Overview of WebRTC

Faisal Chaudhry

fchaudhr@cisco.com

Summary

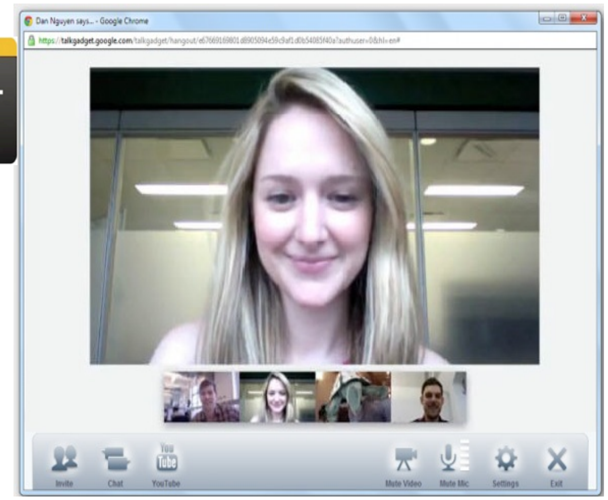
Overview of WebRTC

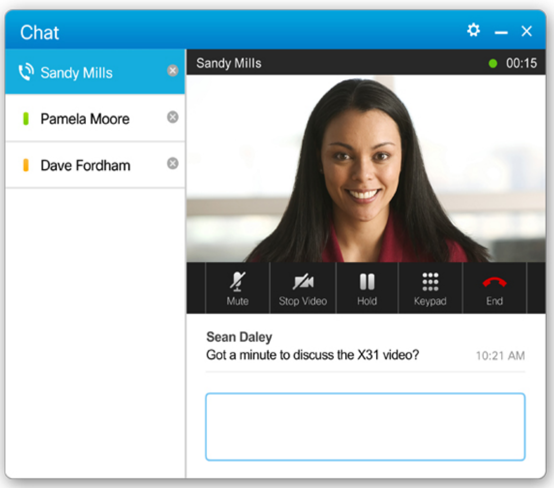
WebRTC – how does it work?

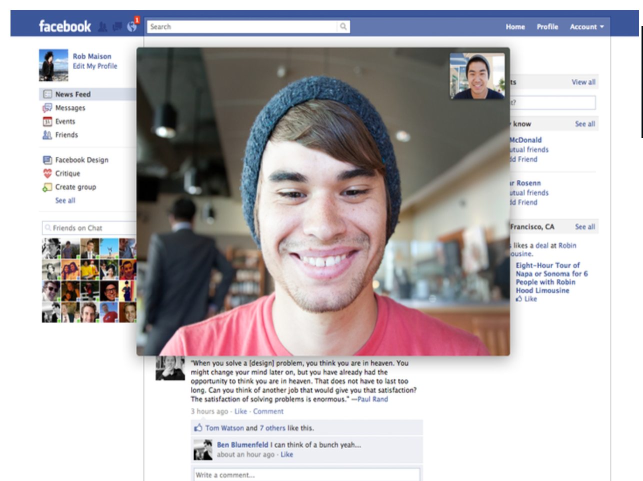
Standards activities

Vendor strategy example

Real Time Communications in Browsers today

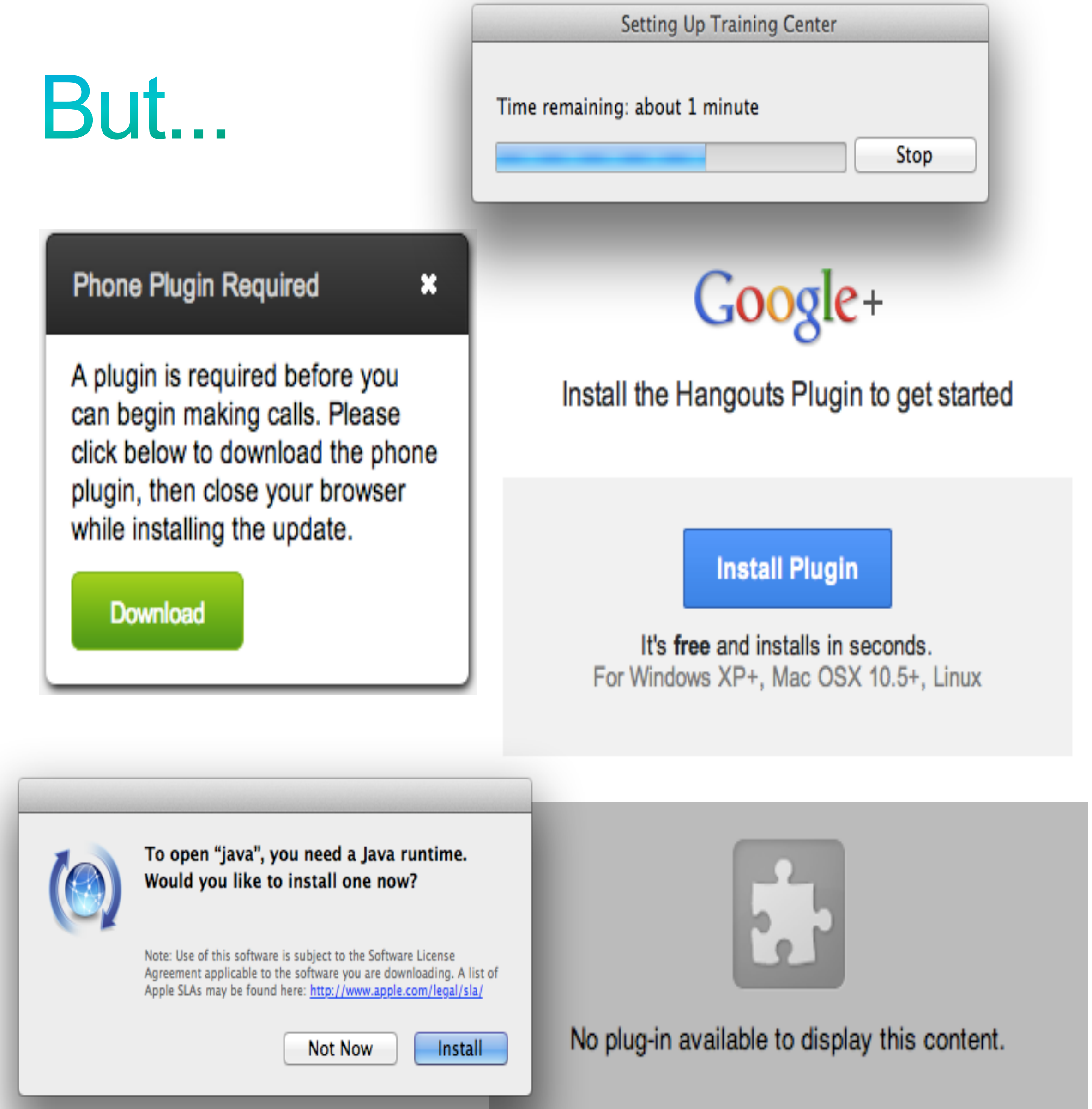






3

But...



- Requires 3rd party plugins
- Proprietary – no interoperability
- Difficult to deploy (permissions, etc...)
- Not available on all platforms

WebRTC to the rescue

- Ability to do real time communications within browsers
  - Browser becomes the softphone with real time data sharing capabilities
  - Real-time = voice, video & data sharing (IM & Presence is not part of WebRTC)
  - No plug-ins involved
- Which standard bodies are involved?
  - IETF & W3C
- IETF Work Group - RTCWeb
  - Defining how browsers will communicate with each other
  - Attempt to largely re-use existing protocols
- World Wide Web Consortium (W3C)
  - Define how Web applications access browser real-time communications i.e. APIs

What's Industry saying about WebRTC?

Hype Justified? flexibility, ease-of-use, pace of evolution … use-cases evaluation

Google, Mozilla Team Up for Skype-Killing Video Call Demo

Mozilla: WebRTC is the Real Future of Communications

WEBRTC: A NEW GAME-CHANGER, DISRUPTING TELCOS AND OTTS

nojitter
Insight for the Connected Enterprise

POSTS WEBINARS WHITEPAPERS VIDEOS SLI

Eric Krapf | March 18, 2013 | 4 Comments

Like 11 Tweet 41 +1 2 share

Enterprise Connect Day One: WebRTC is Hot

Overflow crowds came to hear about how WebRTC will change the enterprise.

ABOUT THE AUTHOR

Eric Krapf is the Program Co-Chair of the Enterprise Connect events, helping to set program content and direction for the...

Read Full Bio >>

SHARE

Enterprise CONNECT
Communications Transforming Business

The Premier Enterprise Conference & Expo
CONFERENCE: March 17 - 20, 2014 | EXPO: March 17-19 | LOCATION: Gaylord Palms, Orlando, FL

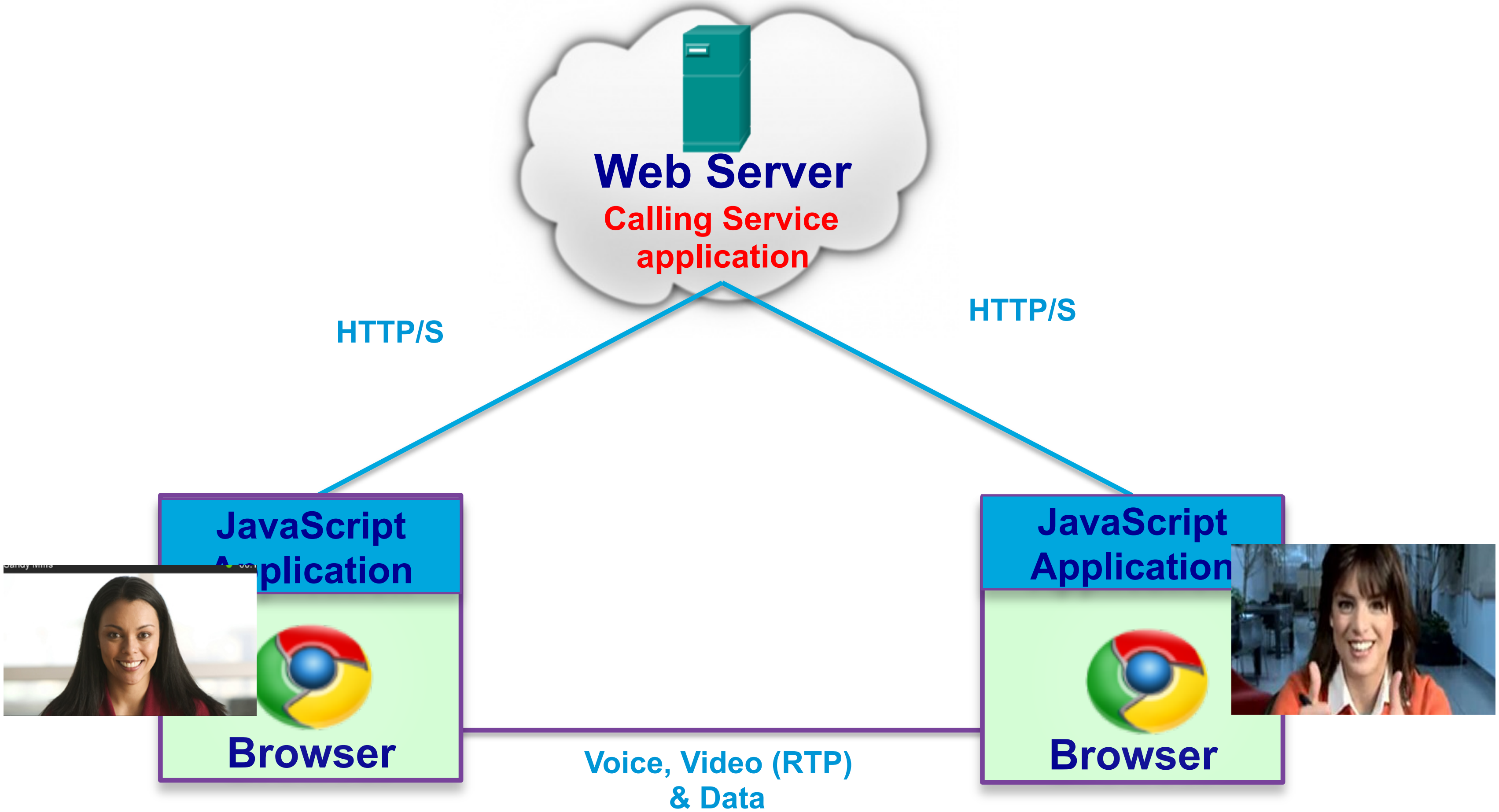
HOME REGISTER ABOUT CONFERENCE EXHIBITION RESOURCES NO JITTER BLOGS

WebRTC: What's the Impact on Your Enterprise?
A Conference-Within-A-Conference at Enterprise Connect 2013

Reasons:
Open standard,
Speed of design & deploy

Overview of WebRTC

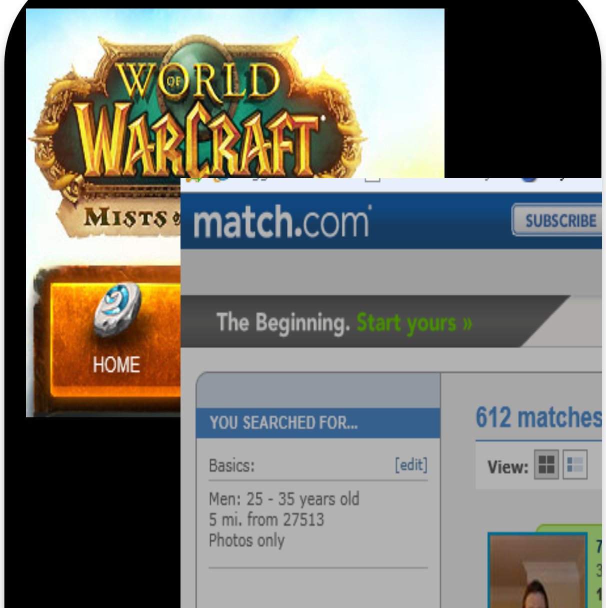
An architecture to provide real-time communications (Voice, Video & Data) within browsers



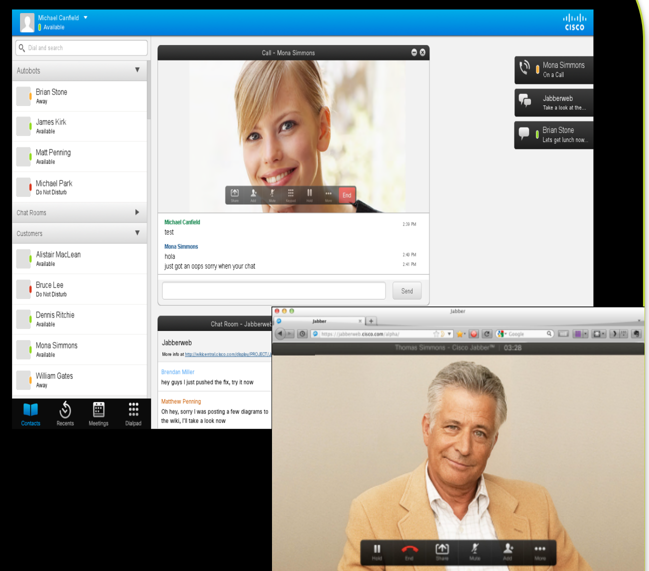
7

WebRTC Use-case

Few examples



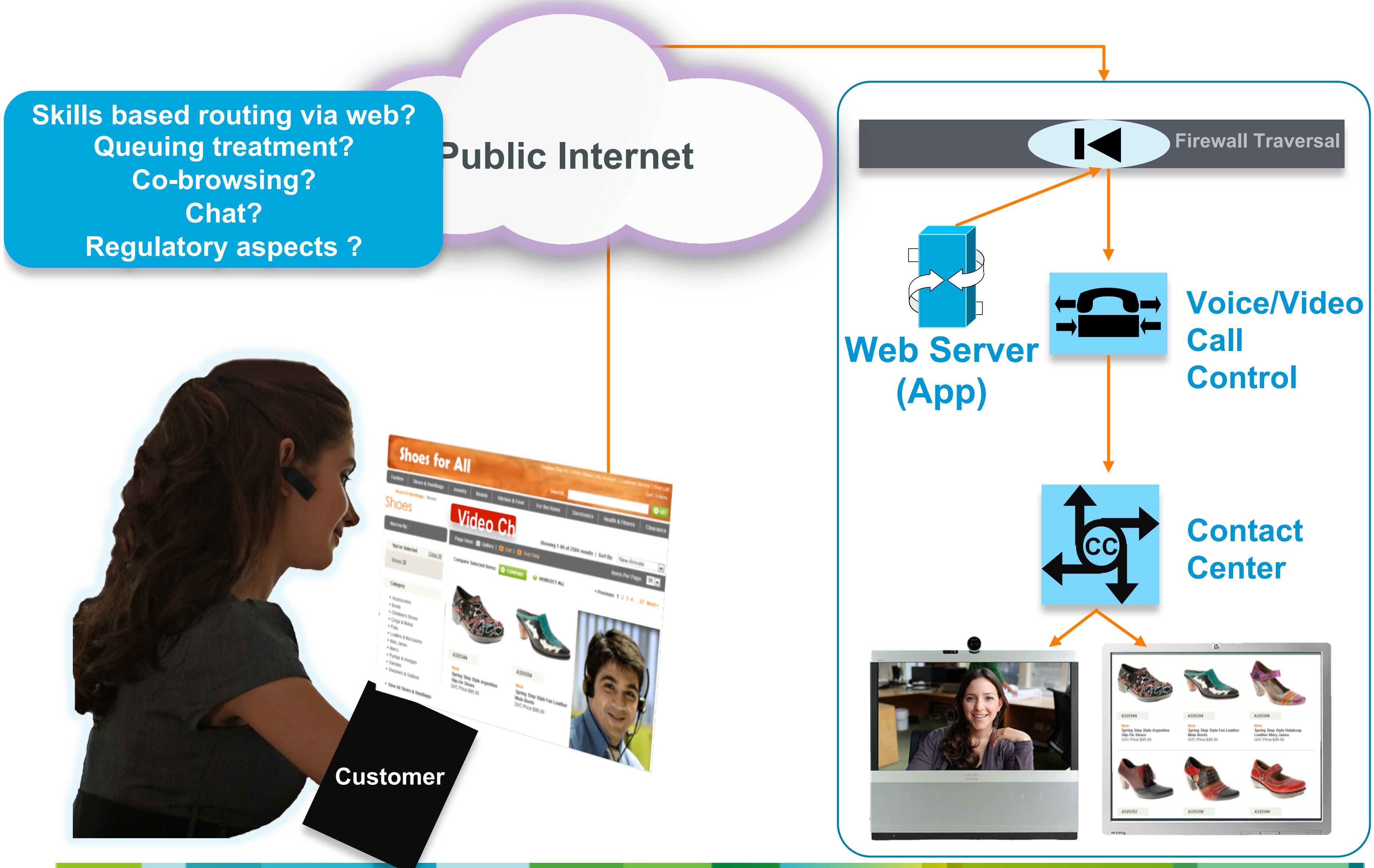
Video Enabled Games & Social Sites



Consumer to Business (B2C) Video Enabled Contact Centers

Video Enabled Browsers interacting with anyone (B2B, Distant Learning, Healthcare …)

Consumer to Video Expert – Architecture



Summary

Overview of WebRTC

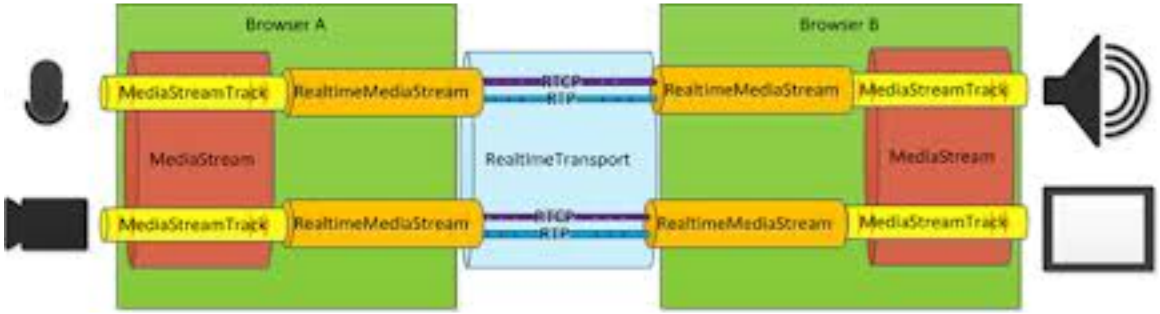
WebRTC – how does it work?

Standards activities

Vendor strategy example

Challenges for RTC in browsers

Softphone engine, Codecs, Signaling, P2P



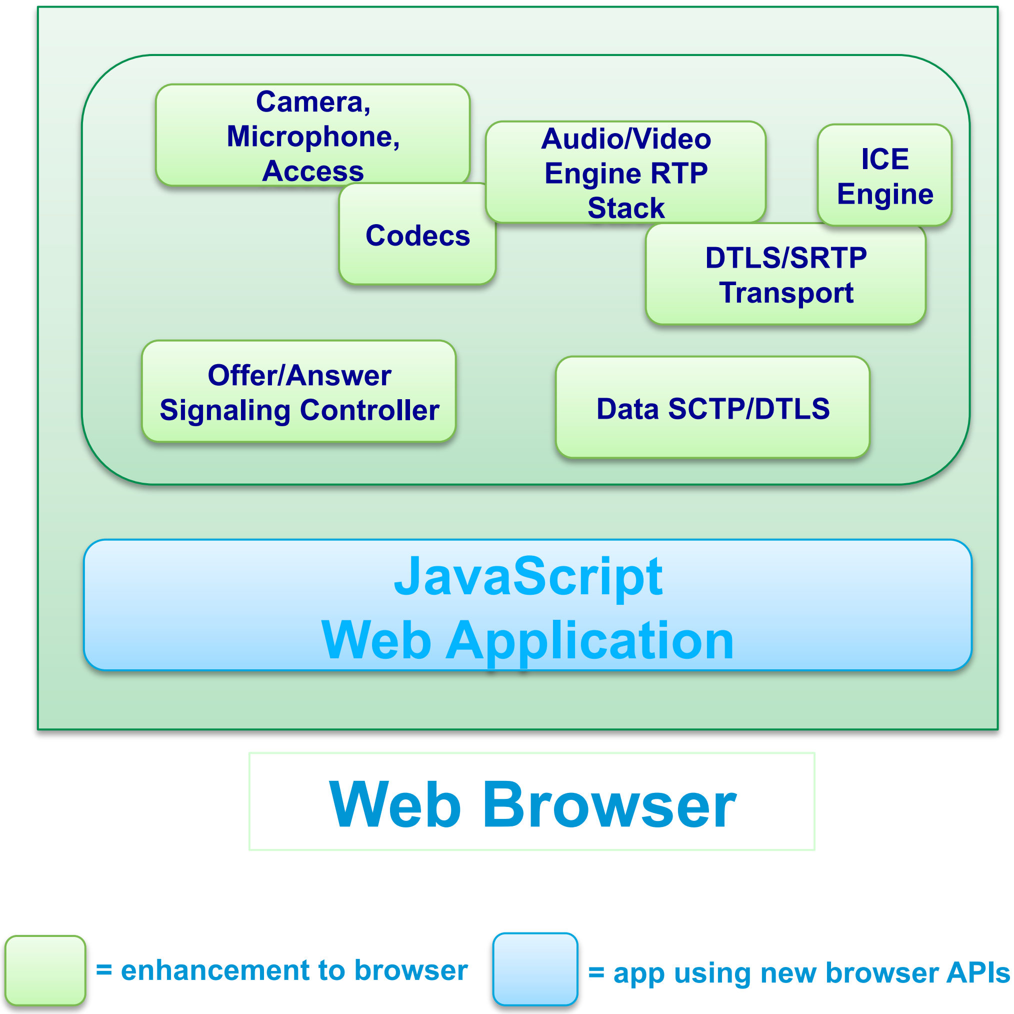
Media traversal through firewalls, NAT

Securing browsers and media

Validating identity of person calling

Optimizing real-time traffic performance

WebRTC – Browser Architecture

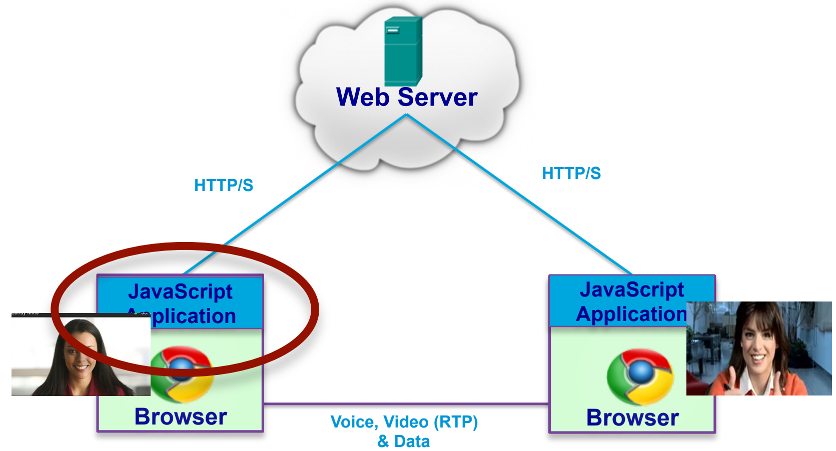


- Access to mic, camera
- Audio & video codecs
- RTP/sRTP stack – Lip sync, packet loss detection, DTLS media keying
- SCTP/DTLS for secure data (real-time data)
- ICE for NAT/firewall traversal
- SDP-based offer/answer negotiation (e.g. ROAP)

12

WebRTC APIs

Three main tasks

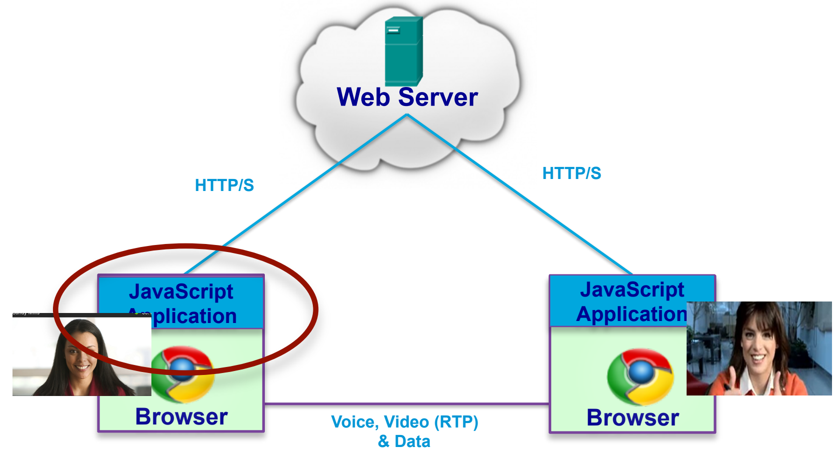


1. Acquire audio and video

2. Communicate audio & video to your peer

3. Communicate data to your peer

WebRTC APIs



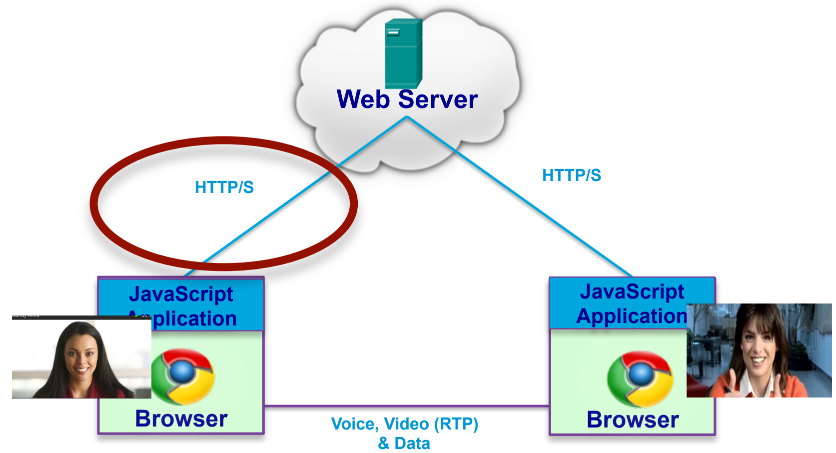
Three main JavaScript APIs

1. Acquire audio and video
   - MediaStream (aka getUserMedia)
   - (JS app to request access to micrphone and camera)
2. Communicate audio & video to your peer
   - RTCPeerConnection
   - (signal Processing, codec handling, P2P communication, Security …)
3. Communicate data to your peer
   - RTCDataChannel

Signaling mechanism

How do you share the details with your peer?

Signaling mechanism



- Need to exchange 'session description objects'

  What I want to send,
  What formats are supported

- Uses SIP Offer & Answer mechanism for setup & negotiation of media between browser & another device (just like SIP)

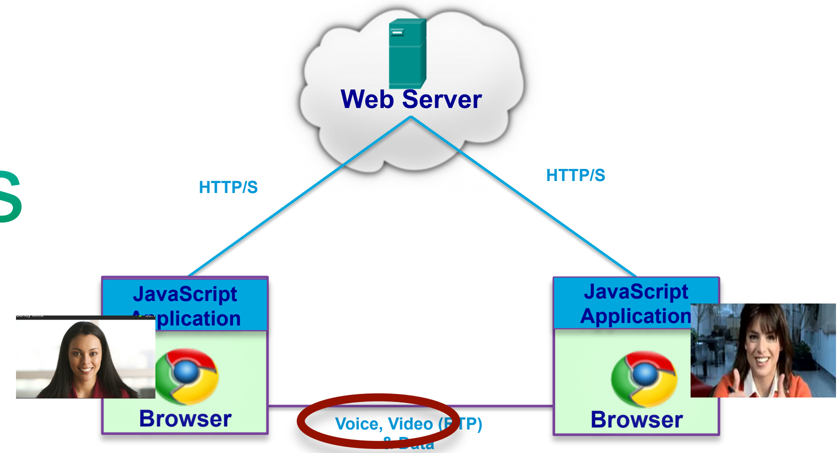
- Uses IETF (RFC 2327) standard Session Description Protocol (SDP) for capabilities:

- Audio
- UDP port # 49172
- Codecs supported: G.711

- Video
- UDP port # 51172
- Codecs supported: H.261, H.263

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.com
c=IN IP4 10.1.3.33
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 51172 RTP/AVP 31 34
a=rtpmap:31 H.261/90000
a=rtpmap:34 H.263/90000
```

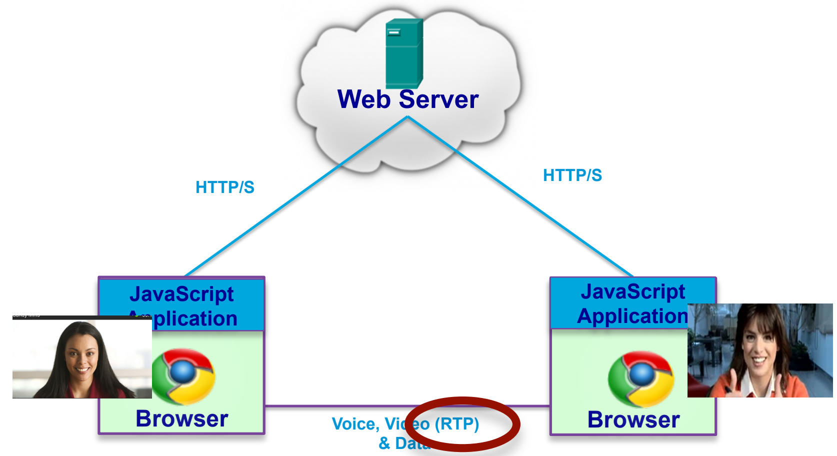
Media – Required Codecs



Mandatory to Implement

- Narrowband Audio: G.711
- Wideband Audio: Opus
- Video: unknown at this time – H.264 vs. VP8

Media Transport – SRTP



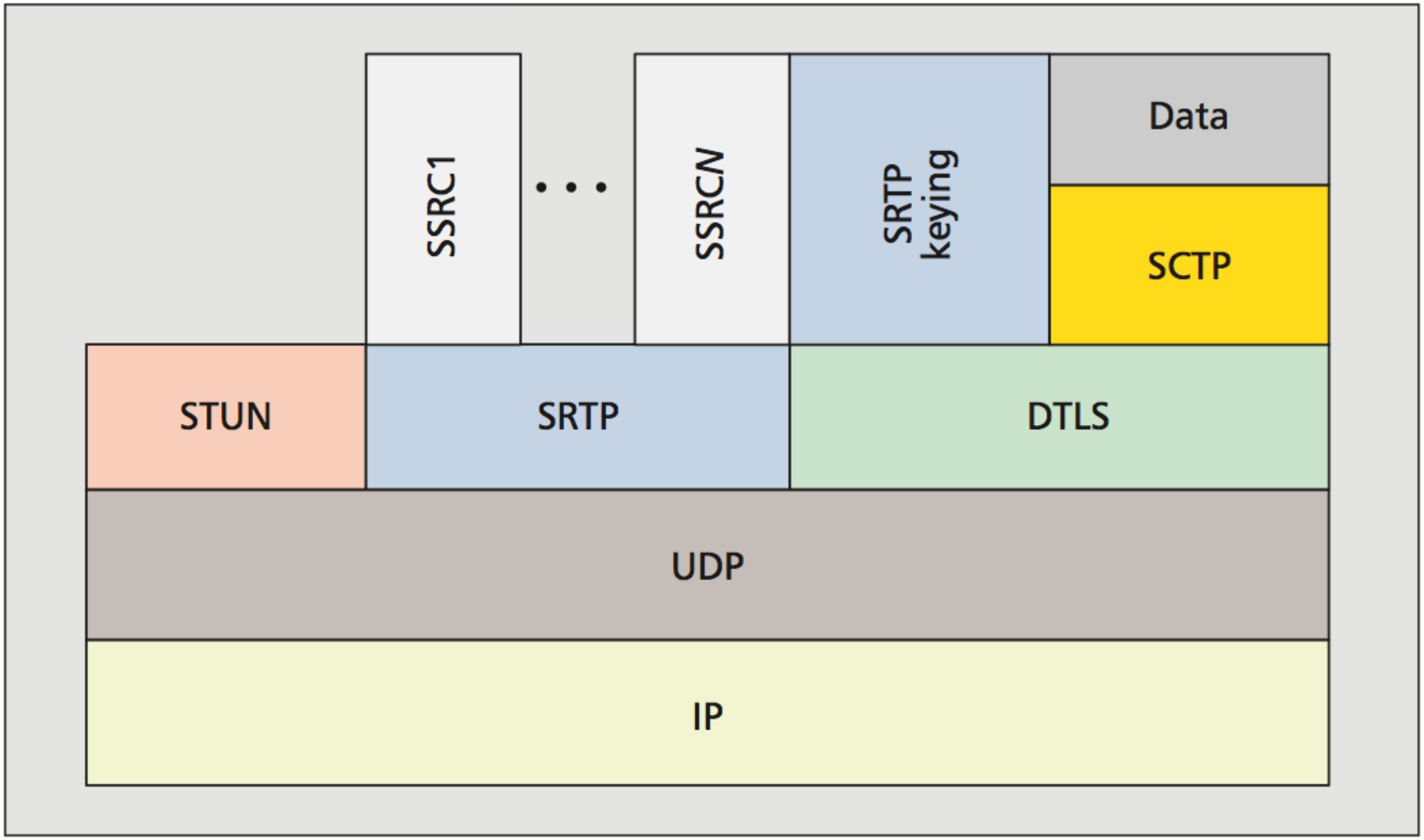
- Same RTP/sRTP used in nearly all other Voice/ Video over IP implementations (Sequence numbers, synchronization, etc...)
- Only Payload is Encrypted

Media Keying – DTLS

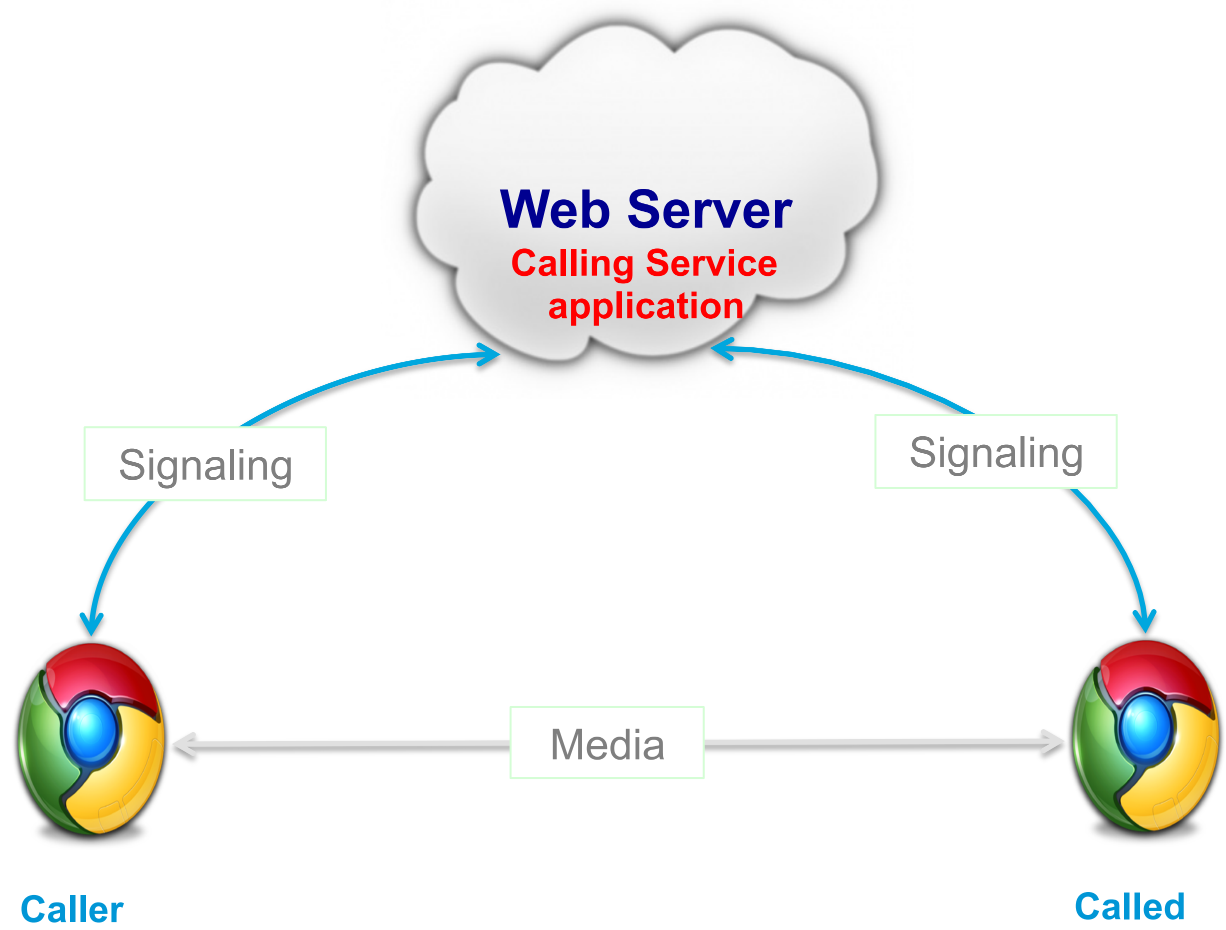
- DTLS handshake between endpoints to negotiate encryption keys
- Uses self-signed certificates (no traditional PKI)

Multiplexing of all Media

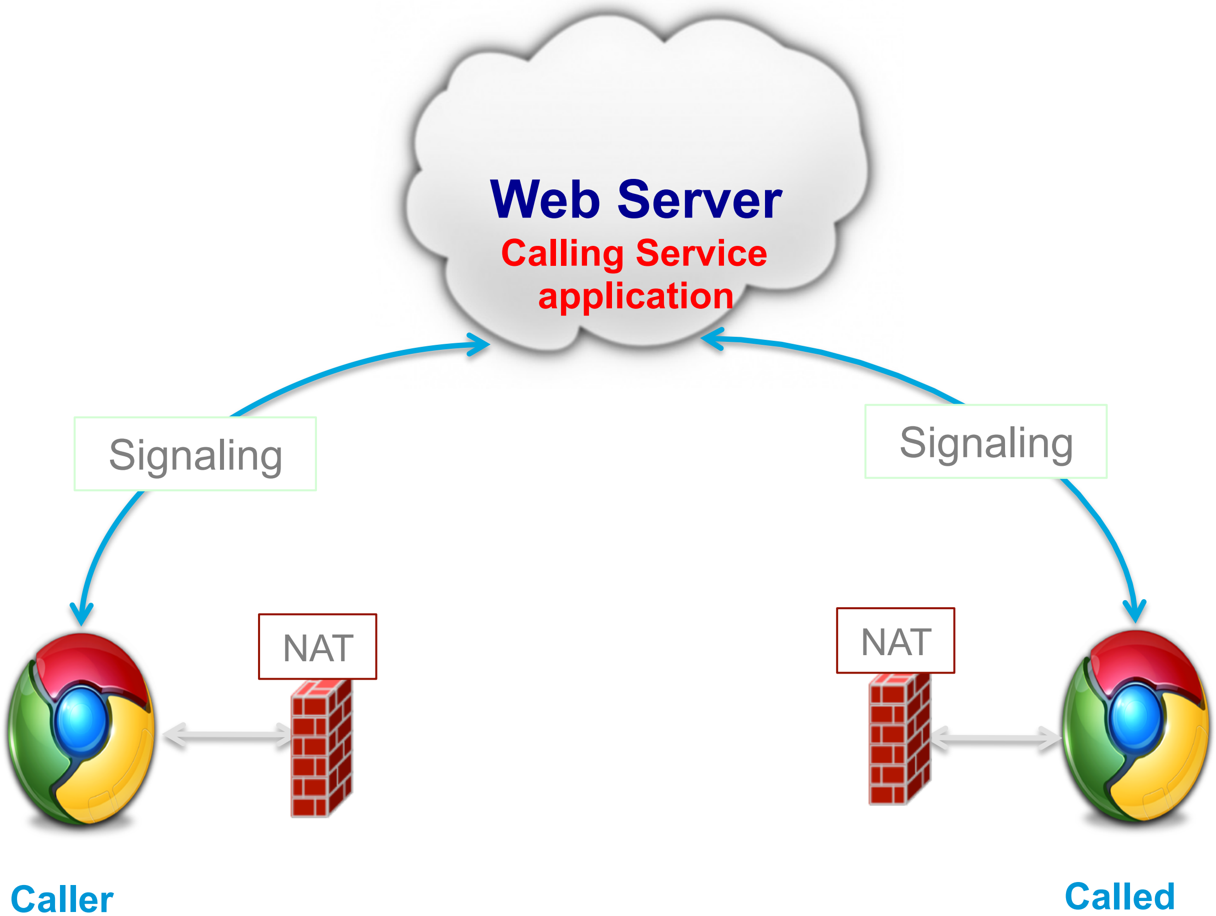
DTLS, SRTP, STUN over UDP



What we hope ..


Web Server
Calling Service application
Signaling
Signaling
Media
Caller
Called

But …

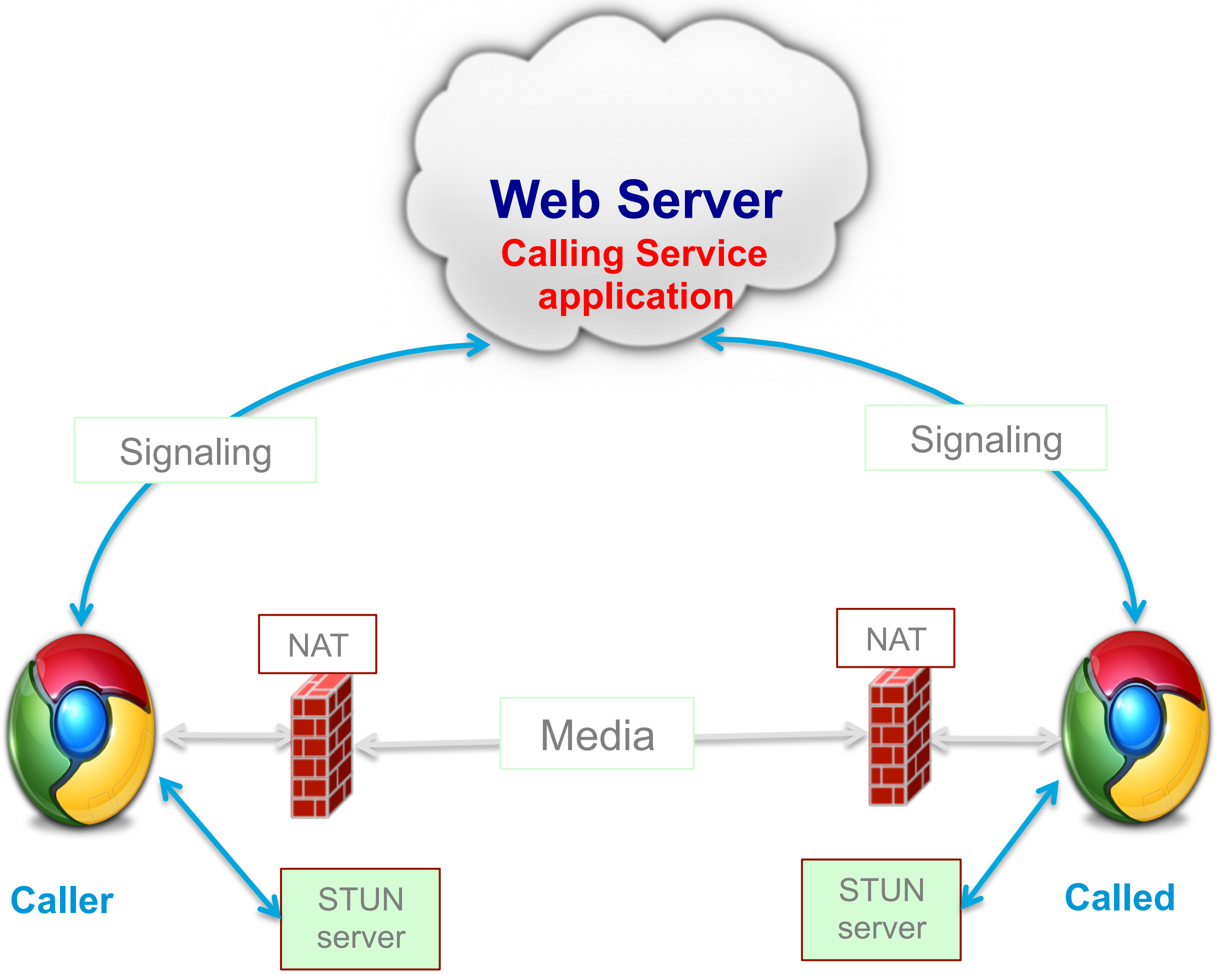


Web Server
Calling Service application
Signaling
Signaling
NAT
NAT
Caller
Called

NAT Traversal – STUN

- STUN (Session Traversal Utilities for NAT)
  - IETF RFC 5389
- A way to ask a server: "what is my Public IP address"
- Data flows peer-to-peer

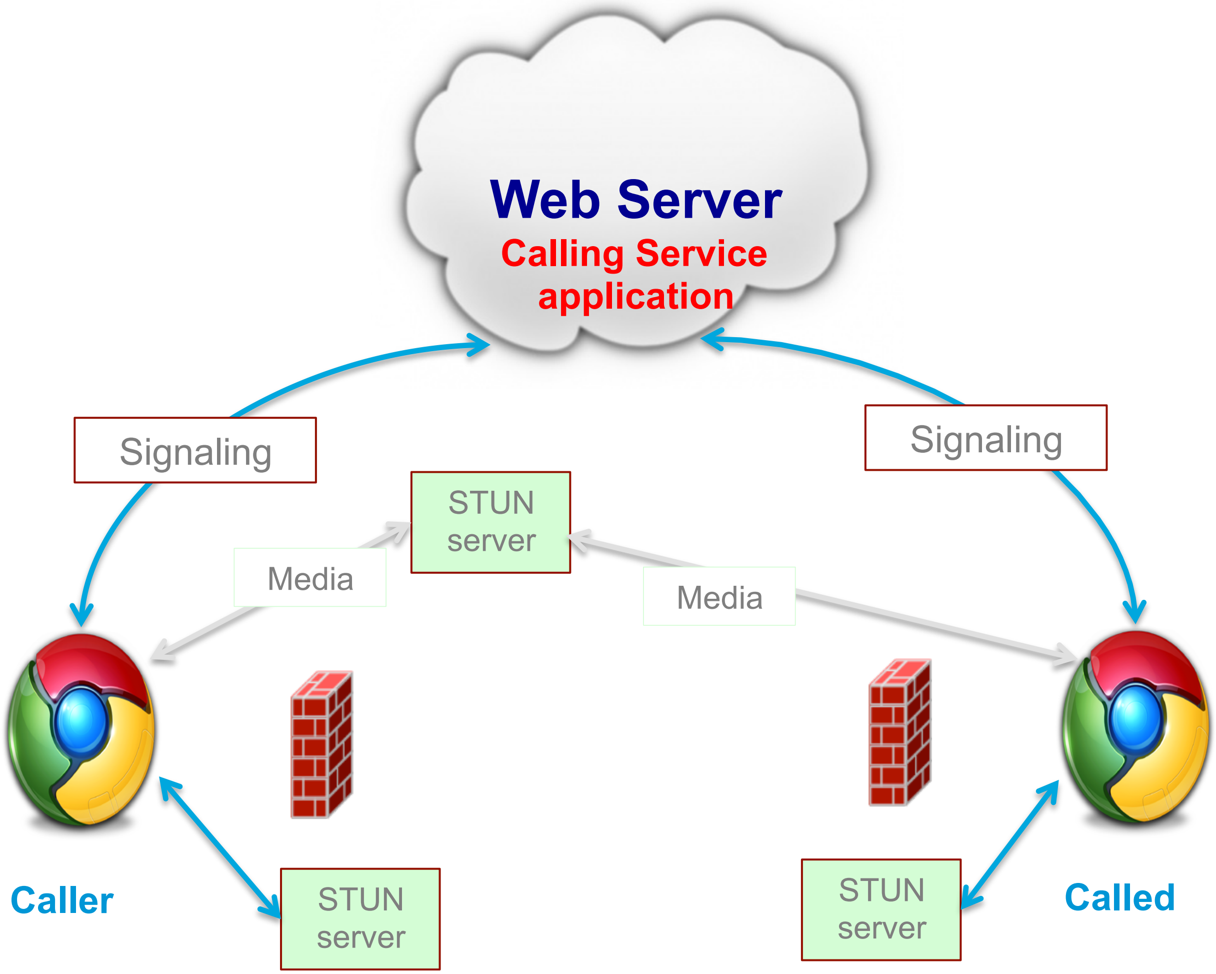
NAT Traversal – STUN



NAT Traversal – TURN

- TURN (Traversal Using Relays around NAT):
  - IETF RFC 5776
  - Extension of STUN – same msg format
- Provides a fallback mechanism if peer-to-peer communications fails
  - Tunnel protocol to tunnel data to and from a public server
- Data is sent through server, uses server bandwidth & adds to the latency (maybe negligible)
- Ensures the call works in almost all environments

NAT Traversal – TURN

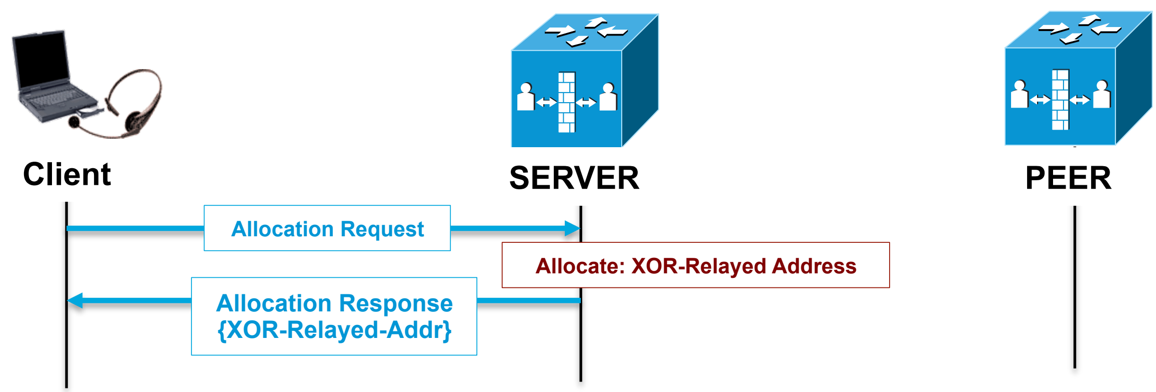


NAT Traversal – TURN protocol

- TURN stands for Traversal Using Relays around NAT
  - IETF RFC 5766
  - Extension of STUN (RFC 5389) and uses STUN formatted messages
- TURN protocol – how a Client communicates to a Relay Server
  - Relay Server allows exchange of data of the Client with it's Peer
  - "*Relayed transport address*" is an IP address & port allocated on Relay Server
- TURN does not cover a mechanism for sharing the "*Relayed transport address*" between Client & it's Peer
  - SIP & ROAP are examples of protocols that allow sharing the addresses

NAT Traversal –
1) TURN protocol – Allocations

- Client (Browser in this case) initiates Allocation-Request towards TURN server
  - All requests are send to Server UDP port 3478
- TURN server allocates relay IP address & responds with Allocation-Response
  - *XOR-Relayed-Address* field, in the response, indicates the IP address & port allocated on the VCS-E

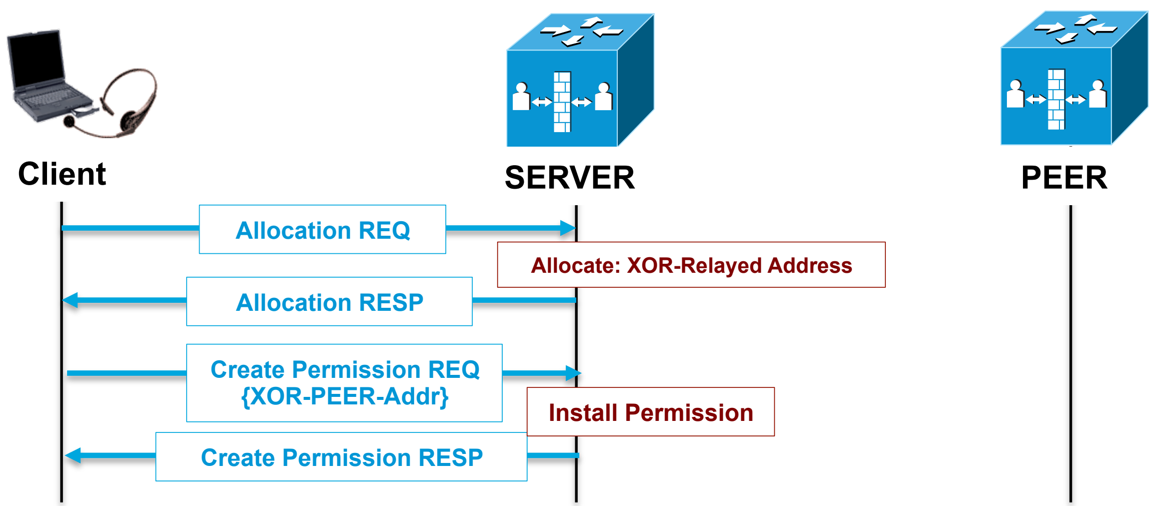


NAT Traversal –
2) TURN protocol – Permissions

- Used as a security mechanism

  Address Restricted Filtering mechanism:

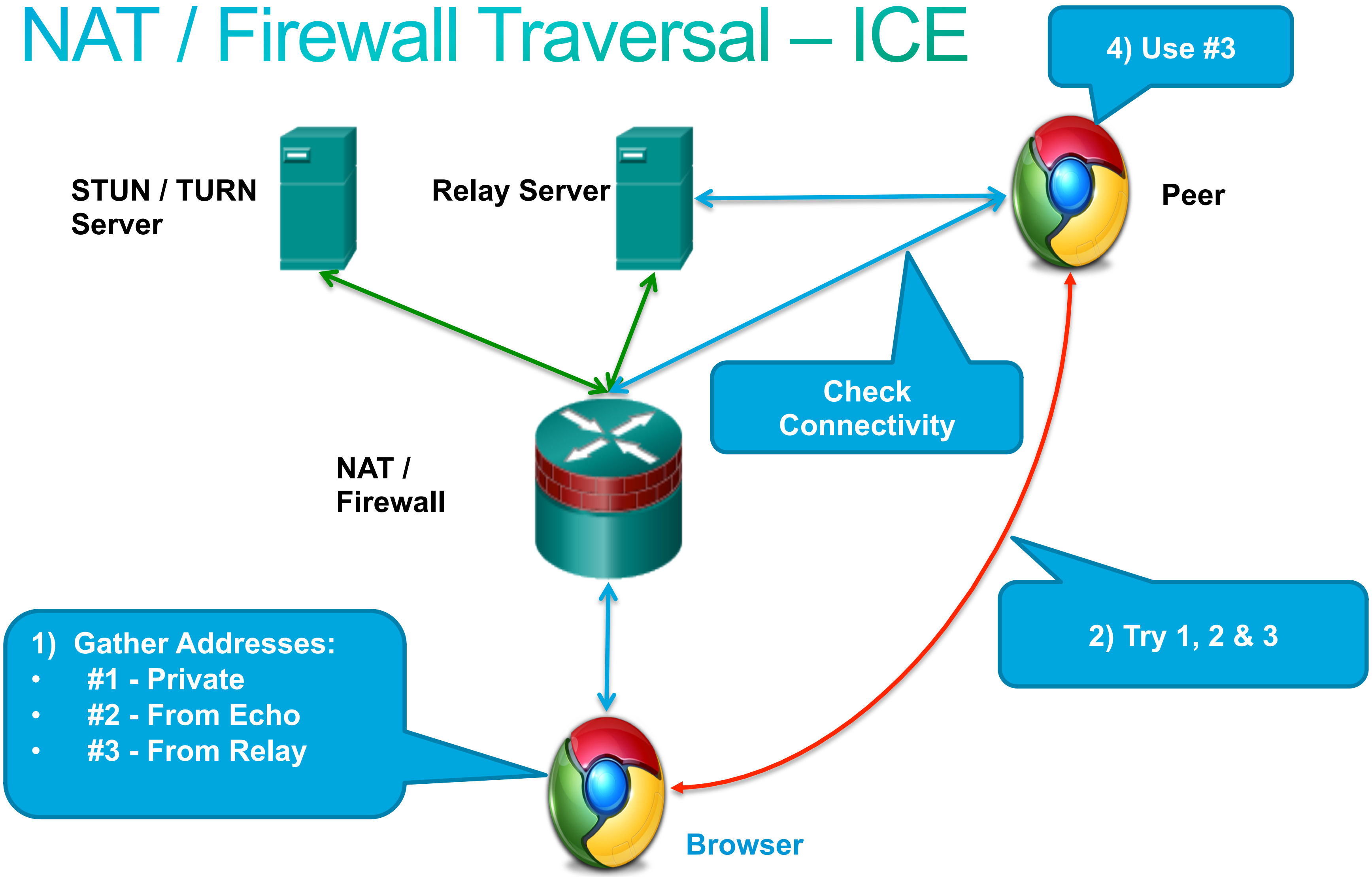
  If source IP address of datagram at Server do not match list-of-permission then data is discarded.



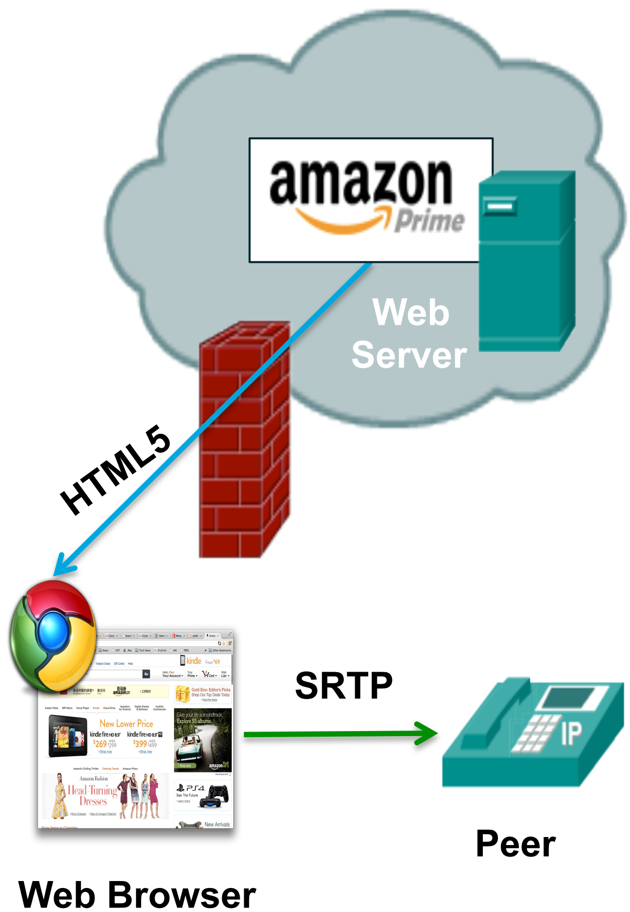
NAT / Firewall Traversal – ICE

- Combination of several components:
  - STUN (Session Traversal Utilities for NAT): a way to ask a public server what a client's apparent IP address is
  - TURN (Traversal Using Relays around NAT): remote relay tunnel protocol to tunnel data to and from a public server
  - ICE (Interactive Connectivity Establishment): approach to take several addresses that might work to communicate to another peer and test them to see which one works

NAT / Firewall Traversal – ICE

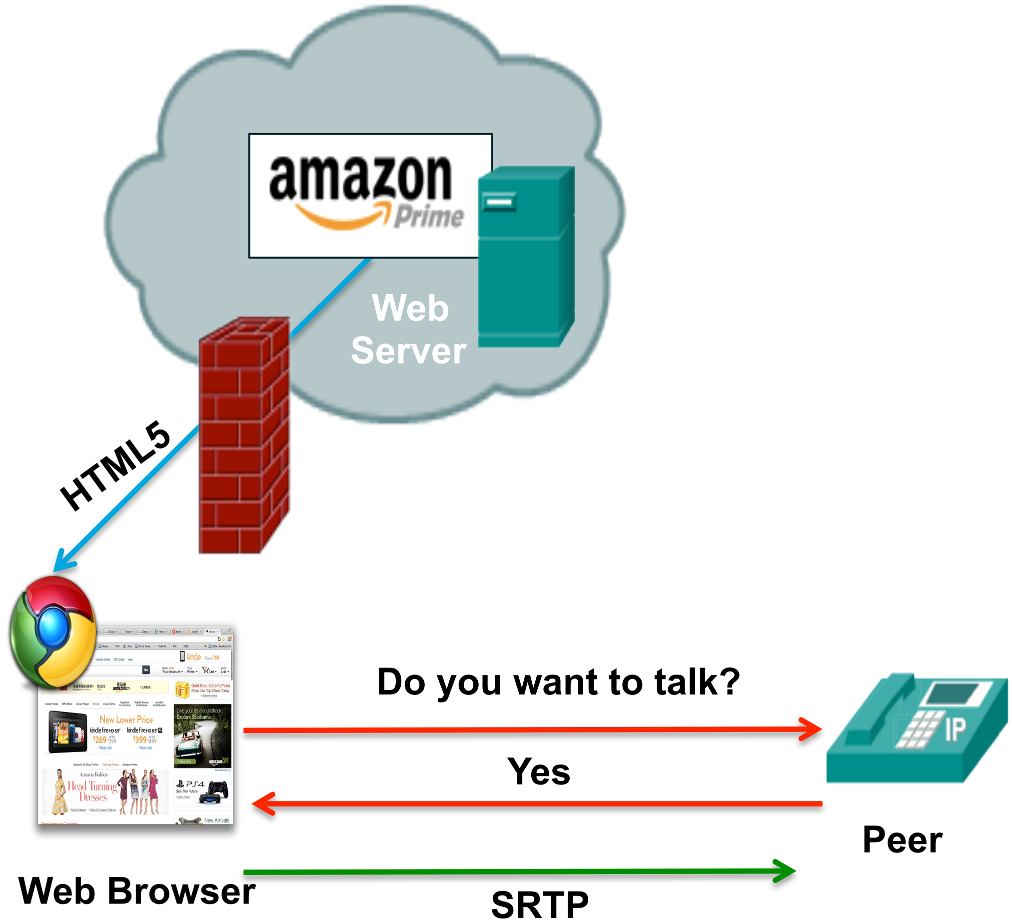


Media Consent

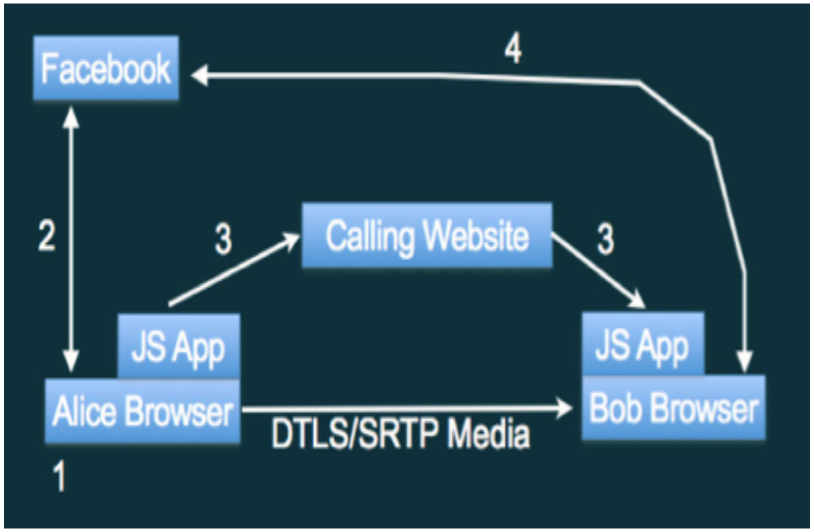


Media Consent

Media Consent – ICE



WebRTC – Identity



- Browser is configured with identity provider(s) for the user
  1. User "logs on" to identity provider
  2. Browser gets assertion from identity provider that binds the DTLS fingerprint to identity
  3. Calling JavaScript passes assertion to far-end
  4. Bob's browser verifies the assertion with identity provider and check DTLS fingerprint matches

Interoperability Considerations

WebRTC & Business Video

How standards-compliant are signaling implementations in the gateway & business system?

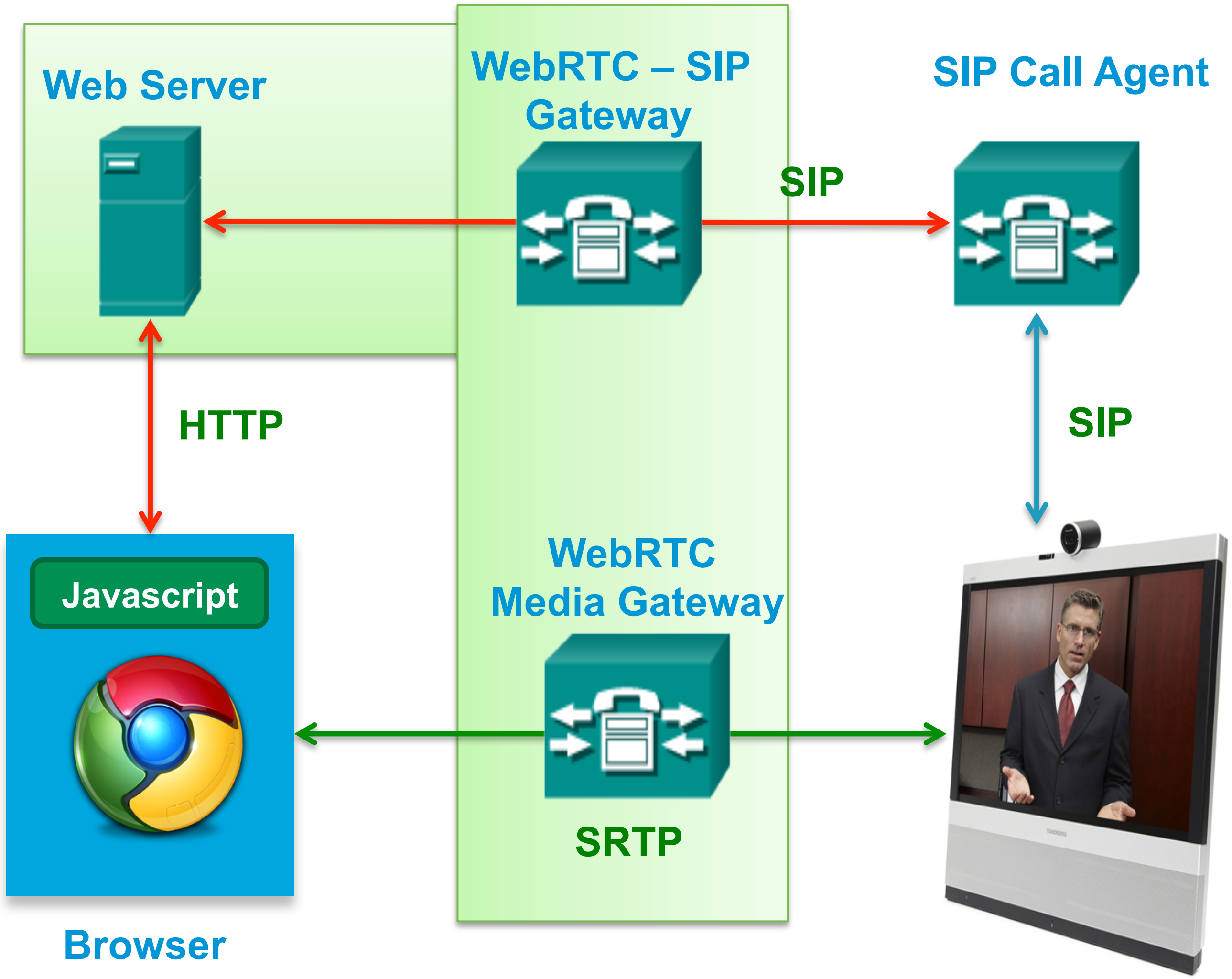
Does the business endpoint support ICE?

Can the business endpoint speak G.711 or Opus for voice?

Can the business endpoint speak common video codec (VP8)?

Does the business endpoint support DTLS SRTP?

WebRTC & Business Video



Summary

Overview of WebRTC

WebRTC – how does it work?

Standards activities

Vendor strategy example

WebRTC / RTCWeb - Standards Efforts

I E T F®

- RTCWeb Working Group
  - Primary effort in IETF
  - April 2011
- Defining how browsers communicate with others … largely re-using existing protocols
- Notable documents …
  - draft-ietf-rtcweb-audio
  - draft-ietf-rtcweb-data-channel
  - draft-ietf-rtcweb-jsep
  - draft-ietf-rtcweb-overview
  - draft-ietf-rtcweb-qos
  - draft-ietf-rtcweb-rtp-usage
  - draft-ietf-rtcweb-security-arch
  - draft-ietf-rtcweb-use-cases-and-requirements

W3C®

- WebRTC Working Group
  - Primary effort in W3C
  - May 2011
- Defining APIs on how Web applications access browser real-time communications, i.e. device control (camera, mic)
- Notable documents …
  - WebRTC 1.0: Real-time Communication Between Browsers
  - Media Capture and Streams
  - Media Capture Scenarios

39

WebRTC standardization

Moving along .. But still there are TBDs

CONVERGING

- Audio Codecs … G.711, Opus
- Signaling … SDP-based offer/ answer using JavaScript
- Firewall/NAT Traversal … ICE, STUN, TURN
- Media Encryption … DTLS-keyed SRTP
- Media Consent … ICE/STUN
- Identity … identity provider model
- QoS … DiffServ Code Point markings to enhance WiFi, residential GWs, LTE links

TBD

- Video Codec(s) … VP8 vs H.264?
- Congestion Control … goals = minimize latency, quick reaction, consistent data flow
- Screen/Application Sharing
- Etc …

40

Standards: Issues & Challenges

Video Codec

- Choice of 'Mandatory-To-Implement' Video Codec(s)
  - Open source codec or royalty based codecs?
  - VP8, H.264, H.261 (what resolution, frequency?)
- One camp pushing for VP8
  - Google open sourced this codec after acquisition of On2
  - Questions about Intellectual Property Rights of VP8
- Impact to Business video vendors e.g.
  - Cisco/Polycom: Existing Video environments using H.264
  - Browser vendors split: Chrome/Opera (VP8) vs Safari/IE (H.264)
    Firefox VP8 & H.264

Browser Implementations of WebRTC

Google Chrome

- Initial implementation in Chrome 23 Stable. Latest v 25
- Actively contributing to standards efforts
- Contributing to open source, e.g. WebRTC.org

Microsoft Internet Explorer

- Implementation status not public

Mozilla Firefox

- Initial implementation in Firefox Aurora channel. Latest in Nightly
- Actively working on open source implementation & contributing to standards efforts
- Cisco contributed open source development, e.g. RFC4566-compliant SDP engine, call control application logic

Apple Safari

- Maintaining strict secrecy

42

Agenda

Overview of WebRTC

WebRTC – how does it work?

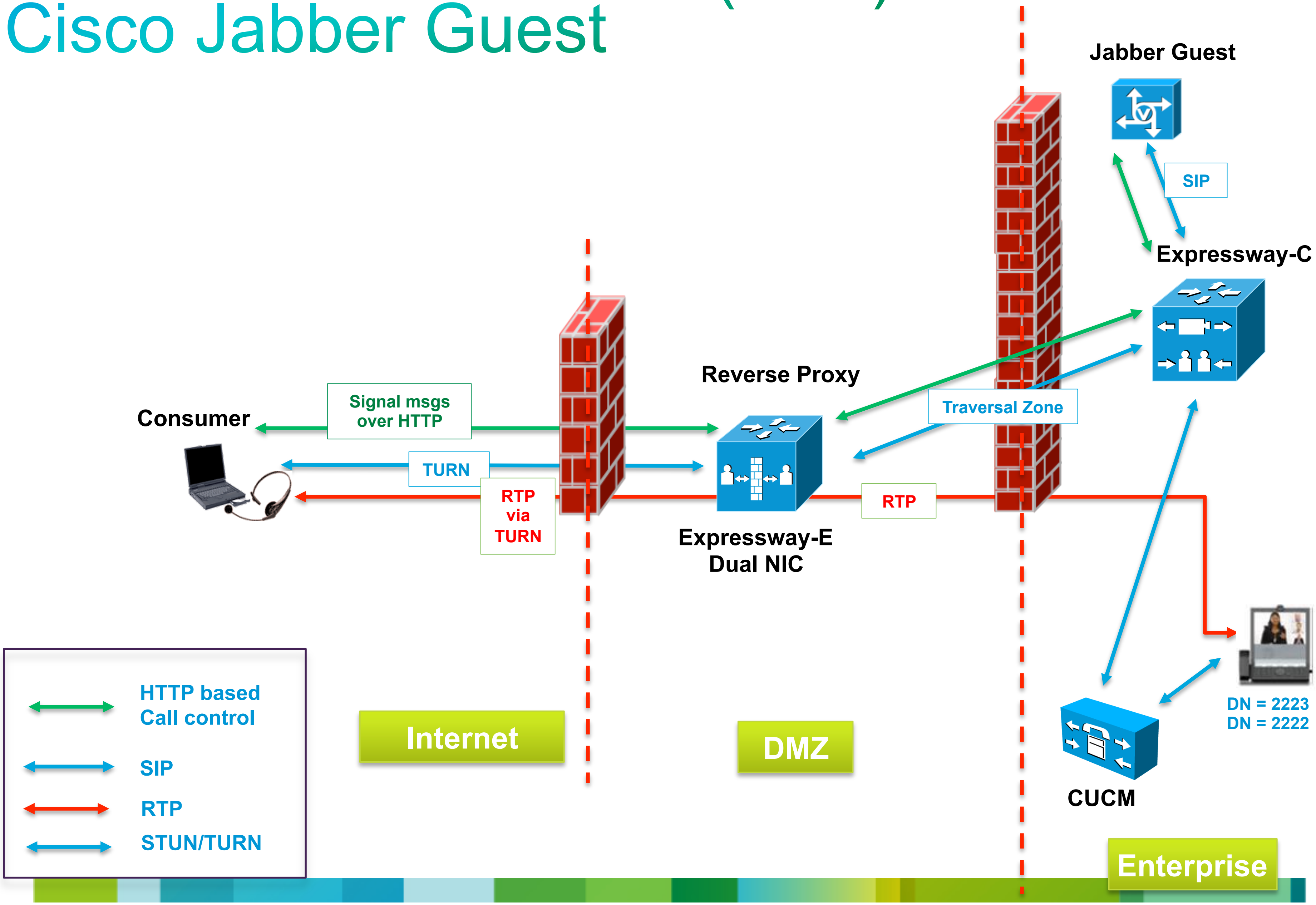
Standards activities

Vendor strategy example

Vendors & WebRTC – summary

- WebRTC browsers vendors: early adopters for desktop
  - Chrome, Firefox, Opera
- Microsoft proposed alternate standard CU-RTCWeb in Aug'12
  - Lack of WebRTC support on IE can cause adoption issues
  - But participating in Standards
- Apple Safari: no plans shared with market
  - Engaged in IETF standardization process
- WebRTC: Browsers on mobile will get there overtime
  - Support *baked-into* Firefox for Android
- WebRTC gateway functionality (WebRTC <-> UC/Video)
  - ACME/Oracle showing interest in developing Signaling & Media gateways – SBC
- Emerging vendors
  - CafeX

Consumer to Business (C2B)
Cisco Jabber Guest



Summary

Overview of WebRTC

WebRTC – how does it work?

Standards activities

Vendor strategy example

Questions

Acknowledgements

- Thanks to Cullen Jennings (Cisco) and Justin Uberti (Google) for various slides