: Try to fix it

When/While necessary (hopefully with the right solution!)

OK, might need to do a couple of fixes to make things work, but…

# HOW?

AS SEEN ON TV

# THE UNMENTIONABLE


# THE GATEWAY

evil inside

# WebRTC Gateway Taxonomy

(great blog post by Chad Hart)

- SIP-over-WebSockets gateway
  - act as a SIP-server and terminate the WebSocket from the browser and convert that to UDP/TCP transport into the SIP network
- Web API to SIP gateway
  - act as a web-server and convert the API calls to SIP
- Media controller
  - ICE/STUN/TURN
  - SRTP/RTP interworking
  - Mux/demux
- Transcoding gateway
  - convert from one codec to another (incl. video!)
- Flash RTMP gateway
  - majority of browsers on the web today support flash and not WebRTC

Several implementation options:

- one or more of the options above
- WebRTC as a feature/product/service
- Other features (?): identity, charging, session recording, LI, middlebox-traversal, etc.

Thank you!