**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 gegenüber einer Web-App im Browser (Zugriff auf Device Features wie z.B. Adressbuch, Kalender, Sensoren und Filesystem), zumal man einem Benutzer nur schlecht 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 den aktuellesten Entwicklungsstand unterstützen, da WebRTC ständig weiterentwickelt wird
* 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
  + Anwendungsmöglichkeiten
    - Echtzeitkommunikation
    - P2P-Filesharing
  + 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) 🡪 Android würde funktionieren, iOS nicht
    - Crosswalk
    - OpenWebRTC

**Entwicklung**

* Aktueller Stand (was wurde in BA1 entwickelt)
  + Kurzübersicht über den Prototyp
    - Funktionsweise (Webanwendung, node.js Server, Verbindungsaufbau über WebSockets)
    - Stärken und Schwächen des Prototyps
* Was muss eine App können, um mit dem Prototyp-Server zusammenzuspielen?
  + Lediglich die WebSocket-Verbindung aufbauen und dann die WebRTC-Signalling-Funktionen aufrufen
  + Bei einer Cross Platform App kann derselbe JavaScript-Code verwendet werden, der für die Webanwendung entwickelt wurde 🡪 minimaler Entwicklungsaufwand in diesem Fall
  + Hauptaufgabe ist somit das richtige Setup des Frameworks
* Beschreibung der Entwicklung für die jeweiligen Apps
  + WebView
  + Crosswalk
  + OpenWebRTC

**Evaluation**

* Vergleich der drei Apps hinsichtlich (vorläufig)
  + Setup Framework
  + Umsetzungsdauer
  + UX (Navigationselemente, …)
  + Tools (wie einfach kann ich mit diesem Framework eine App generieren/deployen)
  + Größe und Speicherbedarf der App
  + 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 (vorläufig)**

*Book*

Loreto, S & Romano SP, 2014, *Real-Time Communication with WebRTC*, 1st edn., O’Reilly Media, Sebastopol. ISBN: 978-1-449-37187-6.

*Paper*

Angulo E & Ferre X, 2014, ‘A Case Study on Cross-Platform Development Frameworks for Mobile Applications and UX’. doi: 10.1145/2662253.2662280

Bertin E, Cubaud S, Tuffin S & Cazeaux S 2013, ‘WebRTC, the day after‘. In *17th International Conference on Intelligence in Next Generation Networks (ICIN).* IEEE. ISBN: 978-1-4799-0980-3. doi: 10.1109/ICIN.2013.6670893

Charkaoui S, Adraoui Z & Benlahmar EH, 2014, ‘Cross-platform mobile development approaches’. doi: 10.1109/CIST.2014.7016616

Charland A & Leroux B, 2011, ‘Mobile Application Development: Web vs. Native’. doi: 10.1145/1941487.1941504

Choudhary SR, Prasad M & Orso A, 2014, ‘Cross-Platform Feature Matching for Web Applications’. doi: 10.1145/2610384.2610409

Ciman M, Gaggi O & Gonzo N 2014, ‘Cross-Platform Mobile Development: A Study on Apps with Animations‘. In Proceedings of the 29th Annual ACM Symposium on Applied Computing (SAC '14). ACM, New York, NY, USA, 757-759. ISBN: 978-1-4503-2469-4. doi: 10.1145/2554850.2555104

Friberg J, 2014, ‘Evaluation of cross-platform development for mobile devices’. <http://www.diva-portal.se/smash/get/diva2:691708/FULLTEXT01.pdf>

Grégoire, JC 2015, ‘On Embedded Real Time Media Communications‘*.* In Proceedings of the 1st Workshop on All-Web Real-Time Systems (AWeS '15). ACM, New York, NY, USA, Article 7 , 4 pages. ISBN: 978-1-4503-3477-8. doi: 10.1145/2749215.2749224

Heitkötter H, Hanschke S & Majchrzak TA, 2012, ‘Comparing Cross-Platform Development Approaches for Mobile Applications’. <https://www.wi1.uni-muenster.de/pi/veroeff/heitkoetter/Comparing-Cross-Platform-Development-Approaches-for-Mobile-Applications.pdf>

Mesfin G, Ghinea G, Midekso D & Gronli TM, 2014, ‘Evaluating Usability of Cross-platform Smartphone Applications’. doi: 10.1007/978-3-319-10359-4\_20

Nurminen J K, Meyn A J, Jalonen E, Raivio Y & Marrero R G, 2013, ‘P2P media streaming with HTML5 and WebRTC’. IEEE International Conference on Computer Communications. doi: 10.1109/INFCOMW.2013.6970739

Singh K & Buford J, 2016, ‘Developing WebRTC-based Team Apps with a Cross-Platform Mobile Framework’. doi: 10.1109/CCNC.2016.7444762

Sredojev B, Samardzija D & Posarec D, 2015, ‘WebRTC technology overview and signaling solution design and implementation’. Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015 38th International Convention on. doi: 10.1109/MIPRO.2015.7160422

Tolkiehn G, Lebedev M & Makariti A, 2015, ‘MyWebRTC, a Free Do-It-Yourself Kit for Secure Real-Time Internet-Communication’. doi: 10.1109/ITIB.2015.7355058

Umuhoza E, Ed-douibi H & Brambilla M, 2015, ‘Automatic Code Generation for Cross-Platform, Multi-device Mobile Apps: Some Reflections from an Industrial Experience’. doi: 10.1145/2846661.2846666

Xanthopoulos S & Xinogalos S, 2013, ‘A Comparative Analysis of Cross-platform Development Approaches for Mobile Applications’. doi: 10.1145/2490257.2490292

Xinogalos S, Psannis KE & Sifaleras A, 2012, ‘Recent Advances Delivered by HTML5 in Mobile Cloud Computing Applications: A Survey’. doi: 10.1145/2371316.2371355

*Web*

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 (Juni 2015)  
<https://webrtchacks.com/webrtc-hybrid-applications/>

Shim to make WebRTC available on iOS (September 2015)  
<https://github.com/common-tater/wkwebview-webrtc-shim>

WebRTC Architecture  
<https://webrtc.org/architecture/>

How To Build an Android WebRTC Video and Voice Chat App (October 2015)  
<https://www.pubnub.com/blog/2015-10-13-build-android-webrtc-video-voice-chat-app/>