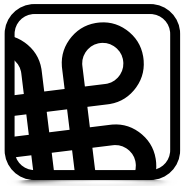


Macoun

Call me Baby: Eine Einführung in WebRTC und CallKit

Christian Braun

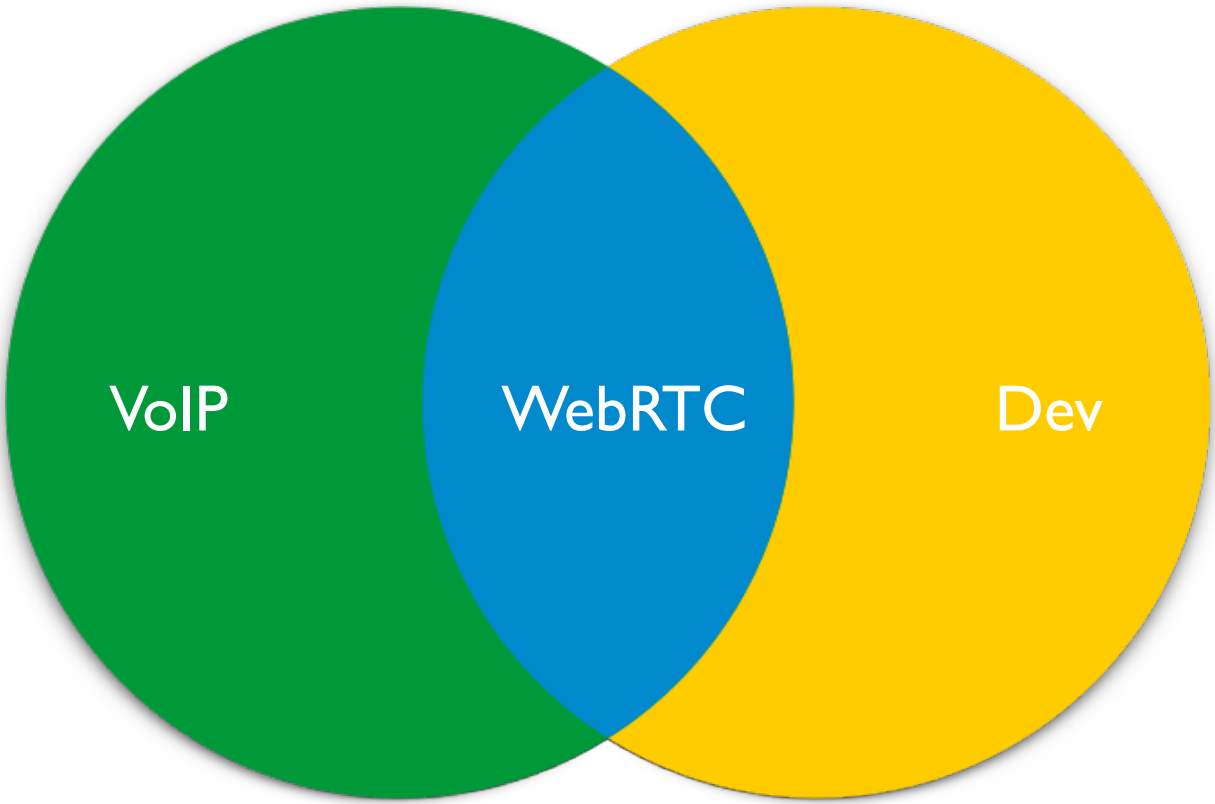
Warum sollte mich das interessieren?

CallKit


SIM fehlt
20:10
Incoming Whale call
Whale-Video ...
Erinnerung
Ablehnen
Annehmen

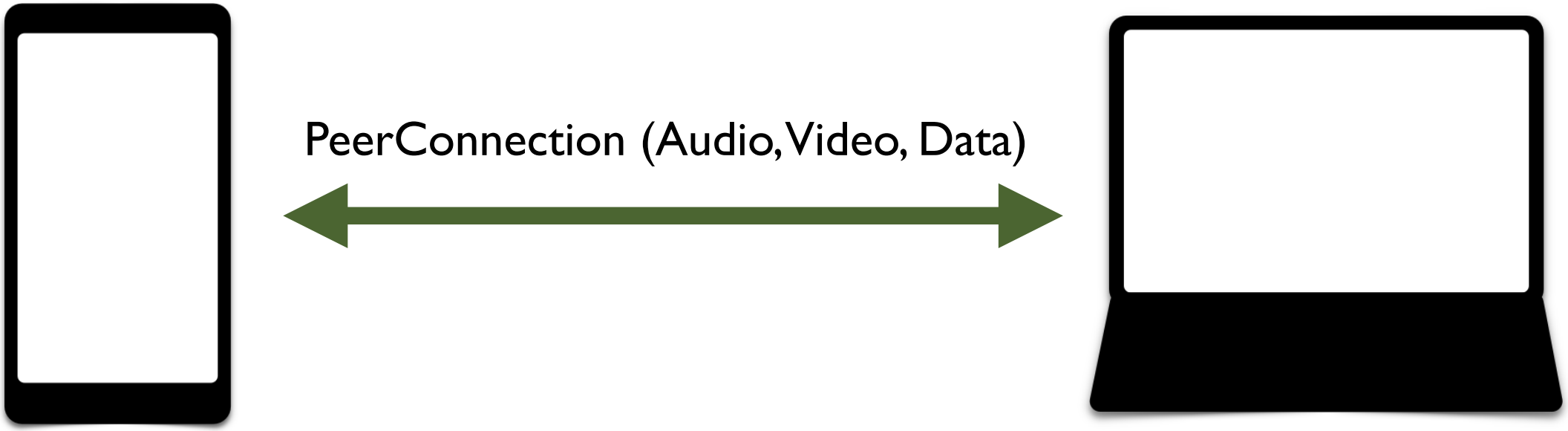

SIM fehlt
20:17
Zurück zum Anruf? Hier tippen! 00:22
Mittwoch
12
Mail
Kalender
Fotos
Kamera
Karten
Uhr
Wetter
Aktien
Wallet
Notizen
Erinnerungen
iTunes Store
App Store
iBooks
Videos
Home
Health
Einstellungen
Produktivität
Extras
MAApp

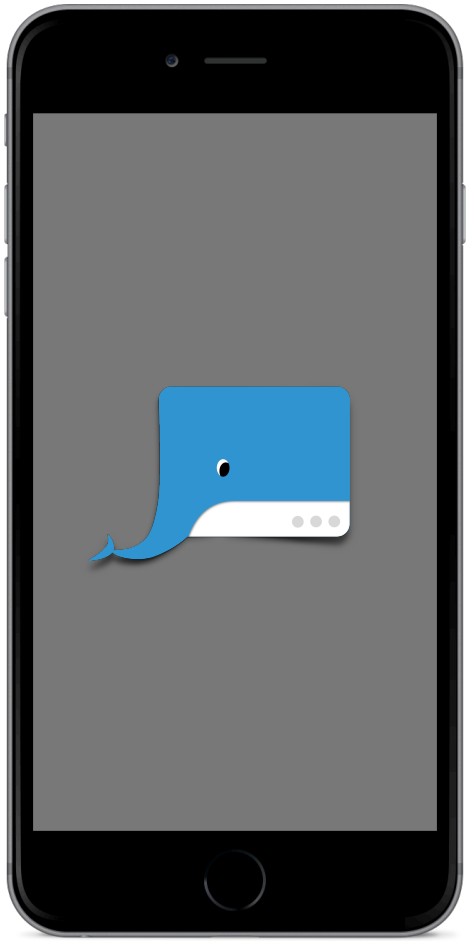

WebRTC


VoIP
WebRTC
Dev

WebRTC

- Leichter Einstieg
- Auf jeder Plattform verfügbar
- Unterstützung vieler Codecs
- Zuverlässiger Verbindungsaufbau
- Peer-to-Peer


PeerConnection (Audio, Video, Data)



Wie?

WebRTC APIs

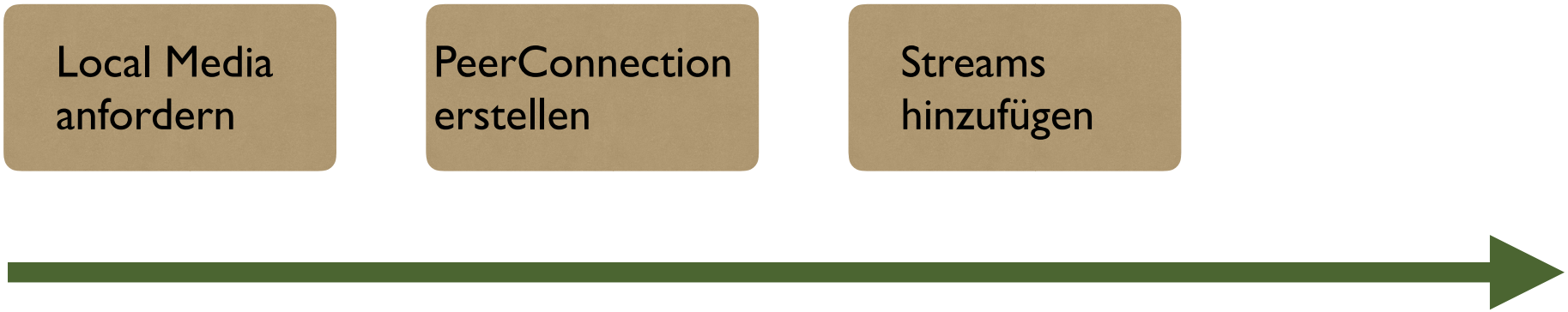
- GetLocalMedia
- RTCPeerConnection
- RTCDataChannel

WebRTC APIs

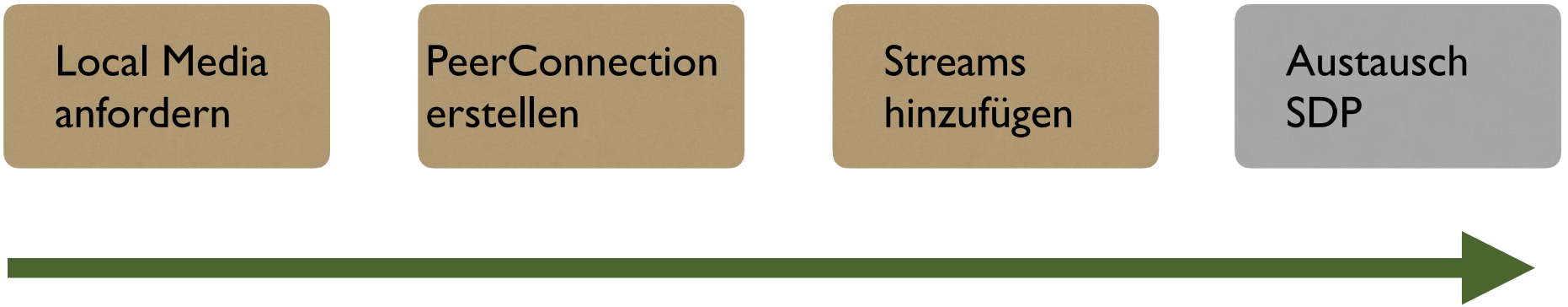
Local Media anfordern

WebRTC APIs

WebRTC APIs


Local Media anfordern
PeerConnection erstellen
Streams hinzufügen

WebRTC APIs


Local Media anfordern
PeerConnection erstellen
Streams hinzufügen
Austausch SDP

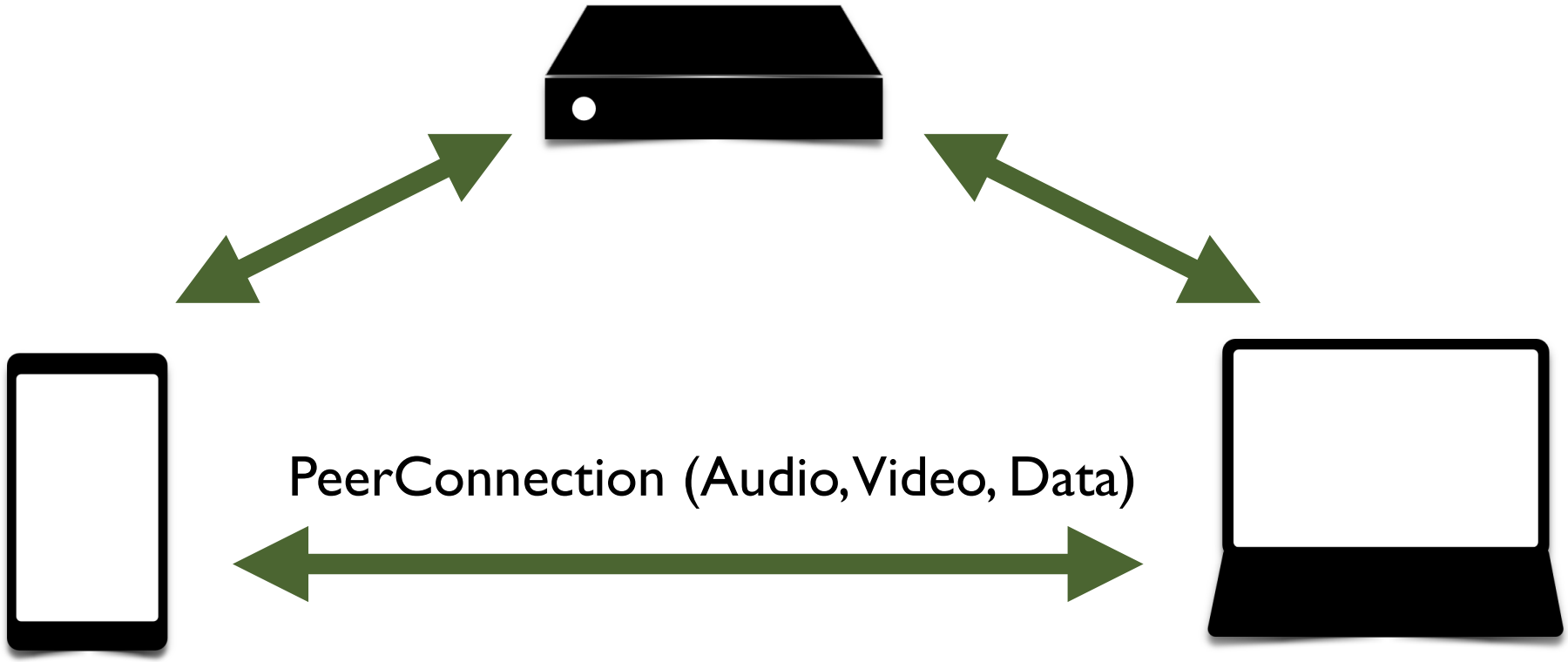
WebRTC APIs

PeerConnection erstellen

Austausch SDP

Signaling

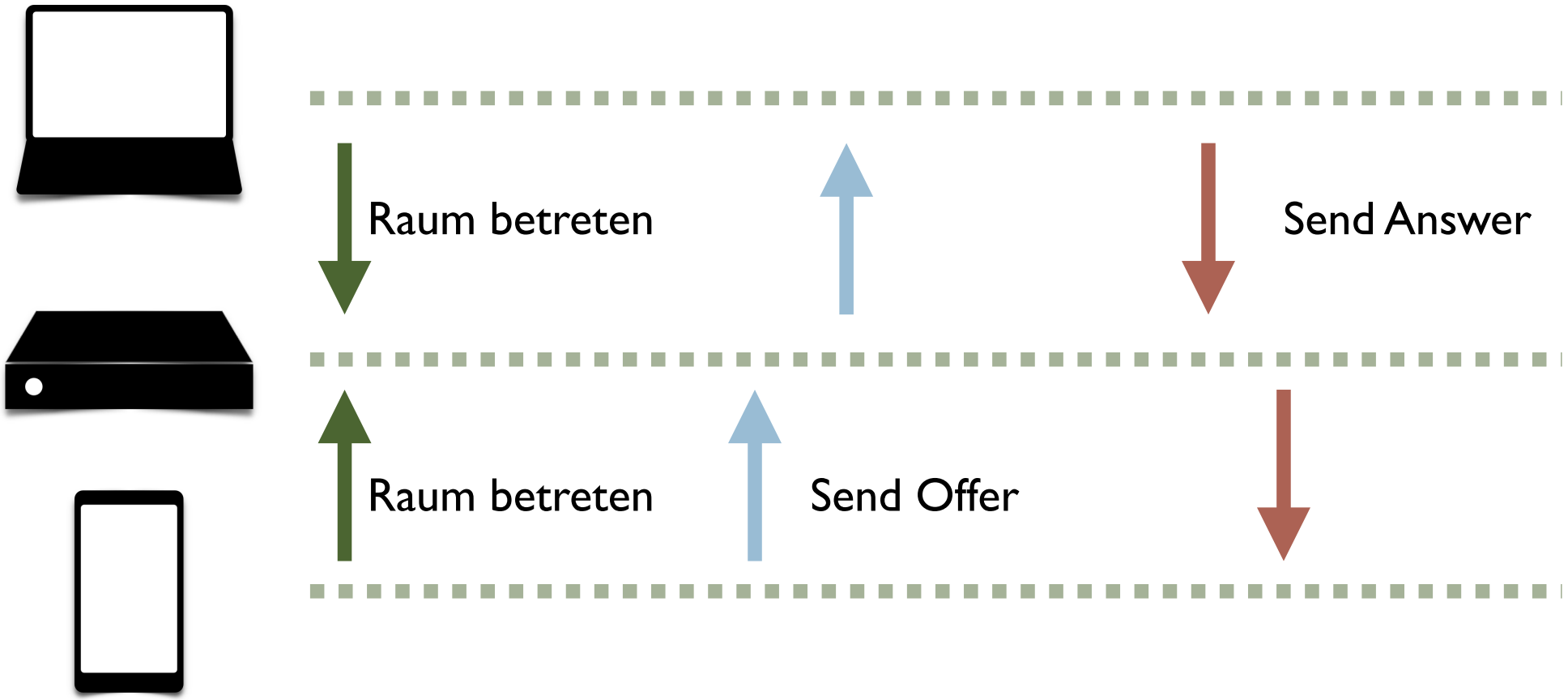
Signaling



Signaling

- WebSockets
- HTTP Long Polling

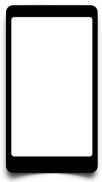
Offer & Answer Austausch


Raum betreten
Send Answer
Raum betreten
Send Offer

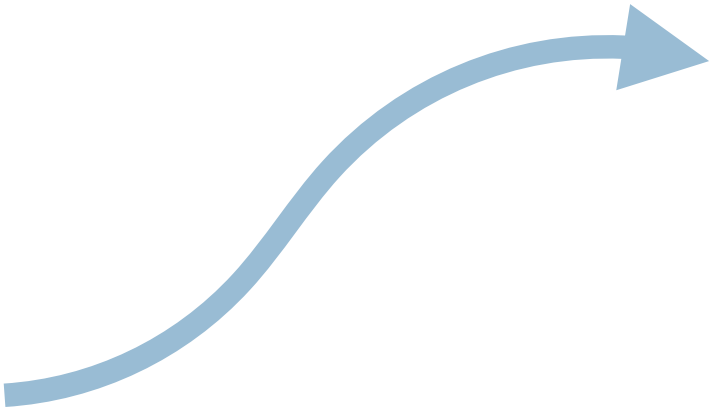
Offer-SDP

```
v=0
o=- 4338197872547430804 2 IN IP4 127.0.0.1
s=-
t=0 0
a=group:BUNDLE audio video data
a=msid-semantic: WMS WaleMS
m=audio 9 UDP/TLS/RTP/SAVPF 111 103
c=IN IP4 0.0.0.0
a=mid:audio
a=sendrecv
a=rtcp-mux
```

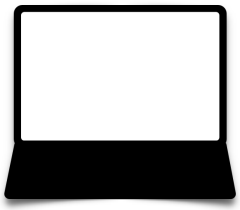
```
m=video 9 UDP/TLS/RTP/SAVPF 96 97 98
c=IN IP4 0.0.0.0
a=mid:video
a=sendrecv
a=rtcp-mux
a=rtcp-rsize
```



- PeerConnection erzeugen
- Offer generieren
- Lokale SDP setzen
- Offer versenden



- Remote SDP setzen



- PeerConnection erzeugen
- Remote SDP setzen
- Answer generieren
- Lokale SDP setzen
- Answer versenden

PeerConnection erstellen

```
let peerConnection = RTCPeerConnectionFactory()
            .peerConnection(
                with: rtcConfig,
                constraints: RTCMediaConstraints(
                    mandatoryConstraints: nil,
                    optionalConstraints: nil),
                delegate: self)
```

Offer generieren

```
peerConnection?.offer(
        for: mandatorySdpConstraints) { sdp, err in
            […]
        }
```

Offer generieren

```
peerConnection?.offer(
        for: mandatorySdpConstraints) { sdp, err in
            […]
        }
```

```
["OfferToReceiveAudio": "true",
"OfferToReceiveVideo": "true"]
```

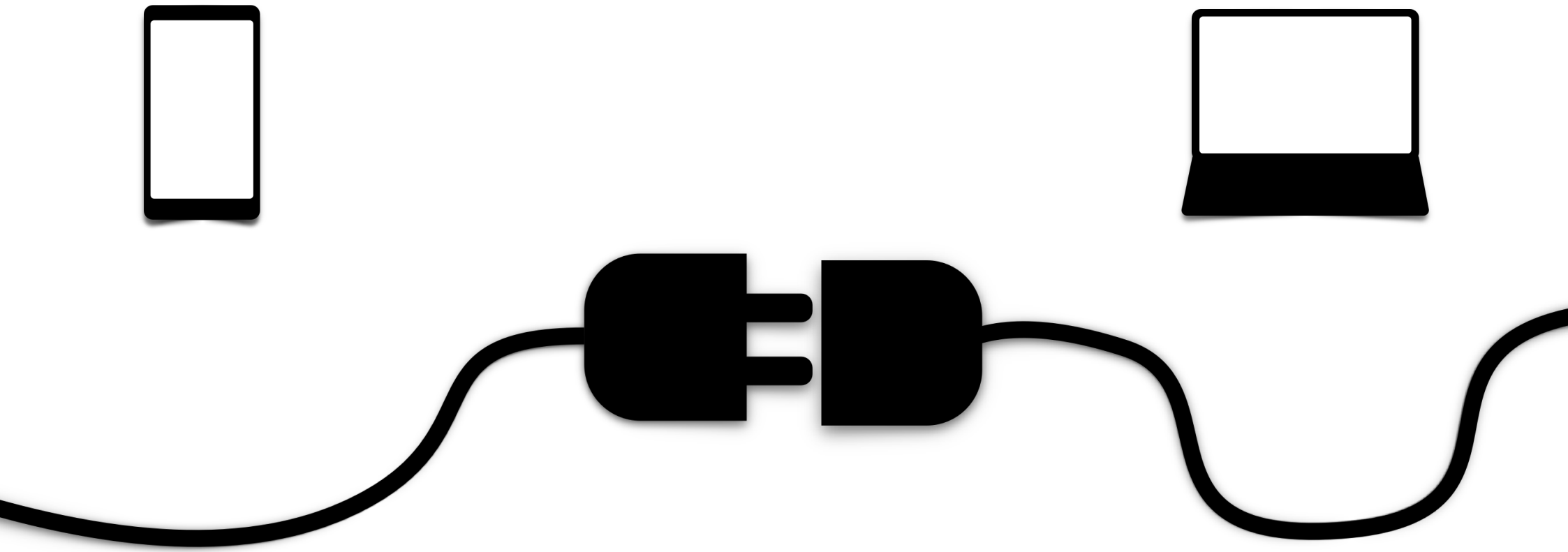
Lokale SDP setzen

```
peerConnection?.offer(
        for: mandatorySdpConstraints) {sdp, err in
            […]
            self.peerConnection?.setLocalDescription(
                sdp,
                completionHandler: { err in
                  // handle error
            })
            […]
        }
```

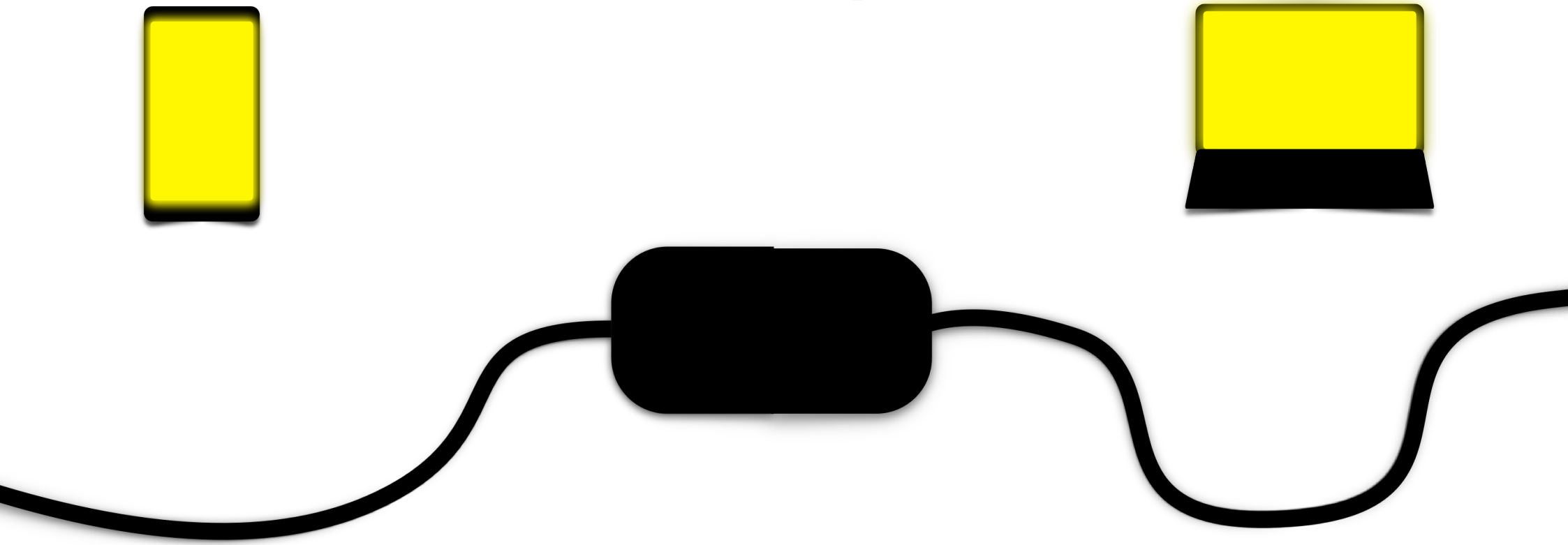
Offer versenden

```
peerConnection?.offer(
        for: mandatorySdpConstraints) { sdp, err in
            […]
            self.signalingClient?.send(
                offer: sdp,
                to: userId)
        }
```

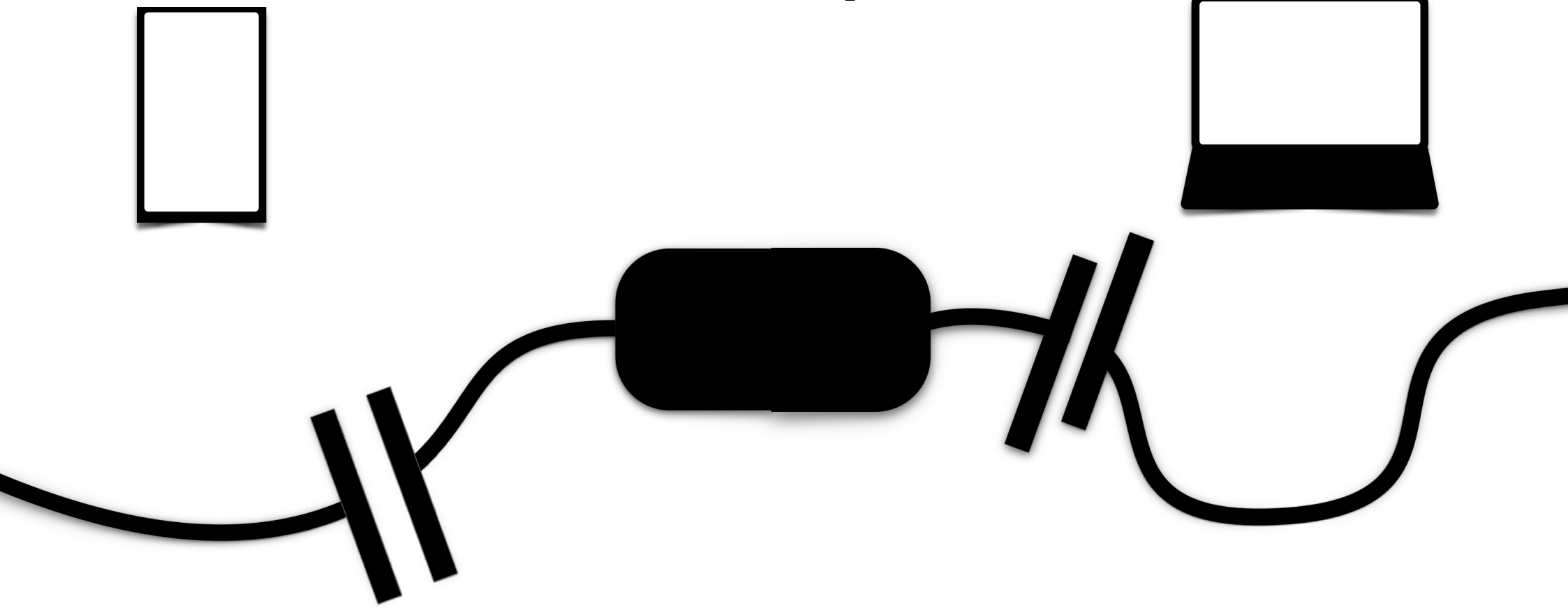
Für Answer analog



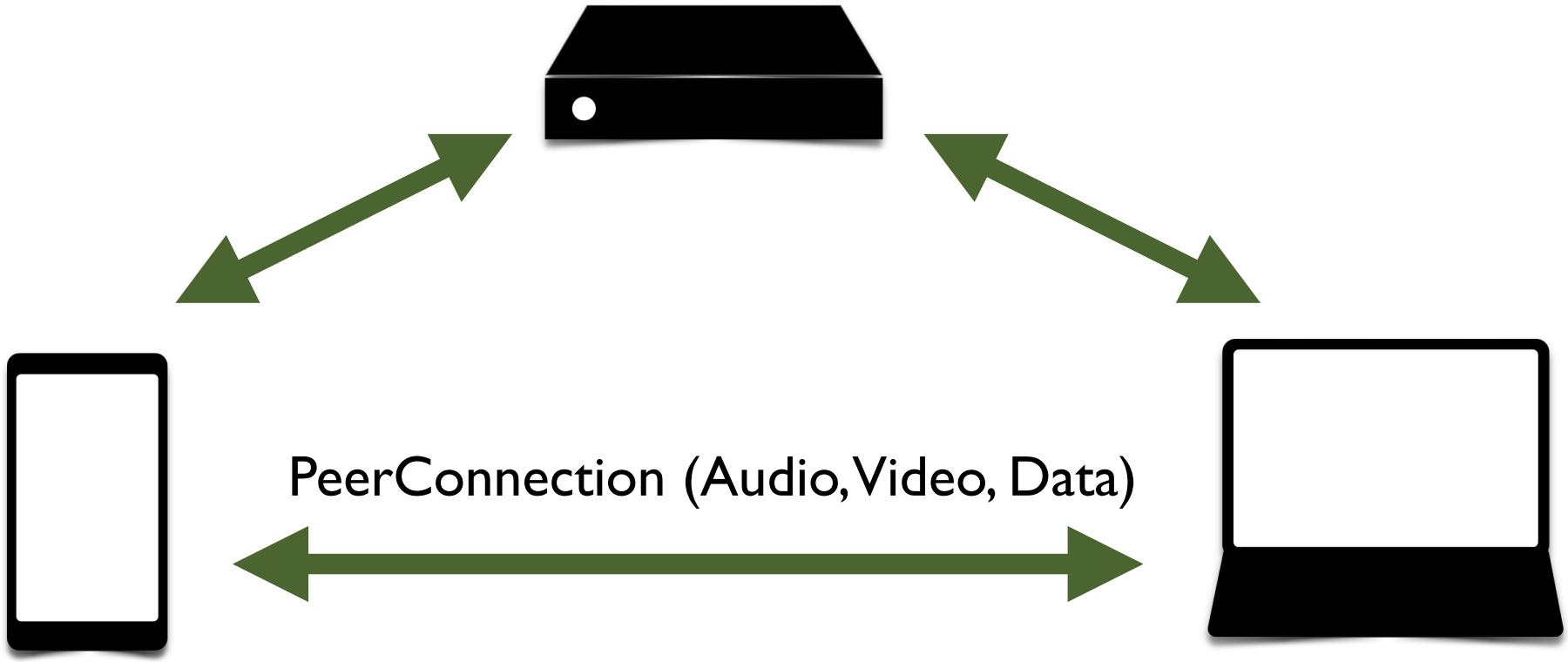
Hurray!!



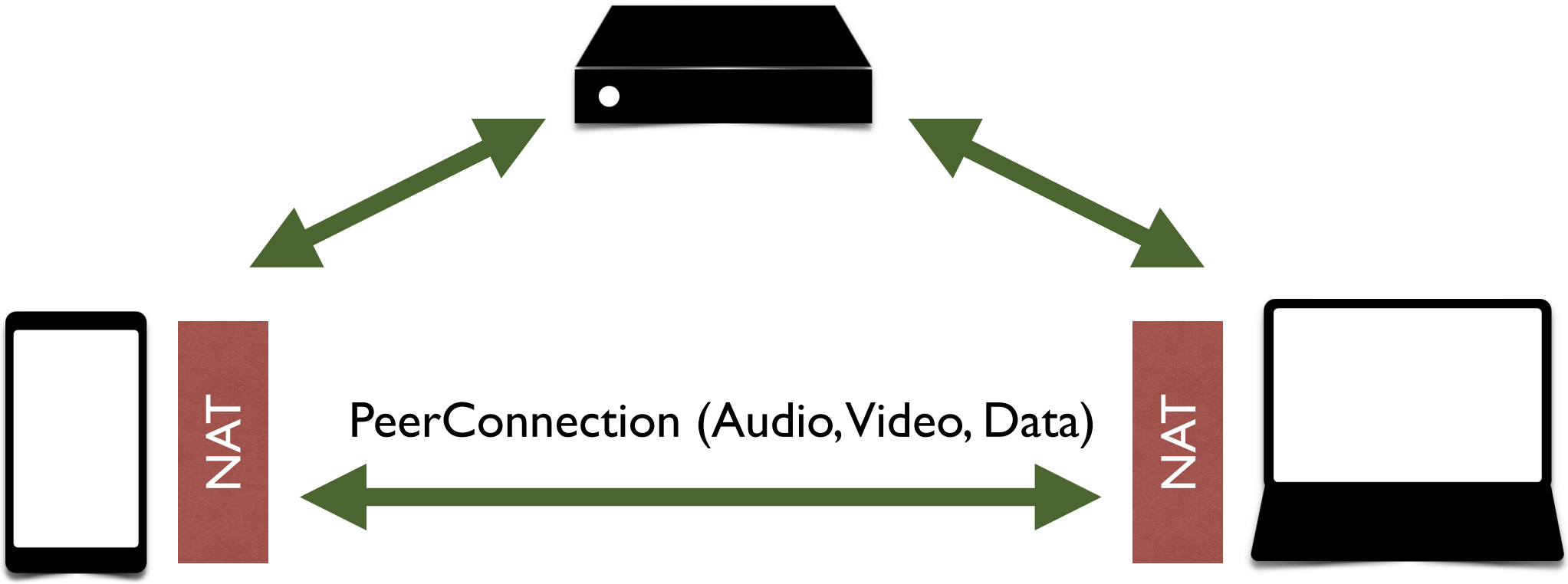
Hurray??



Signaling



Signaling

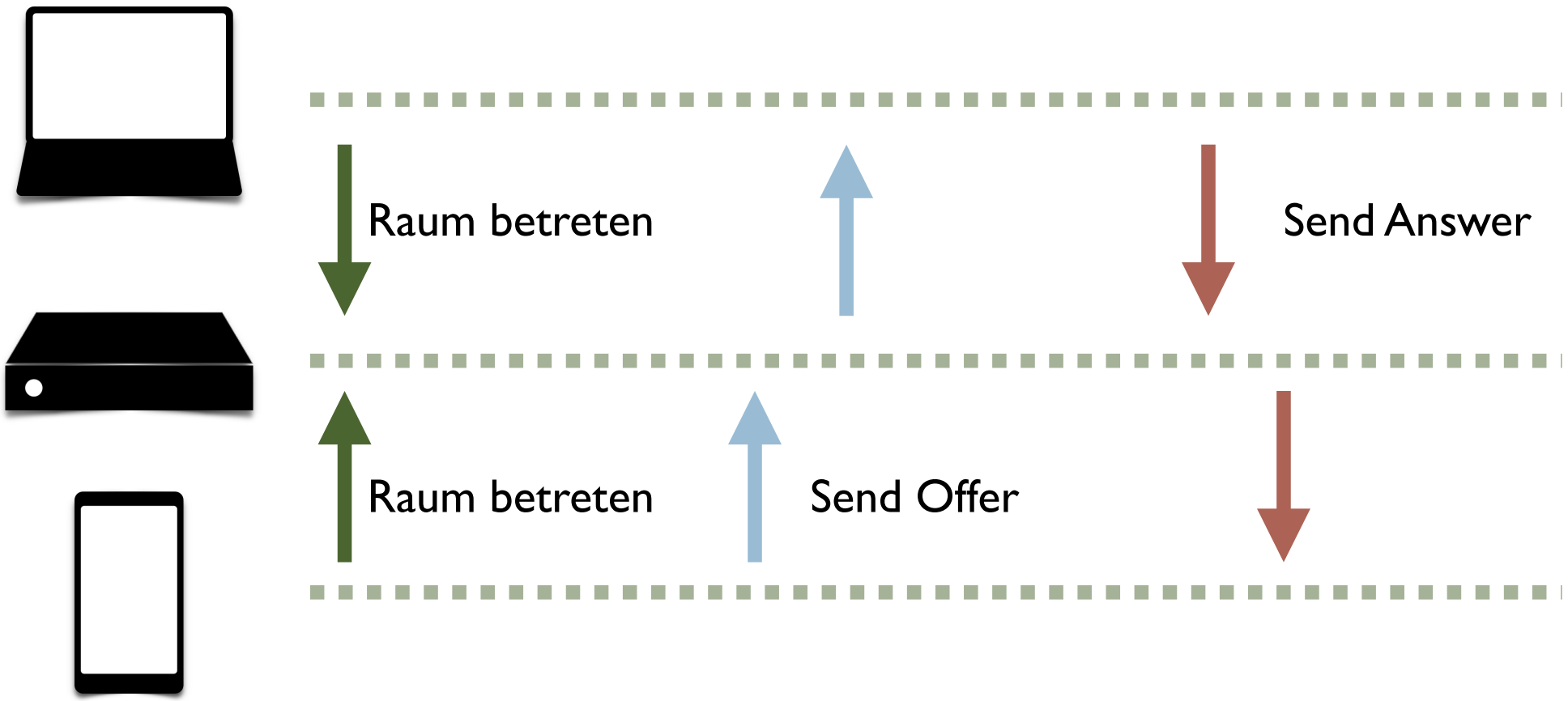


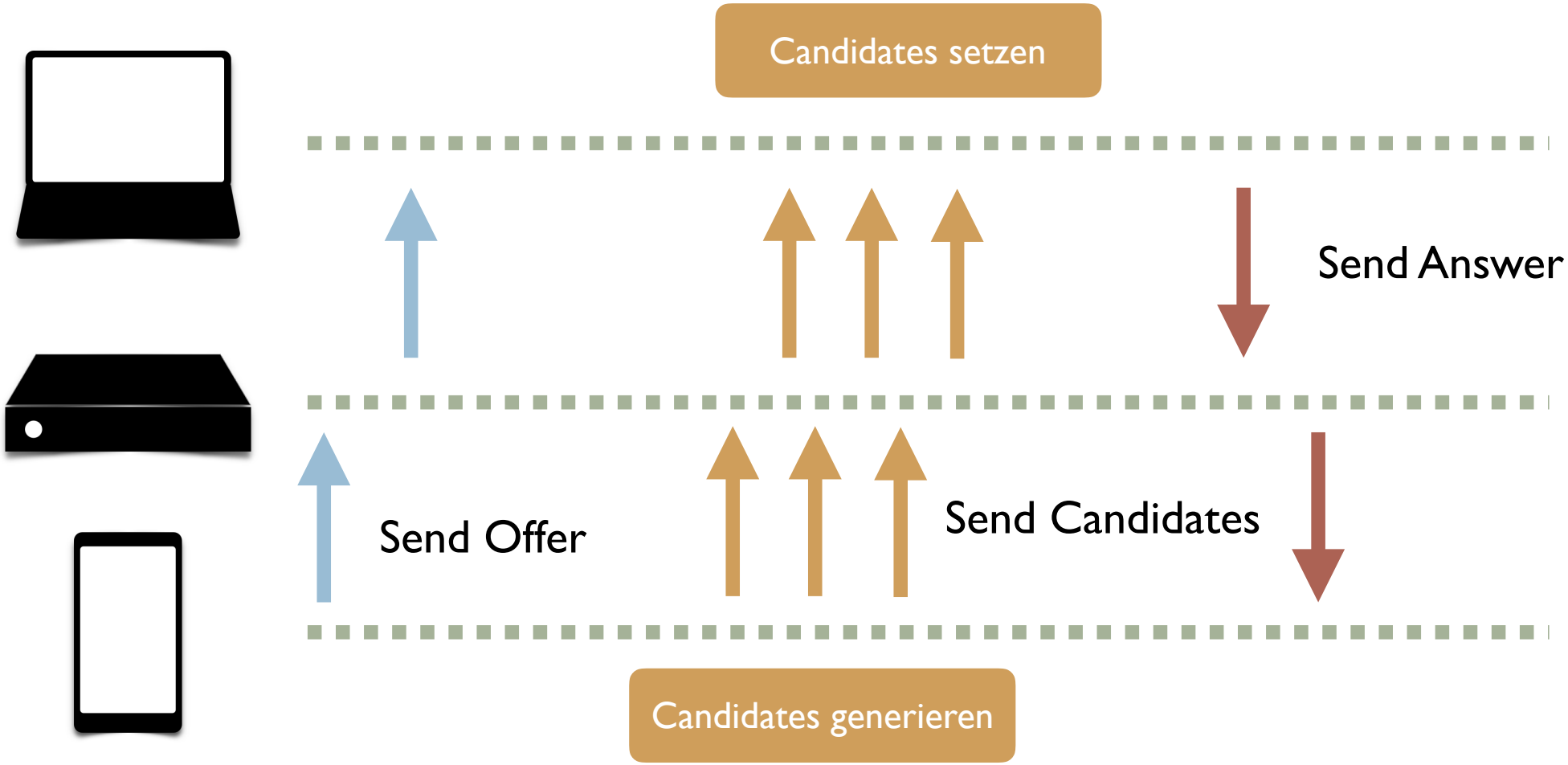
NAT (Network Address Translation)

- Bilden viele lokale IP-Adressen auf eine öffentliche ab
- Kümmern sich um die Zuordnung der ein- und ausgehenden Pakete
- Ein Gerät kann hinter mehreren NATs liegen
- Peer-to-Peer-Verbindungen müssen besonders behandelt werden

Interactive Connectivity Establishment (ICE)

- Methode zur Überwindung von NATs
- Bestimmung verfügbarer Adressen
- Bestimmung verfügbarer Protokolle (tcp, udp)


Raum betreten
Send Answer
Raum betreten
Send Offer


Candidates setzen
Send Answer
Send Offer
Send Candidates
Candidates generieren

Candidate

- Lokale Adresse
- Öffentliche Adresse, außerhalb des NATs
- Die Adresse eines Relay-Servers

Candidate

- Candidate-Paare erzeugen
- Priorisieren
- ICE-Checking durchführen

Candidate-SDP

Lokale Adresse

```
candidate:2240123056 1 udp 2122260223 10.199.1.127 58817 typ host
candidate:3406195776 1 tcp 1518280447 10.199.1.127 58341 typ host
```

Remote Adresse

```
candidate:1939204452 1 udp 1686052607 82.194.116.230 59344 typ srflx raddr
```

Relay Adresse

```
candidate:938911850 1 udp 41885439 173.212.192.23 56367 typ relay raddr
```

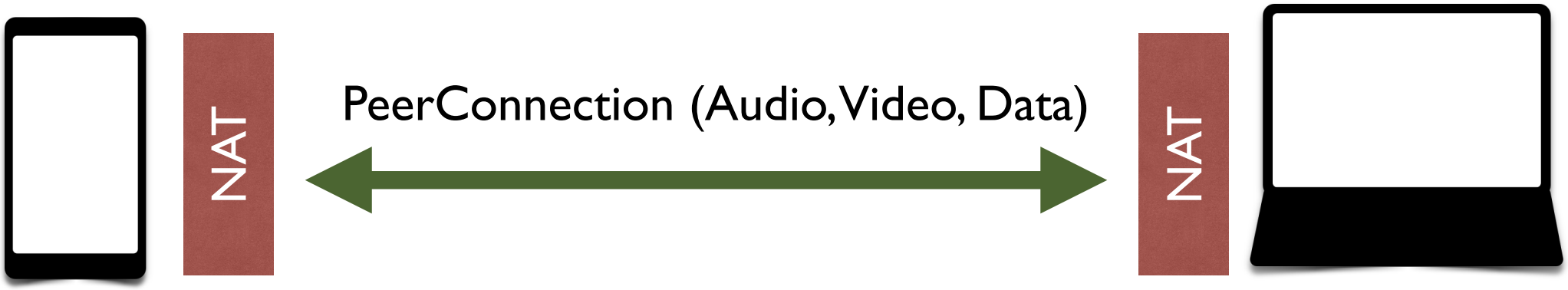
STUN

- Session Traversal Utilities for NAT
- Liefert öffentliche Adresse des Clients
- Bei mehreren NATs die Adresse des Äußersten
- `stun:awesomeurl.com:3478`

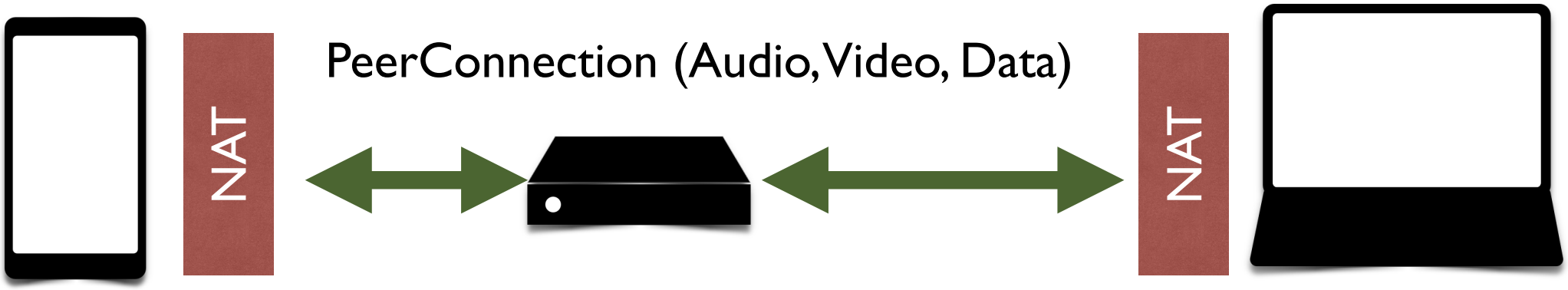
TURN

- Traversal Using Relay around NAT
- TURN steht zwischen Peers
- Ermöglicht Verbindung auch in schwierigen Situationen
- `turn:awesomeurl.com:3478?transport=udp`

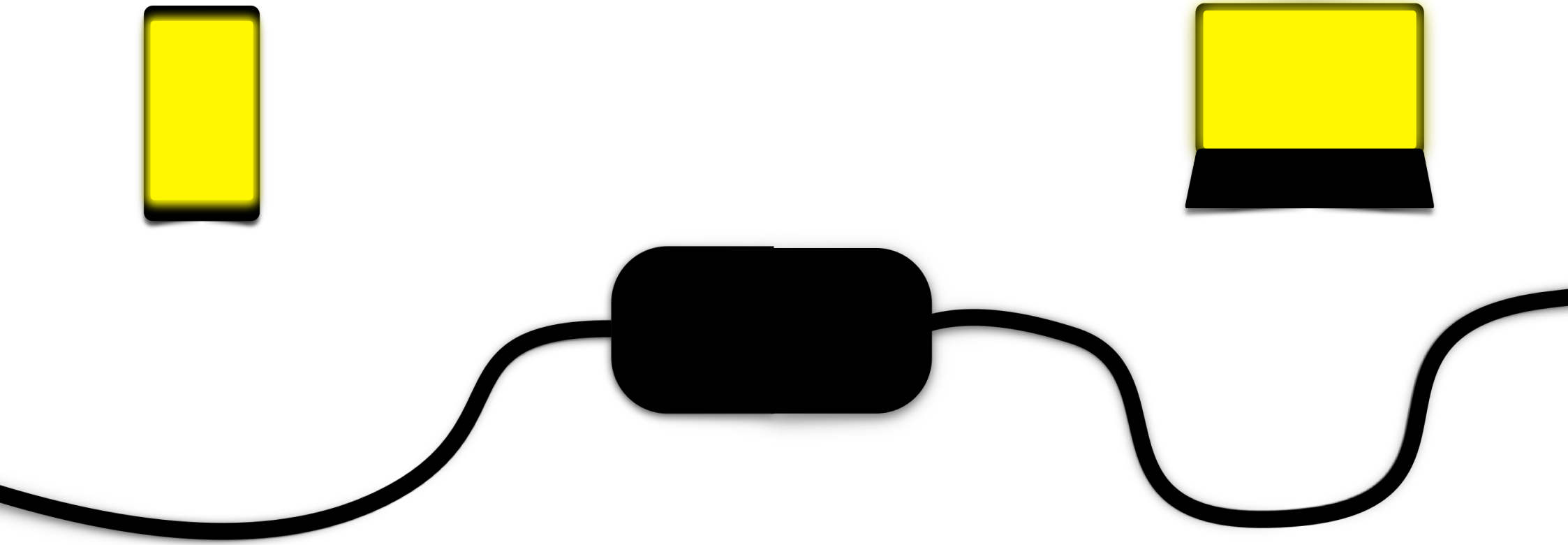
TURN



TURN



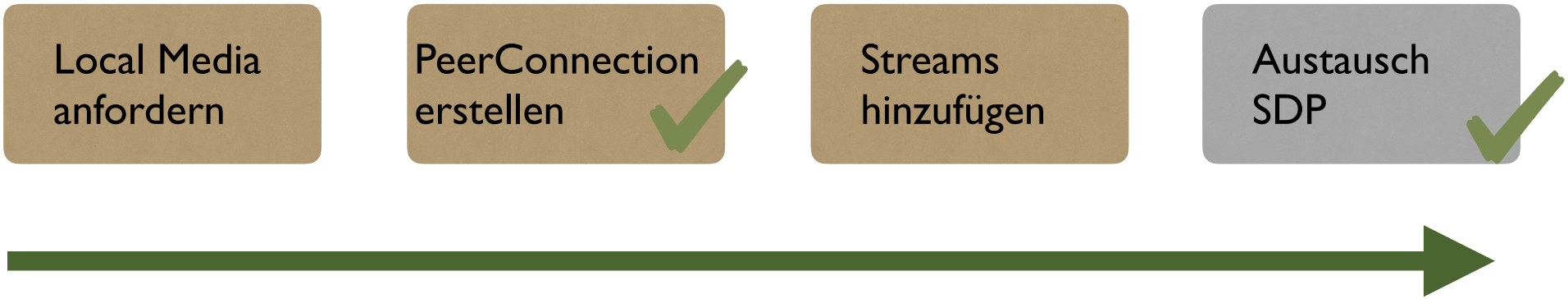
Hurray!!



Signaling

- Austausch der SDPs
- Ist nicht standardisiert
- STUN & TURN
- Reihenfolge der Operationen ist wichtig
- Sehr sicherheitskritisch!

WebRTC APIs


Local Media anfordern
PeerConnection erstellen
Streams hinzufügen
Austausch SDP

WebRTC APIs

Local Media anfordern

Local Media

- Grundlage sind `RTCMediaTrack`s
- Beschreibt Medienquelle eines Typen (Audio, Video)

Audio

```
let audioSource = peerConnectionFactory.audioSource(
          with: RTCMediaConstraints(
              mandatoryConstraints: nil,
              optionalConstraints: nil))
let audioTrack = peerConnectionFactory.audioTrack(
    with: audioSource,
    trackId: audioTrackId)
```

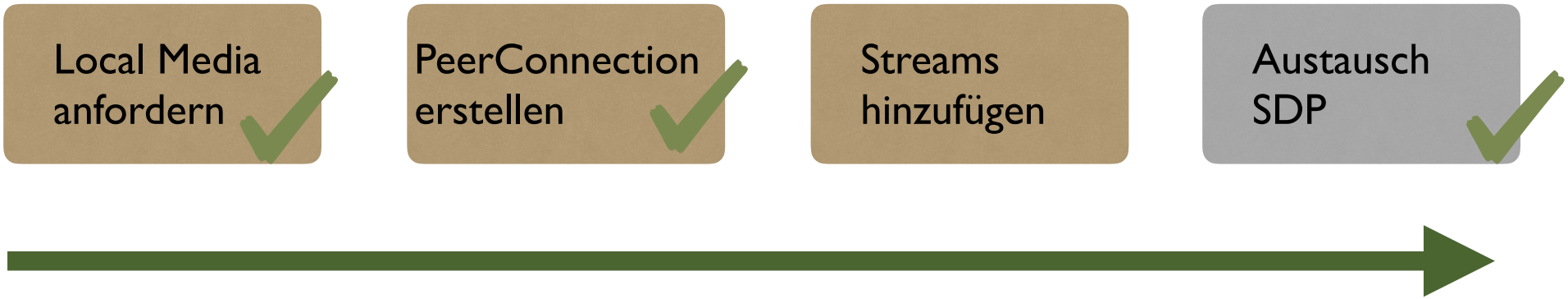
Video

```
let videoSource = peerConnectionFactory.videoSource()
let videoTrack = peerConnectionFactory.videoTrack(
    with: videoSource,
    trackId: videoTrackId)

localCapturer = RTCCameraVideoCapturer(delegate: videoSource)
guard let device = RTCCameraVideoCapturer
    .captureDevices()
    .first(where: {$0.position == .front}) else {
        fatalError()
}
[…]
```

Video

```
[…]
let format = RTCCameraVideoCapturer.format(
             for: device,
             constraints: config.formatConstraints)
localCapturer?.startCapture(
    with: device,
    format: format,
    fps: RTCCameraVideoCapturer.fps(for: format))
```

WebRTC APIs


Local Media anfordern
PeerConnection erstellen
Streams hinzufügen
Austausch SDP

WebRTC APIs

Streams hinzufügen

MediaStream

- Beinhaltet mehrere MediaTracks
- Wird über eine ID gekennzeichnet
- Der gleiche MediaTrack kann in mehreren MediaStreams vorkommen

MediaStream hinzufügen

```
let mediaStream = peerConnectionFactory
    .mediaStream(withStreamId: mediaStreamId)
mediaStream.addAudioTrack(audioTrack)
mediaStream.addVideoTrack(videoTrack)
peerConnection?.add(mediaStream)
```

MediaStream empfangen

```
public func peerConnection(
    _ peerConnection: RTCPeerConnection,
    didAdd stream: RTCMediaStream) {
    if let remoteVideoTrack = stream.videoTracks.first {
        self.remoteVideoTrack = remoteVideoTrack
    }
    if let remoteAudioTrack = stream.audioTracks.first {
        self.remoteAudioTrack = remoteAudioTrack
    }
}
```

DataChannel

- Peer-to-Peer-Datenübertragung
- Wird beim Peer automatisch aufgebaut außer anders verlangt
- Entweder mit Empfangsbestätigung oder ohne (SCTP)
- Nachricht sollte 16kiB nicht überschreiten

DataChannel erstellen

```
let dataChannelConfig =
RTCDataChannelConfiguration()
dataChannelConfig.isOrdered = true
// dataChannelConfig.isNegotiated = true

datachannel = peerConnection?.dataChannel(
    forLabel: "WhaleDataChannel",
    configuration: dataChannelConfig)

datachannel?.delegate = self
```

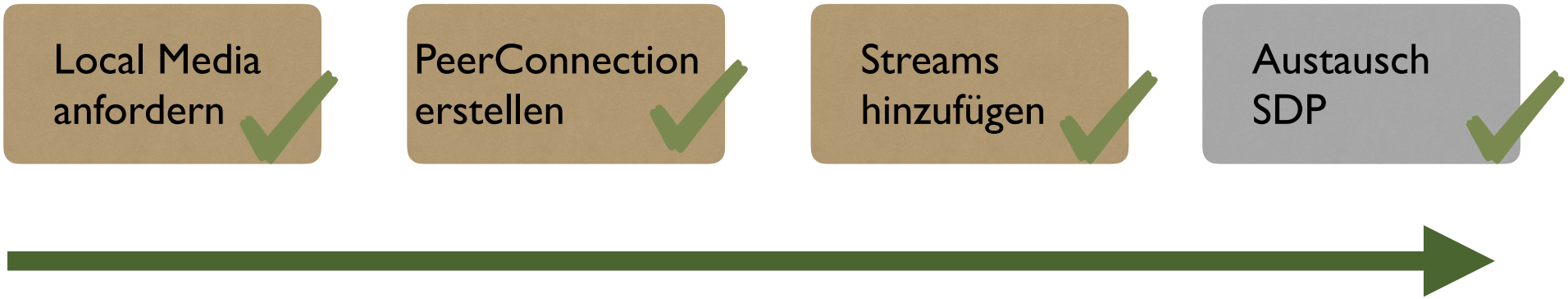
DataChannel bei Peer

```
public func peerConnection(
    _ peerConnection: RTCPeerConnection,
    didOpen dataChannel: RTCDataChannel) {
    self.datachannel = dataChannel
    self.datachannel?.delegate = self
}
```

Daten senden

```
datachannel?.sendData(
    RTCDataBuffer(data: data, isBinary: false)
)
```

WebRTC APIs


Local Media anfordern
PeerConnection erstellen
Streams hinzufügen
Austausch SDP

CallKit

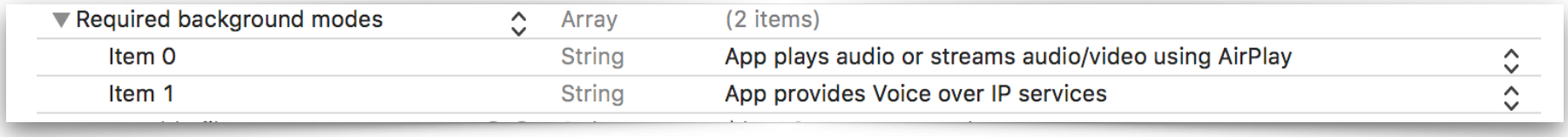
CallKit


SIM fehlt
20:10
Incoming Whale call
Whale-Video ...
Erinnerung
Ablehnen
Annehmen

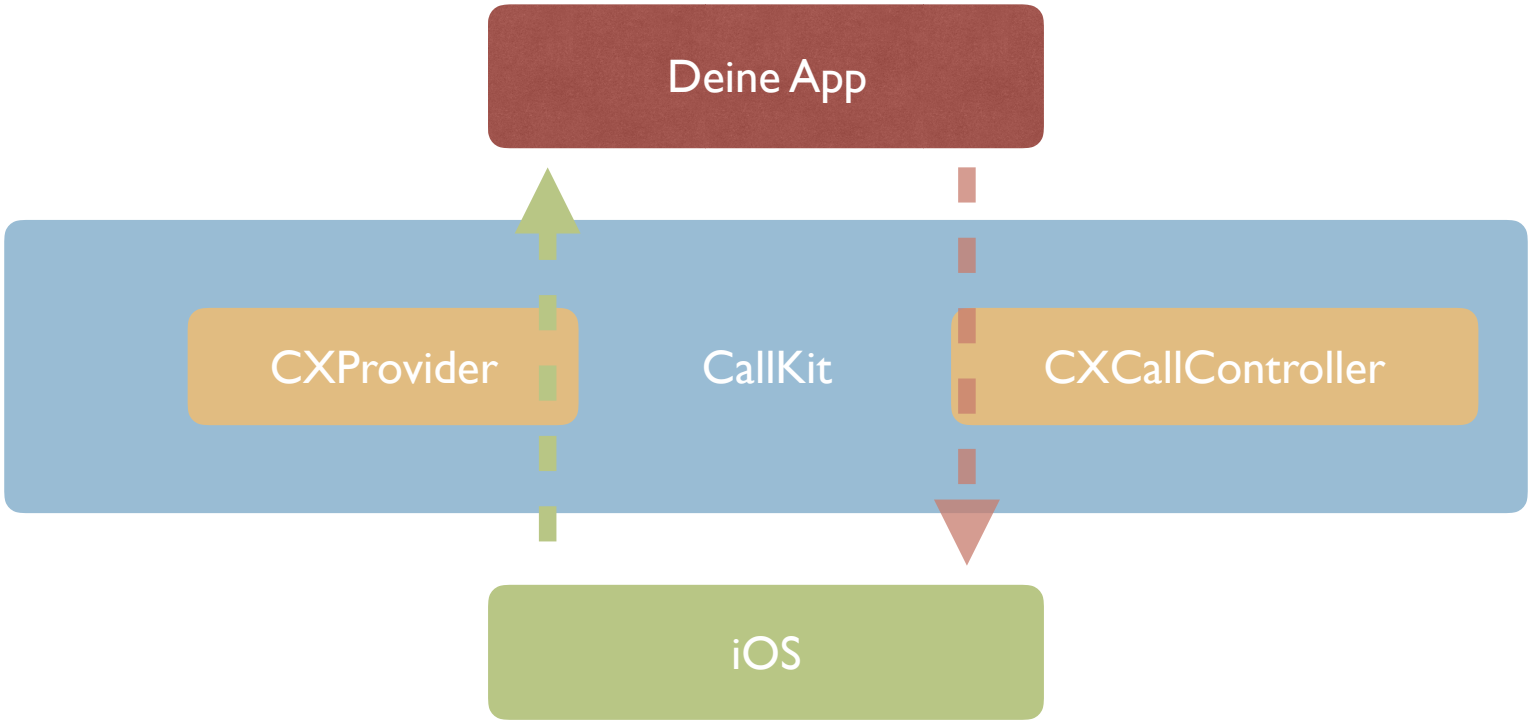

SIM fehlt
20:17
Zurück zum Anruf? Hier tippen! 00:22
Mittwoch
12
Mail
Kalender
Fotos
Kamera
Karten
Uhr
Wetter
Aktien
Wallet
Notizen
Erinnerungen
iTunes Store
App Store
iBooks
Videos
Home
Health
Einstellungen
Produktivität
Extras
MAApp

Projekt Vorbereitung

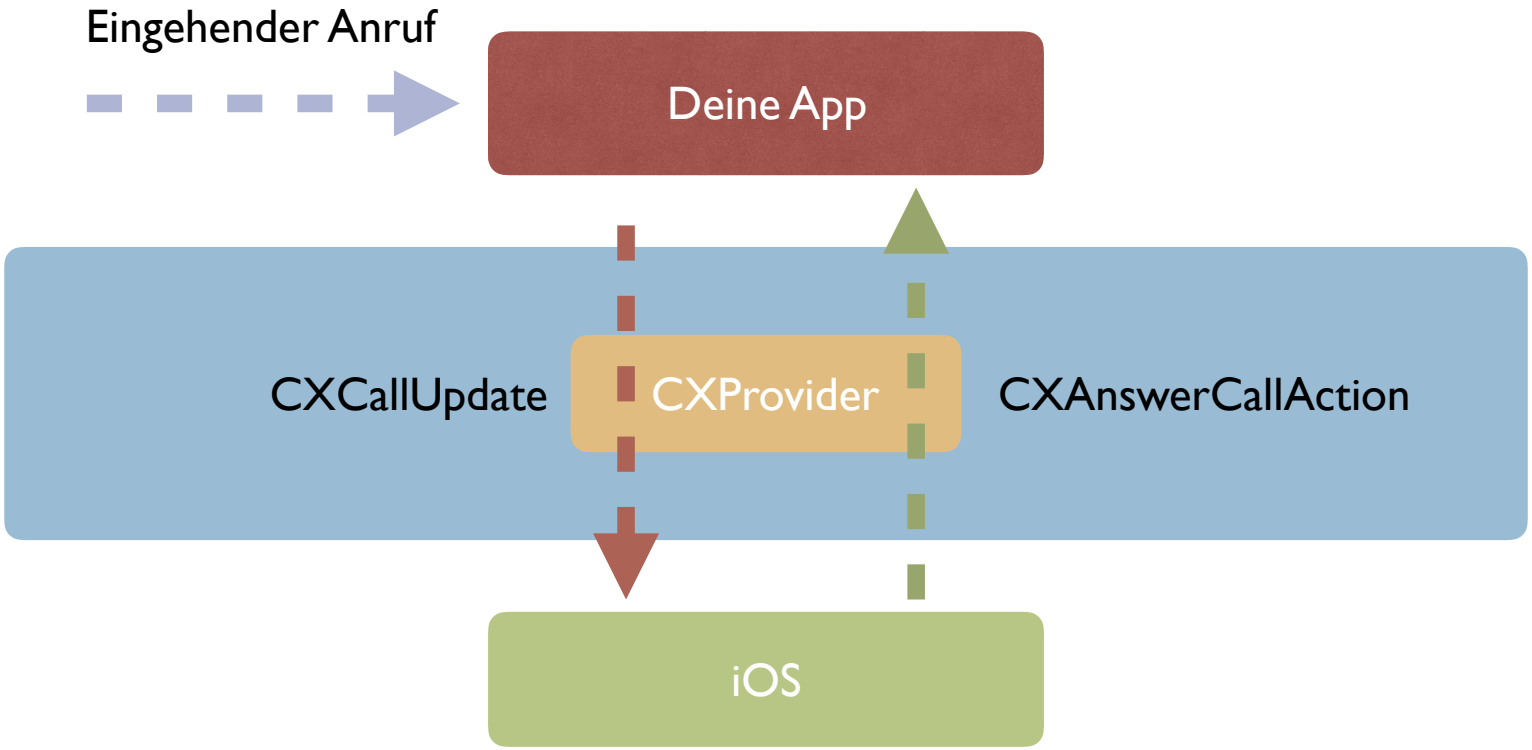
- Entitlement in Info.plist setzen (VoIP)



CallKit



Eingehender Anruf



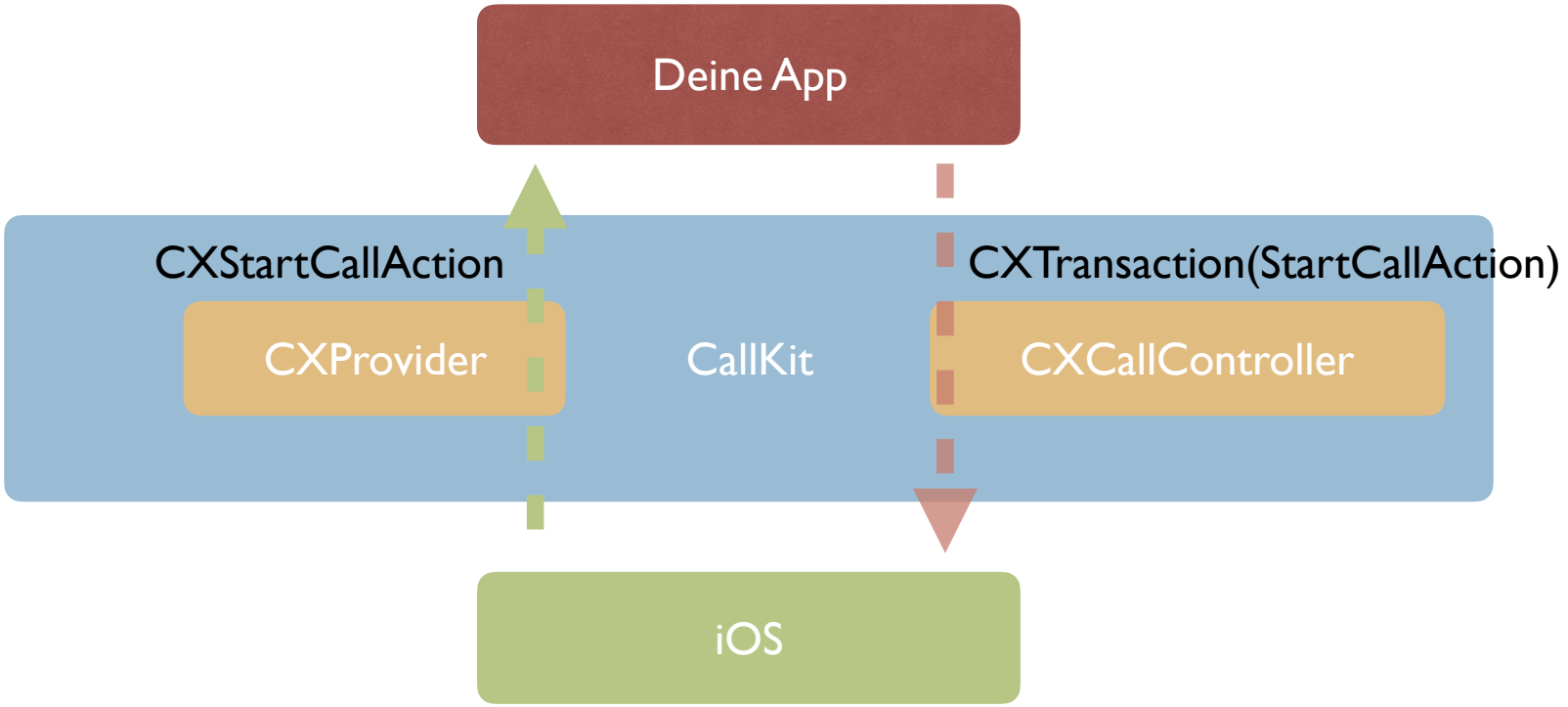
CXProvider

```
let callUpdate = CXCallUpdate()
callUpdate.remoteHandle = CXHandle(type: .generic, value: call.handle)
callUpdate.hasVideo = true

provider.reportNewIncomingCall(
    with: UUID(),
    update: callUpdate,
    completion: { _ in })
```

CXProviderDelegate

```
func providerDidReset(_ provider: CXProvider)

func provider(_ provider: CXProvider, perform action: CXStartCallAction)
func provider(_ provider: CXProvider, perform action: CXAnswerCallAction)
func provider(_ provider: CXProvider, perform action: CXEndCallAction)
func provider(_ provider: CXProvider, didActivate audioSession: AVAudioSession)
func provider(_ provider: CXProvider, didDeactivate audioSession: AVAudioSession)
func provider(_ provider: CXProvider, perform action: CXSetMutedCallAction)
func provider(_ provider: CXProvider, perform action: CXSetHeldCallAction)
```

Ausgehender Anruf



CXCallController

```
let cxhandle = CXHandle(type: .generic, value: call.handle)
let startCallAction = CXStartCallAction(call: call.uuid, handle: cxhandle)
startCallAction.isVideo = true
let transaction = CXTransaction(action: startCallAction)
callController.request(transaction) {error in}
```

Zu Beachten

- AudioSession muss für CallKit richtig konfiguriert werden

```
let session = AVAudioSession.sharedInstance()
try? session.setCategory(AVAudioSessionCategoryPlayAndRecord)
try? session.setMode(AVAudioSessionModeVideoChat)
try? session.setPreferredIOBufferDuration(0.005)
try? session.setPreferredSampleRate(44_100)
```

CallKit

- SystemUI für eigene VoIP App nutzen
- Anrufe Initiieren, Beenden, Annehmen

Was kann CallKit noch?

- Anrufer identifizieren
- Anrufer blockieren
- Sollte mit PushKit kombiniert werden

WebRTC

- Aufbau einer PeerConnection
- Hinzufügen von Streams
- Überwinden von NATs
- Nutzen des DataChannels

Was kann WebRTC noch?

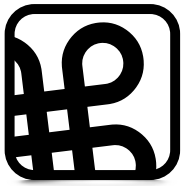
- Multi-User Call
- Kommunikation mit standardisierten VoIP Protokollen
- Nutzen von TRAM (Besserer TURN)
- Auslesen von Stats

Wie gehts weiter?

- https://github.com/kurzdigital/Whale
- WebRTC.org
- AppRTC WebRTC Demo für iOS
- Apples CallKit Demo Speakerbox

Fragen?

Vielen Dank



Macoun