**Abstract**

**Introduction, Problem, Idea**

* WebRTC auf Desktop-Browser läuft gut (wo es unterstützt wird), dasselbe gilt für Browser auf mobilen Geräten
* Um WebRTC auf mobilen Geräten möglichst verlässlich verwendbar machen zu können, bringt eine mobile App Vorteile (Zugriff auf Device Features wie z.B. Adressbuch, Kalender, Sensoren und Filesystem), zumal man einem Benutzer nicht einen bestimmten Webbrowser „aufzwingen“ kann.
* WebRTC in nativen Apps geht, erhöht aber den Entwicklungsaufwand erheblich (es muss die WebRTC-Funktionalität im jeweils nativen Code implementiert werden). WebRTC in nativen Apps mit WebView bringt dieselben Einschränkungen wie für WebRTC im Browser (keine iOS-Unterstützung).
* Cross Platform Development Frameworks können hier Abhilfe schaffen, um WebRTC auf iOS-Geräten auch verwenden zu können. Sie leiden teilweise aber auch unter dem Problem, dass sie nicht immer die aktuellste Technologie unterstützen
* deshalb Idee: Evaluierung der Möglichkeiten, um WebRTC verlässlich und möglichst flächendeckend in einer Cross Platform App verwenden zu können (welche Plugins/Frameworks gibt es?)
* Methode der Evaluierung der verschiedenen Möglichkeiten
* Aufbau der Bachelor-Arbeit

**Cross-Platform App Development**

* Verschiedene Arten des mobilen App Developments:
  + Native
  + Hybrid
  + Web App
* Unterschiede zwischen Native und Hybrid App Development (detailliert)
* Vor- und Nachteile der verschiedenen Arten
* Beispiele für Cross-Platform Frameworks
  + Apache Cordova, (PhoneGap)
  + Xamarin
  + Titanium
  + Ionic

**WebRTC**

* Überblick, Grundsätzliche Funktionen
  + Entstehungsgeschichte
  + Wie funktioniert es?
    - Skizze Verbindungsaufbau
  + Wichtigste Komponenten
    - Kamera-/Mikrofon-Zugriff
    - RTCPeerConnection
    - RTCDataChannel
  + Aktueller Entwicklungsstand, wer verwendet es schon?
* Vorteile
  + Alle Verbindungskomponenten 100% verschlüsselt
  + Im Browser ohne Plugin verwendbar
  + Durch P2P geringere Latenz, Daten laufen nicht über externen Server
* Nachteile
  + Nicht alle Browser/Betriebssysteme unterstützen es
  + Noch in Entwickling (API-Komponenten könnten sich ändern)
* Verbreitungsgrad, Unterstützung (iOS-Problem!)
  + Aktuelle Statistik (wie viele Geräte sind WebRTC-fähig, bezogen auf Browser und Apps)
* Möglichkeiten, WebRTC auf mobilem Gerät zu verwenden
  + Webanwendung im Browser (Browser-Support?)
  + Native App
  + Cross-Platform App
    - WebView (iOS/Android)
    - Crosswalk
    - OpenWebRTC

**Evaluation**

* Vergleich der Verwendungsmöglichkeiten von Cross-Platform App hinsichtlich
  + UX
  + Tools
  + Größe und Speicherbedarf der App
  + Umsetzungsdauer
  + (Setup Framework)
  + Zugriff auf Device Features

**Results**

**Outlook**

* Mögliche Erweiterungen
  + Ausbau des Webservers um User-Management-System, Authentifizierung
  + Erweiterung des Webservers und der App um Sessions mit mehr als zwei Benutzern zu ermöglichen (Multipoint Control Unit notwendig, die Streams verteilt)

**Conclusion**

**Sources:**

Hybrid Vs Cross-Platform Vs Native Application Development  
<http://www.blog.provab.co.in/hybrid-vs-cross-platform-vs-native-application-development/>

Developing mobile WebRTC hybrid applications  
<https://webrtchacks.com/webrtc-hybrid-applications/>

Shim to make WebRTC available on iOS  
<https://github.com/common-tater/wkwebview-webrtc-shim>