WEBRTC,
MOBILE CONSIDERATIONS
AND VOICE OVER IP

IETF
W3C
Google
Microsoft
Apple
...

open-source

WebRTC 1.0

2011

2017

RFC7675 RFC5888

RFC7635 RFC2119 RFC6236

RFC7656 RFC3264 RFC3986

RFC5389 RFC7874 RFC6464

RFC7515 RFC5245

RFC7065 RFC7064

RFC4572 RFC4566

RFC3550 RFC5761

RFC6749 RFC6544 RFC6465


RFC7675
RFC5888
RFC7635
RFC2119
RFC6236
RFC7656
RFC3264
RFC3986
RFC5389
RFC4572
RFC4566
RFC3550
RFC5761
RFC6749
RFC6544
RFC6465

WebRTC (Real-Time Communications)

- Acquiring audio and video
- Communicating audio and video
- Communicating arbitrary data

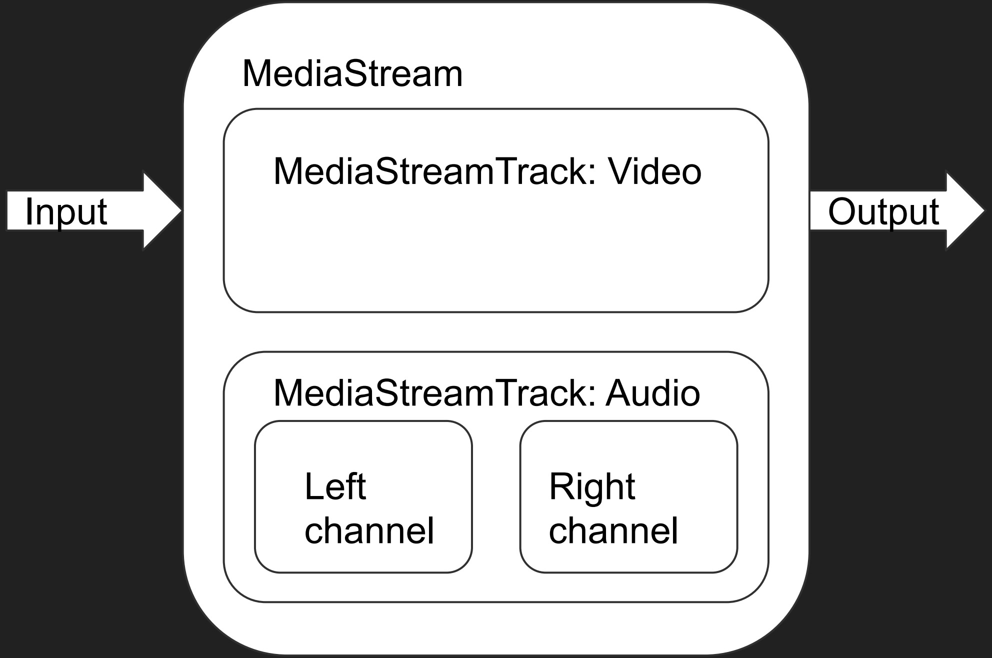
WebRTC (Real-Time Communications)

Acquiring audio and video	MediaStream (aka getUserMedia)
Communicating audio and video	RTCPeerConnection
Communicating arbitrary data	RTCDataChannel

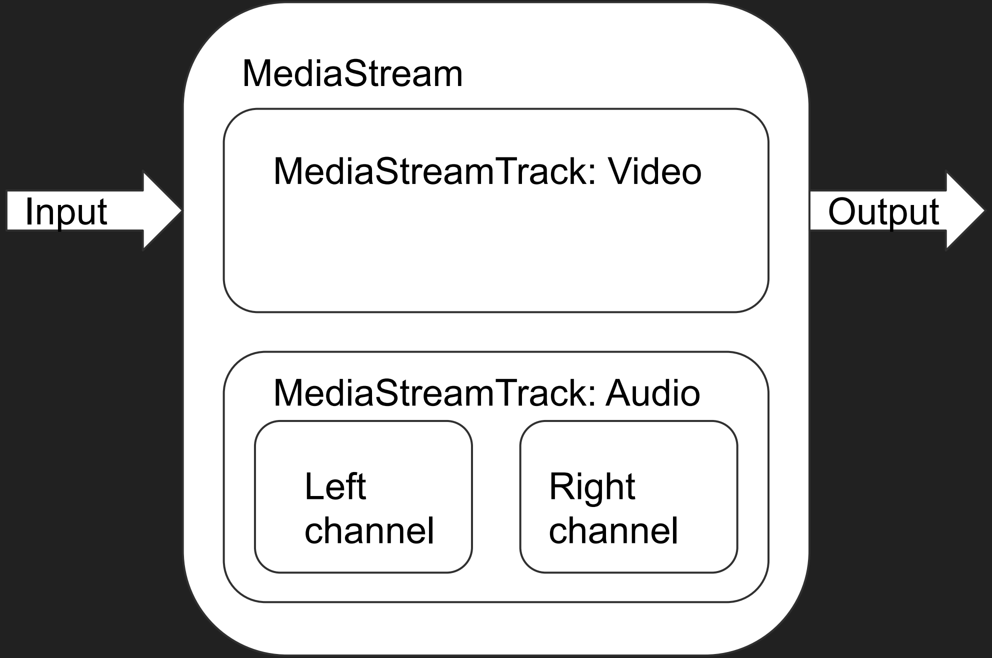
WebRTC (Real-Time Communications)

- Acquiring audio and video — MediaStream (aka getUserMedia)
- Communicating audio and video — RTCPeerConnection
- ~~Communicating arbitrary data~~ — ~~RTCDataChannel~~

MediaStream



MediaStream



Constraints

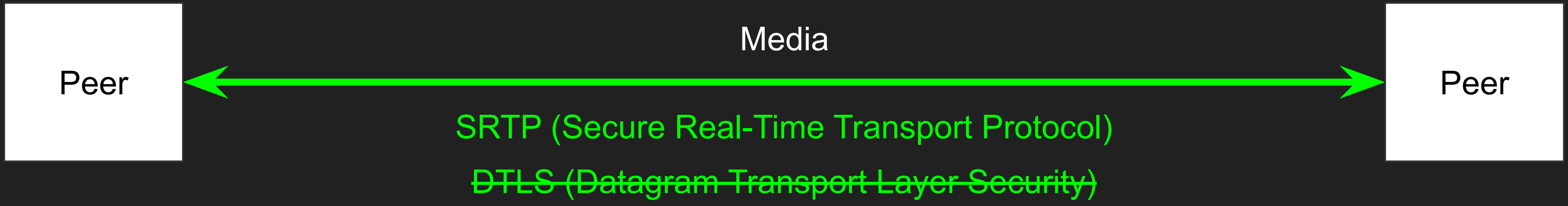
- Media Type
- Resolution
- Frame rate

RTCPeerConnection

- Signal processing
- Codec handling
- Peer-to-peer connection
- Security (Encryption)
- Bandwidth management

RTCPeerConnection

- Signal processing
- Codec handling
- Peer-to-peer connection
- Security (Encryption)
- Bandwidth management



Signalling

- Exchange Session Description Object
    - Codec to use
    - Security keys
    - Network information
- Any messaging mechanism (HTTPS, Websockets, XHR, ...)
- Any messaging protocol (SIP, XMTP, JSON, ...)

Call

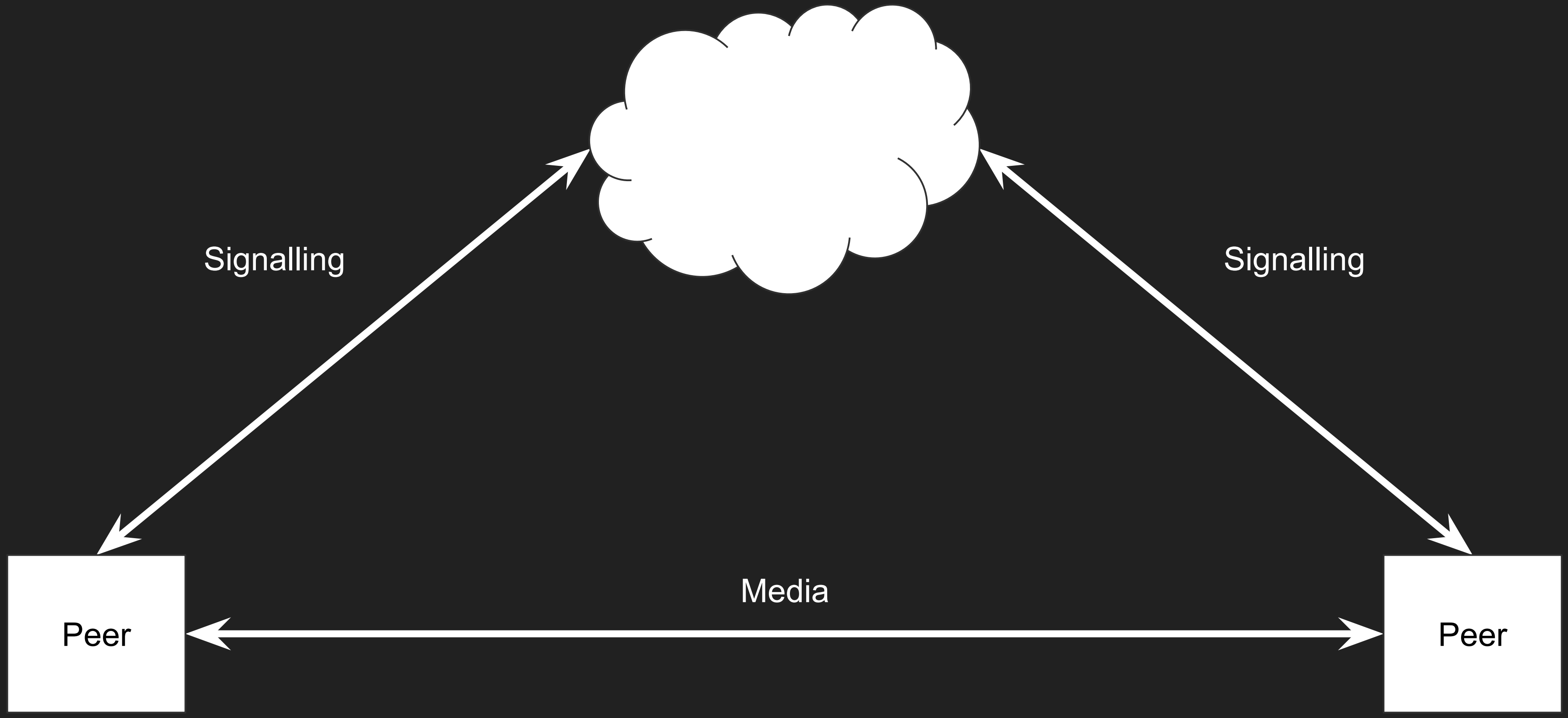
Show/Hide | List Operations | Expand Operations

GET	/sessions/{sessionId}/calls	Retrieve calls
POST	/sessions/{sessionId}/calls	Create a new call
DELETE	/sessions/{sessionId}/calls/{callId}	Call hangup
GET	/sessions/{sessionId}/calls/{callId}	Retrieves the call information
POST	/sessions/{sessionId}/calls/{callId}/answer	Call answer
POST	/sessions/{sessionId}/calls/{callId}/ringing	Send ringing notification
POST	/sessions/{sessionId}/calls/{callId}/reject	Reject the call
POST	/sessions/{sessionId}/calls/{callId}/mute	Mutes the call
POST	/sessions/{sessionId}/calls/{callId}/unmute	Unmutes the call
POST	/sessions/{sessionId}/calls/{callId}/hold	Places the call on hold
POST	/sessions/{sessionId}/calls/{callId}/unhold	The call is no longer on hold
PUT	/sessions/{sessionId}/calls/{callId}/offer	The call is modified
POST	/sessions/{sessionId}/calls/{callId}/dtmf	The dtmf tone is sent
POST	/sessions/{sessionId}/calls/{callId}/merge	Merge two ongoing calls
POST	/sessions/{sessionId}/calls/{callId}/warm-transfer	Warm transfer two calls
POST	/sessions/{sessionId}/calls/{callId}/blind-transfer	Blind transfer the call

RTCSessionDescription (SDP)

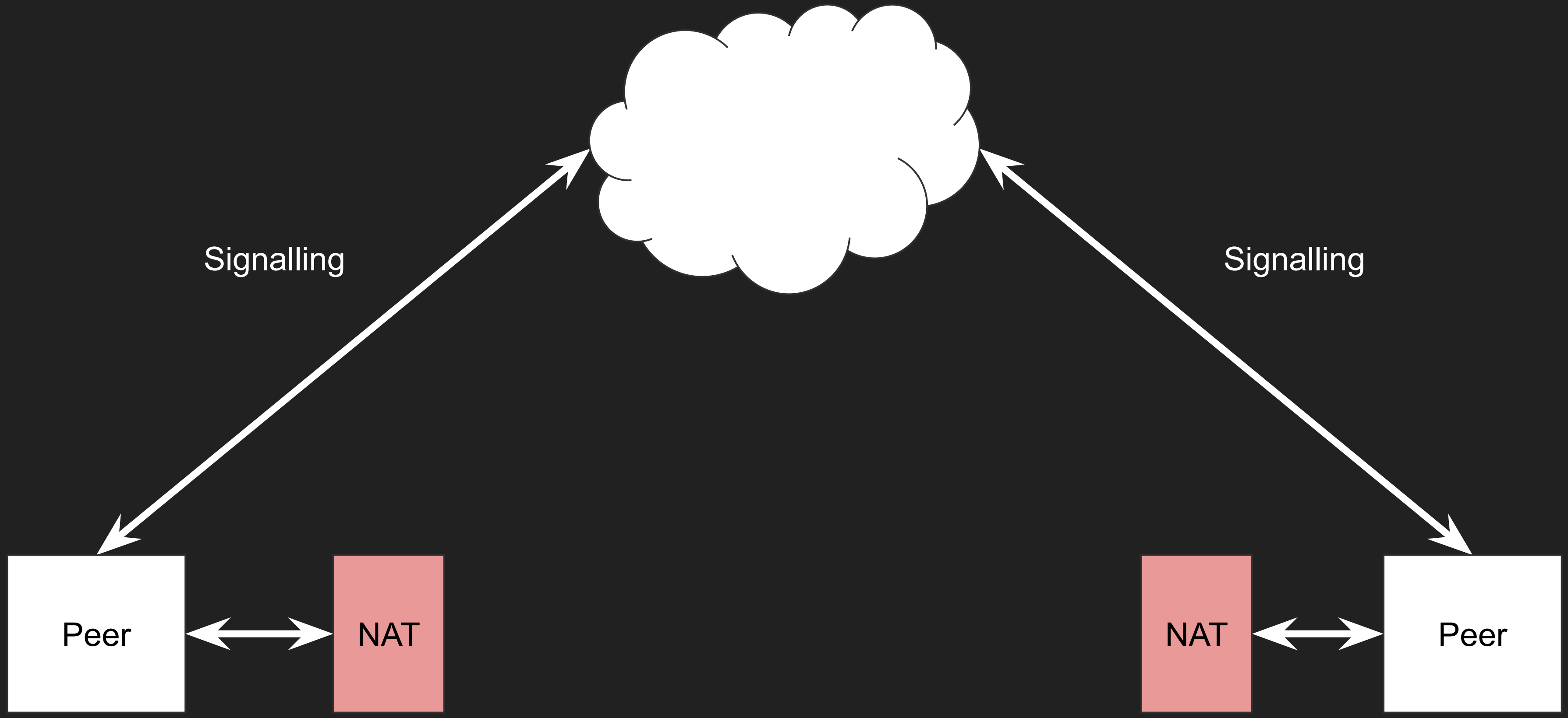
```
[OFFER]
v=0
o=alice 2890844526 2890844526 IN IP4 host...
s=
c=IN IP4 host.atlanta.example.com
t=0 0
m=audio 49170 RTP/AVP 0 8 97
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
m=video 51372 RTP/AVP 31 32
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
```

```
[ANSWER]
v=0
o=bob 2808844564 2808844564 IN IP4 host…
s=
c=IN IP4 host.biloxi.example.com
t=0 0
m=audio 49174 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 49170 RTP/AVP 32
a=rtpmap:32 MPV/90000
```


Signalling
Signalling
Media
Peer
Peer

NAT (network address translation)

- Let multiple computers share the same IP address
- IPv4 address exhaustion

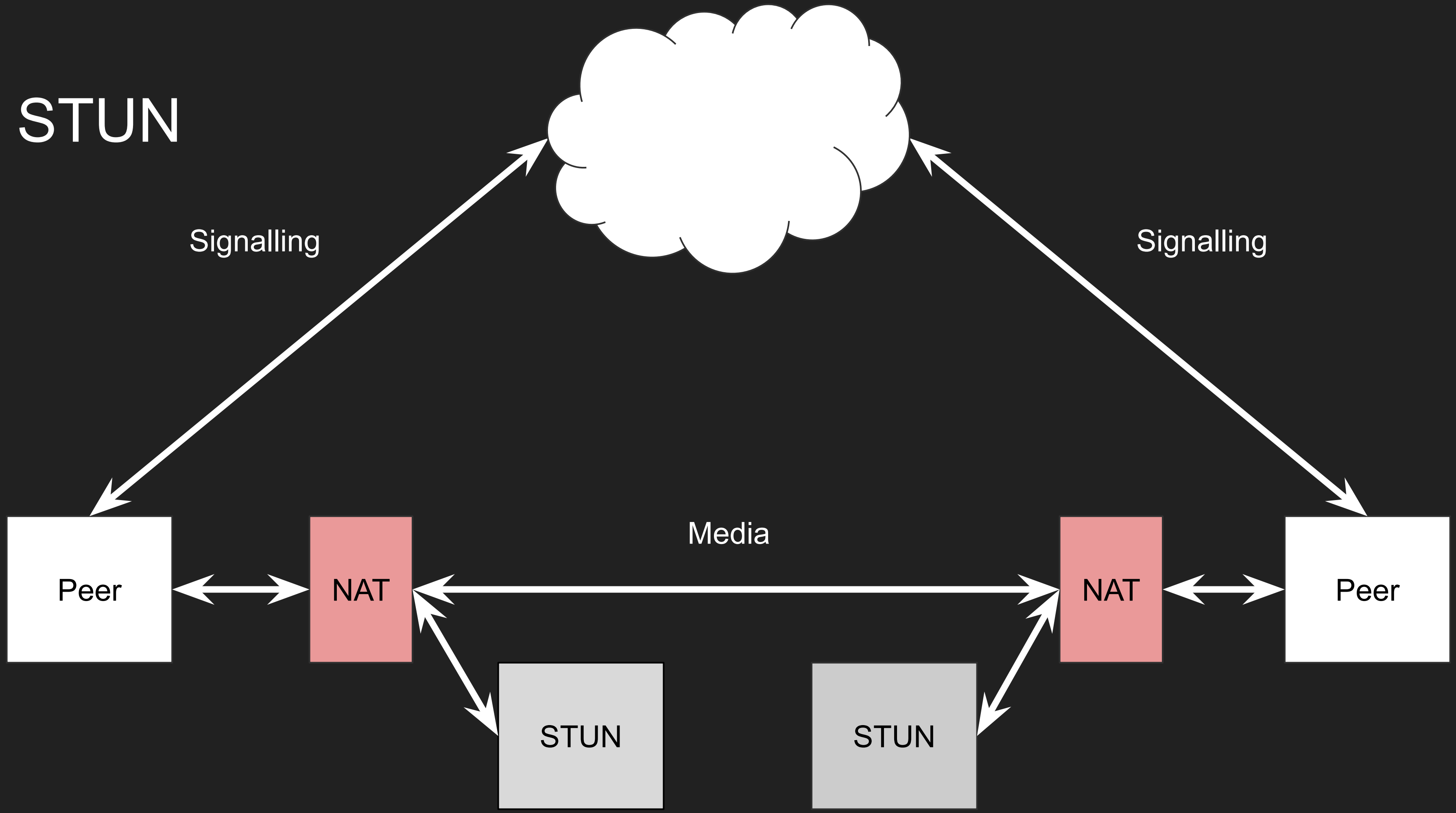


Signalling
Signalling
Peer
NAT
NAT
Peer

STUN (session traversal utilities for NAT)

- What is my IP address?
- Simple server
- CHEAP

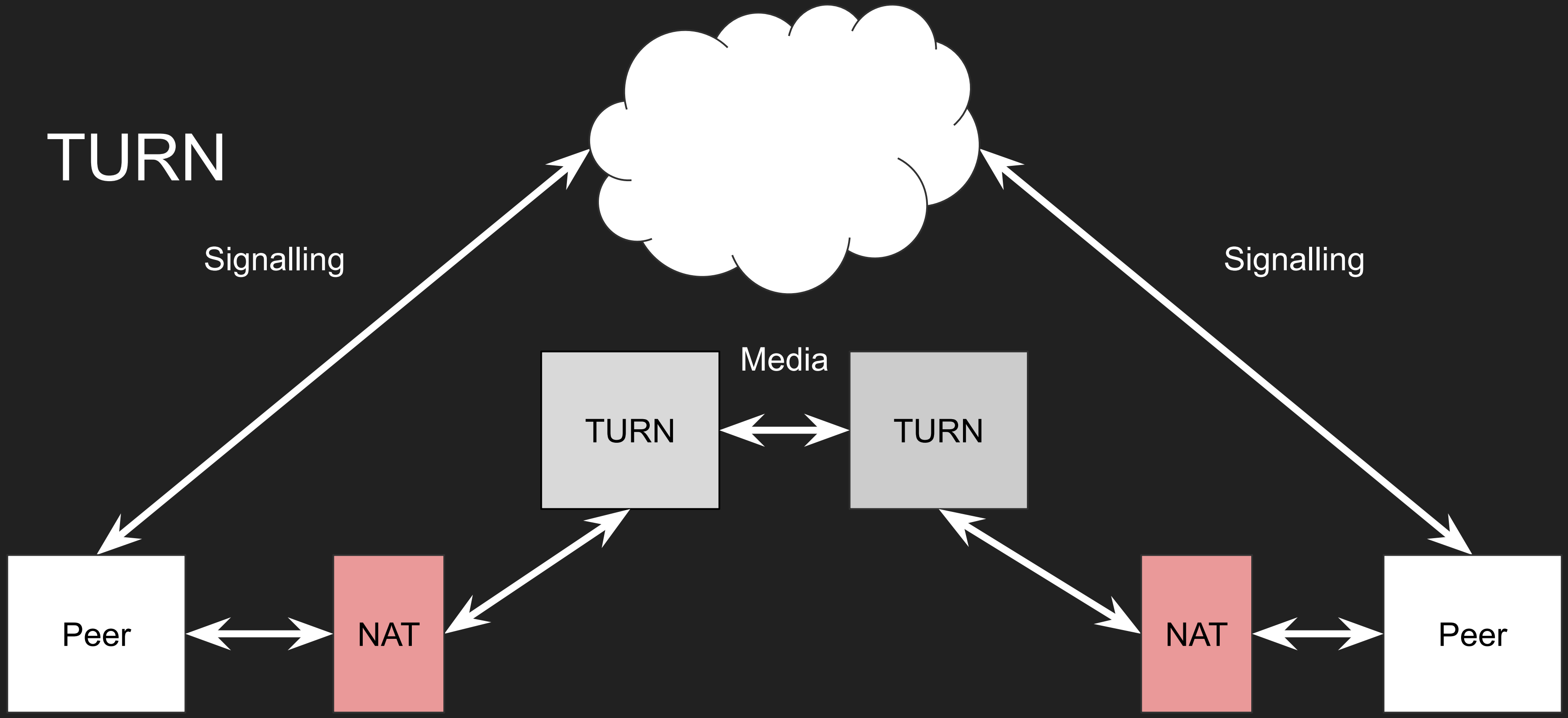
STUN



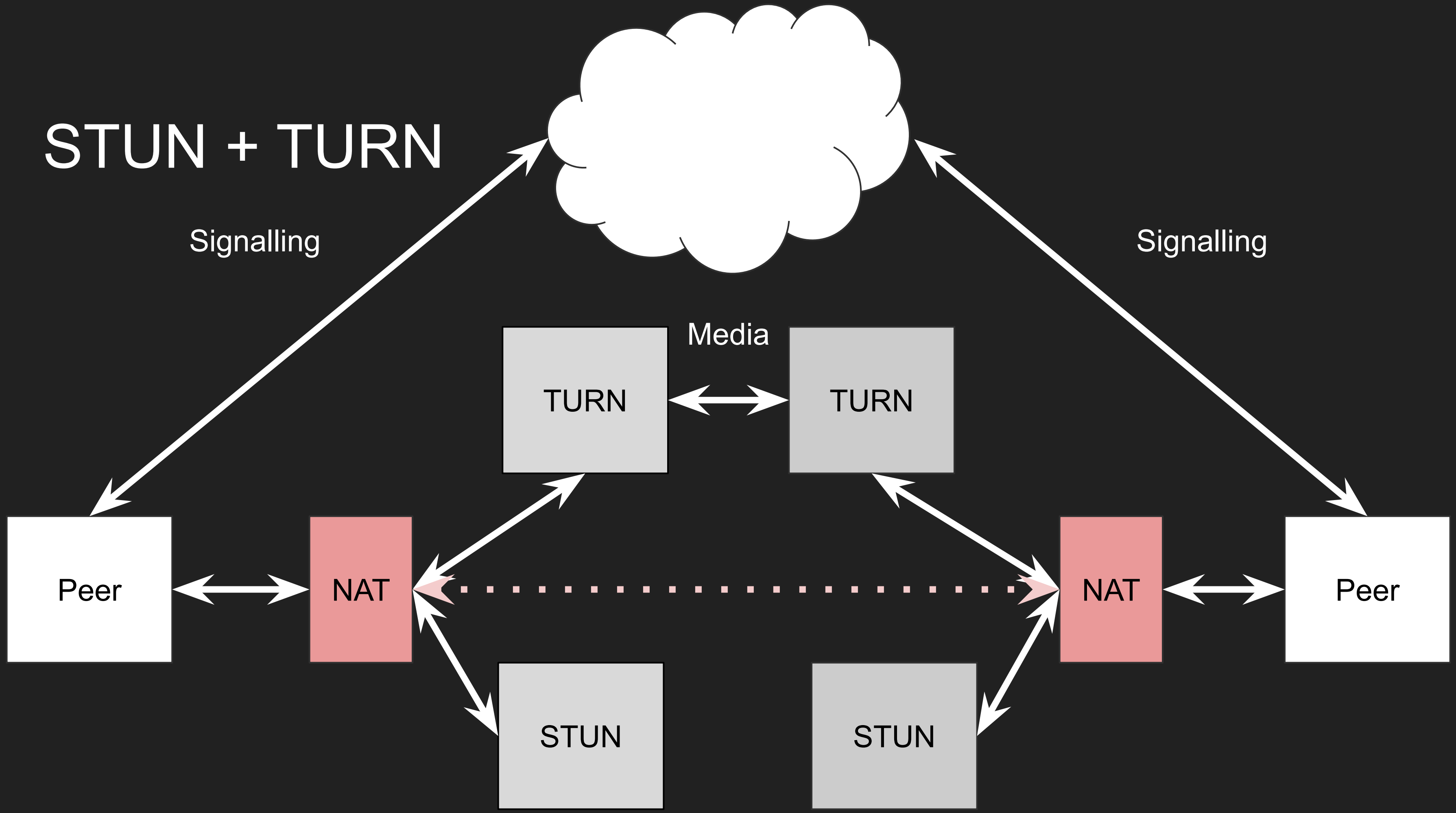
TURN (traversal using relays around NAT)

- Cloud fallback if peer-to-peer fails
- Data sent through the server
- Ensure call works in almost any environments
- EXPENSIVE

TURN



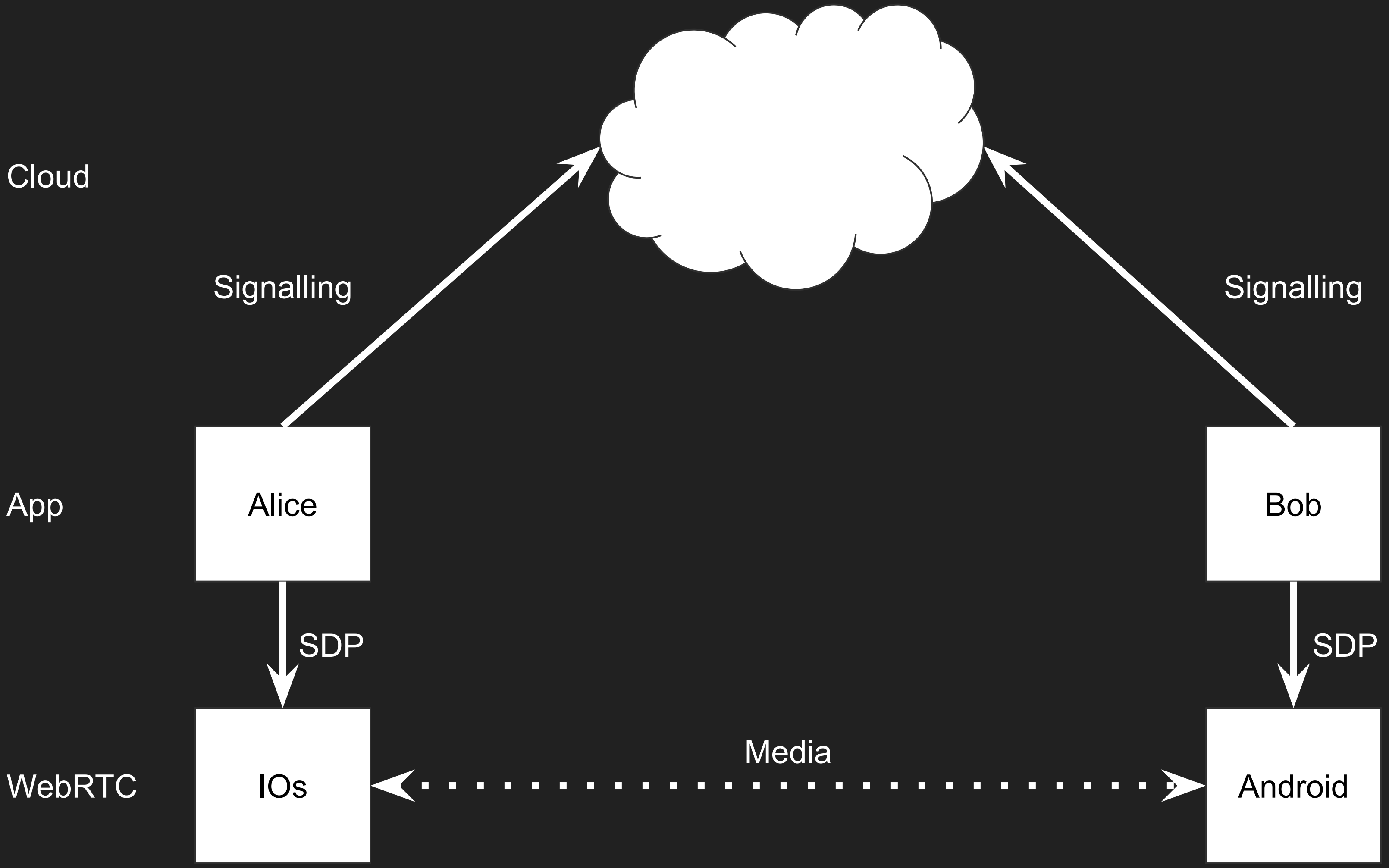
STUN + TURN



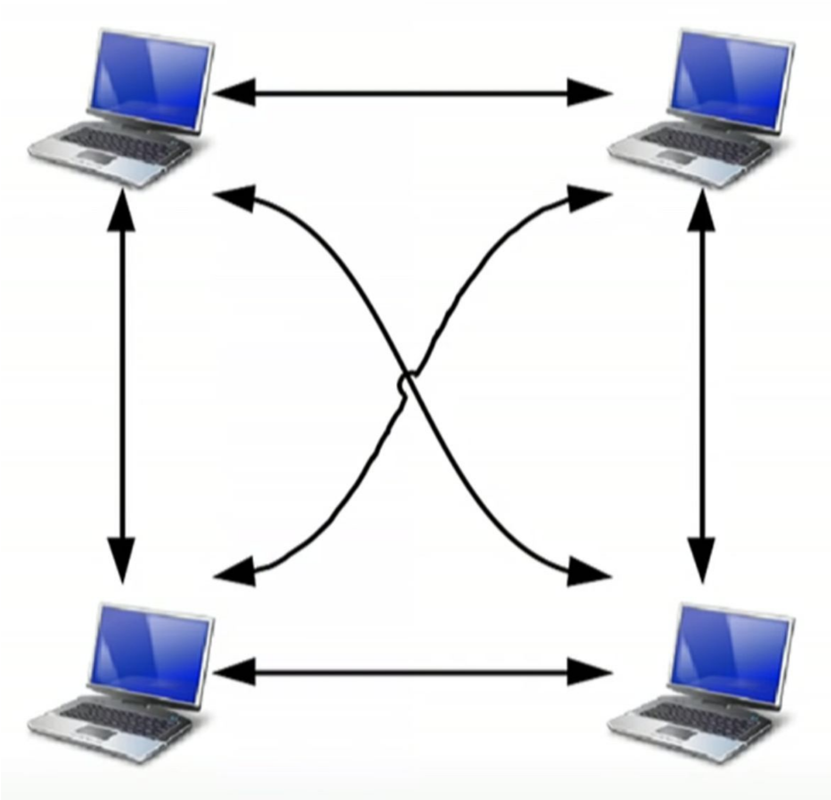
Signalling
Signalling
Media
TURN
TURN
Peer
NAT
NAT
Peer
STUN
STUN

ICE (interactive connectivity establishment)

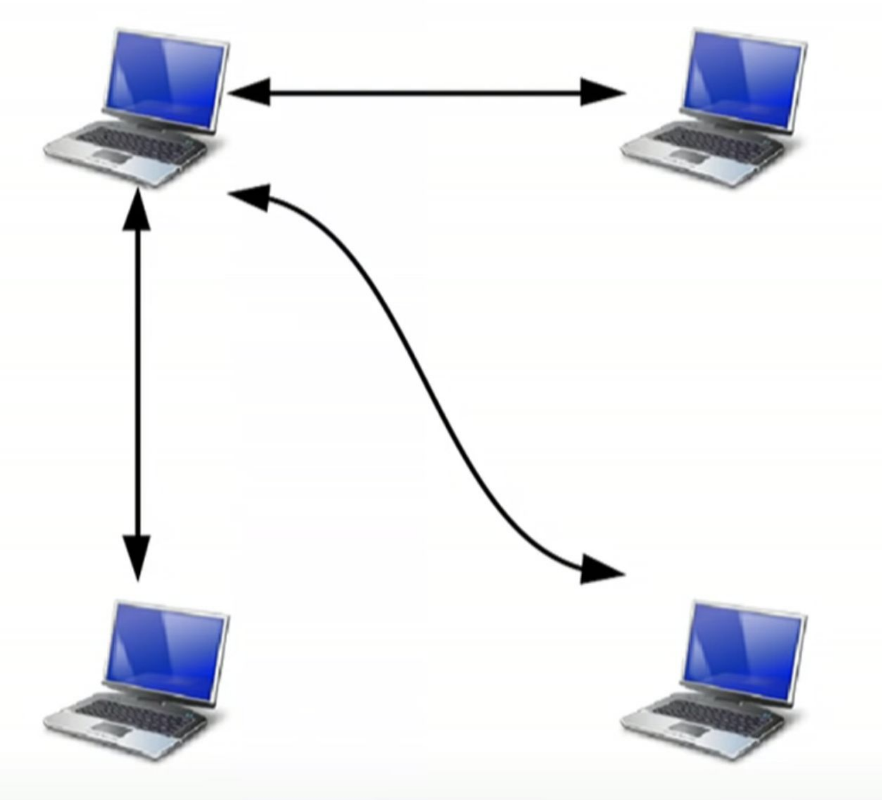
- Framework for connecting peers
- Find the best path for each call
- How?
  - Gathering candidates
    - IP address + port + transport protocol
      - Directly attached network interface
      - Server reflexive (STUN)
      - Relayed address (TURN)
  - Connectivity checks
    - Sort the candidate pairs in priority order
    - Send checks on each pairs in priority order
    - Acknowledge checks received from the agent
  - Nominating Candidate Pairs and concluding ICE


Cloud
Signalling
Signalling
App
Alice
Bob
SDP
SDP
WebRTC
IOs
Media
Android

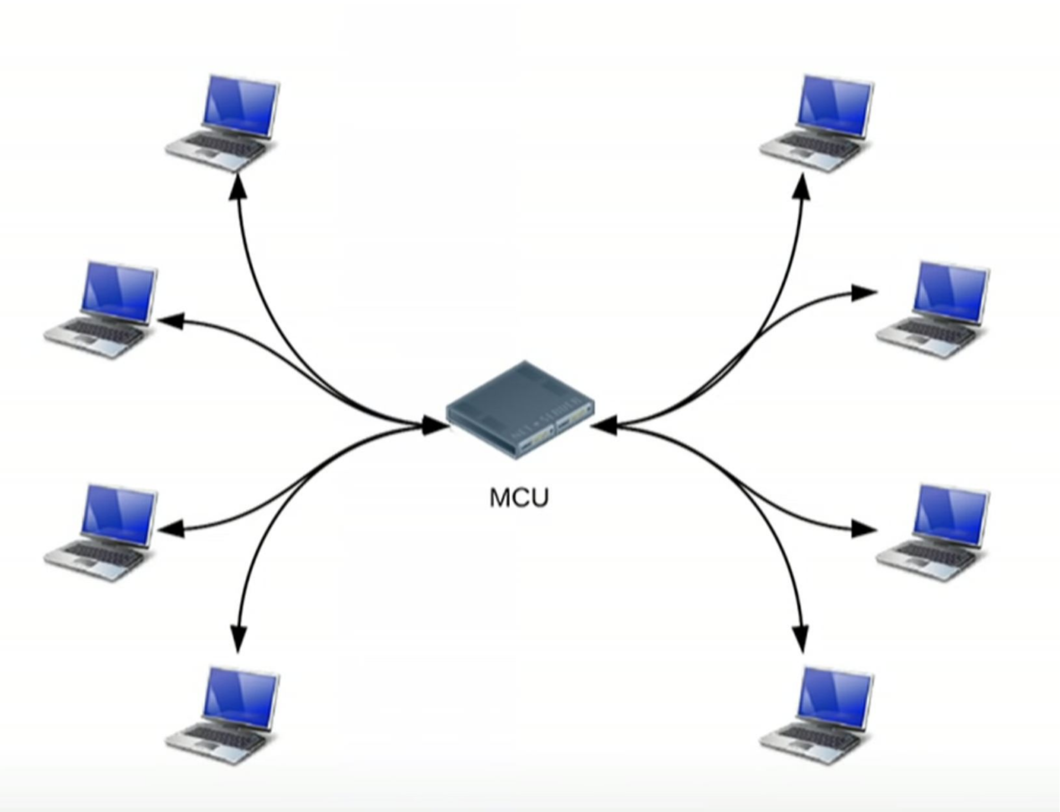
Architecture: Small call



Architecture: Medium call

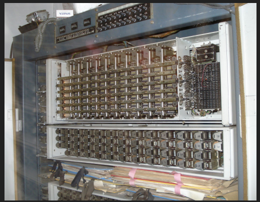


Architecture: Big call



VoIP (Voice over Internet Protocol)

PBX (Private Branch Exchange)



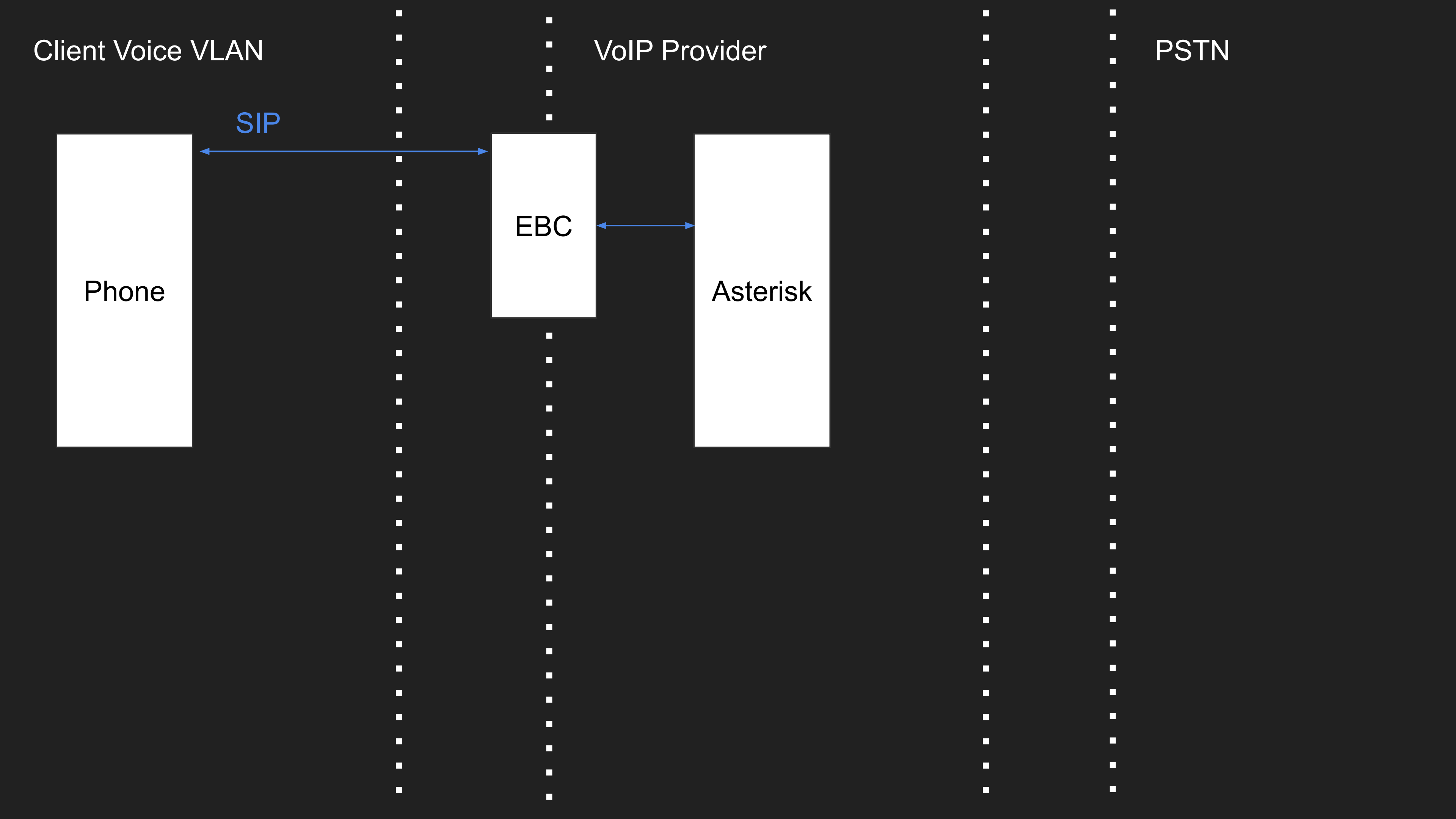
Asterisk

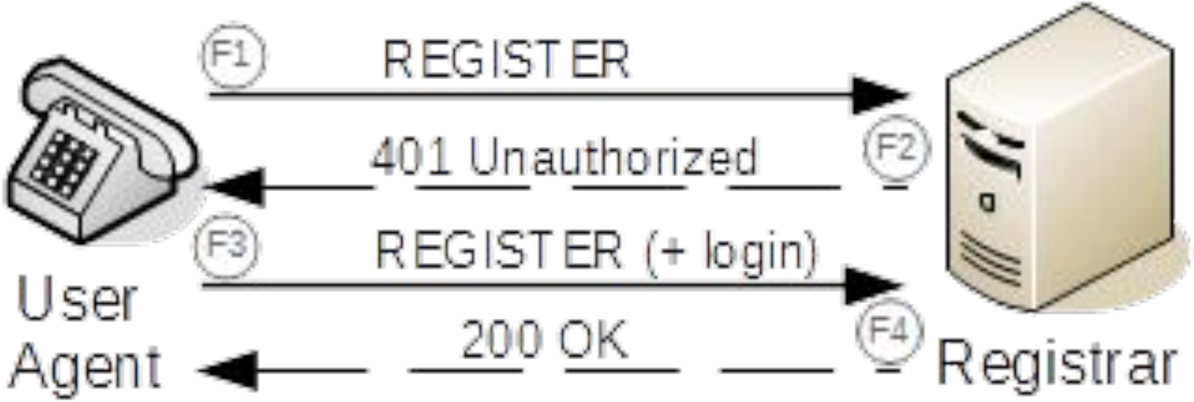
JIVE

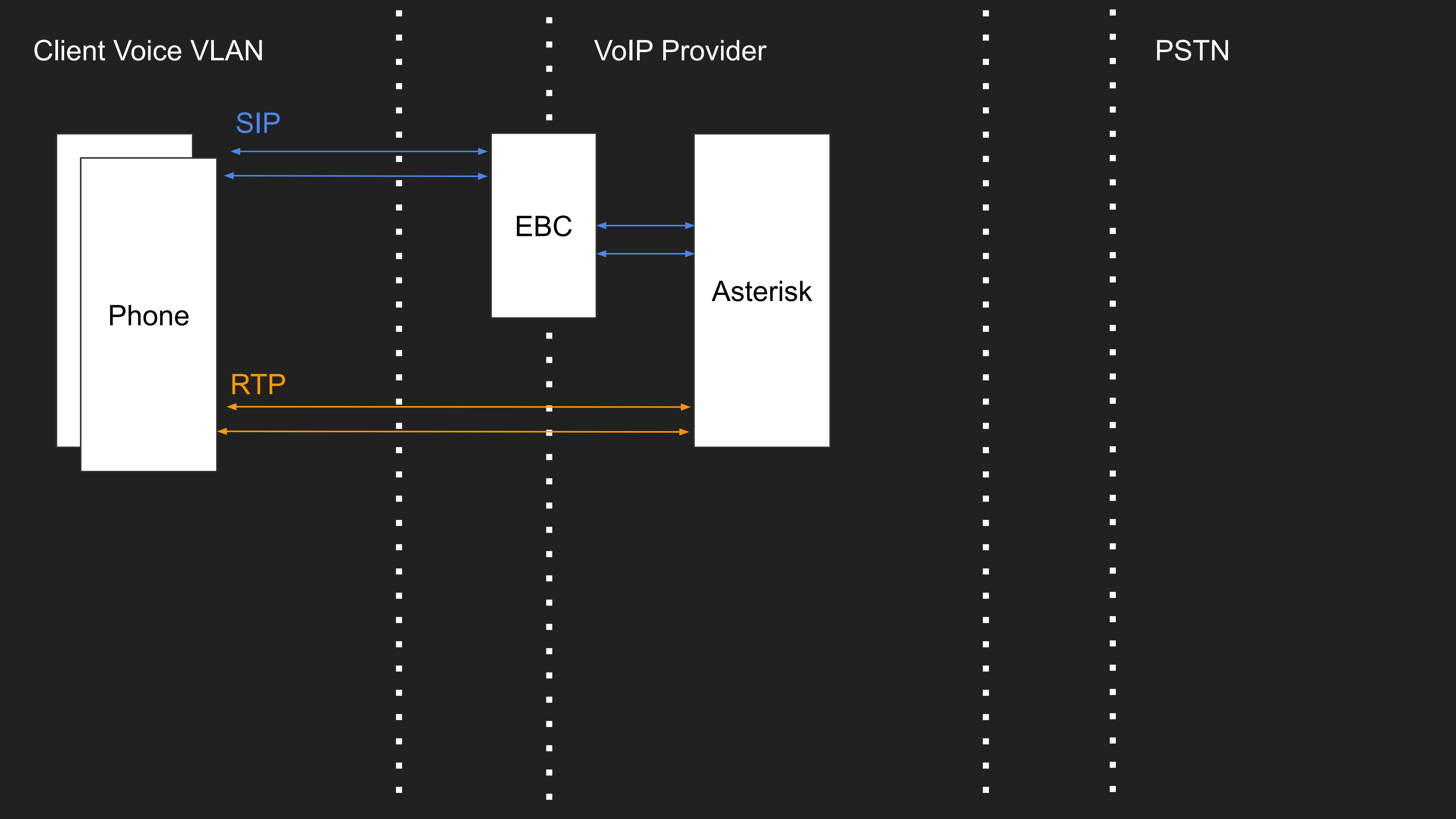
JIVE by LogMeIn

1940 1970 1999 2006 2018

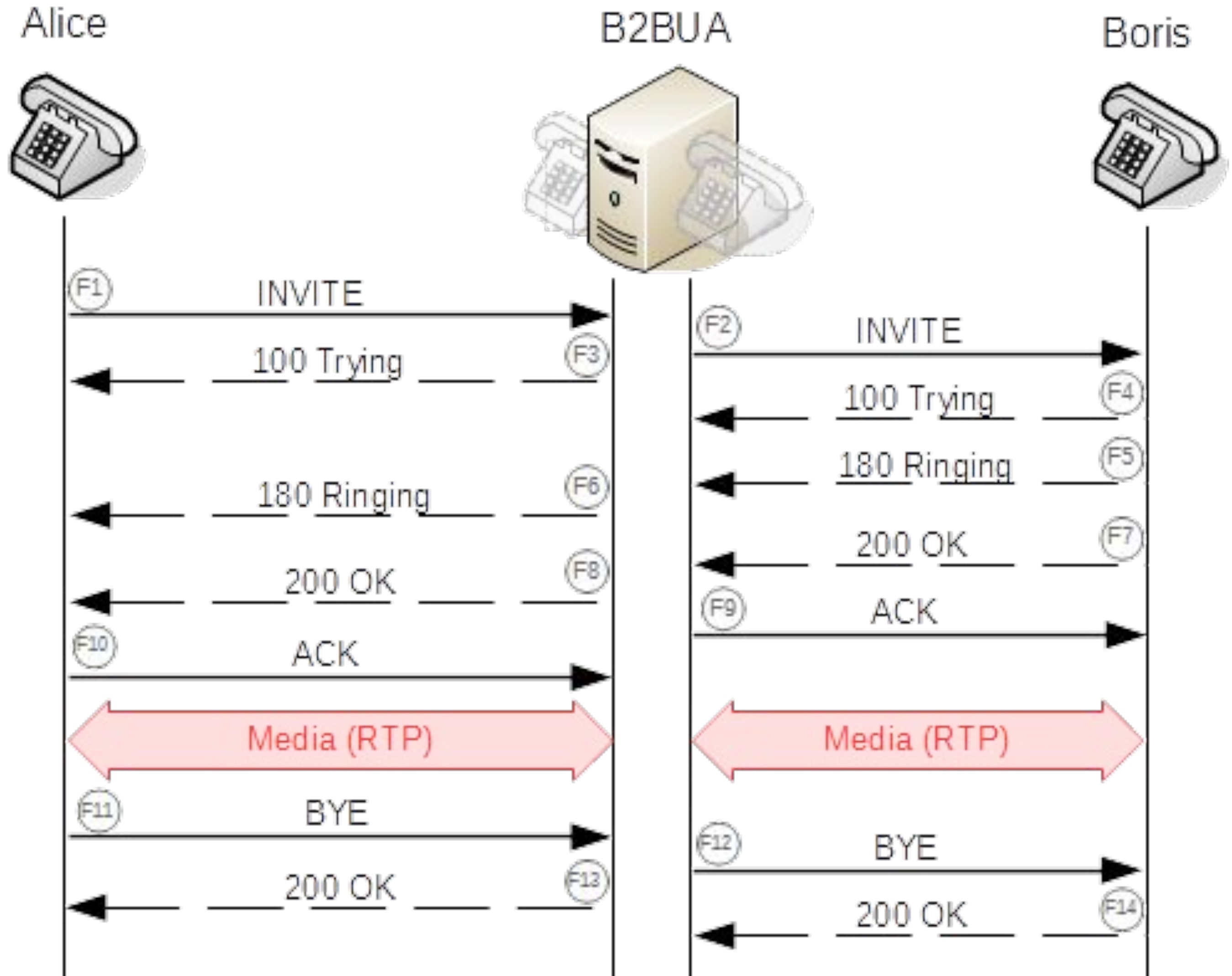
SIP (Session Initiation Protocol)

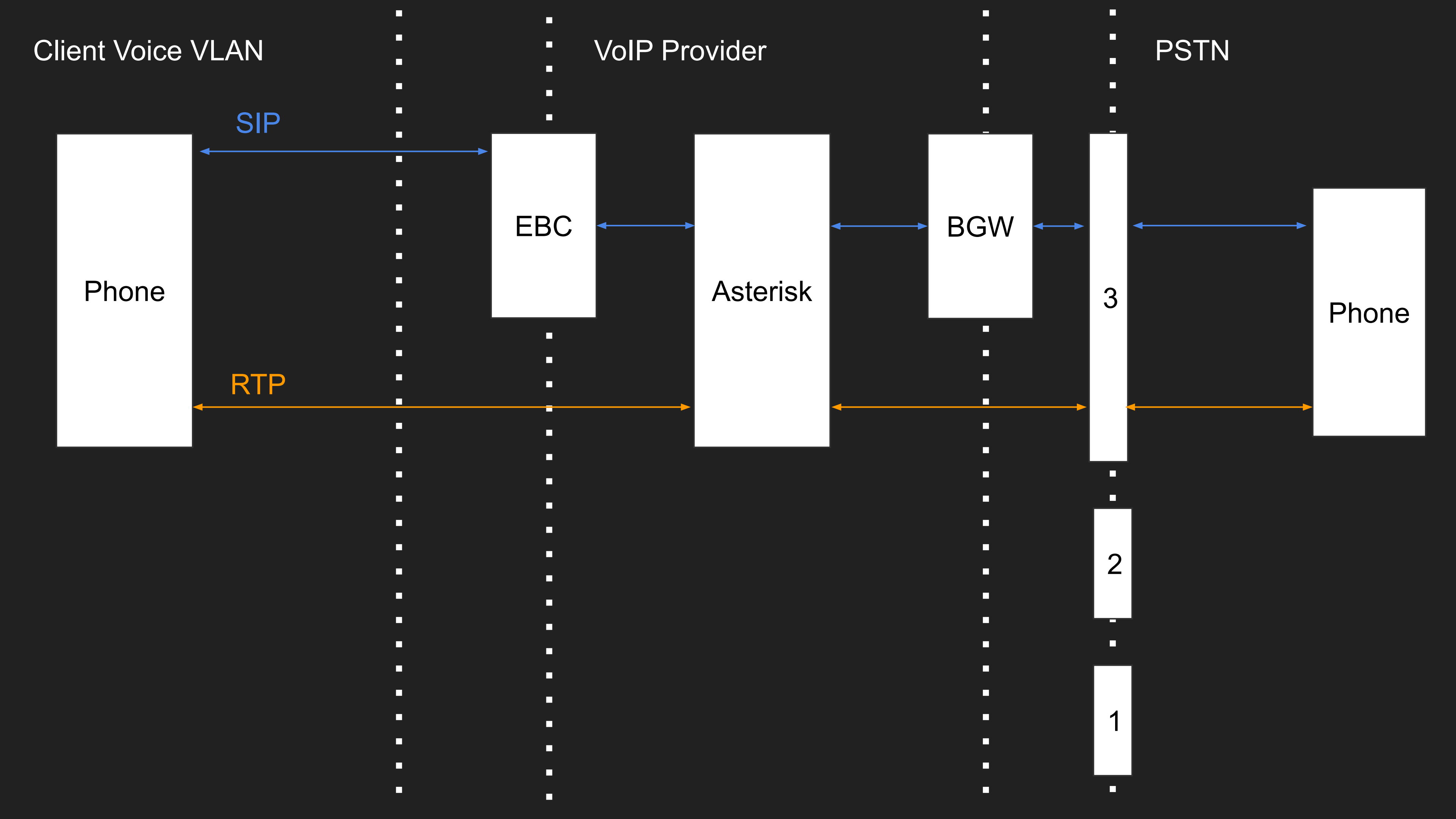

Client Voice VLAN
VoIP Provider
PSTN
SIP
Phone
EBC
Asterisk


F1
REGISTER
F2
401 Unauthorized
F3
REGISTER (+ login)
F4
200 OK
User
Agent
Registrar



Client Voice VLAN
VoIP Provider
PSTN
SIP
Phone
EBC
Asterisk
RTP


Alice
B2BUA
Boris
F1
INVITE
F2
INVITE
100 Trying
F3
100 Trying
F4
180 Ringing
F5
180 Ringing
F6
200 OK
F7
200 OK
F8
F9
ACK
F10
ACK
Media (RTP)
Media (RTP)
F11
BYE
F12
BYE
200 OK
F13
200 OK
F14


Client Voice VLAN
VoIP Provider
PSTN
SIP
Phone
EBC
Asterisk
BGW
3
Phone
RTP
2
1

WebRTC + VOIP

2006

open-source

2011

Jive web

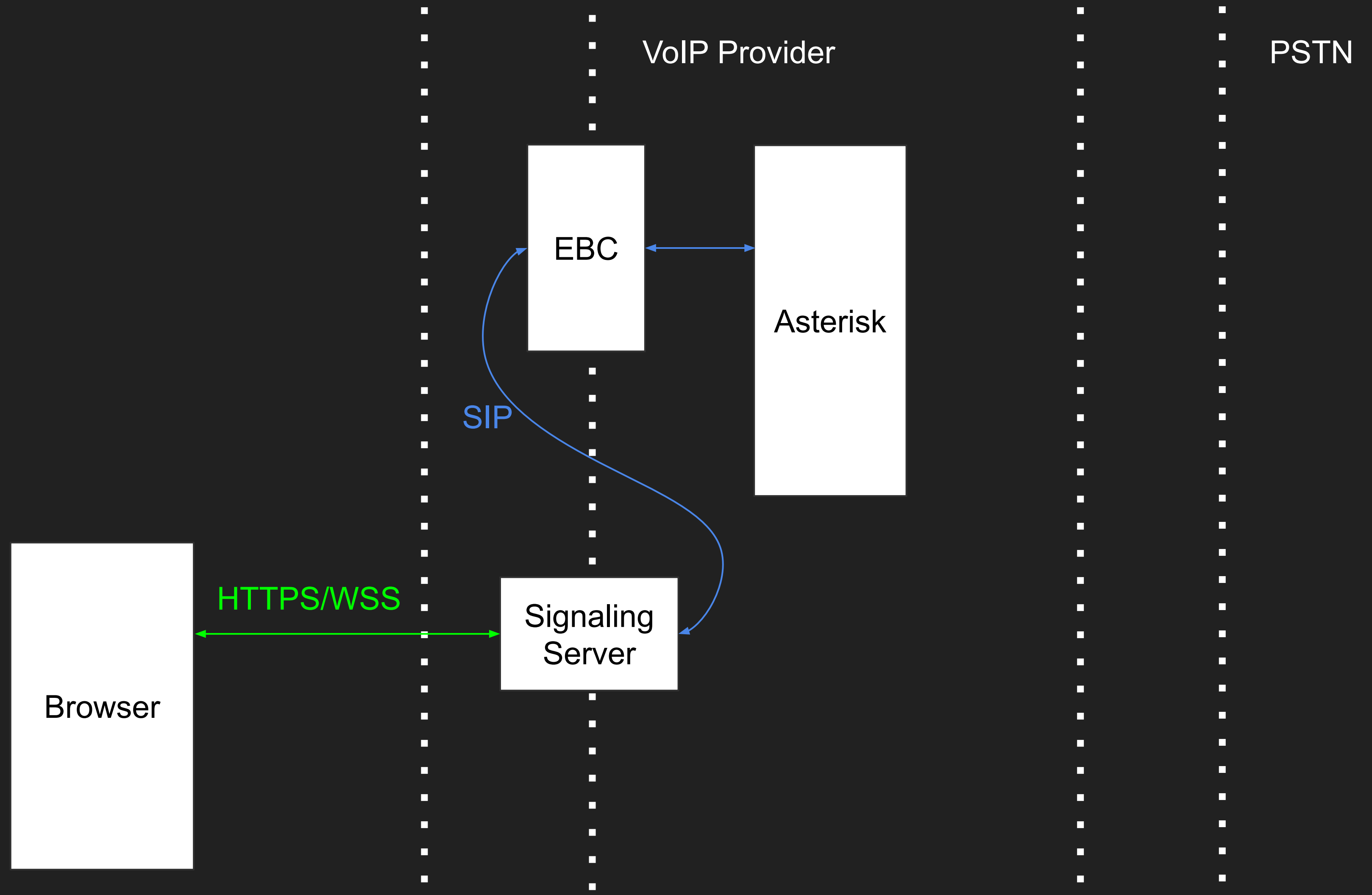
2016

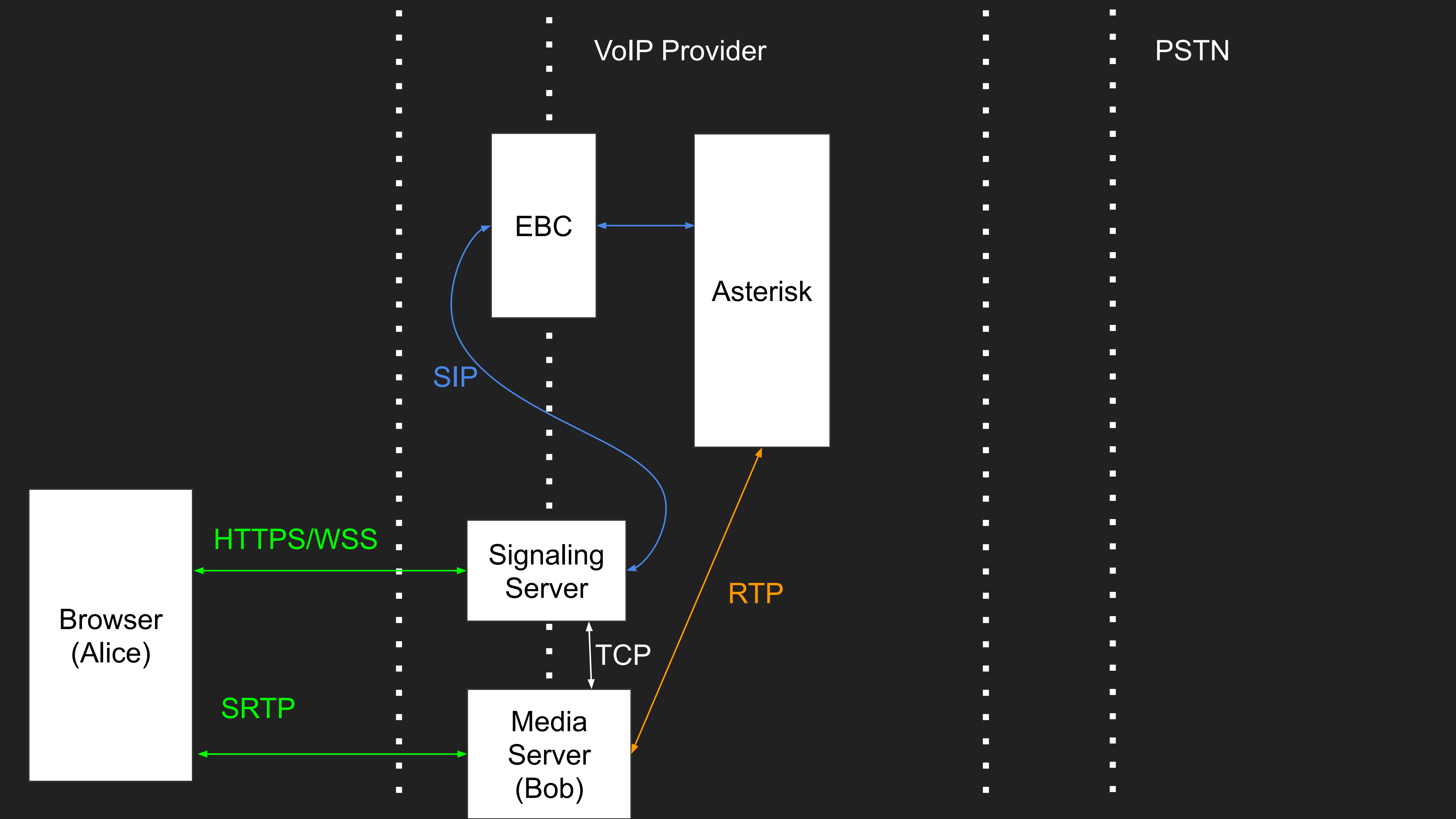
WebRTC 1.0

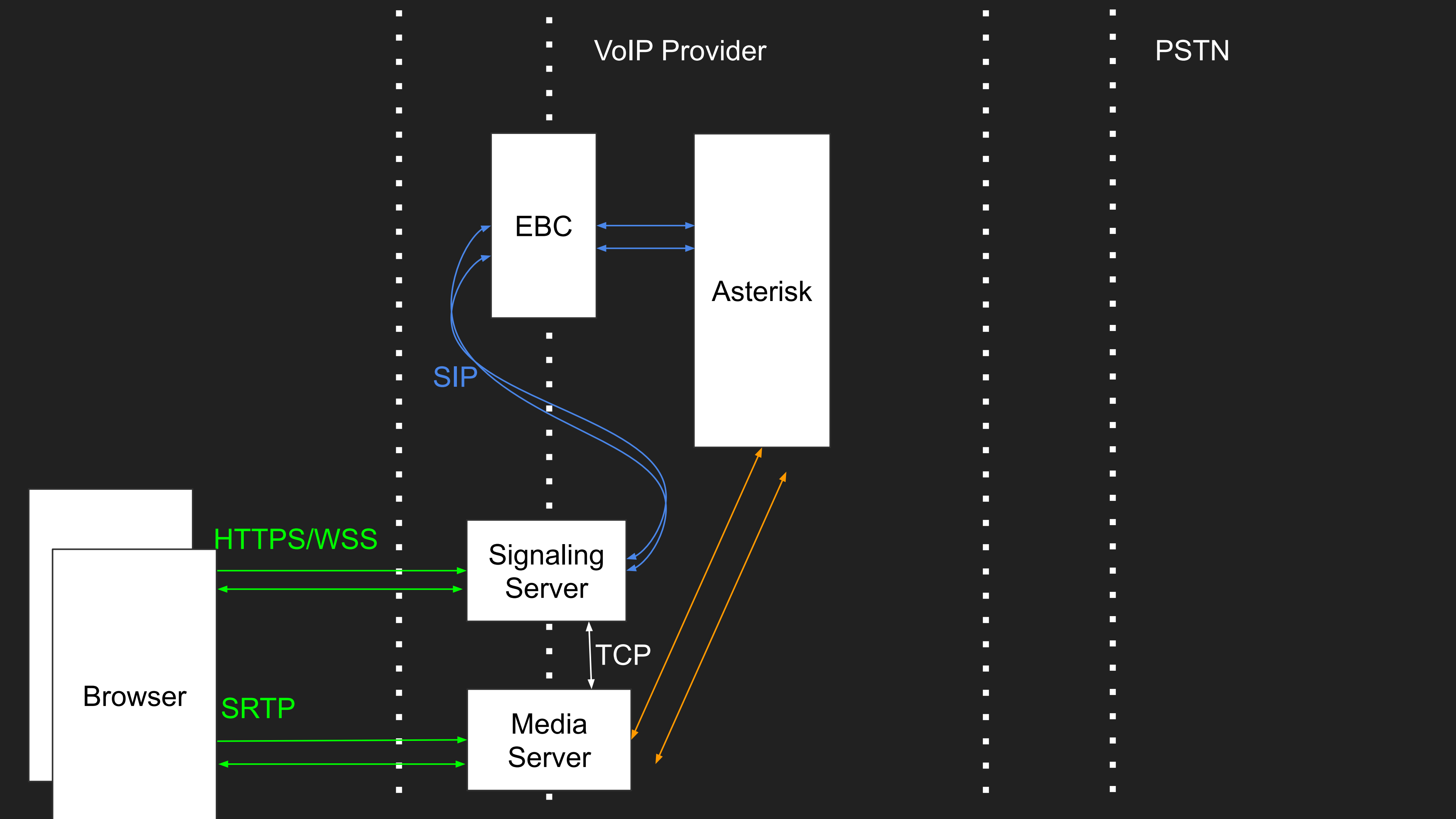
2017

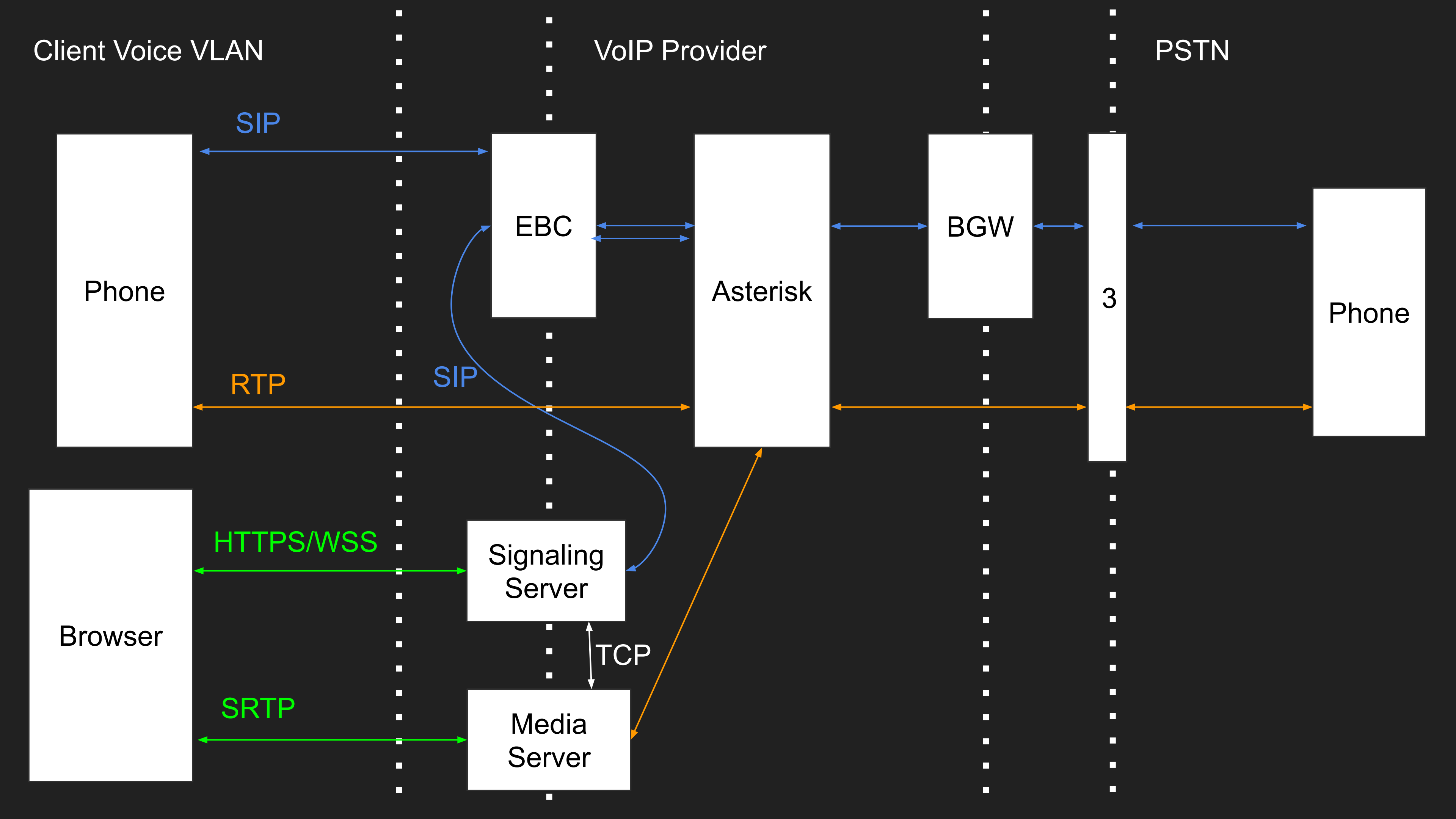
by LogMeIn

2018


VoIP Provider
PSTN
EBC
Asterisk
SIP
HTTPS/WSS
Signaling Server
Browser


VoIP Provider
PSTN
EBC
Asterisk
SIP
HTTPS/WSS
Signaling Server
Browser (Alice)
RTP
TCP
SRTP
Media Server (Bob)


VoIP Provider
PSTN
EBC
Asterisk
SIP
HTTPS/WSS
Signaling Server
TCP
Browser
SRTP
Media Server


Client Voice VLAN
VoIP Provider
PSTN
SIP
Phone
EBC
Asterisk
BGW
3
Phone
RTP
SIP
Browser
HTTPS/WSS
Signaling Server
TCP
SRTP
Media Server

WebRTC + VOIP + Mobile

open-source

2011

Jive

2016

web/mobile

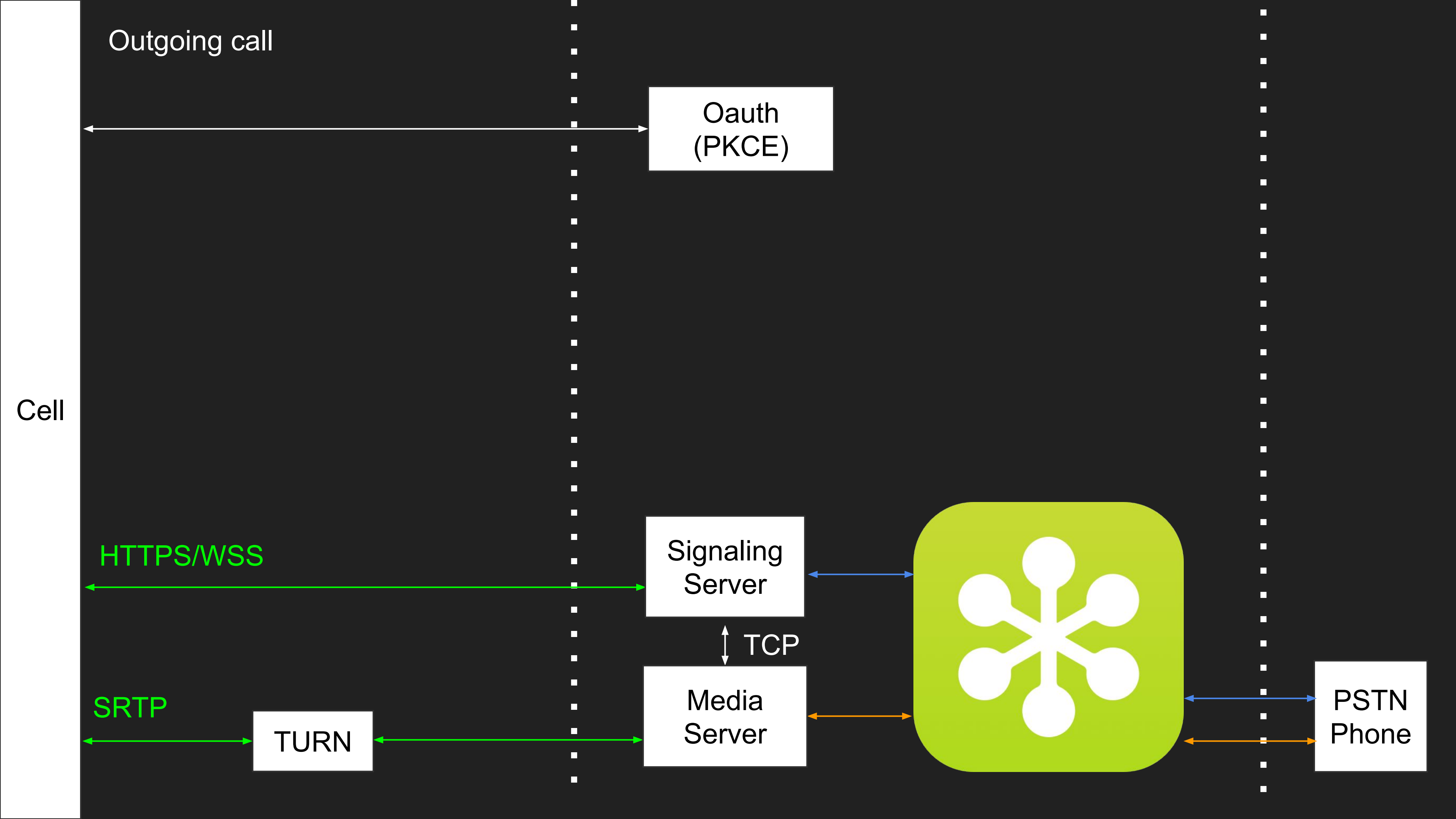
2017

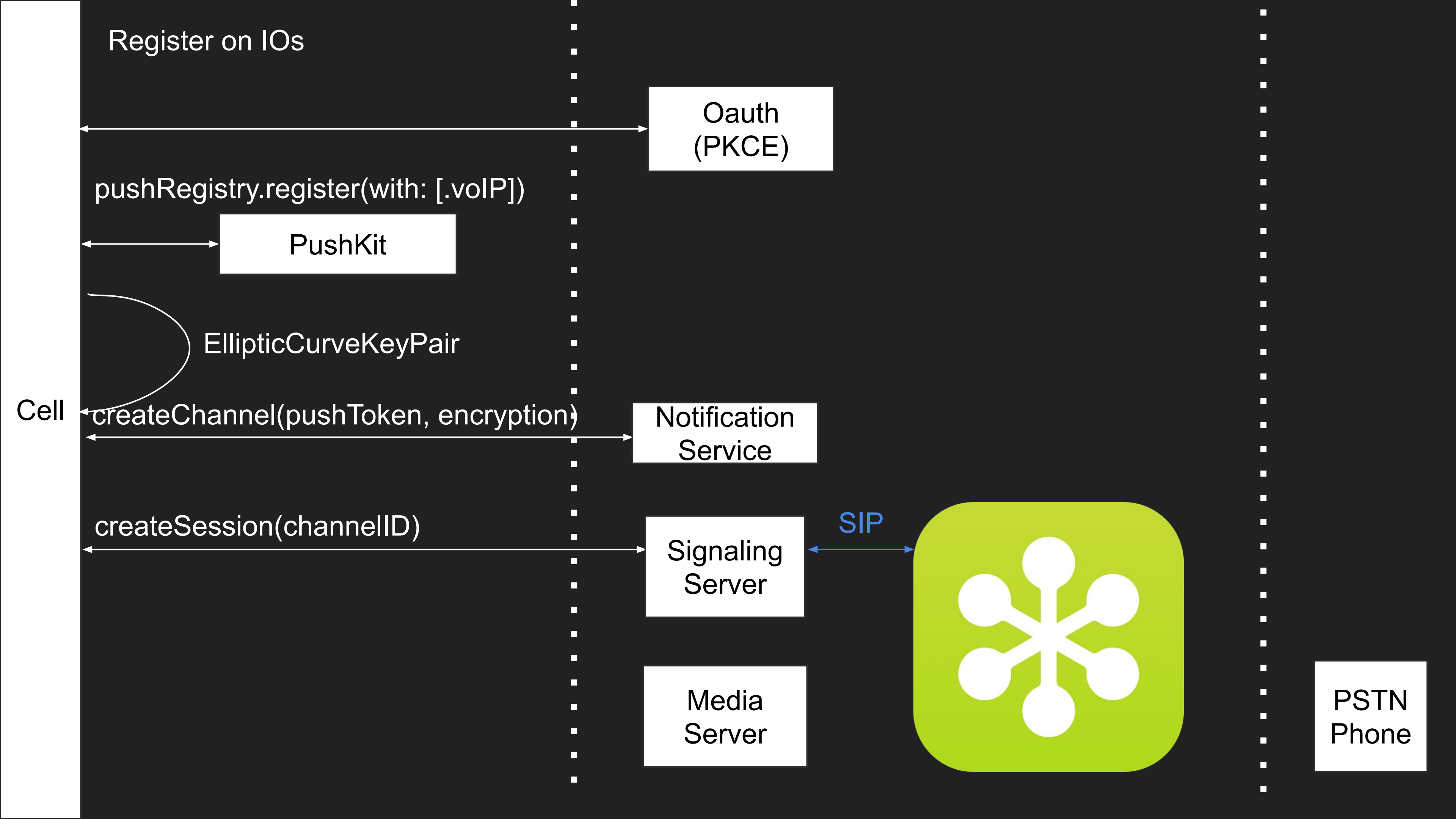
WebRTC 1.0

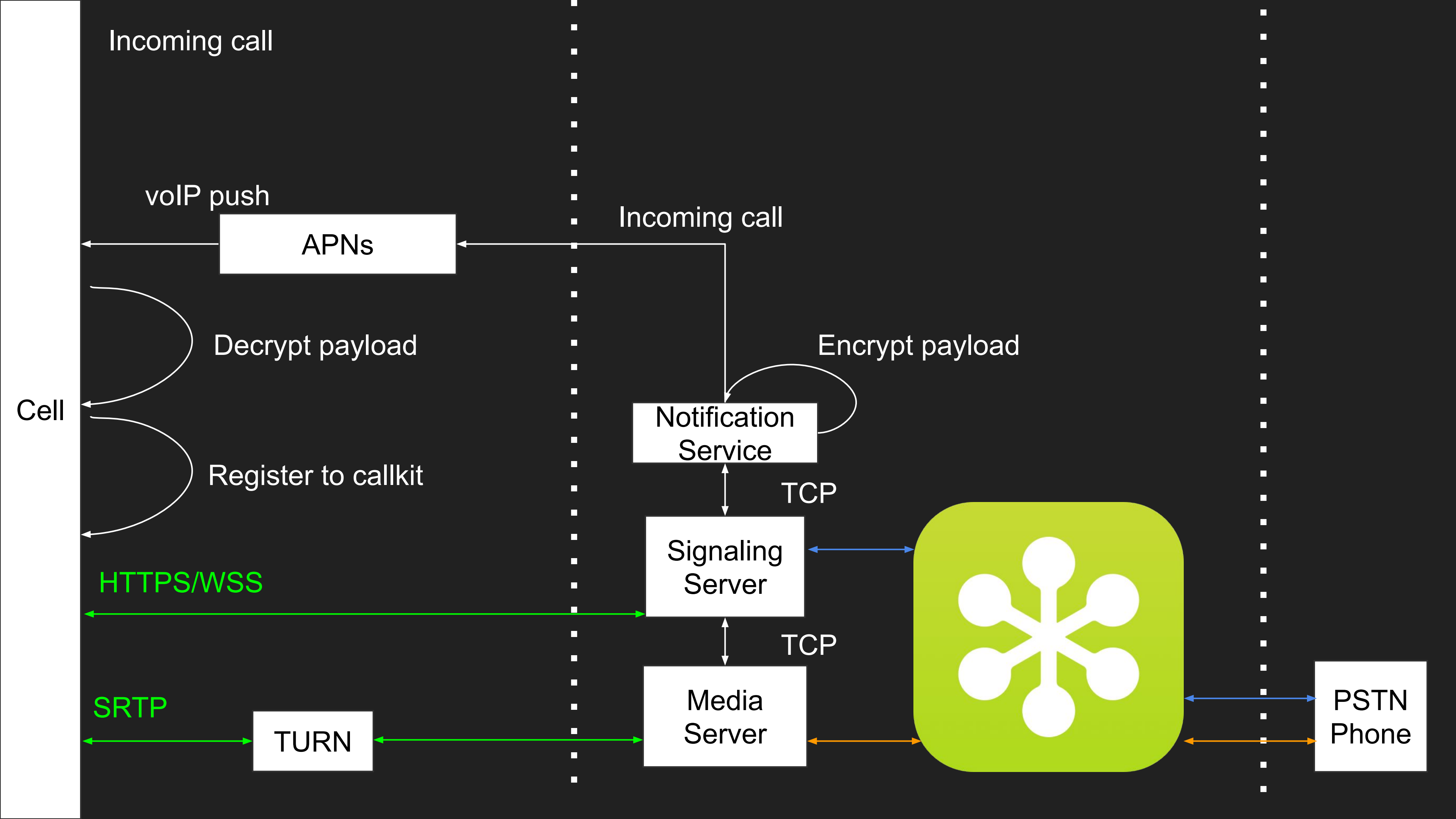
2018

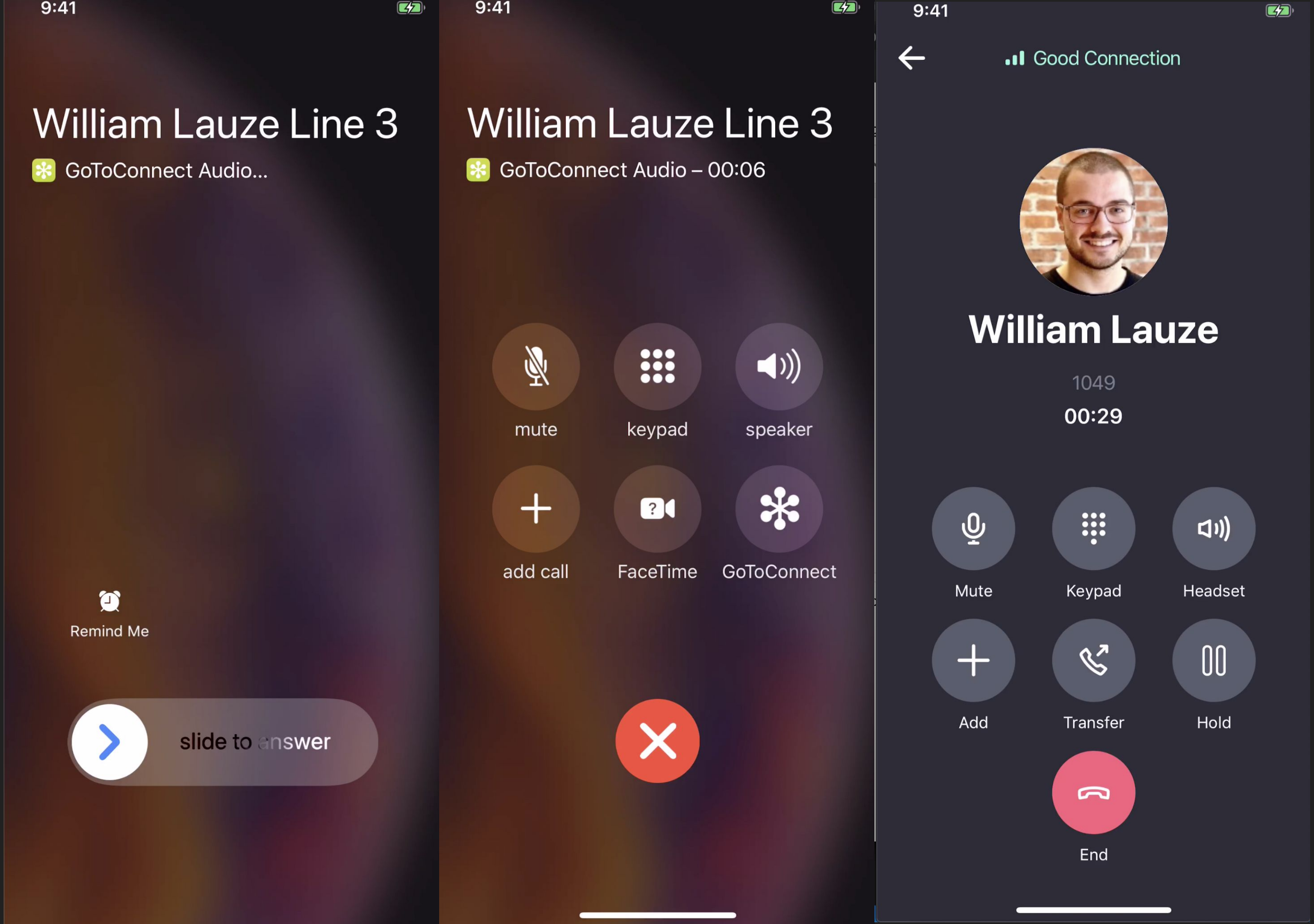
JIVE by LogMeIn

2019


Outgoing call
Cell
Oauth (PKCE)
HTTPS/WSS
Signaling Server
TCP
SRTP
TURN
Media Server
PSTN Phone


Register on IOs
Cell
Oauth (PKCE)
pushRegistry.register(with: [.voIP])
PushKit
EllipticCurveKeyPair
createChannel(pushToken, encryption)
Notification Service
createSession(channelID)
Signaling Server
SIP
Media Server
PSTN Phone


Incoming call
voIP push
APNs
Incoming call
Decrypt payload
Encrypt payload
Cell
Notification Service
Register to callkit
TCP
Signaling Server
HTTPS/WSS
TCP
Media Server
SRTP
TURN
PSTN Phone


9:41
William Lauze Line 3
GoToConnect Audio...
Remind Me
slide to answer
9:41
William Lauze Line 3
GoToConnect Audio – 00:06
mute
keypad
speaker
add call
FaceTime
GoToConnect
9:41
Good Connection
William Lauze
1049
00:29
Mute
Keypad
Headset
Add
Transfer
Hold
End

Pitfalls

- IPv6 mobile provider
  - TURN
- IOs13
  - VoIP push must register to callkit
  - DND must be server side
- Callkit
  - No customization
- Background
  - Callkit + ConnectionService
- Bandwidth + CPU
  - Frame rate
  - Resolution
  - Pause streams
  - Batch update participants
- Android
  - Audio Routes
  - Proximity sensors

Questions?

https://www.linkedin.com/in/williamlauze/

https://github.com/wilau2

https://twitter.com/WLauze

cabane-io.slack.com