

Datagram (Node.js)

- Signaling Server ist nicht notwendig, aber IP muss dann anderweitig ausgetauscht werden
- Keine Media-Metadaten \Rightarrow Video, Audio etc benötigt eigene SDP Implementation falls gewünscht

Geeignet für Lan-Play

TCP (Websockets): - Einfach und zuverlässig

- Reliable
- Ordered
- Volle bi-directionale Kommunikation über TCP
- Low Bandwidth → braucht nur einen Header pro Verbindung
- Kompatibel mit allen Browsern
- Standardisiert
- Kann Event-basiert programmiert werden
- Echtzeit
- Kompatibel mit existierenden Technologien

⇒ Unter Umständen für Games geeignet

→ Kann nicht mit REST kommunizieren

⇒ Load balancing fragwürdig

⇒ Server kann in vielen Sprachen aufgesetzt werden

- mit Arduino kompatibel als Mini-Server

Socket.io als aufgesetzte API

→ Fallback Handling: Protokoll upgrade failed, dann Wechsel auf long polling

→ Reconnection möglich

→ Custom Events supported

→ Namespaces für Client Organisation als Gruppen

→ Kann Bottleneck werden, nur begrenzt skalierbar

Geeignet für Latenz unabhängige Echtzeitprojekte
z.B. Chat

Server Sent Events (SSE, basierend auf long polling) (bsp GitHub)

- one-directional, Daten fließen nur von Server zu Client
- leicht weight im Vergleich mit Websockets
- Server sendet Events an Clients die subscriben
 - > Client to Server erfordert neuen Request (POST)

Geeignet für nicht Echtzeit games, zB um States in einem rundenbasierten Game zu updaten, wenn wenig Client to Server nötig ist

HTTP/2

- Alternative zu Websockets durch Streams
- neuerer Standard

Nicht kompatibel mit Websockets

entweder oder

Now, if we compare HTTP/2 against WebSocket, we can see a lot of similarities:

	HTTP/2	WebSocket
Headers	Compressed (HPACK)	None
Binary	Yes	Binary or Textual
Multiplexing	Yes	Yes
Prioritization	Yes	No
Compression	Yes	Yes
Direction	Client/Server + Server Push + <i>bidirectional via streams</i>	Bidirectional
Full-duplex	Yes	Yes

Web RTC

Pseudo P2P: benötigt Signaling Server (evtl. STUN/TURN Server)

Signaling Server: sendet SDP an peers um ein offer weiter zu
leiten (Session Description Protocol)

Stun: Gibt public IP des Anfragenden aus + Port

SDP: Metadata, notwendig für Negotiation, Med. metadataen
müssen vereinbart werden selbst bei binary / arch. binary Daten

Transport der SDP von Browser zu Browser braucht Signaling

Sig Server kann sehr lightweight sein, je nach Anwendung
→ für einfaches P2P reichen 40 Zeilen Code
→ kann stark customized werden für fortgeschrittene
Anwendungszwecke

Issues: Authentication, Control-Level, Push Services, Scale

Verfügbarkeit: - CPaaS anfragen

- G.4thub

- Messaging Services (Firebase, Google Cloud Messenger)

NAT-Traversal:

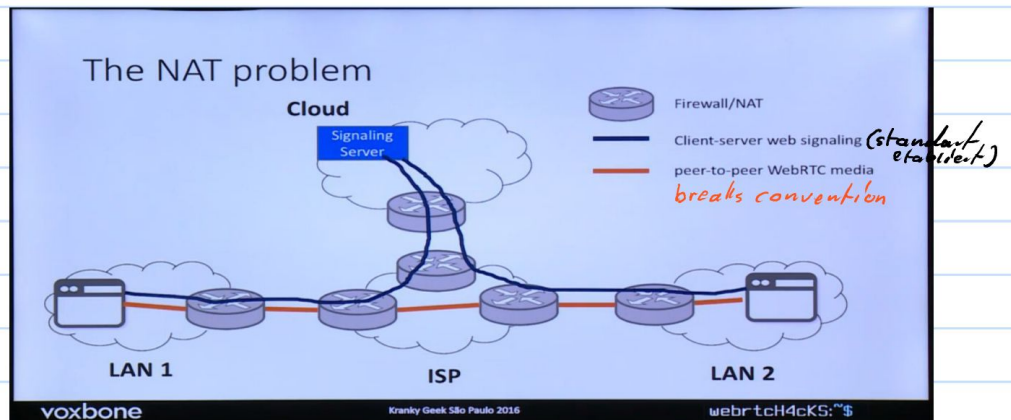
- UDP/TCP haben
Port-Ranges, Firewalls
blockieren Ports und
UDP

↓

Lösung: ICE, erlaubt Kommunikation zw. Peers hinter versch. NATs,
ICE wird per STUN/TURN etabliert

Geeignet für:

- Online Games die von
geringer Latenz abhängig
sind und kein



Comparing STUN vs. TURN

	STUN Server	TURN Server
What it does?	Returns an external IP address	Relays media
When is it needed?	Almost always	Infrequently
Operating costs?	Inexpensive	Expensive
Quality impact?	No	Possible

voxbone

Kranly Geek São Paulo 2016

webrtcH4cK5:~\$