WebRTC: Real Time Communications for the Web

Dr. Cullen Jennings
May 2015
fluffy@cisco.com

WebRTC Motivations

- Easy for developers to put communications where needed
  - Enable contextual communications
- Easy to deploy across many operating systems and types of devices
- Strong security
  - Communications users can trust
- Faster to get new features from developer to user
- Peer 2 Peer

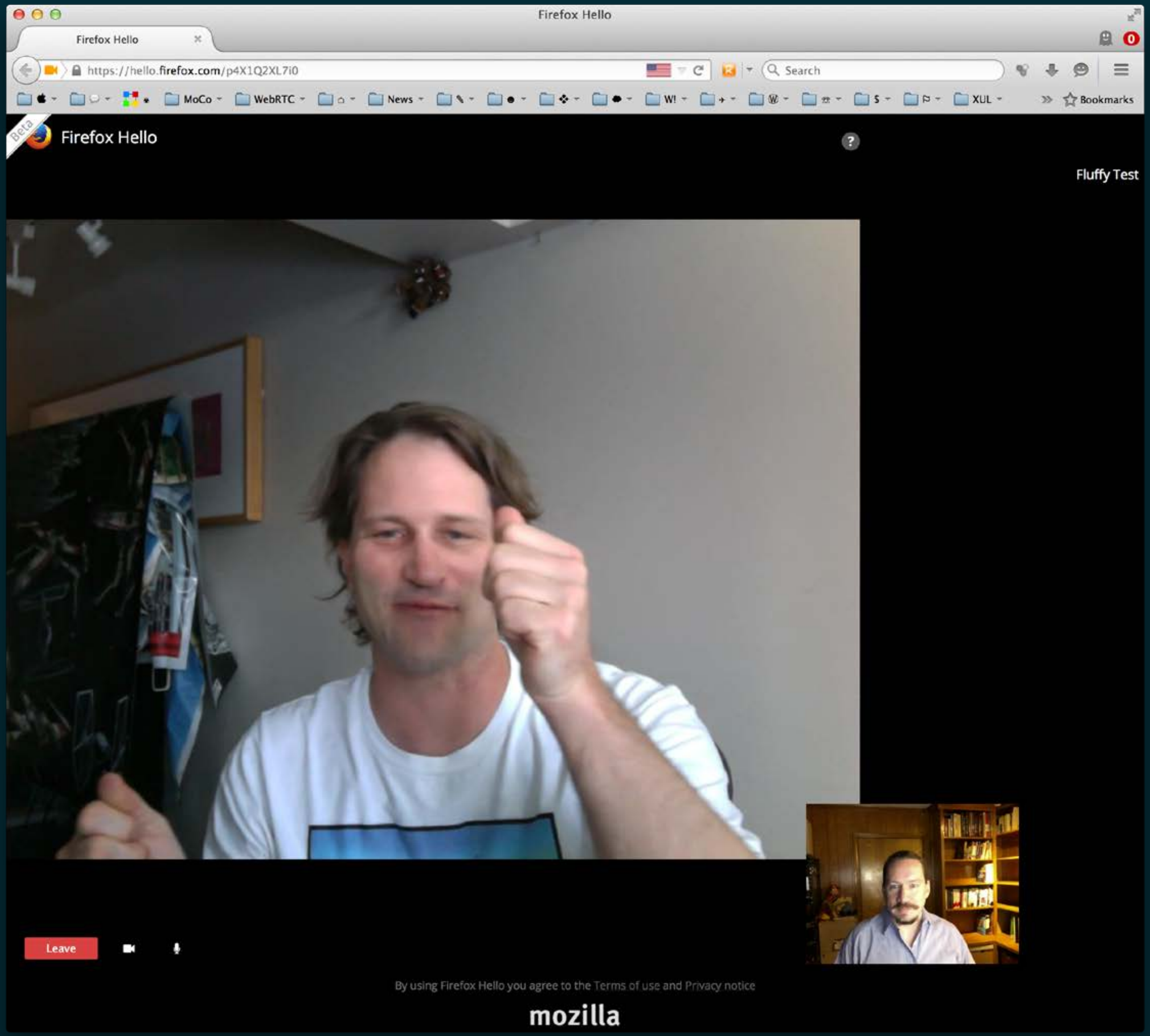
Plan

- Take the guts of a SIP soft phone
- Stuff it into a browser
- Wrap it with an programming interface in the browser that any website can use
- TBD
- Profit

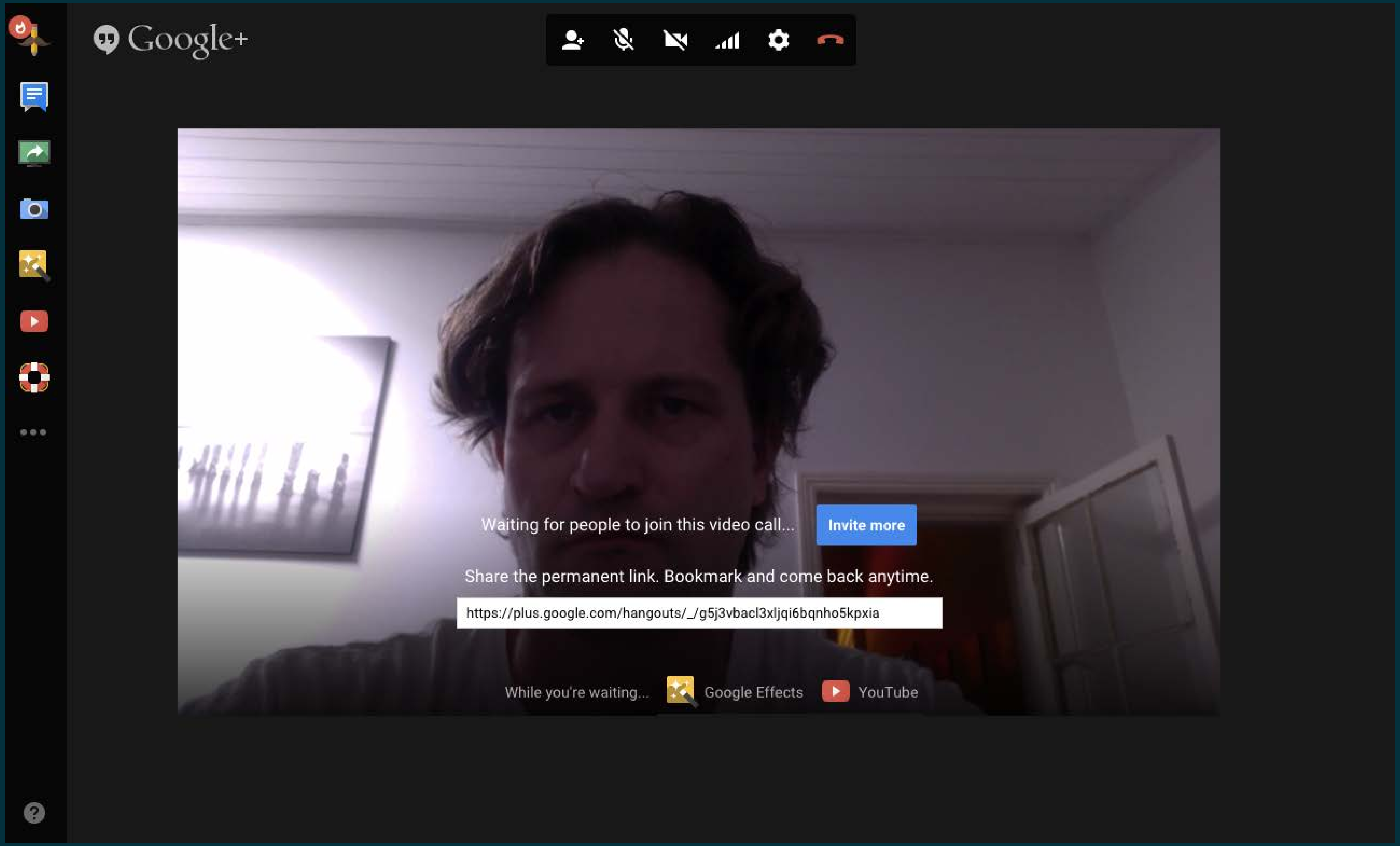
How to Think About WebRTC

- Technology
  - It’s a technology that enable voice, video, and data sharing in a peer to peer fashion between applications running in a browser
- Peer 2 Peer
  - Traditionally browsers only sent data in client server fashion, now they can talk browser to browser
- Big Eco System
  - It is interoperable with modern unified communication systems
  - Accessible to many developers
- Zero Install
  - It’s part of the web browser and does not require installing any extra plugins

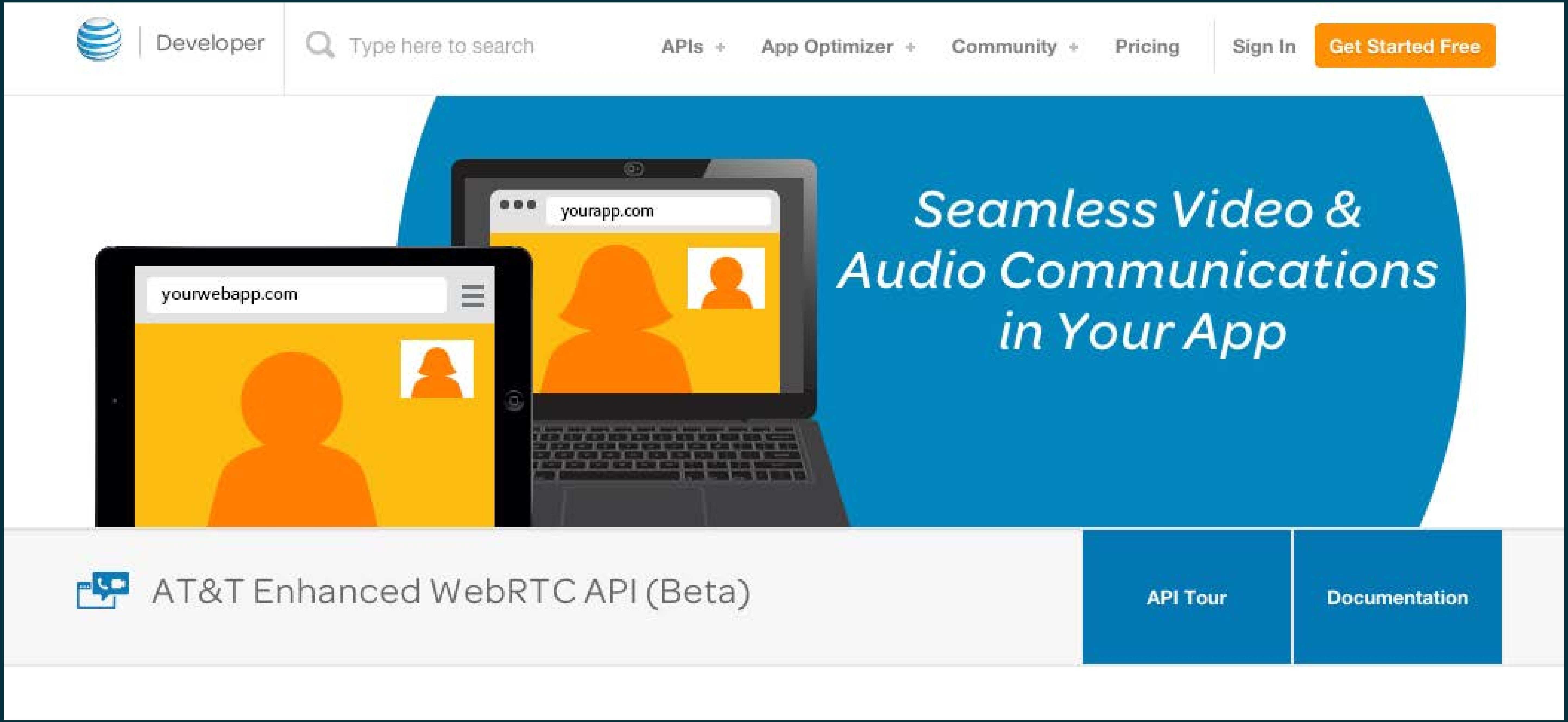
Firefox / Telefonica Hello



Google Hangouts

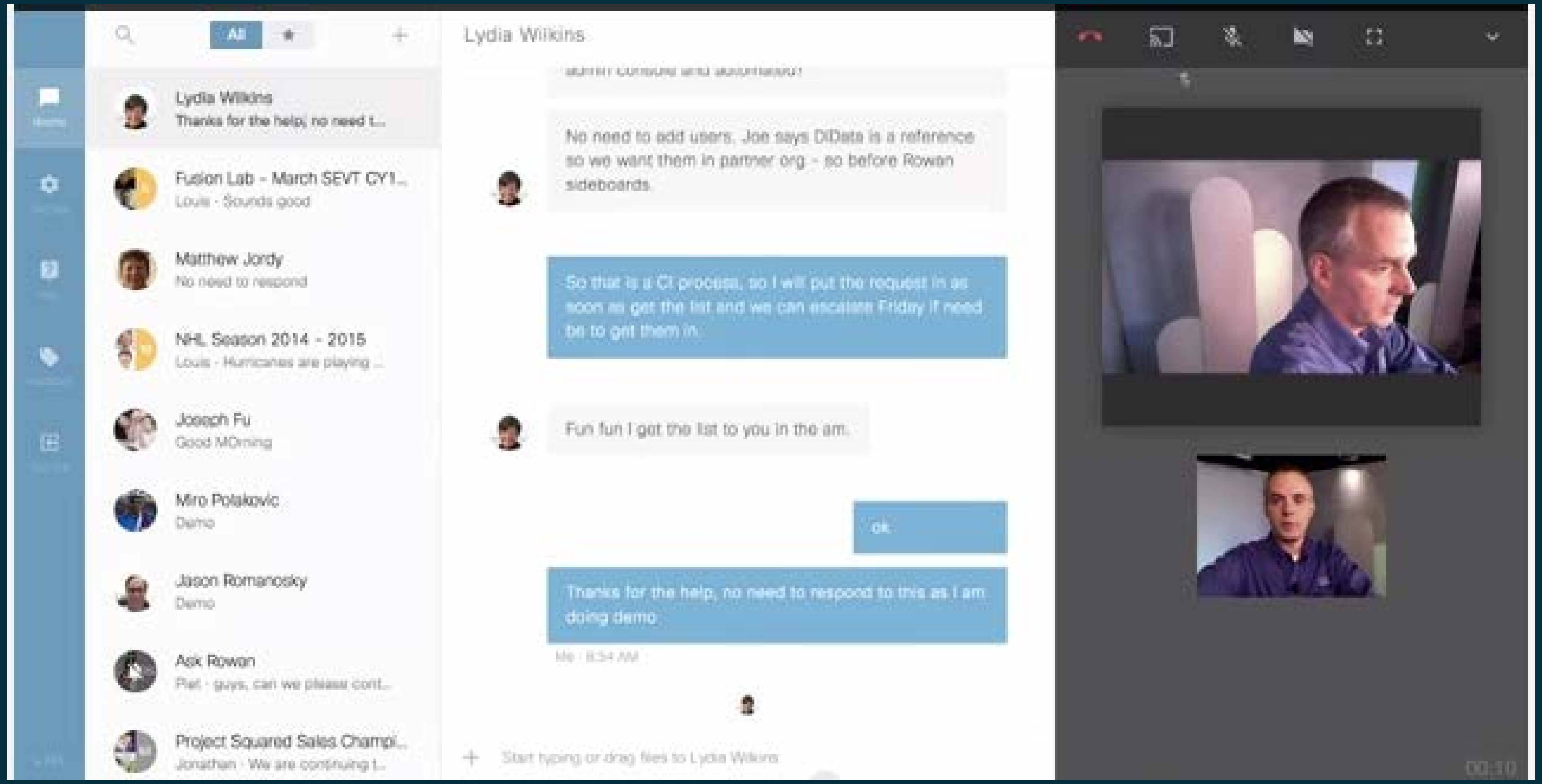


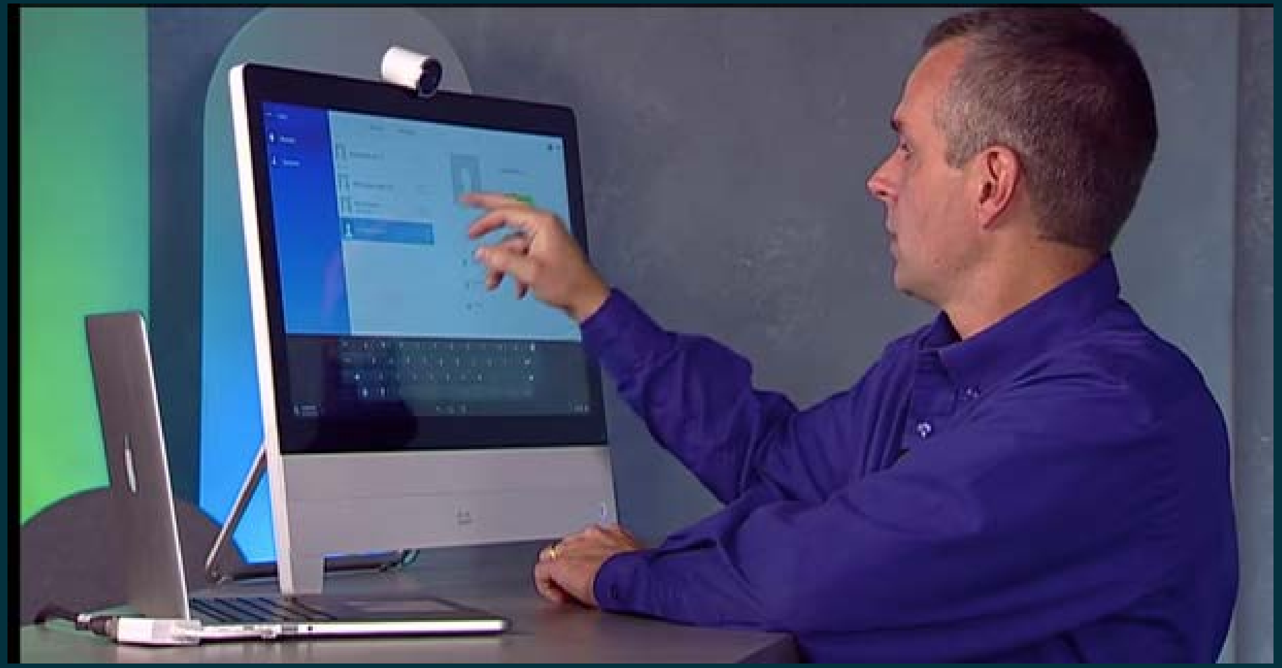
AT&T



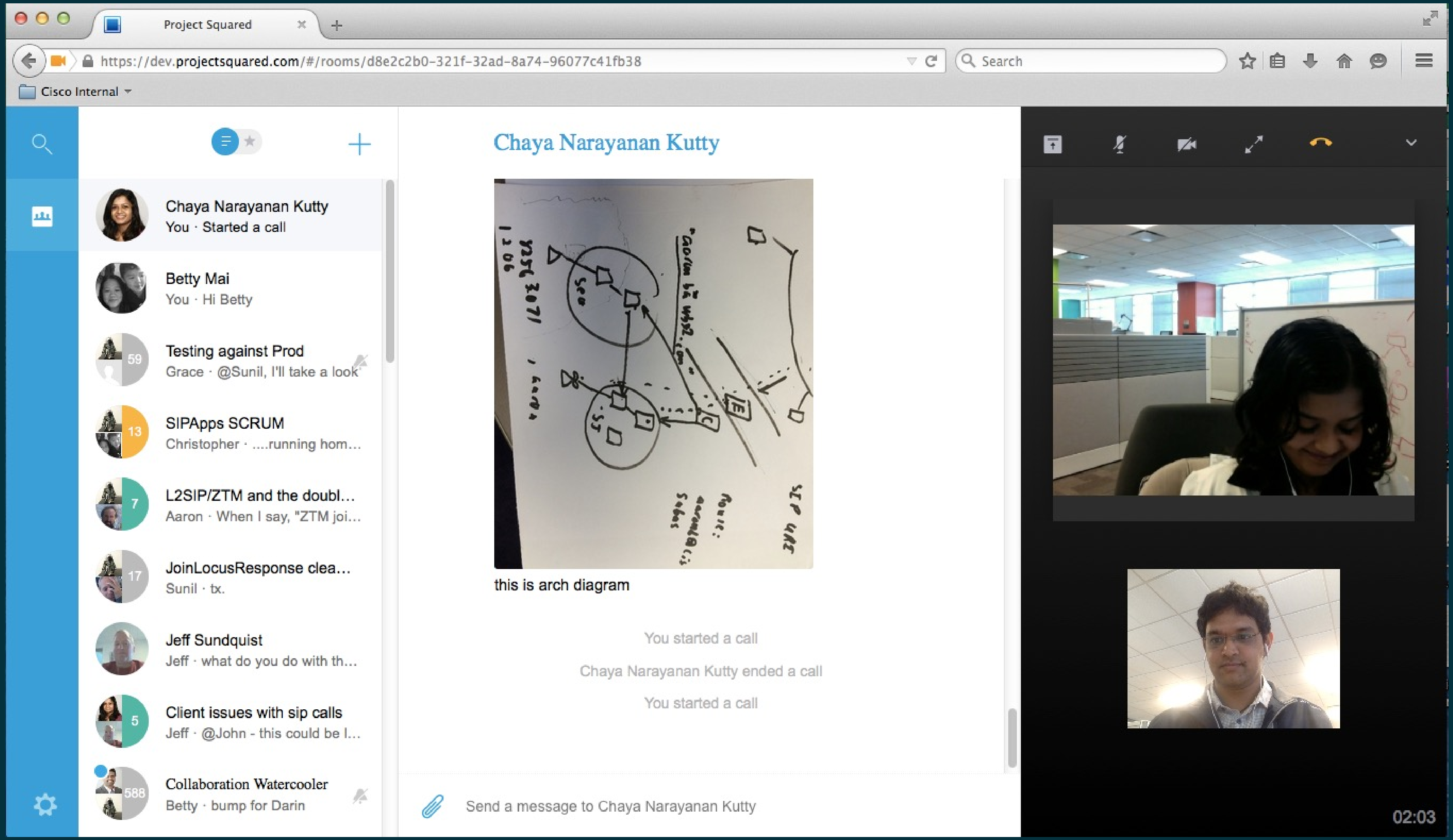
Developer
Type here to search
APIs +
App Optimizer +
Community +
Pricing
Sign In
Get Started Free
yourapp.com
yourwebapp.com
Seamless Video & Audio Communications in Your App
AT&T Enhanced WebRTC API (Beta)
API Tour
Documentation

Cisco Spark


All
Lydia Wilkins
Lydia Wilkins
Thanks for the help, no need t...
Fusion Lab - March SEVT CY1...
Louis · Sounds good
Matthew Jordy
No need to respond
NHL Season 2014 - 2015
Louis · Hurricanes are playing ...
Joseph Fu
Good MOrning
Miro Polakovic
Demo
Jason Romanosky
Demo
Ask Rowan
Piet · guys, can we please cont...
Project Squared Sales Champi...
Jonathan · We are continuing t...
No need to add users. Joe says DiData is a reference so we want them in partner org - so before Rowan sideboards.
So that is a CI process, so I will put the request in as soon as get the list and we can escalate Friday if need be to get them in.
Fun fun I get the list to you in the am.
ok
Thanks for the help, no need to respond to this as I am doing demo
Start typing or drag files to Lydia Wilkins

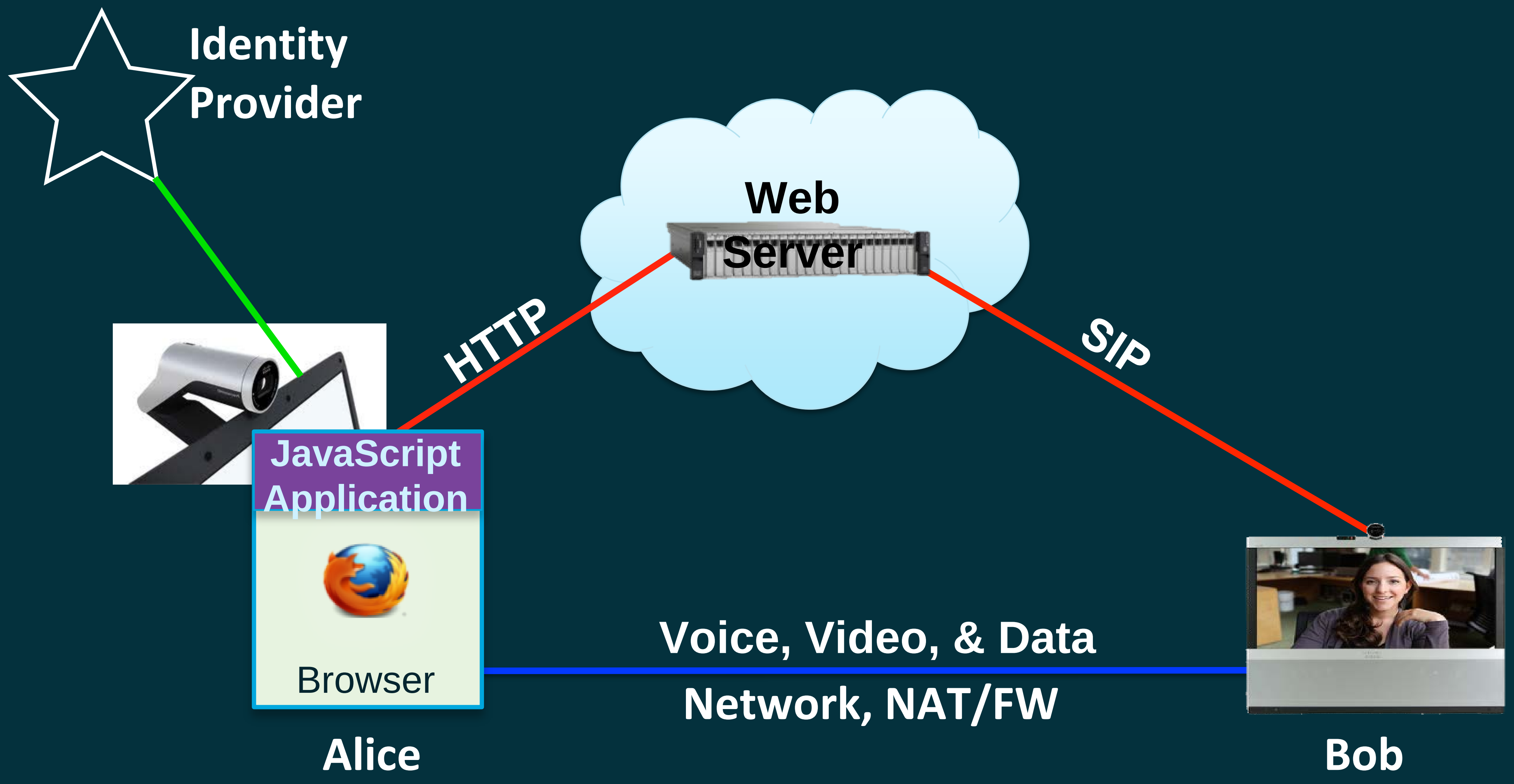


Cisco Spark


Project Squared
https://dev.projectsquared.com/#/rooms/d8e2c2b0-321f-32ad-8a74-96077c41fb38
Search
Cisco Internal
Chaya Narayanan Kutty
You · Started a call
Betty Mai
You · Hi Betty
Testing against Prod
Grace · @Sunil, I'll take a look
SIPApps SCRUM
Christopher · ....running hom...
L2SIP/ZTM and the doubl...
Aaron · When I say, "ZTM joi...
JoinLocusResponse clea...
Sunil · tx.
Jeff Sundquist
Jeff · what do you do with th...
Client issues with sip calls
Jeff · @John - this could be l...
Collaboration Watercooler
Betty · bump for Darin
Chaya Narayanan Kutty
this is arch diagram
You started a call
Chaya Narayanan Kutty ended a call
You started a call
Send a message to Chaya Narayanan Kutty
02:03

How WebRTC Works

Architecture



The Parts of WebRTC

- WebRTC API
- Identity
- SDP
- ICE/STUN/TURN
- DTLS/SRTP
- CODECs

Network Protocols

Media - Codecs

- Either end can have many codecs and a negotiation picks the best possible that both ends support
- Audio Codecs
  - Narrowband audio: G.711
  - Wideband audio: Opus
- Video standards:
  - Browser required to support both VP8 and H.264

Data Channel

- WebRTC isn't just voice and video
  - It also provides direct P2P data channels
  - Useful for games, file sharing, P2P networks, etc.
- How does this relate to Web Sockets?
  - Similar API but data goes direct
  - This makes it easy to polyfill WebRTC DC apps to WebSockets
- Lots of apps will just use Data Channels

Media Transport - SRTP

- SRTP provides a sequence number and timestamp for each media packets
- This allows synchronization of play out of differ media streams (lip sync)
- It also allows detection of lost packets

- SRTCP provides feedback on packet loss rates and SRTP statistics
- SRTP support many forms of error recovery and forward error correction

- SRTP uses symmetric key cryptography to provide confidentiality and integrity
- Ongoing IETF work to multiplex multiple SRTP over same UDP flow

Media Keying - DTLS

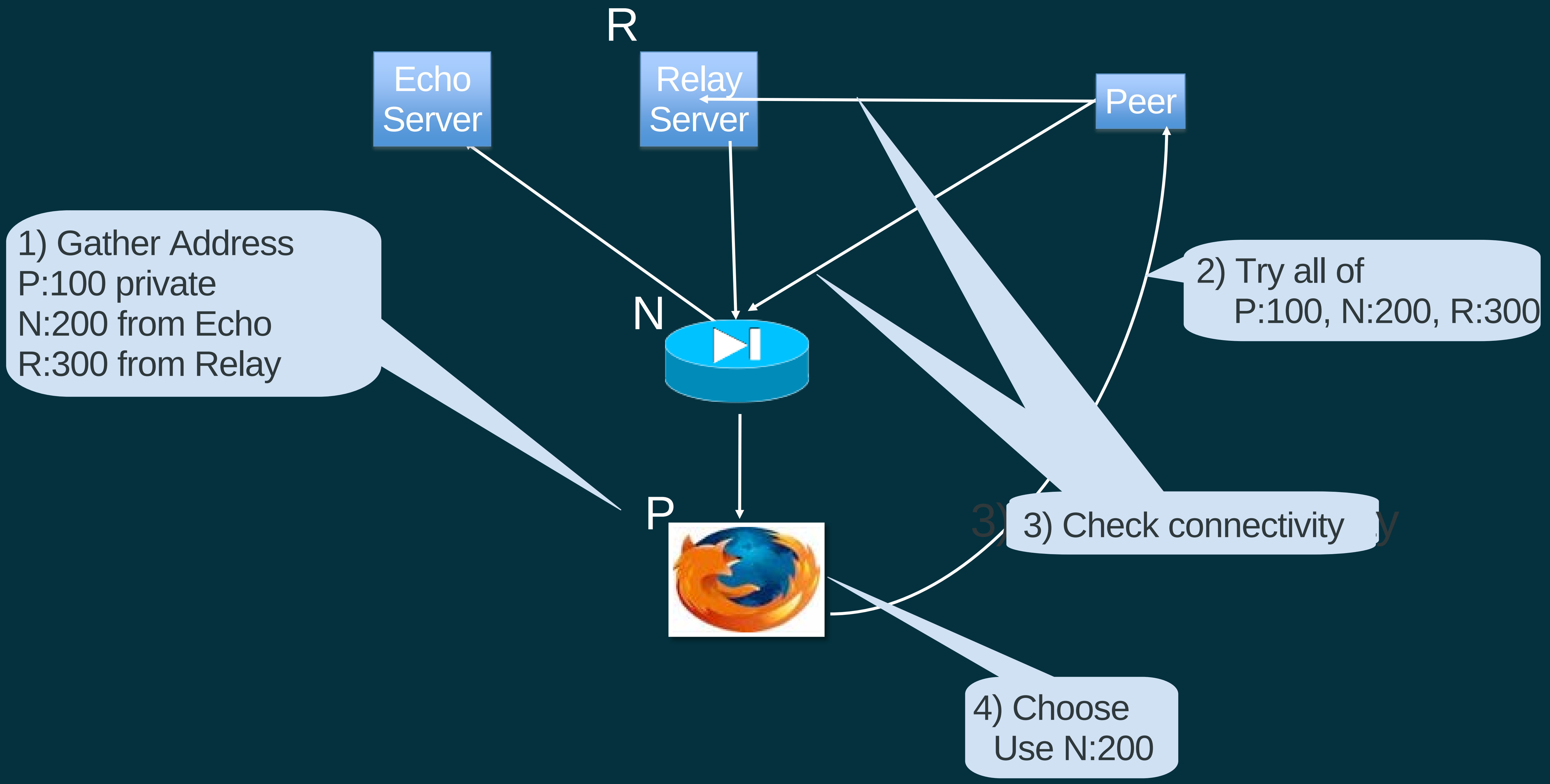
- DTLS is simply the same TLS used for HTTPS adapted for UDP
- DTLS handshake is used to form the session keying material for the SRTP media encryption
- Used with self signed certificates. Each certificate has a fingerprint which is bound to a user identity in a way described later in this presentation

NAT / Firewall Traversals - ICE

- ICE provides a way to get media between two devices that are both behind NATs and some firewalls
- It also forms a way to detect changing network conditions and switch from an interface such as WiFi to a different interface such as LTE
- Finally it is used for media consent to make sure unwanted traffic is not sent to devices

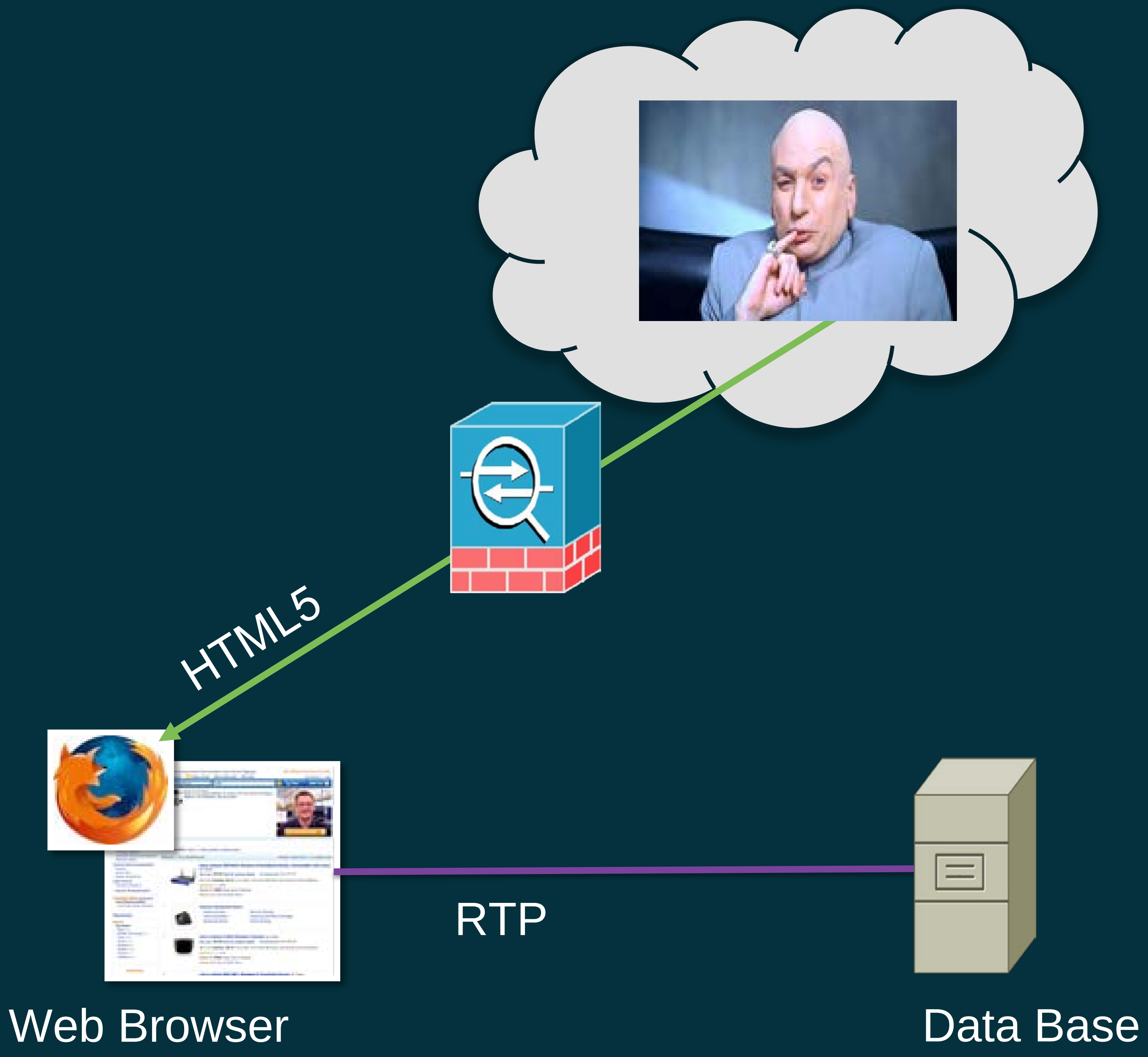
- Combination of several components
  - TURN: is a remote relay tunnel protocol to tunnel data to and from a public server
  - STUN: is a way to ask a public server what a client's apparent IP address is
  - ICE: an approach to take several addresses that might work to communicate to another peer and test them to see which one works

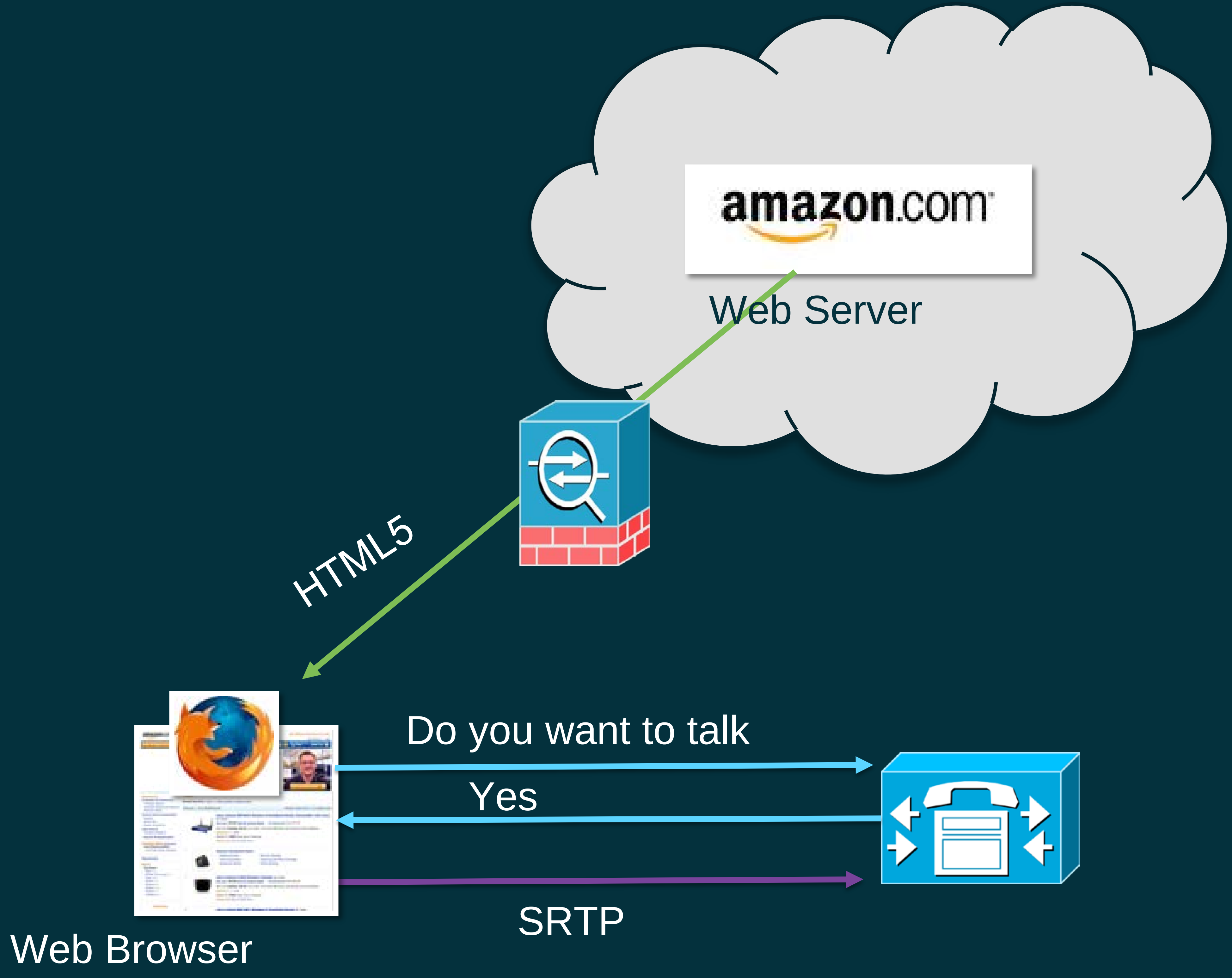
ICE


R
Echo Server
Relay Server
Peer
1) Gather Address
P:100 private
N:200 from Echo
R:300 from Relay
N
2) Try all of
P:100, N:200, R:300
P
3) Check connectivity
4) Choose
Use N:200

Media Consent


amazon.com
Web Server
HTML5
SRTP
Web Browser
Video Phone


HTML5
RTP
Web Browser
Data Base


amazon.com
Web Server
HTML5
Do you want to talk
Yes
SRTP
Web Browser

Signaling - SDP

- The SDP offer/answer protocol used by SIP is used for media negotiation
- Rich interface to describe what codecs, network transports, and media options one side can support (the offer) and which ones the other sides wants to select (the answer)

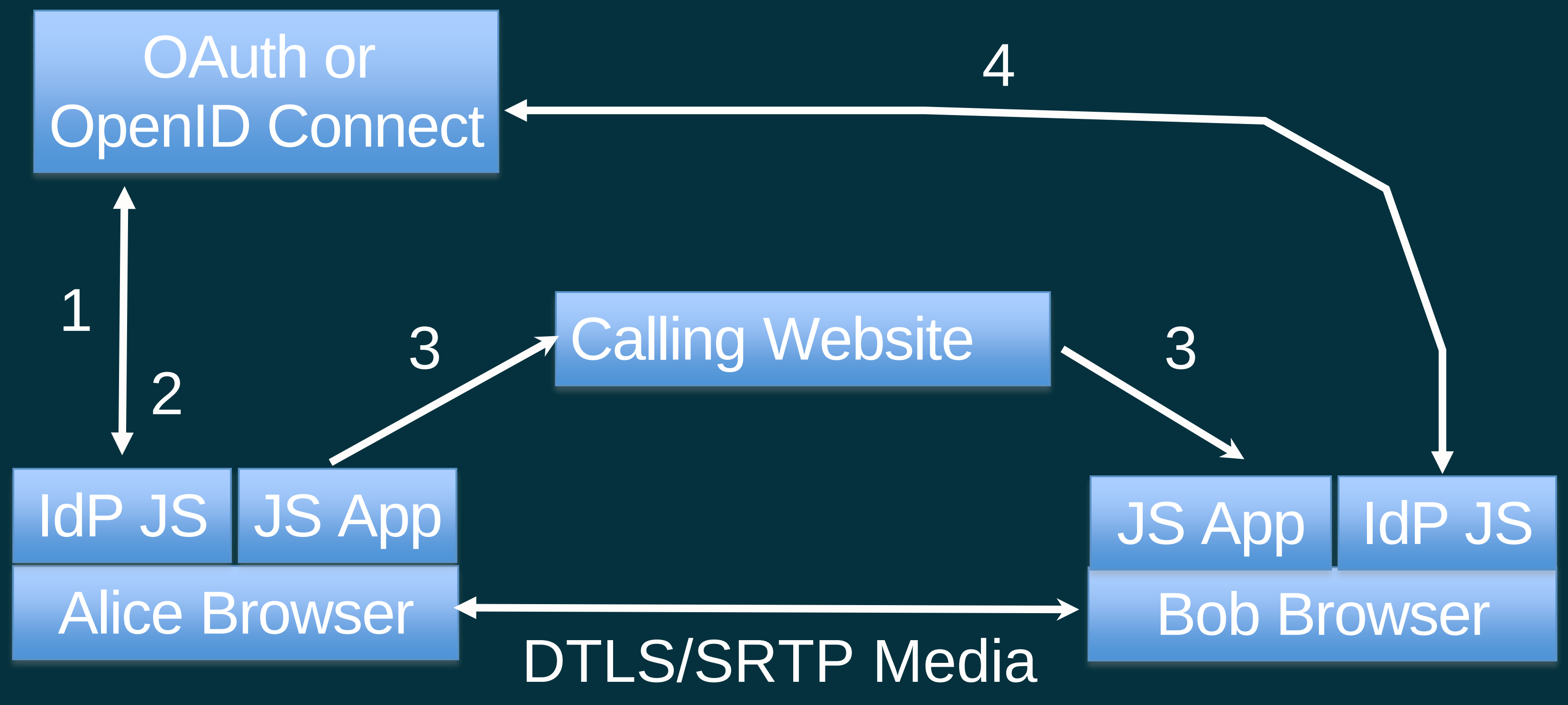
```
v=0
o=- 292742730 29277831 IN IP4 131.163.72.4
s=
c=IN IP4 131.164.74.2
t=0 0
m=video 52886 RTP/AVP 31
a=rtpmap:31 H261/90000
a=content:slides
m=video 53334 RTP/AVP 31
a=rtpmap:31 H261/90000
a=content:main
```

Identity

Who is fluffy@cisco.com

- Who is in the best position to make strong assertions about who fluffy@cisco.com is?
  - Cisco.com allocated the address fluffy to Cullen
  - They provided a way for Cullen to prove his identity with logon password, secure token card, etc.
  - Having a certificate authority (CA) assert that some random person can receive email sent to fluffy@cisco.com is a weak assertion of identity
- Who knows who cisco.com is?
  - The CA can verify with DNS registrars who has been given that name and can get appropriate contacts for it

Identity



- Browser is configured with identity provider(s) for the user

1. User "logs on" using protocol downloaded from identity provider in JavaScript/HTML
2. Browser get an assertion from identity provider which binds the DTLS fingerprint to the identity such as fluffy@cisco.com
3. The calling JavaScript passes the assertion to far side
4. Bob's browser verifies the assertion with identity provider and check DTLS fingerprint matches the assertion
5. Browser display "secure to fluffy@cisco.com"

Quality of Service (QoS)

- Based on Differentiated Service Code Point markings set on media packets
  - JS Application can provide hints about relative priority of media streams
  - Browser knows media type of packets
  - Browser sets the DSCP appropriately
  - Network may take DSCP into account when prioritizing packets

Congestion Control & Rate Adaptation

- Goals:
  - Be “fair” with TCP - i.e.. don’t push TCP traffic to floor and don’t be pushed to floor by TCP
  - Minimize latency
  - React to changing network conditions quickly
  - Provide a consistent flow of data

- Variety of algorithms combined:
  - Losing too many packets, slow down
  - Not losing many packets, speed up
  - Packet delay starts going up, slow down
  - If up shifted, then promptly downshifted, wait awhile for next upshift

Transitions

Industry Transitions

- Viruses / malware / industrial spying
  - reduce willingness to run plugins or new software
- Dev Op
  - driving a need for rapid deployment
- Embedded communications
  - put communications in the tools and systems that need it
- Internet of Things
  - enable more “thing” to “people” communications

Cloud Data

- Huge amount of data in the cloud which WebRTC further adds too
- Large amounts of collection by governments and less legal entities
- Continuous stream of financial losses

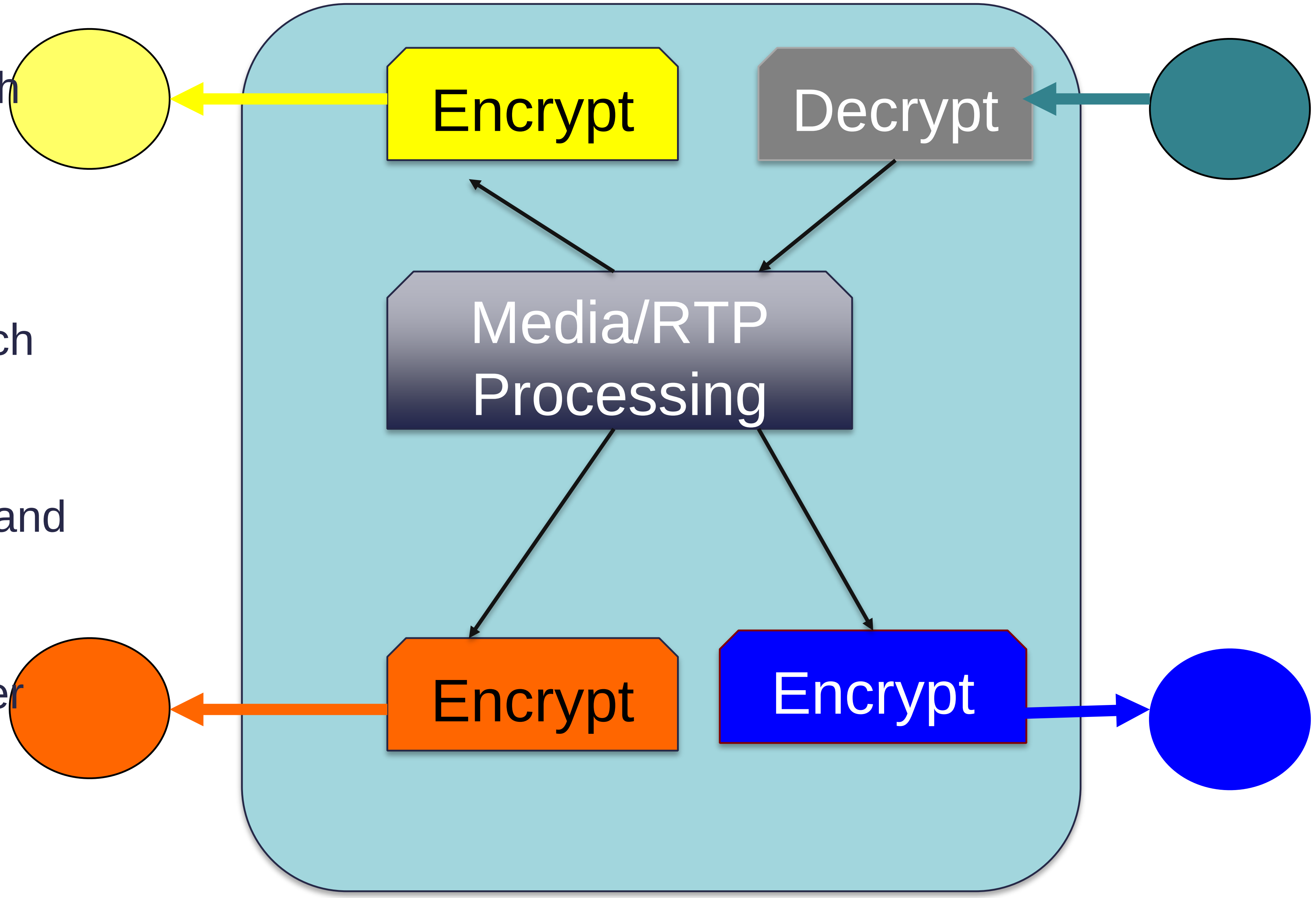
If you can’t protect data, don’t collect it

Securing the Cloud

Conference Bridges with the Keys

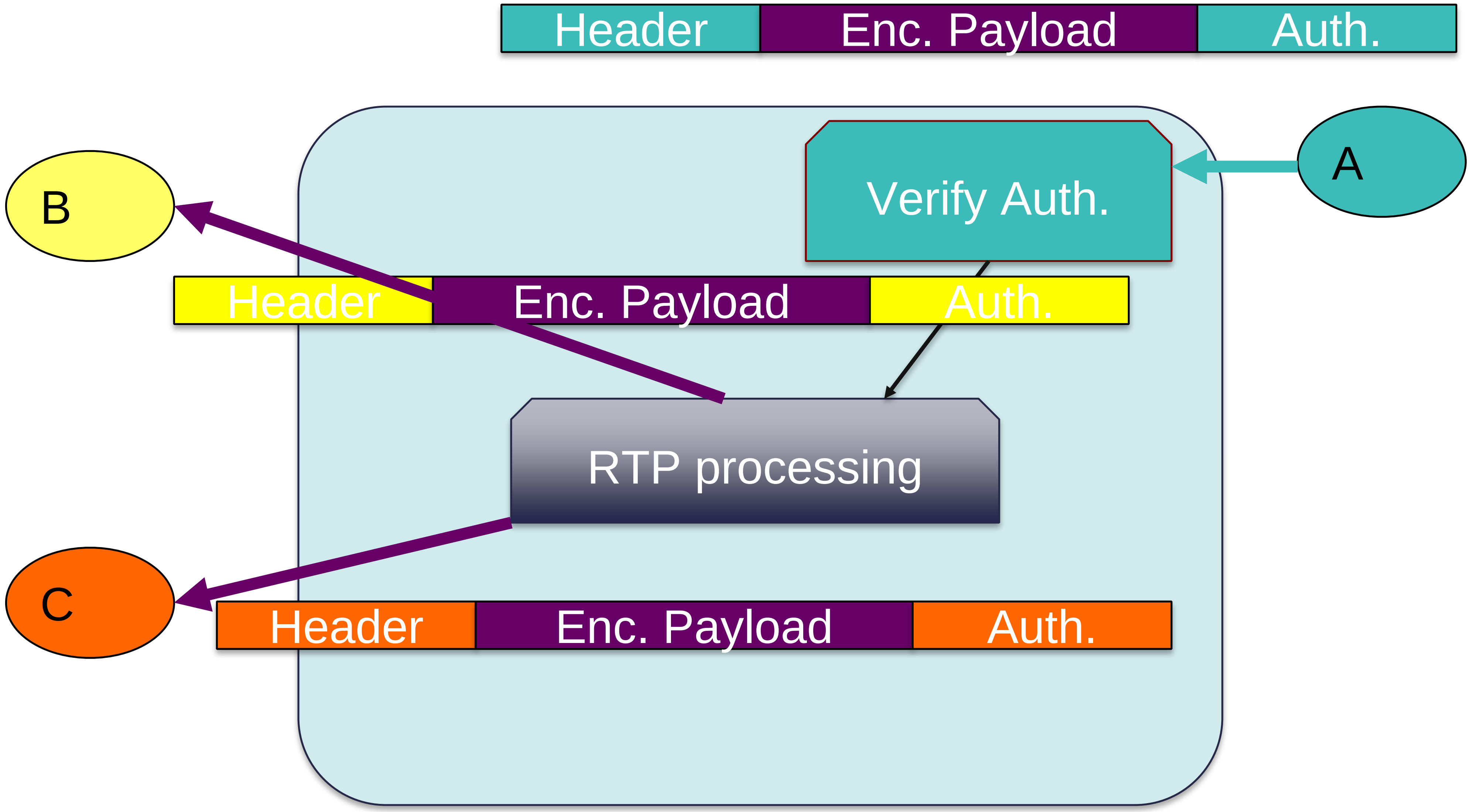
The Old way to do Multi-User Security Using SRTP

- Endpoint encrypts/authenticates using SRTP with its own key and unique SSRC per stream
- Multi-point server verifies authentication and decrypts each stream
- Multi-point server generates a unique key for each endpoint and a unique SSRC per stream per endpoint
- Multi-point server generates a new RTP header and encrypts and authenticates prior to forwarding
- SRTP context is managed between endpoint transmitters and server as well as between server and endpoint receivers



Multi-User Security with Content Privacy (the New Way)

- Endpoint transmitter encrypts and authenticates content
- Multi-point server verifies authentication, modifies RTP header and re-authenticates
- Key used for media encryption is not known to server
- Endpoint receiver authenticates packet and decrypt media

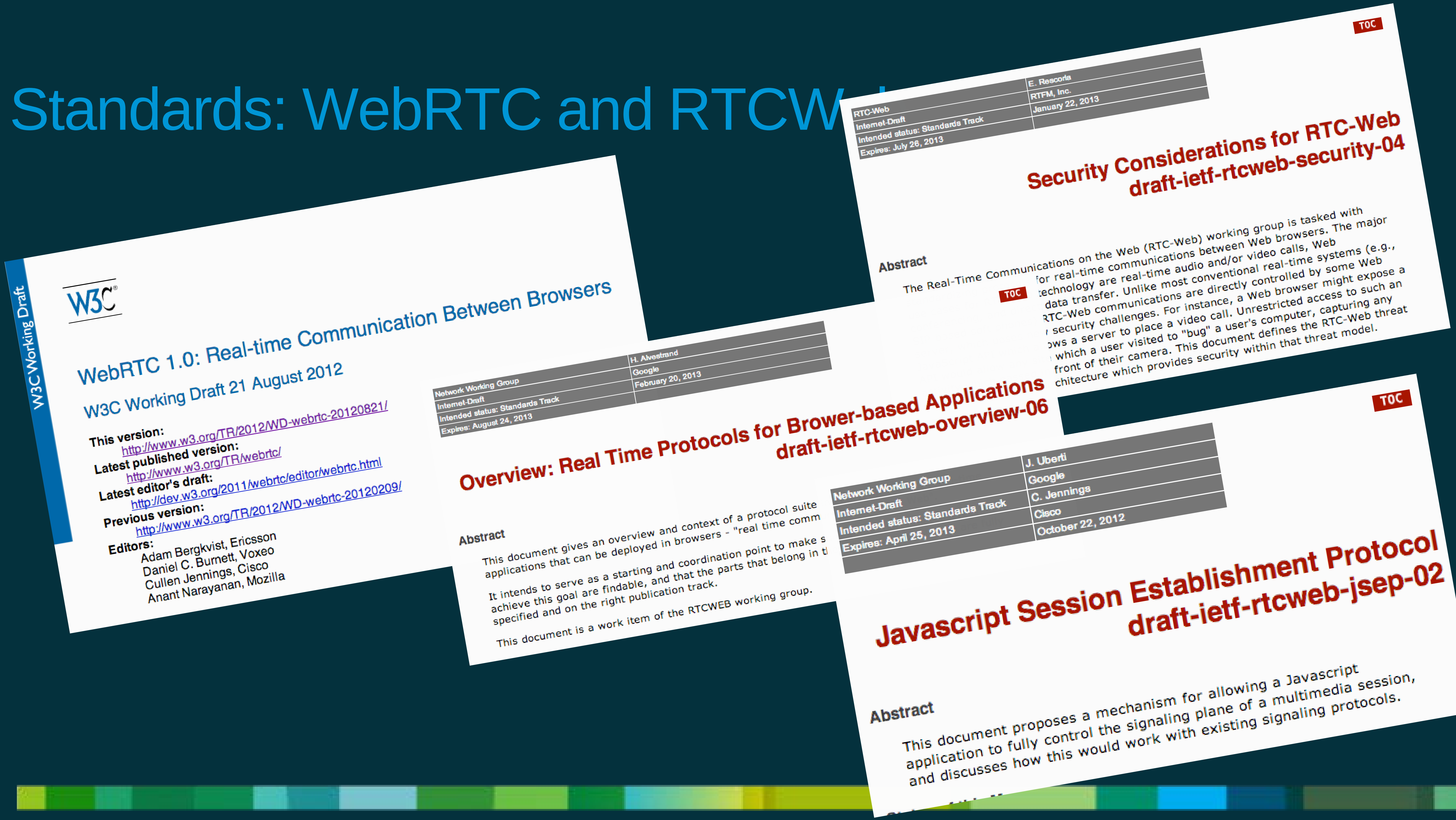


WebRTC, Privacy, TOR, and VPNs

- The WebRTC API allows a webpage to get your IP addresses
  - This includes, public, private, and multi-homed
  - Needed to provide these to the other side to send peer to peer traffic
  - Web servers have always got your public address
- If you run a split tunnel VPN, it reveals both external interfaces
  - If you are in Canada, and have a VPN into the US so you look american to netflix, a netflix web client might be able to figure out that one of your public IPs is in Canada and one is in the US
- If you are using a VPN to hide your location, don't use a split tunnel
  - Many enterprises have a policy against using split VPN

Standards & Implementations

Standards: WebRTC and RTCW



IETF RTCWeb WG

- Main IETF work is done in the RTCWeb working group
- Key documents are
  - draft-ietf-rtcweb-audio
  - draft-ietf-rtcweb-audio-codecs-for-interop
  - draft-ietf-rtcweb-constraints-registry
  - draft-ietf-rtcweb-data-channel
  - draft-ietf-rtcweb-data-protocol
  - draft-ietf-rtcweb-fec
  - draft-ietf-rtcweb-jsep
  - draft-ietf-rtcweb-overview
  - draft-ietf-rtcweb-rtp-usage
  - draft-ietf-rtcweb-stun-consent-freshness-11.txt
  - draft-ietf-rtcweb-transports
  - draft-ietf-rtcweb-use-cases-and-requirements
  - draft-ietf-rtcweb-video

W3C WebRTC WG

- W3C work is done in WebRTC working group
- Key documents are:
  - http://w3c.github.io/webrtc-pc/
  - http://w3c.github.io/mediacapture-main/

Implementations

- Mozilla - Firefox
    - Working implementation with
        - audio / video
        - data channels
    - Ongoing work on evolving standards
- Google - Chrome
    - Working implementation with
        - audio / video
        - data channels
    - Ongoing work on evolving standards

- Apple - Safari
    - Maintaining strict secrecy
- Microsoft - IE
    - Very active in contributing to standards
    - Released a plugin that can provide limited functionality via polyfill
    - Conflicting statements about will do WebRTC 1.1 / will not do SDP

ORTC

- WebRTC always recognized they could do both a high level and low level API
  - Decided to start with high level API and later do low level API
  - Microsoft had desired a low level API first but that proposal was rejected by the WG
- Microsoft formed a community group to push it’s low level API called ORTC
  - this is not a standards forming group
- Once WebRTC 1.0 is done, the WebRTC WG would like to start working on a low level API
  - The low level API would sill keep the high level API as well and become WebRTC 1.1
  - ORTC would be relevant input to this
  - Microsoft has objected to the WG charter update to do this

Ongoing Major Items

- Screen Capture API
- Depth Camera (3D range images)
- Control of coding for video on particular Peer Connection (Adding new JS object)
- Congestion Control
- Recording
- Simulcast Video
- Trickle ICE
- Port reduction with Bundle
- Partial Offer / Answer

Summary

The Power to Create

Ease of Developmen t → **Ease of Deployment** → **Many Devices** →

- No VoIP expertise needed
- Enables huge web developer population
- New applications
- Mashable components
- Cross platform

- Distribution = URL
- Datacenter, not individual devices
- Low maintenance
- Rapid updates

- Click to access
- Any device
- Reduced need for plugins/native apps
- Extends business comm. systems

Digging Deeper

- Read the specifications at :
  - http://w3c.github.io/webrtc-pc/
  - http://w3c.github.io/mediacapture-main/
  - http://tools.ietf.org/wg/rtcweb/
- Read the books:

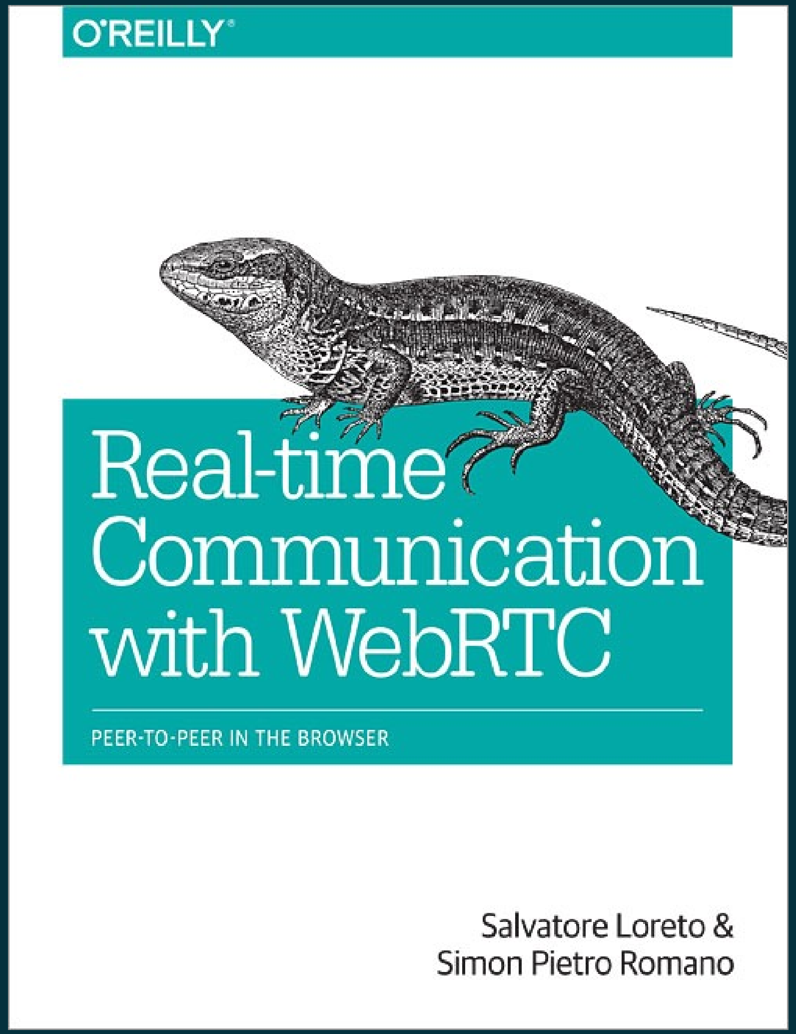
  http://shop.oreilly.com/product/0636920030911.do

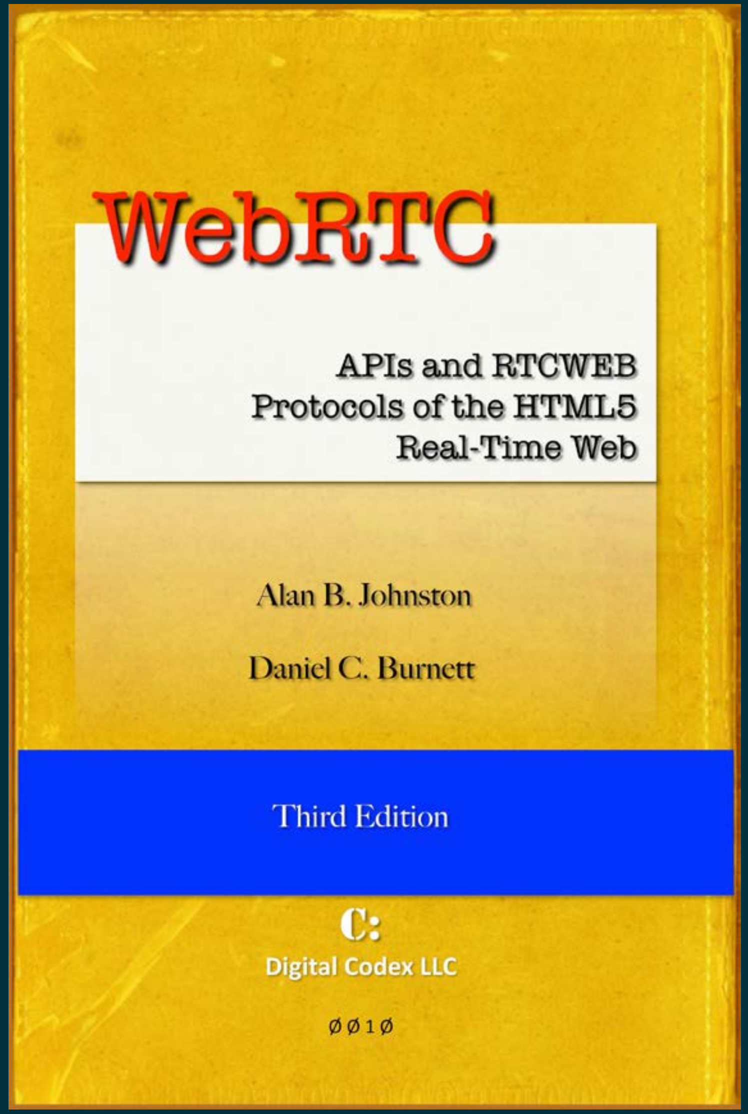
  http://webrtcbook.com/

  (and many more)
- Join the community mailing lists of ISOC supported standards organizations

  W3C: Send email with "subscribe" to public-webrtc-request@w3.org

  IETF: https://www.ietf.org/mailman/listinfo/rtcweb





Credits and Usage

- If you want to use these slides, please contact me at fluffy@cisco.com and I will be happy to get you some slides you can use
- Thanks to many people for contributions to these slides including Eric Rescorla, Ethan Hugg, Suhas Nandakumar, Darin Dunlap and Martin Thomson

Thank you.

CISCO