

# **WebRTC Web Real Time Communication**

## **WebRTC Web Real Time Communication**

- Echtzeit Kommunikation erfordert:
  - geringe Latenz (wenige 100 ms)
    - sonst sind Gespräche kaum möglich, man fällt sich im Mund
- Für größere Runden (mehr als 3 Teilnehmer) empfehlen sich Systeme mit ein zentralen Server.
- Der zentrale Server nutzt eine MCU oder eine SFU
- Idealerweise ist der Installation von Client nicht notwendig, der Browser reicht aus.

# Beispiele für Konferenz-Server

- BigBlueButton
- **Jitsi-Meet**
- **Galène**
- **eduMeet**
- Peer-Calls
- **Wirow**
- openvidu
- wire
- MiroTalk-sfu
- ...

## MCU (Multipoint Control Units)

- Jeden Client sendet sein Streams zur MCU
- Die MCU kann:
  - Transkodieren,
  - Die Streams mischen.

Dies bewirkt sich vorteilhaft, die Bandbreite kann optimal genutzt werden.

Die MCU bedingt allerdings sehr leistungsfähigen Server, die CPU Leistung kann sehr hoch werden, vor allem wenn es sich um eine Software Lösung handelt

BigBlueButton verwendet eine Software MCU

## SFU (Selective Forwarding Unit)

- Jeden Client sendet sein Video und Audio zur SFU
- Die SFU sendet diese Streams zu den andere Clients
  - Weniger Verbrauch im Upload,
  - Im Download gleiche Verhältnis wie bei P2P Systeme

Galène, Jitsi-Meet PeerCall und Wirow verwenden eine SFU. Jitsi-Meet verwendet die SFU bei mehr als 3 Teilnehmer, bei weniger wird Mesh-Betrieb genutzt.

## WebRTC Server für den private Gebrauch

- SFU basierten WebRTC Server eignen sich am besten
  - Ein passende Server von ein deutschen Anbieter kostet nur einige Euro pro Monat, wenn es ein virtuelle Server ist.
    - Ein Server in Deutschland sollte eine gute Latenz garantieren
    - Das Datenschutz kann garantiert werden.

# Verbindungsaufbau zum Server

- Ein **ICE Server** (Interactive Connectivity Establishment) ist notwendig.
  - So ein Server hilft beim Überwinden einer Firewall und ein NAT.  
Ohne solch ein Server können keine Verbindung aufgebaut werden.
  - Ein externe Server kann bemüht werden, ein eigene Server kann jedoch installiert betrieben werden.
- STUN (Session Traversal Utilities for NAT)  
Wenn IPv6 vorhanden ist reicht ein STUN Server
- TURN (Traversal Using Relays around NAT)  
Ein TURN Server ist, normalerweise nur für IPv4 von Nöte.

Falls ein TURN-Server verwendet wird muss kein STUN-Server deklariert werden.

## STUN Server in Deutschland

- `stun:stun.1und1.de:3478`
- `stun:stun.t-online.de:3478`
- ...

Mit der STUN Server erfahren die Teilnehmern die externe Adresse und der Port der infrage kommt. Meistens können Verbindung zwischen den Teilnehmern aufgebaut werden. In bestimmte Fälle ist es nicht möglich (symmetrische NAT, UDP-Verkehr blockiert, ...).

## TURN Server

Da sieht es schlecht aus!

- <https://www.metered.ca> bietet ein freie TURN Dienst.
- `relay.backups.cz`

Der Client befindet sich hinter ein NAT/Firewall der sehr restriktiv eingestellt wurde. So etwas kann man bei Behörden oder Institutionen die, wenn überhaupt, nur ein sehr begrenzten Internetzugriff erlauben. Dies sollte respektiert werden, also sollte TURN ausreichend sein, zumal IPv6-Adressen, in praktisch allen private Haushalte vorhanden sind. Sollte der privaten Anwender Hirnrissige Firewall-Regeln konfiguriert haben, ist er selbst schuld.

Galène beinhaltet ein integrierten TURN Server der nur auf IPv4 erreichbar ist.

## Verbindungsarten gemäß ICE

- Host: Adresse ist öffentlich oder im selben LAN.
- Srflix: NAT, Ports werden nicht verändert.
- Rrflx: NAT, Ports werden verändert.
- Relay: Die Verbindung muss über ein Relay stattfinden. Administratoren, Hersteller oder Anbieter haben eine tolle Arbeit geleistet!

# SDP (Session Description Protocol)

**\*\*SDP\*** ist Teil vom ICE (im STUN und TURN Server berücksichtigt).

- Per SDP verständigen sich die Teilnehmer über die unterstützte CODEC (Codierer, Entcodierer).

Die Browser deklarieren die unterstützten Audio und Video in der Reihenfolge der Präferenzen, beispielsweise VP9, VP8, H264. Der Server meldet, das zu verwendeten Format zurück. Bei SFU-basierten Konferenzsysteme sollte immer VP8 den Vorzug gegeben werden, sonst besteht die Gefahr, dass einige Teilnehmern sich nicht verbinden können.

## WebRTC und Audio

- Standardmäßig wird **opus** verwendet, es gibt kein Grund einem der weiteren aufgeführten Codec zu verwenden.
  - **Opus** skaliert sehr gut, und bittet Übertragungsbandbreite von Telefonqualität bis Musikqualität.

## WebRTC und Video: VP8

- VP8 wird von alle Systeme unterstützt.

## WebRTC und Video: H.264

- H.264 wird nicht von alle Clients unterstützt (Debian Systeme, manche Android, ...).

## WebRTC und Video: VP9

- Bessere Kompression bei gleiche Bildqualität als VP8 oder H.264
- Unzureichend unterstützt:
  - VP9 muss in Firefox freigeschaltet werden (about:config)
  - Firefox unterstützt nur profile-id 0, Chromium basierten Browser unterstützen die profile-id 0 und 2.
  - Firefox unterstützt keine Bildschirmfreigabe unter VP9
  - Ältere Systeme (u.a. Android Smartphone) bieten nicht ausreichende Leistung.

## WebRTC und Video: AV1

- Noch bessere Komprimierung als VP9
- Zurzeit keine Unterstützung für Videokonferenzen

# WebRTC Media Konfiguration

- Der Server sollte nur **opus** und **VP8** verwenden
- Alle Browser und OS unterstützen es.
- Andere Formate führen zu Inkompatibilität.

## Größe der Video in Pixel

- Um Bandbreite sollte keine zu große Auflösung verwendet werden
- Meistens werden die Teilnehmer in 4:3 Format angezeigt
- Die meisten WebCam unterstützen die Größe 640\*480 bei 30 Bilder/Sek.
- Ein Format 16:9 verschwendet nur Bandbreite.

## Datenschutz

- Der Server sollte keinerlei Daten speichern
- Aufnahmen der Konferenz dürfen nur mit explizite Erlaubnis alle Teilnehmer erfolgen.
- Die Kommunikation erfolgt immer dtls verschlüsselt (Jitsi-Meet scheint hier eine Ausnahme zu bilden)

## Cookies

- Manche Systeme verwenden Cookies (Jitsi-Meet, Wirow, eduMeet)
- Galène verzichtet auf Cookies.
- Die verwendete Cookies erlauben es User-Name/alias, Media Einstellungen und ähnliches lokal zu speichern. Dies erhöht die Ergonomie.

Wirow merkt sich die Sitzung ID jeden einzelne Teilnehmer, damit kann eine unterbrochene Sitzung ohne weitere Aktion wieder zu betreten.

# Betriebs-Arten

- Konferenz
  - Alle Teilnehmer können sich sehen und hören
- Seminar (Webinar)
  - Normalerweise ist nur ein Sprecher mit Webcam oder Screensharing aktiv.
  - Keine Notwendigkeit alle Teilnehmer zu sehen zu hören.
  - Wenn nur ein System Audio und Video sendet, können wesentlich mehr Zuschauer/Hörer teilnehmen.
  - Die Kommunikation kann per Chat erfolgen.

Der Anzahl an Sende-Verbindung die zu den Klienten (Meeting-Modus) aufgebaut wird beträgt  **$N \cdot (N-1)$**  Video Streams und den gleicher Anzahl an Audio Streams.

Angenommen, dass wir 10 Meeting Klienten bedienen können (Bandbreite des Servers maßgebend), könnten wir im Webinar-Modus 90 Klienten bedienen. Sollte es 30 Meeting-Klienten sein, die wir bedienen können, hätten wir die Möglichkeit  $30 \cdot 29 = 870$  Zuschauer zu erreichen.

Auf ein VPS, bei Ionos beträgt die Bandbreite bis zu 400 MB/s, realistisch sind vielleicht 300 MB/s erreichbar es bedeutet in dem Fall  $300/2.5 = 120$  Zuschauer mit volle Bildqualität oder 300 mit geringere Bildqualität.

# Server im Konferenz Betrieb

Merkmal	Wirow	Jitsi-Meet	eduMeet	Galène
Raumreservierung	Ja	Nein	Nein	Jein
Moderation	Nein	???	???	?
Mediageräte aus	Nein	Ja	Nein?	Nein
Hand heben	Nein	Ja	Ja	Ja
Chat	Ja	Ja	Ja	Ja
Bildschirm teilen	Ja	Ja	Ja	Ja
Darstellung umschalten	Ja	Ja	Ja	Ja
Hintergrund Änderung	Nein	Ja	Nein	Nein
Sprache Deutsch	Ja	Ja	Ja	Nein
Barrierefreiheit	Bedingt	Bedingt	Nein	Nein

Die Tastatur Navigation muss unterstützt sein und der Anwender muss die Möglichkeit haben im Chatbereich zu gelangen.

Wirow und Jitsi-Meet erlauben es, auch wenn es nicht optimal gelöst ist.

## Moderation / Raum Reservierung

- Jitsi-Meet und EduMeet reklamieren es, wirklich einleuchtend ist es nicht.
- Galène ist hier vielseitiger, auf den Server kann einiges konfiguriert werden.
- Wirow Kann Räume reservieren, der Zugriff kann nur stattfinden wenn eine berechtigten Anwender sich angemeldet und das Meeting gestartet hat.

## Hand heben

- Alle Systeme bis auf Wirow haben eine Handheben Funktion.

Ein Handzeichen kann gegeben werden, damit ist es zwar schön, ist aber nicht absolut notwendig.

## Bildschirm Teilen

*Jitsi-Meet, EduMeet und Galène öffnen ein neuer zusätzliche Ansicht.* Jitsi-Meet schaltet die Ansicht der anderen Teilnehmer, so dass die Teilnehmern als Miniature an der Rechte Seite dargestellt werden. \* Wirow ersetzt die Ansicht der Camera durch die des geteilten Inhalt \* Bis auf Jitsi-Meet müssen die Anwender die Darstellung des "share" auf volle Größe bringen, es ist jedoch schnell gelernt (intuitiv).

## Internationalisation

- Bis auf Galène bitten alle Systeme die Sprache Deutsch.
  - Fremdsprachen Begriffe können einige Menschen verwirren, dadurch entstehen Barrieren.

## Seminar Betrieb

- Jitsi-Meet, eduMeet haben nicht wirklich solch ein Modus
  - Das Meeting Modus wird dazu benutzt.
- Wirow hat ein extra Webinar Modus. Die Teilnehmern können weder Ton, noch Video senden. Als Kommunikationsmittel steht nur der Chat zur Verfügung.

Ein Vorteil des eigenständigen *Webinar* Modus, ist dass den Anzahl an Teilnehmern wesentlich höher sein kann.

## Hintergrund ersetzen

- Einzig Jitsi-Meet bietet diese Funktion.
- Mit ein kleiner Programm (backscrub) kann diese Funktionalität ergänzt werden.

## Installation

- Wirow ist der Server der am leichtesten zu installieren ist
  - Zusatzdienste sind nicht erforderlich.
  - Sehr einfache Konfigurationsdatei.
  - TLS-Zertifikat Behandlung integriert.
- Galène ist auch einfach zu Installieren, die Konfiguration ist jedoch umfangreicher (unterschiedliche Textdateien)
- EduMeet scheint mit mehr Aufwand verbunden zu sein.
- Jitsi-Meet dürfte den größeren Aufwand mit sich ziehen.



# Raum Verwaltung

- Jitsi-Meet und eduMeet neue Räume werden automatisch erstellt.
- Wirow kann Anwender verwalten, eingetragene Anwender können Räume anlegen.
- Galène Räume werden mittels Dateien auf der Server angelegt, jede Raum kann
  - ein Admin (**opérateur**) s
  - ein **Presenter**
  - und oder **other** (Gäste) beinhalten.

## Raumsicherheit

- Vorraum **Jitsi-Meet, eduMeet**
- Passwort **Jitsi-Meet** (möglich), **galène**(konfigurierbar)

Jitsi-Meet Sieht es vor freie Raum Namen einzugeben, dies ist schön, es kann aber zur Kollision führen, beispielsweise wenn als Name test gewählt wird. eduMeet schlägt ein Name in der Art und Weise "**cptivjv5**" vor es kann aber mit ein Klartextname ersetzt werden.

Galène kann auf viele Arten konfiguriert werden, wenn ein Raum von Gäste erreichbar sein soll, muss eine entsprechende Datei angelegt werden. Die Namensgebung ist (fast) keine Grenze gesetzt.

Wirow erlaubt es sich als Gast einzubinden, die Links auf den Raum sind aber ziemlich lang und kompliziert. Wer den Link kennt kann an ein Meeting oder Webinar teilnehmen, die Sitzung muss allerdings von eine berechnigte Person (regulär eingeloggten Anwender gestartet werden). der Gast wird angeboten als Gast oder mit Login im Raum (mit klar Name) einzutreten.

## Audienz der Server

- Jitsi-Meet: Jedermann.
- eduMeet: Institutionen, bedingt Jedermann.
- Galène: Institutionen (hochschulen, Universitäten), bedingt Jedermann.
- Wirow: Unternehmen, Jedermann.