WebRTC multipoint conferencing

with recording using a Media Server

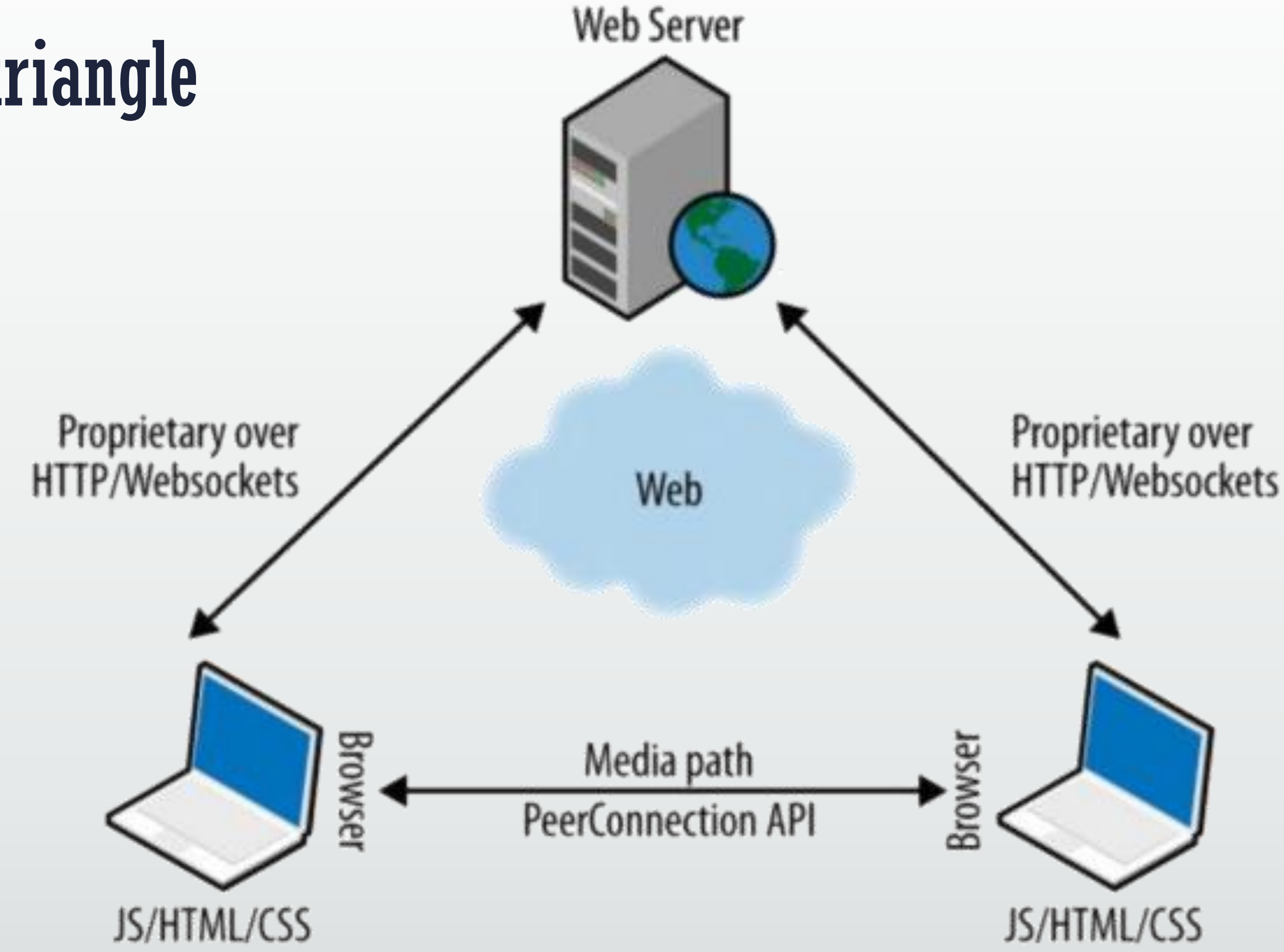
Goal: WebRTC conferencing prototype that

1. allows more than four participants to communicate simultaneously
2. supports recording of conversations
3. allows participants in restrictive network environments to take part in conversations
4. optionally allows SIP-Clients (soft/hard phones), and Teleconferencing systems to connect to a conference

WebRTC introduction

- Peer-to-Peer based communication solution for all devices
- Not dependent on OS, proprietary plugins or other programs
- Most known for video chats between Google Chrome and Mozilla Firefox
- Uses SDP to instantiate sessions
- No signaling method is defined
- Jointly defined by the W3C WebRTC Working Group (browser API) and the IETF RTCWEB Work Group (protocols)

WebRTC triangle



Source: Iya Grigorik, 2013
High-performance browser networking

Important bits – The API

- **RTCPeerConnection**
  main object used by a web application in the browser
  - Create connections between peers
  - Handle SDP offers and answers
  - ICE agent to find and negotiate usable IP addresses and port numbers (ICE candidates)
  - Receive and send media streams
- **media stream**
  requested using the MediaStream API
  - Multiple video and audio tracks
- **RTCDataChannel**
  TCP based, used to send arbitrary non-media data

Device types

- **WebRTC device**
  conforms to the protocol specifications
- **WebRTC browser**
  WebRTC device that also supports the full ECMAScript API.
- **WebRTC gateway**
  WebRTC device that mediates media traffic to non-WebRTC devices and may not conform to all protocol specifications.

UDP based protocols for the peer connection and TCP based ones for the data channel including encryption with TLS and DTLS

Connect WebRTC browser to a multicast endpoint

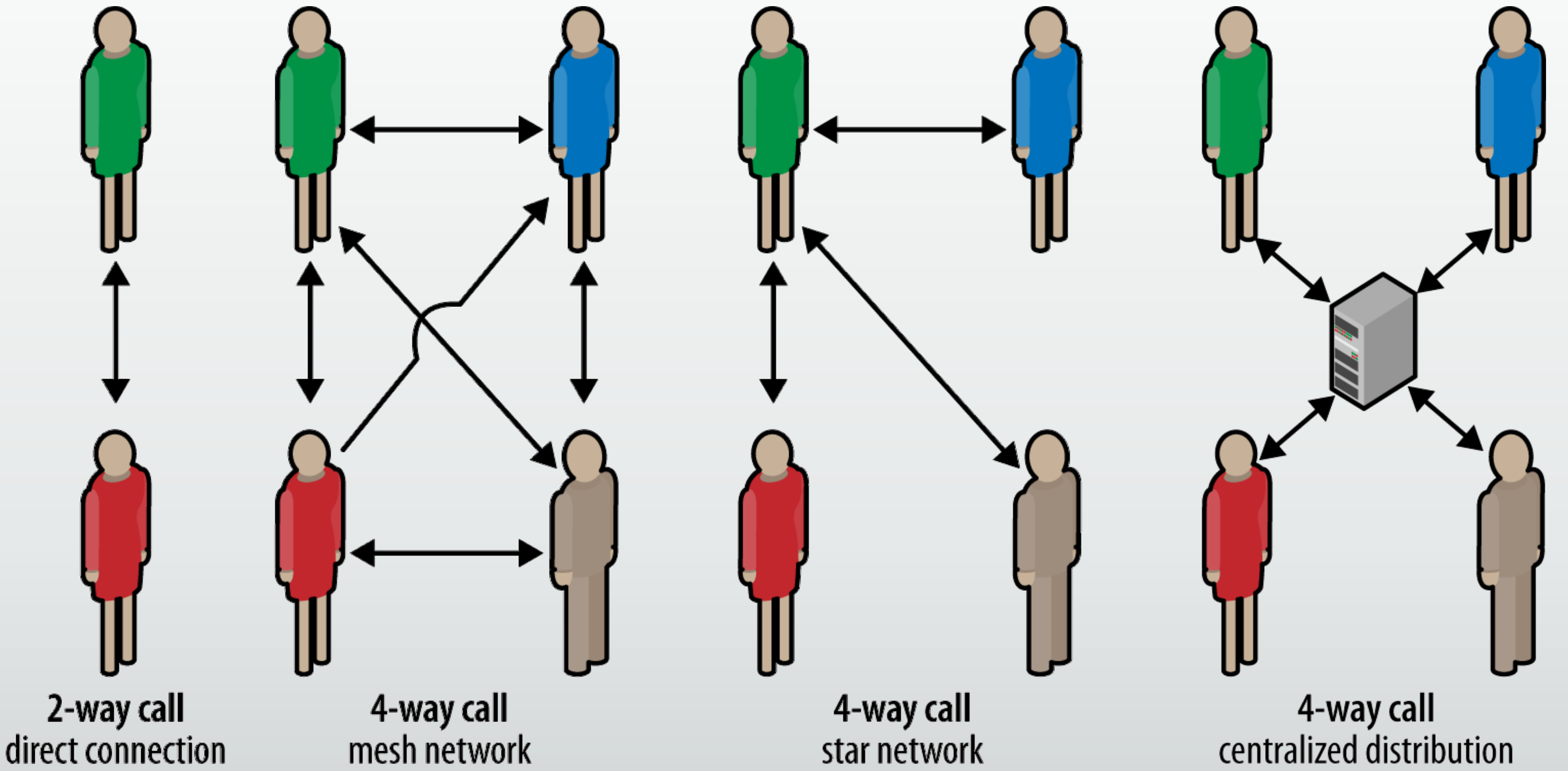
Connect PSTN telephone and WebRTC browser

Goals: WebRTC conferencing prototype that

1. allows more than four participants to communicate simultaneously
2. supports recording of conversations
3. allows participants in restrictive network environments to take part in conversations
4. optionally allows SIP-Clients (soft/hard phones), and Teleconferencing systems to connect to a conference

Multipoint conferencing

Source: Ilya Grigorik, 2013
High-performance browser networking



Who records a conference

conference participant

- High processing power on client
- Distribution complicated

dedicated client

- Additional device needed
- Distribution complicated
- One more connection for every participant

Recording with a centralized entity

Each participant records himself

Uploads recording to center

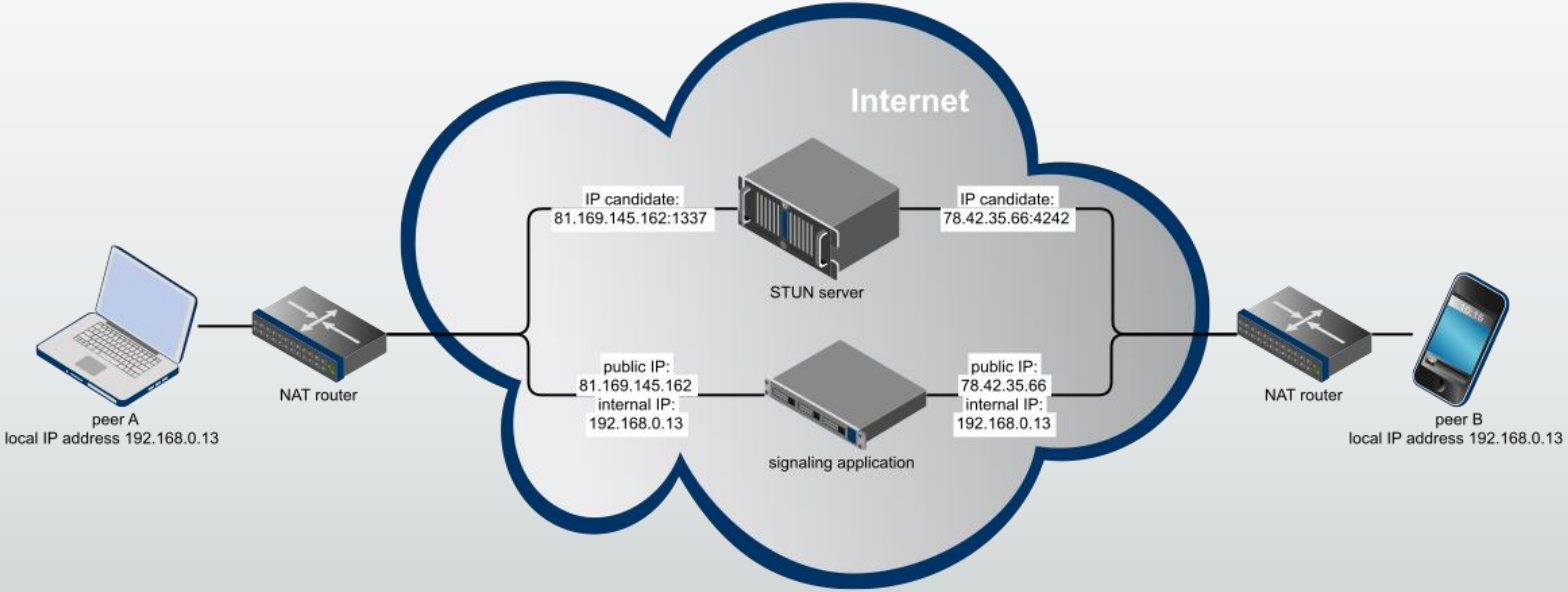
Entity distributes streams to participants

Stores the streams

Restrictive network environments

- To establish a direct connection, a publicly accessible IP address is needed
- ICE (Interactive Connectivity Establishment)
  - STUN (Session Traversal Utilities for NAT) UDP, TCP, http, https
  - TURN (Traversal using Relays around NAT)

Restrictive network environments



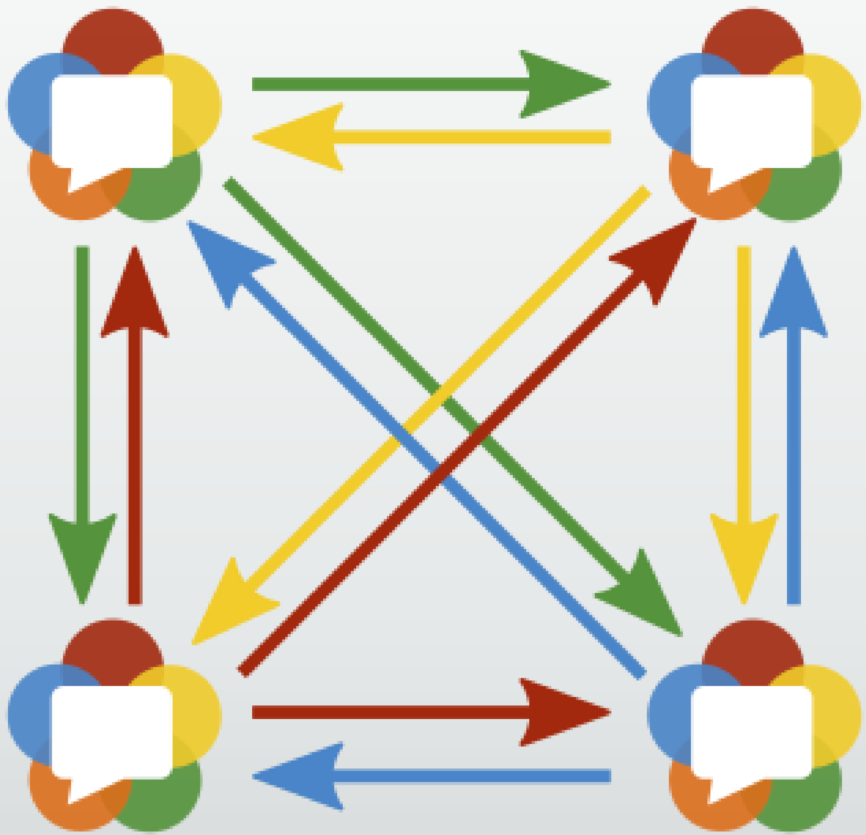
Connecting SIP devices

- Session Instantiation Protocol
- Text based protocol
- Uses SDP to describe media streams (and their initialization parameters)
- No mandatory codecs – transcoding is needed
- PSTN telephone behaves the same as a teleconferencing solution with four cameras and microphones

Media Server

Comparison of multipoint architectures

Participant in a full mesh conference

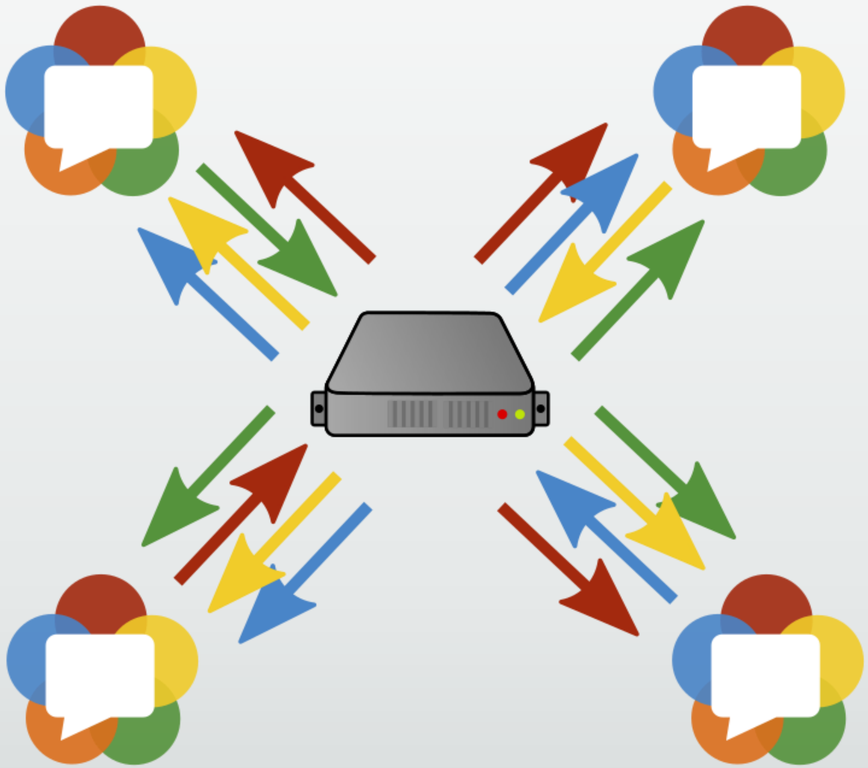


- 4 Participants
- Each participant
  - 3 uploads
  - 3 downloads
  - 3 Mbit/s bandwidth each up- and downstream for VGA video

7 participants:
Bandwidth of 12Mbit/s

WLAN 802.11b
does not suffice

Participant in a relayed conference

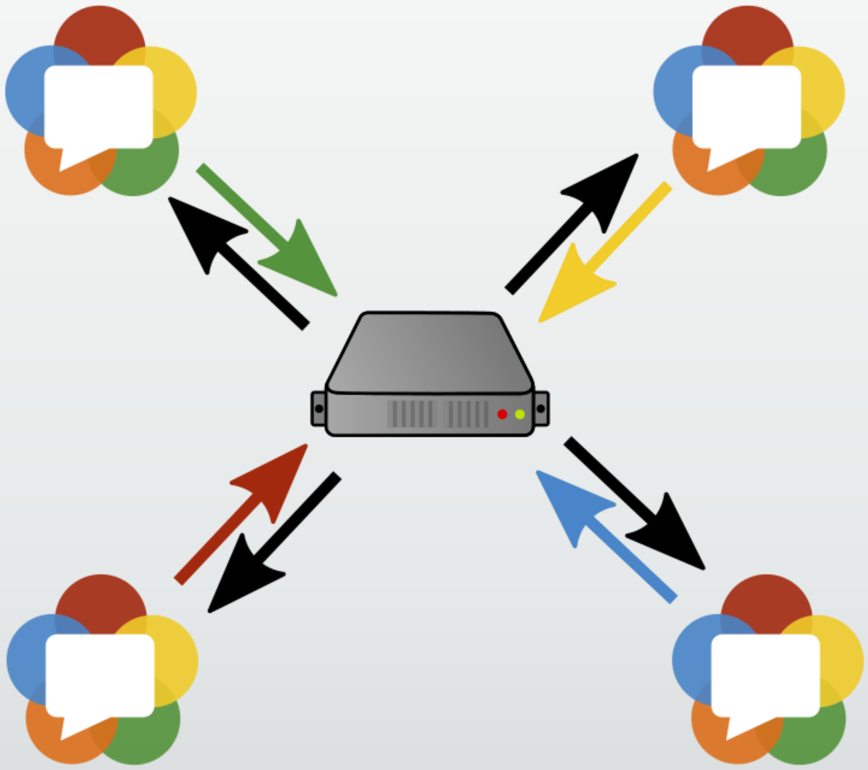


- 4 Participants
- Each participant
  - 1 upload
  - 3 downloads
  - 4 Mbit/s bandwidth for VGA video

10 participants:
Bandwidth of 18.4Mbit/s

WLAN 802.11g
needed

Participant in a mixed conference



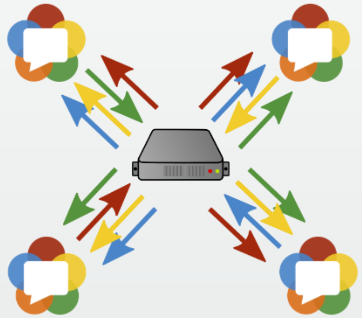
- 4 Participants
- Each participant
  - 1 upload
  - 1 download
  - 2 Mbit/s bandwidth for VGA video

25 participants:
Bandwidth of 2 Mbit/s

Might work on 3G

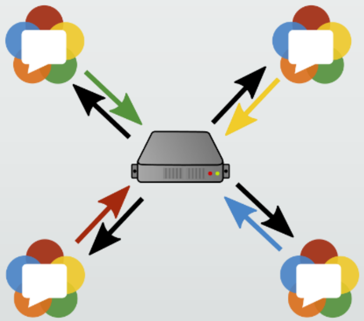
Media Server

- Relayed conference
  - 2 participants: 4.1 Mbit/s needed
  - 4 participipants: 16.4 Mbit/s needed
  - 7 participants: 50.2 Mbit/s needed
  - 10 participants: 102.4 Mbit/s needed



High strain on network device

- Mixed conference
  - 2 participants: 4.1 Mbit/s needed
  - 4 participants: 8.2 Mbit/s needed
  - 7 participants: 14.3 Mbit/s needed
  - 10 participants: 20.5 Mbit/s needed



High processing power needed

Bandwidth recommendations (at 15fps)

Entry	Resolution	Recommended bandwidth	Minor hickups (1 freeze per minute)	Maximum used bandwidth
QCIF	176x144	384kbit/s	200kbit/s	700kbit/s
CIF	352x288	700kbit/s	384kbit/s	2000kbit/s
VGA	640x480	1024kbit/s	512kbit/s	2100kbit/s
HD 720p	1280x720	1900kbit/s	1024kbit/s	2500kbit/s
HD 1080p	1920x1080	-	-	-

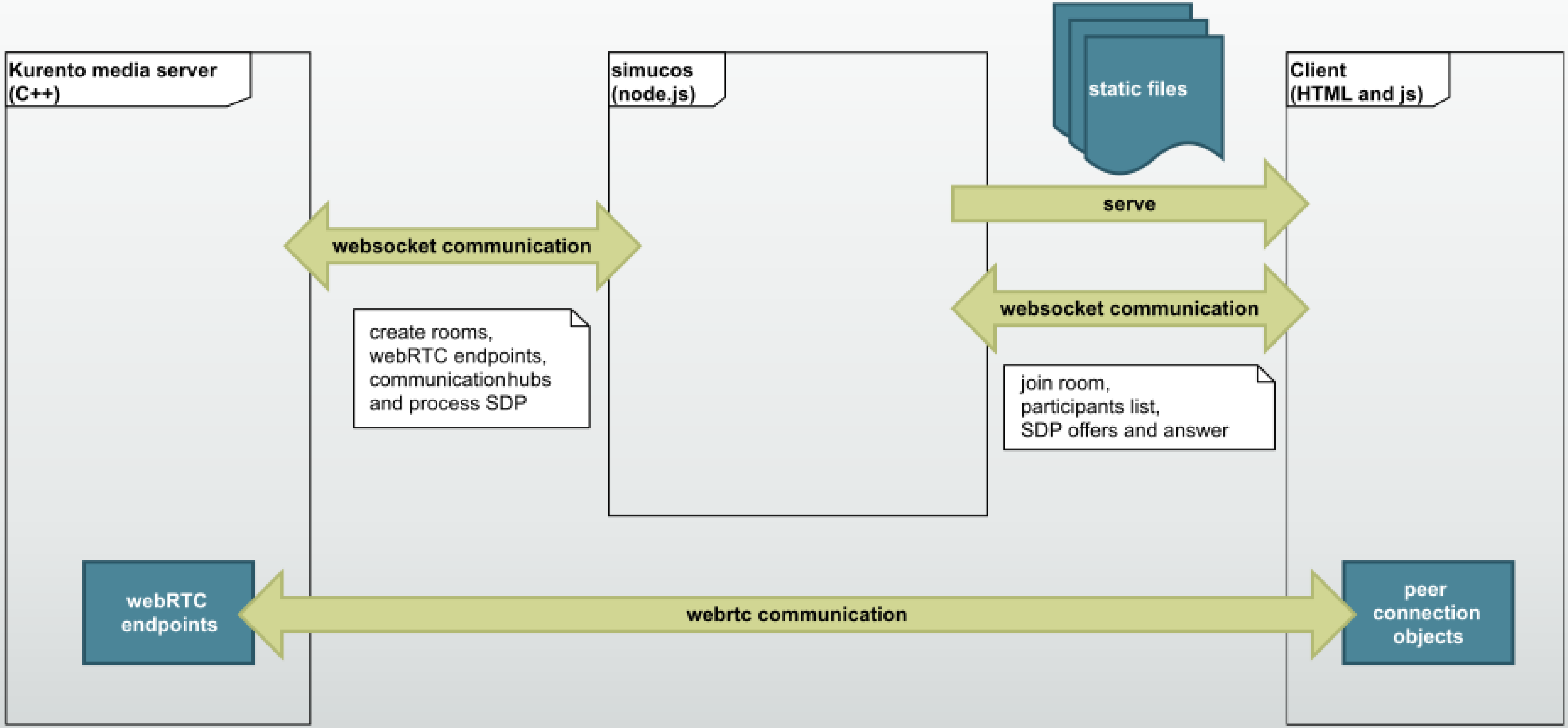
One stream was tested on a virtual network device that was limited using the Network Emulator Toolkit

Open Source Media Servers

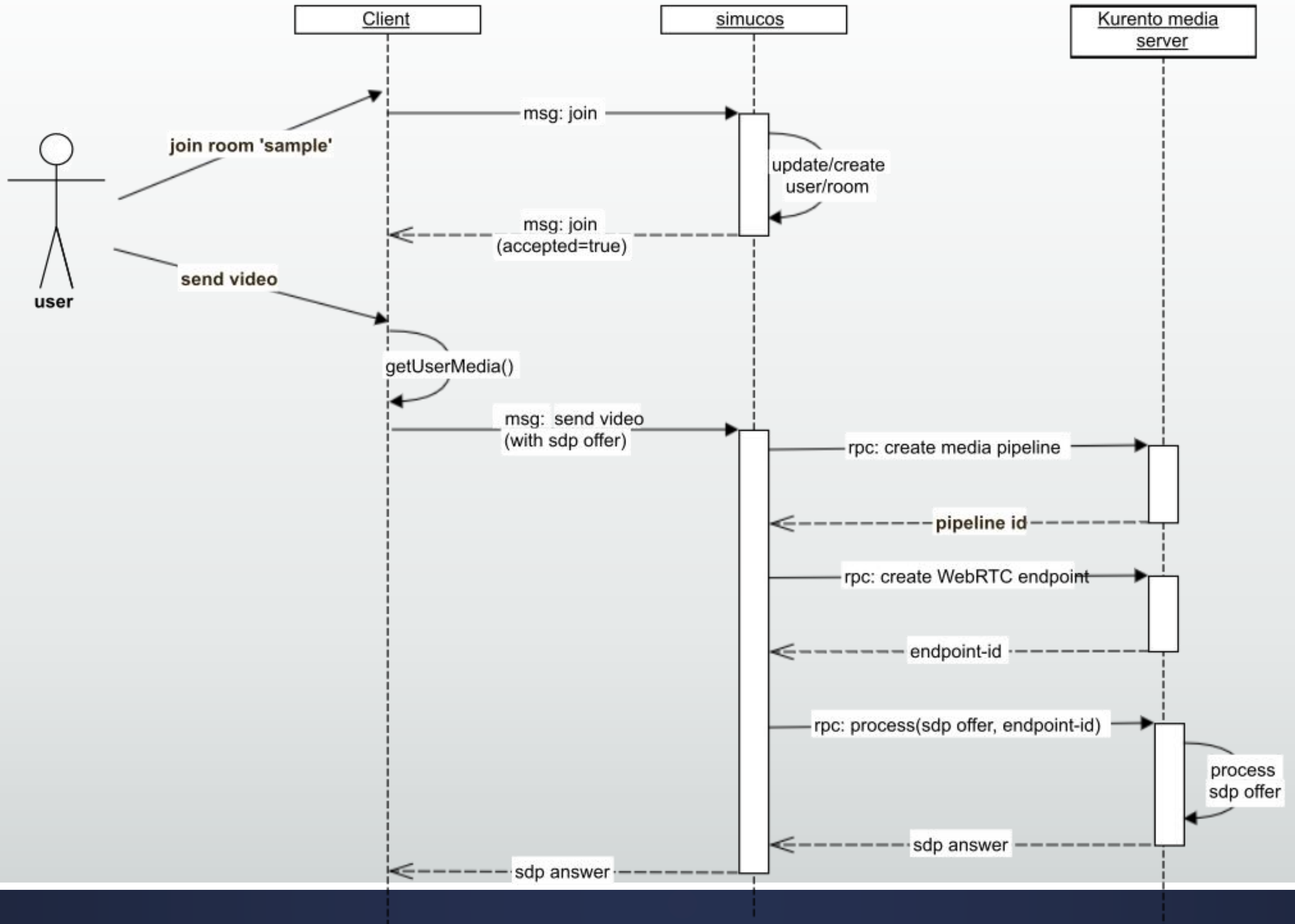
Kurento http://www.kurento.org/ (LGPL)

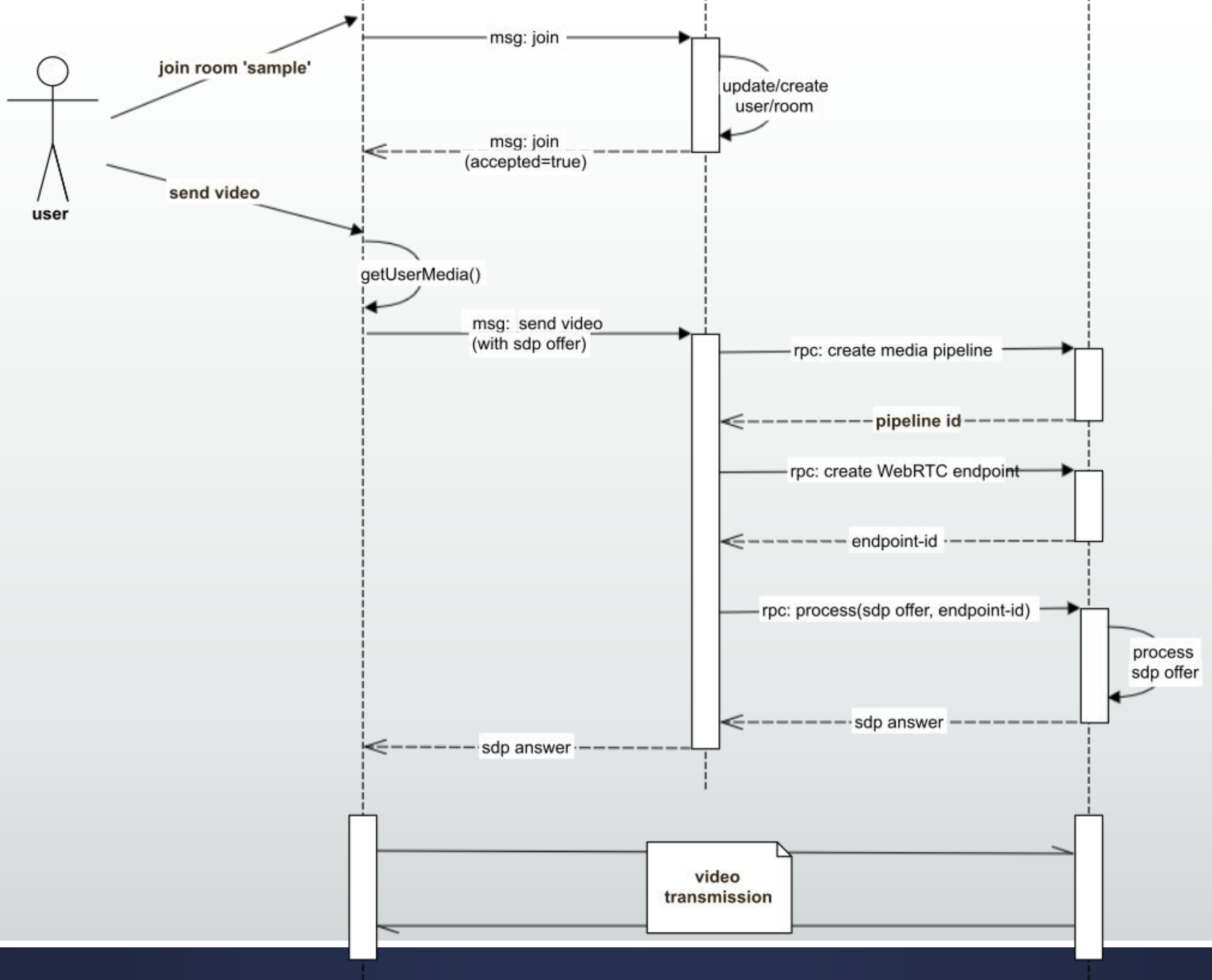
- **Doubango** https://code.google.com/p/telepresence/ (GPL)
- **Jitsi** https://jitsi.org/Projects/JitsiVideobridge (MIT)
- **Licode** http://lynckia.com/licode/ (MIT)

Architecture



Example: Start a conversation




user
join room 'sample'
msg: join
update/create
user/room
msg: join
(accepted=true)
send video
getUserMedia()
msg: send video
(with sdp offer)
rpc: create media pipeline
pipeline id
rpc: create WebRTC endpoint
endpoint-id
rpc: process(sdp offer, endpoint-id)
process
sdp offer
sdp answer
sdp answer
video
transmission

Goals

Most of the goals for this thesis were achieved with the simucos prototype:

- Conferences can be recorded on the media server as VP8 or h.264 video files.
- Many clients may participate in a conference, the highest number tested was 25 in a mixed conference, and 14 in a relayed conference.
- Participants in restrictive network environments can connect to a conference using a STUN or TURN server.
- Only the optional goal to add support for SIP clients into simucos was not implemented due to the chosen Node.js architecture.

Conclusion and prospect

- WebRTC is a mature technology
- WebRTC has disruptive qualities:
  - Solutions are achievable that would need more experience, more man-power and more time with other communication solutions

Future ideas

- Switching a peer-to-peer conference to a media server
- Hybrid architecture approach
  - Lessen strain on media server by using peer-to-peer meshes
- Dynamic architecture approach
  - Develop algorithms to optimize the hybrid architecture approach
- Peer-to-peer broadcasting of one presenter
  - Tree based structure
  - Ring structure

Bright future for WebRTC

- New features in WebRTC 1.0
  - Unified statistics API
  - Promises instead of callbacks
- Object RTC (ORTC)
  - Effort to create the next version of WebRTC (backed by Microsoft and Google)
  - Exchange text-based SDP with a JS object model
  - Developers gain access to lower-level functions (e.g. Codec settings per track)
  - More powerful and flexible JavaScript API

www.mikogo.com

Questions?

Thesis available at:
https://github.com/marc136/thesis

Email: mwalter@mikogo.com

More Time?

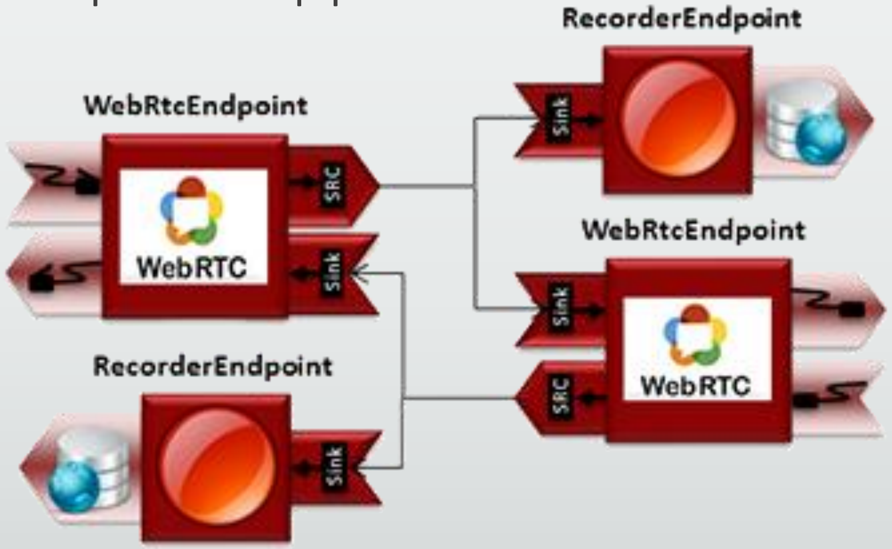
Some other stuff...

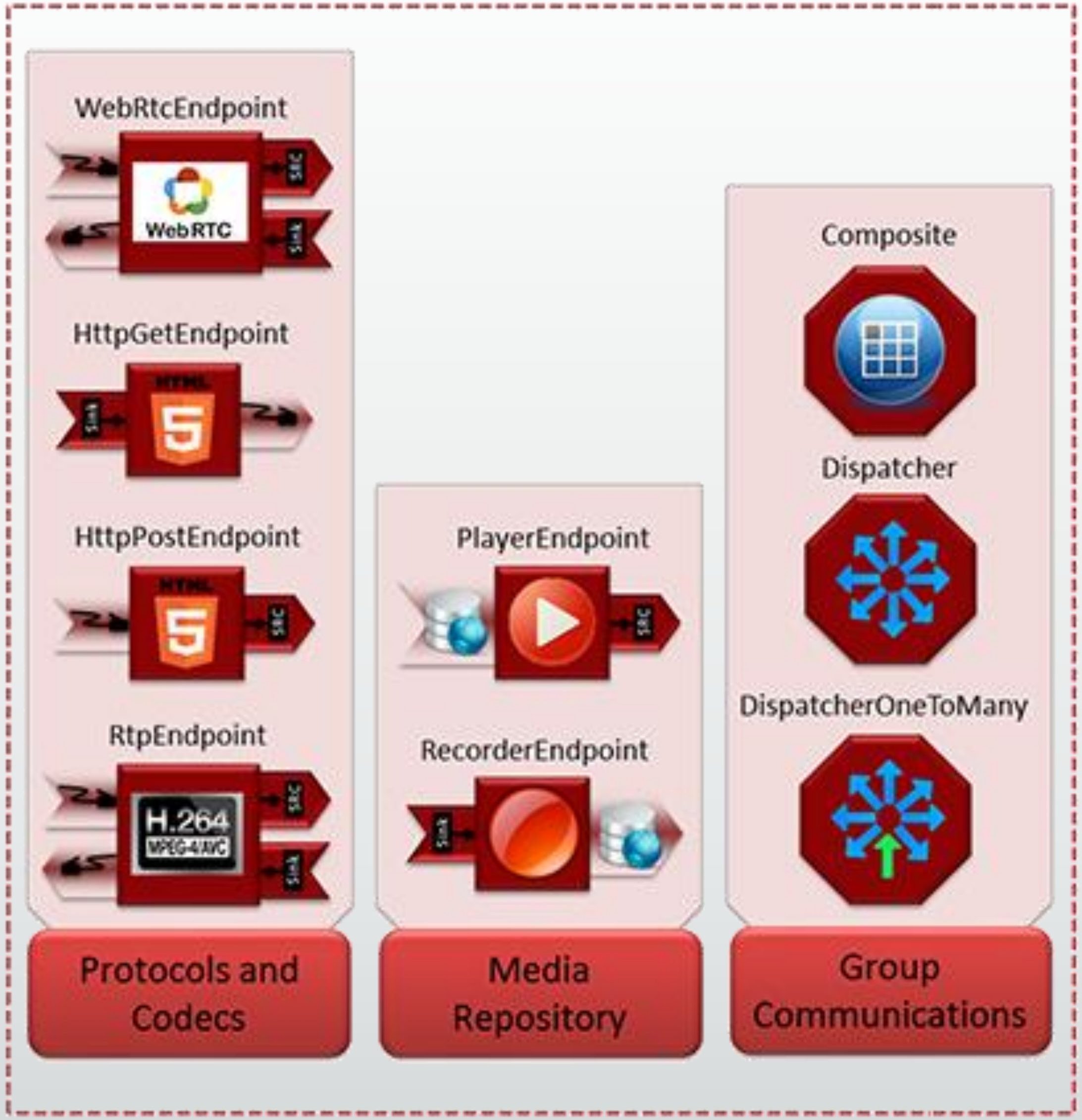
Image sources

- Title page:
  - Top left: http://lifestyle.beiruting.com/wp-content/uploads/2011/09/business-man-resume.jpg
  - Top center: http://www.jenningswire.com//wp-content/uploads/2014/01/Businessman1.jpg
  - Top right: http://bizblog.cosmobc.com/files/2015/03/Business-Man.jpg
  - Bottom left: http://ak3.picdn.net/shutterstock/videos/4885424/preview/stock-footage-businessman-enjoying-book-on-digital-ebook-reader-device-on-train-journey.jpg
  - Bottom center: http://www.careerealism.com/wp-content/uploads/2011/11/Mature-Businessman-Thinking.jpg
  - Bottom right: http://thumbs.dreamstime.com/z/picture-confused-woman-smartphone-bright-32589001.jpg
- 28: http://www.kurento.org/docs/current/introducing_kurento.html

Kurento Media Server

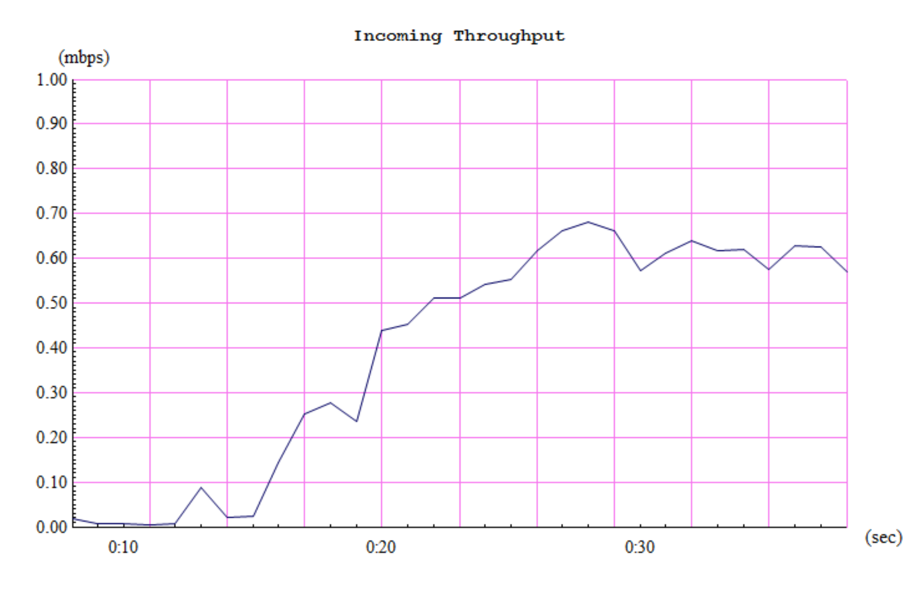
Sample media pipeline

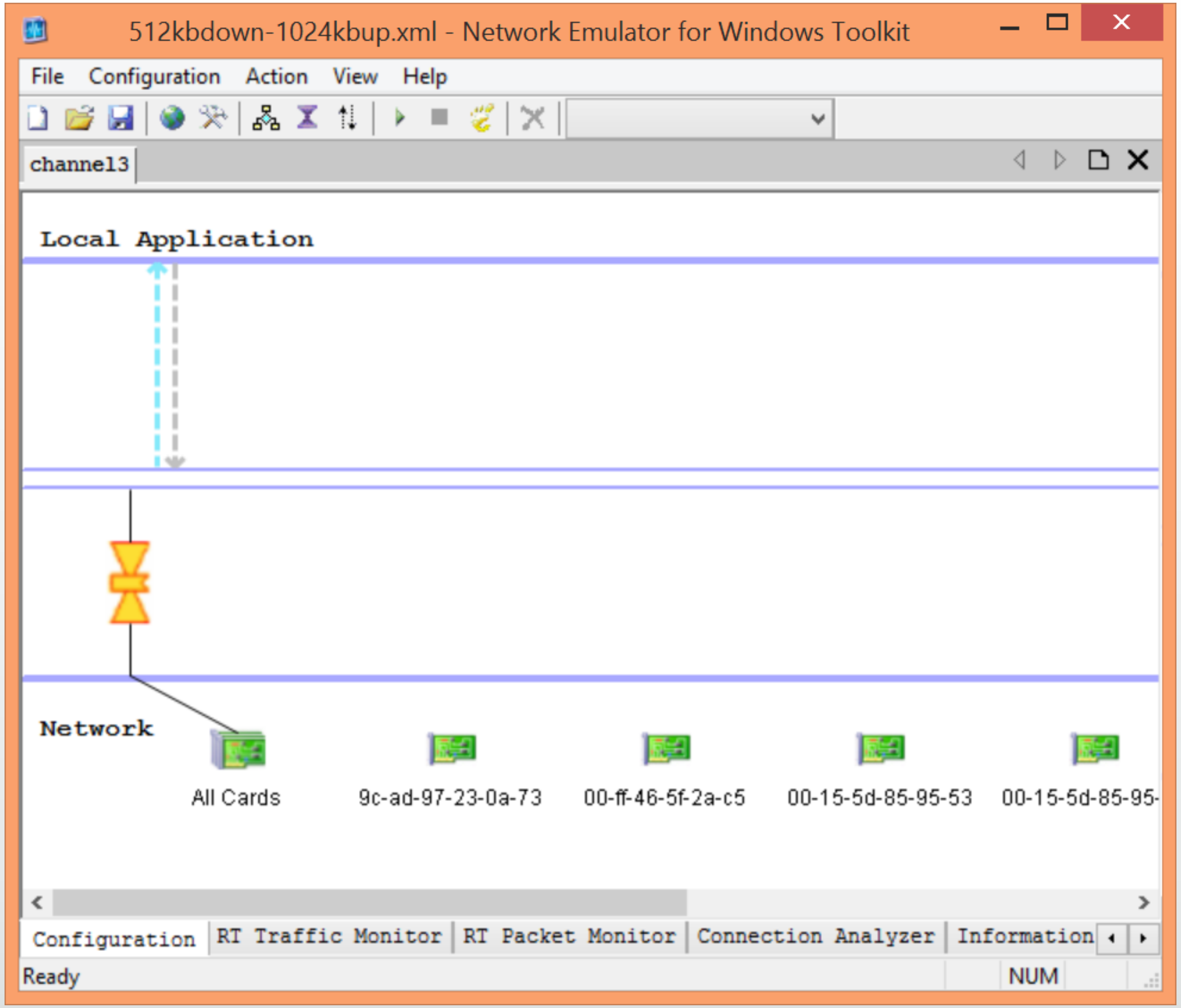




Network Emulation Toolkit (NEWT)

- Limit bandwidth
- Simulate high traffic
- Simulate packet loss

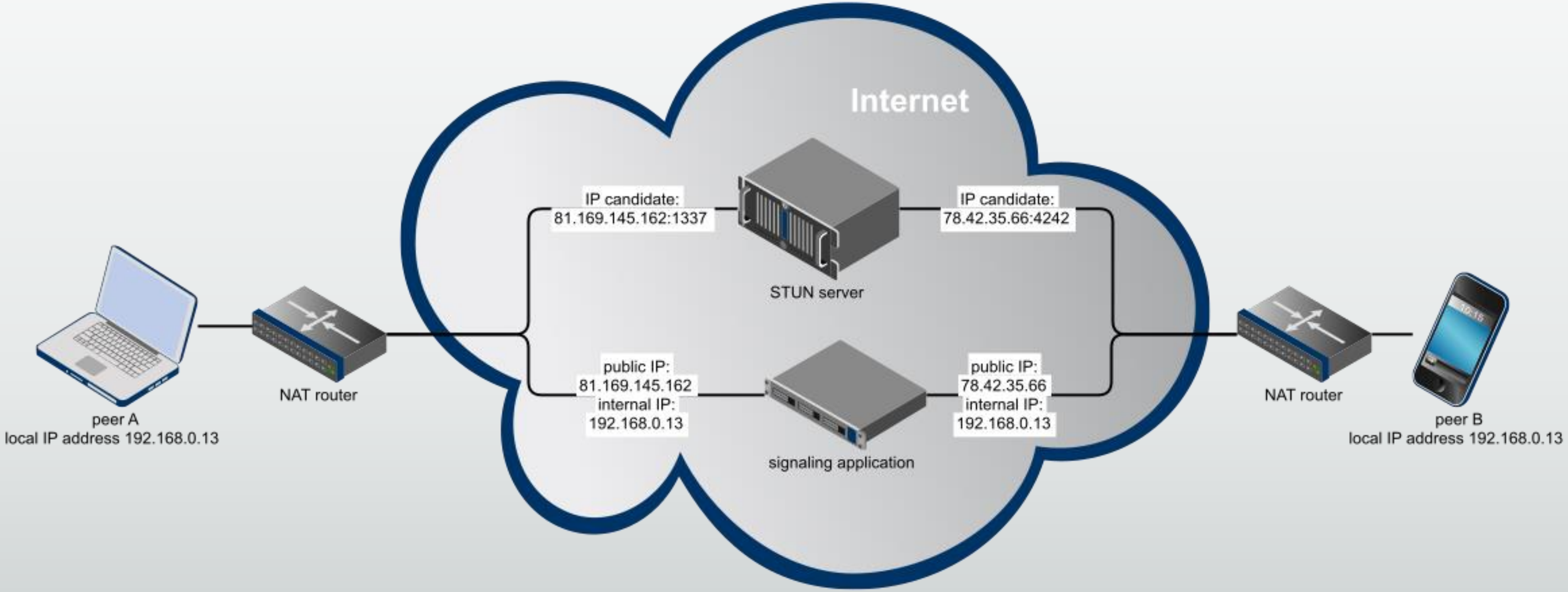




http://blogs.msdn.com/b/lkruger/archive/2009/06/08/introducing-true-network-emulation-in-visual-studio-2010.aspx

https://blog.mrpol.nl/2010/01/14/network-emulator-toolkit/

ICE Candidate Gathering



Symmetric NAT