|  |
| --- |
| FH Joanneum |
| **Screensharing** |
| Seminararbeit Heterogene Systeme |

|  |
| --- |
| Kevin Pfeifer  [Datum] |

Inhalt

[1. Abstract 3](#_Toc442289908)

[2. Einleitung 4](#_Toc442289909)

[3. Airplay 5](#_Toc442289910)

[4. Intel WIDI 6](#_Toc442289911)

[5. AMD AWD 7](#_Toc442289912)

[6. Miracast 8](#_Toc442289913)

[7. Chromecast, Firetv 9](#_Toc442289914)

[8. Fazit 10](#_Toc442289915)

# Abstract

**Entweder den abgegebenen oder nach der seminararbeit neu formulieren**

**Und links 3,5 cm abstand, sonst 3 - erst am schluss machen für verlängernden effekt ;)**

Im Rahmen dieser Seminararbeit werden verschiedene Screenshare-Techniken behandelt und verglichen. Der Vergleich erfolgt zum einen nach der zu Grunde liegenden Technologie und Funktionsweise aber auch in Bezug auf die Interoperabilität zwischen verschiedenen Geräten und Plattformen. Weiters werden die Punkte Kosten und Einsatzgebiete behandelt.

Es soll ermittelt werden welche Software nur auf einer einheitlichen Plattform läuft, welche auf verschiedenen Plattformen laufen und welche Einschränkungen es dabei gibt (Performance, Funktionalität, Sicherheit, Workarounds). Dabei werden Informationen aus den einzelnen Spezifikationen, Erfahrungsberichten sowie Tests gewonnen.

Zusätzlich soll speziell für mobile Plattformen (Android und iOS) ermittelt werden welche Möglichkeiten zum Screenshare auf Desktop-, Mobile- sowie TV-Geräte vorhanden sind. Spezielles Augenmerk liegt dabei auf der Performance sowie den Einschränkungen der einzelnen Methoden auf mobilen Plattformen.

Diese Arbeit liefert eine Zusammenfassung und Übersicht über alle bearbeiteten Technologien, deren Vor- und Nachteile sowie deren effektiven Nutzen.

# Einleitung

Allgemeiner bullshit

Warum wir des thema behandeln

Welche software wir genau untersuchen

Auf was wir sie untersuchen

Was wir davon eigentlich wollen (gscheider vergleich, sinnvolle software)

Im prinzip eine erweiterung und präzisierung des abgegebenen abstracts

# AirPlay

Proprietäres protokoll für apple geräte

Zwischen mac, iphone, ipad etc.. für tv: spezielle set-top box

Airserver: für windows.. kann airplay und miracast streamen

AirPlay ist ein speziell von Apple entwickeltest Protokoll, welches das streamen von Audio und Video Daten zwischen unterstützten Geräten ermöglicht. Im Vergleich zu Bluetooth hat AirPlay eine weitaus höhere Datenübertragungsgeschwindigkeit (120 MBit/s) welche bei verlustfreien Audio- und Videoformaten benötigt werden. AirPlay löste 2010 das Vorgängerprojekt AirTunes ab. (GRAVIS 2011; Hoffman 2013)



Abbildung X: iPad und iPhone AirPlay Auswahl (Apple 2016)

Prinzipiell ist es ohne weitere Zusatzsoftware mögliche von allen mobilen Apple Geräten zu einem Apple TV oder eine AirPlay fähigen Gerät Videos bzw. Audio zu senden. Dies geschieht entweder über ein LAN oder über eine P2P-Verbindung. Lautsprechersysteme oder Fernsehapparate können diese AirPlay-Funktionalität auch direkt implementiert haben um kein weiteres Gerät, wie einen AirPort Express o.Ä. kaufen zu müssen. (Apple 2016)

Andere Betriebssysteme wie Windows müssen Zusatzsoftware wie 5KPlayer installieren um auf AirPlay-fähigen Geräten zu streamen oder als AirPlay-Empfänger zu dienen.

Durch zusätzliche Applikationen wie AirFloat können auch mobile Apple Geräte als AirPlay Empfänger dienen.

# Intel WIDI

## Ursprünge

Intels Wireless Display (WiDi) wurde ursprünglich entwickelt, um den Bildschirm eines Notebooks beziehungsweise einzelne Inhalte kabellos zu einem Monitor oder Projektor zu übertragen. Die erste Version wurde 2010 vorgestellt und zielte vor allem darauf ab, bei Präsentationen (Powerpoint, Slideshows) eingesetzt zu werden (vgl. Müssig 2010).

Wie sich bereits aus dem Namen erschließen lässt, wurde diese Technologie von Intel entwickelt. Um WiDi nutzen zu können wurde ursprünglich eine CPU der Arrendale Generation mit integrierter Grafikeinheit sowie ein Intel Centrino 6000 Wlan Modul benötigt um WiDi nutzen zu können. Intel integrierte nur in diese Treiber die entsprechenden Funktionen für WiDi. Durch diese Einschränkungen war es nicht möglich WiDi zu nutzen, wenn der Sender andere Prozessoren beziehungsweise Grafikeinheiten wie Nvidia oder AMD nutzte (vgl. Müssig 2010).

Um Daten per WiDi zu empfangen war eine eigene Set-Top Box notwendig. Diese dekodierte das per WiDi gesendete Signal und leitete dies per Kabel an das Endgerät weiter.

Die Probleme dieser ersten Versionen von WIDI waren zahlreich. Zum einen war die Anzahl an WIDI fähigen Sendern sehr begrenzt und zum anderen gab es nur wenige Empfänger, die zudem auch noch sehr teuer (damals ca. 100USD) waren. Zusätzlich war WiDi nur für Windows 7 und darüber verfügbar. Weiterhin war es nicht möglich, kopiergeschützte Daten wie Filme von BluRays zu übertragen, da WiDi High-Bandwith Digital Content Protection (HDCP) nicht unterstützte. Die Übertragungsqualität war außerdem auf 720p und Stereo Sound beschränkt. Zudem entstand bei der Nutzung von WiDi eine Latenzzeit im Bereich von mehreren Sekunden. Die Latenzzeit entspricht der verzögerten Darstellung beim Empfänger aufgrund der zum Übertragen benötigten Zeit. Eine hohe Latenzzeit ist vor allem bei Anwendungen welche Benutzereingaben verlangen störend (vgl. Müssig 2010; Ziesecke o.J.).

Da WiDi die Grafikeinheit der CPU nutzt, wird diese sehr stark belastet. Vor allem bei schwächeren Prozessoren (niedrige Taktrate, kleine Caches) kann dies zu Problemen führen (vgl. Paine 2014).

## Entwicklung

In den darauffolgenden Generationen wurden viele dieser Probleme beseitigt. So wurde die Latenzzeit verbessert. Dadurch wurde sowohl das Arbeiten mit WiDi vereinfacht als auch das Anzeigen von Spielen ermöglicht. Zusätzlich unterstützt WiDi ab der zweiten Generation auch HDCP, wodurch das Abspielen von kopiergeschütztem Material ermöglicht wird. Auch in Sachen Qualität wurde WiDi verbessert. Neben 5.1 Surround Sound wird auch die Übertragung von höheren Auflösungen ermöglicht (vgl. Ziesecke o.J.).

WiDi ist weiterhin nur für Windowsgeräte mit entsprechender Intel Hardware verfügbar, wobei jetzt wesentlich mehr Prozessoren und Grafikeinheiten unterstützt werden. Ob ein Gerät WiDi unterstützt kann mit einem Tool[[1]](#footnote-1) festgestellt werden.

Ab der Version 3.5 unterstützt WiDi den Screensharing Standard Miracast und gilt als eine konkrete Implementierung dessen. Ab Windows 8.1 benützt WiDi Prozesse des Betriebssystems um Miracast Verbindungen aufzubauen (vgl. Länger 2015).

Der Vorteile der Implementierung von Miracast liegt in der breiteren Unterstützung von Empfänger Geräten. Da für Miracast wesentlich mehr Empfänger verfügbar sind, fällt es den Benutzern leichter, ein passendes Empfangsgerät zu finden. Allerdings implementiert nicht jedes Gerät dieselben Aspekte von Miracast, was oft zu Inkompatibilität führt (Siehe Kapitel 6).

Zusätzlich zu normalem WiDi hat Intel auch noch eine Pro Version speziell für Business Kunden entwickelt. Diese inkludiert zusätzliche Sicherheitsfeatures sowie Kollaborationstools (vgl. Paine 2014).

Generell ist WiDi eher dafür konzipiert, einen Notebook- oder Tabletbildschirm auf einen größeren Monitor oder Beamer mit entsprechendem Receiver (integriert oder extern) zu übertragen, nicht aber um Bildschirme zwischen beispielsweise zwei Notebooks zu übertragen.

Da die Verfügbarkeit auf Windows Geräte beschränkt ist, und die Implementierung von Miracast nicht zwangsläufig Kompatibilität mit allen Miracast Receivern bedeutet, ist WiDi in Bezug auf heterogene Systeme eher ungeeignete Lösung zu betrachten.

# WiGig, WHDI, WiFi Display, Wireless HD

Wifi display: untermenge von wifi direct

Whdi: wireless home digital interface: von geräten auf fernseher, von vernseh herstellern

Wigig: erweiterung von wifi direct: sehr geringe latenz durch hohe frequenz, dadurch aber auch geringe reichweite

# Miracast

## Allgemeines

Miracast wurde designed um den Bildschirm eines Smartphones, Tablets oder PCs auf einem anderen Gerät, beispielsweise einem Fernseher, wiederzugeben. Dabei wird keine direkte Kabelverbindung benötigt, die Übertragung erfolgt kabellos. Im Vergleich zu Apple’s AirPlay und Google’s Chromecast wurde Miracast als cross-platform standard entwickelt und sollte auf vielen unterschiedlichen Geräten laufen.

Da Miracast nur den Bildschirm des Senders wiedergibt muss dieser für die gesamte Übertragung aktiv sein. Auf einem Smartphone ein Video von Netflix zu starten, dieses per Miracast auf den Fernseher übertragen und den Smartphone Bildschirm für etwas anderes nutzen oder sperren funktioniert nicht. Geht der Smartphone Bildschirm aus wird es auch am Fernseher dunkel. Miracast wirkt sich dementsprechend stark auf die Laufzeit des Sendegerätes aus.

Miracast kann durchaus mit einem wireless HDMI Kabel verglichen werden: Es fehlen die „smarten“ Funktionen von anderen Protokollen, welche es ermöglichen Inhalte auf andere Geräte zu streamen und zeitgleich den Bildschirm des Smartphones oder Tablets für andere Funktionen zu verwenden. (vgl. Chris Hoffman 2014)

Die Idee, einen einheitlichen Standard für das teilen eines Bildschirmes zu entwickeln, ist großartig, an der Umsetzung scheitern bisher jedoch alle Hersteller. Obwohl Miracast als einheitlicher Standard entwickelt wurde gibt es nur wenige Geräte die Miracast unterstützen und dabei keinerlei Einarbeitungszeit des Nutzers benötigen. Trotz Miracast Zertifizierung funktionieren viele Geräte von unterschiedlichen Herstellern nicht mit anderen zertifizierten Geräten wie in Abbildung 6.1 zu erkennen ist. Dabei wurden verschiedene Smartphones als Sender mit unterschiedlichen Empfängern getestet. Für einen Standard der von jedem Gerät unterstützt werden sollte ist das Ergebnis erschreckend.

Miracast ist aktuell ein Standard der kein Standard ist. Dies fällt schon auf den Verpackungen von Fernsehern auf. Bei LG nennt sich Miracast „SmartShare“, bei Samsung „AllShare Cast“, bei Sony „screen mirroring“ und bei Panasonic „display mirroring“. Potentielle Käufer werden durch die verschiedenen Begriffe unnötig verwirrt und finden ohne zusätzliche Recherche selten heraus, dass damit Miracast gemeint ist.

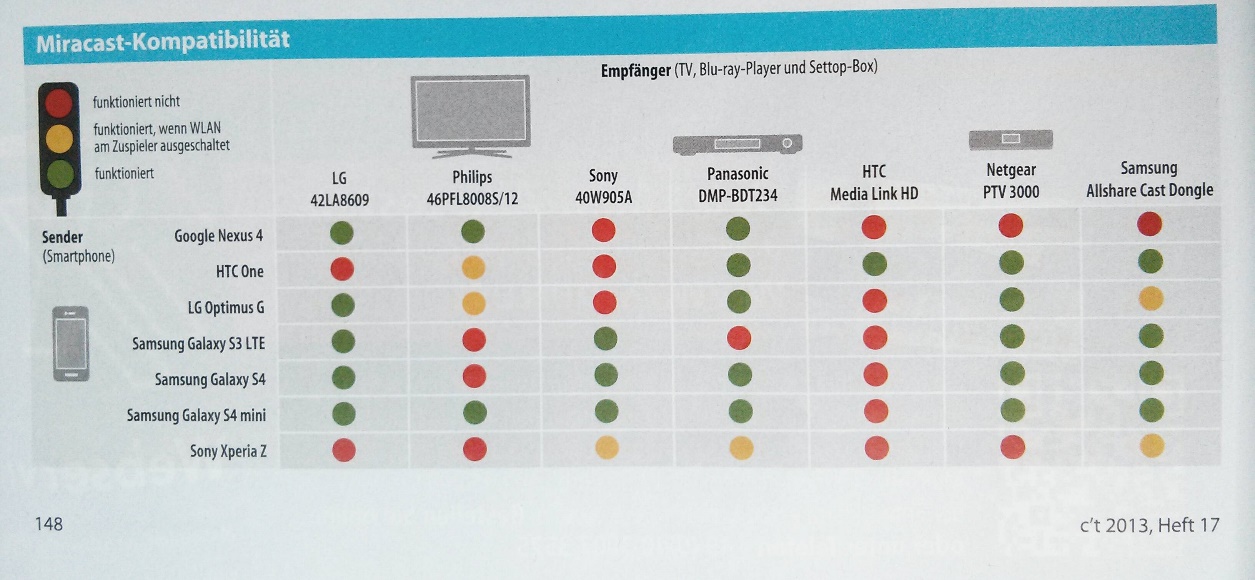


Abbildung 6.1 Miracast Kompatibilität (Czerulla and Hansen 2013)

## Technisches

Miracast wird von der Wi-Fi Alliance, einem 17 Jahre alten Netzwerk hunderter Unternehmen mit dem Ziel Standards rund um Wi-Fi zu entwickeln, als kabellose Lösung zum Teilen eines Bildschirmes vorgestellt.

Miracast baut auf Wi-Fi Direct auf. Wi-Fi Direct ermöglicht es zwei Geräten eine direkte, peer-to-peer Verbindung aufzubauen um dadurch automatisch wireless router zu finden und sich mit diesen zu verbinden. Miracast ist somit nicht von einem Netzwerk abhängig sondern erstellt ein eigenes Netzwerk zwischen den verbundenen Geräten. Für die Übertragung von 1080p Video und 5.1 Surround Sound wird der H.264 Codec verwendet. Desweiteren wurde in Miracast ein DRM Layer implementiert um Copyright geschützte Inhalte, beispielsweise DVDs oder Musik, vom Sender zum Empfänger zu senden. (vgl. Betters 2015)

Um Wi-Fi Display (WFD) und Miracast nutzen zu können gibt es einige Grundvoraussetzungen für WFD (Wi-Fi Display) Geräte:

* 802.11n Zertifizierung (impliziert WPA2)
* Wi-Fi Protected Setup Zertifizierung
* Wi-Fi Direct Zertifizierung

Um Wi-Fi Display über ein „Tunneled Direct Link Setup“ (TDLS) zu unterstützen muss das WFD Gerät zusätzlich folgende Zertifizierung besitzen:

* Wi-Fi TDLS Zertifizierung

## Encoder/Decoder

##### Video

Ein WFD fähiges Gerät sollte den H.264 Video Codec unterstützen. Des Weiteren sollte das Gerät eine Auflösung von 640x480 p60 mit dem Constrained Baseline Profile (CBP) Codec von H.264 auf Level 3.1 unterstützen. Wenn ein Gerät höhere Auflösungen unterstützen soll dann muss die Auflösung 720x480 p60 betragen und der H.264 CBP Codec zwischen Level 3.1 und 4.2 verwendet werden. (vgl. Wi-Fi Alliance)

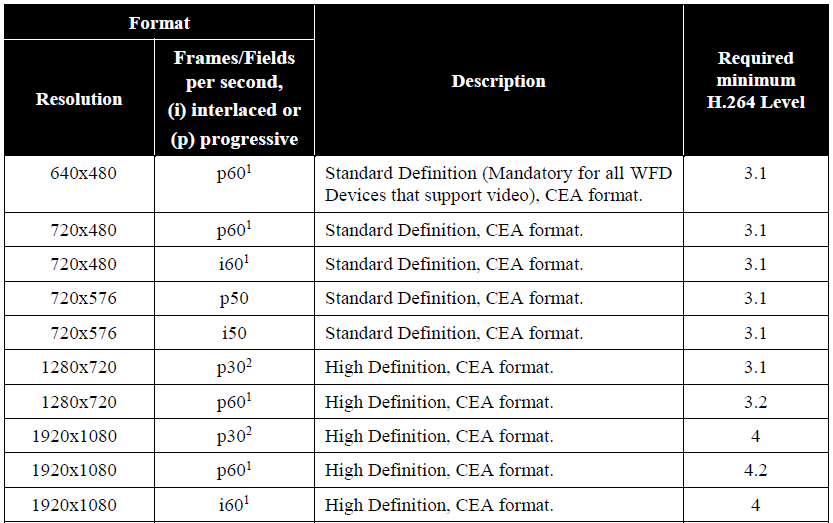


Abbildung 6.2 Ausschnitt der unterstützten Video Auflösungen und Bildwiederholraten (Wi-Fi Alliance)

##### Audio

Ein WFD Gerät das Audiowiedergabe ermöglicht sollte ein 2 Channel Linear Pulse Coded Modulation (LPCM) Audiosignal mit 16 bits pro Sample und 48000 Samples pro Sekunde unterstützen. Weitere unterstützte Audioformate sind in Abbildung 6.3 zu erkennen. (vgl. Wi-Fi Alliance)

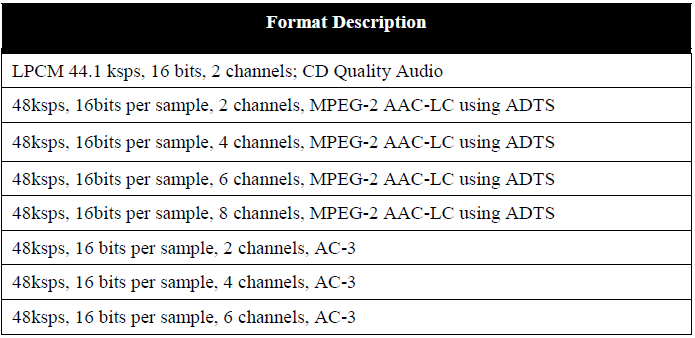


Abbildung 6.3 Unterstützte Audioformate (Wi-Fi Alliance)

## Verbindungsaufbau und Management

Generell folgt der Verbindungsaufbau für eine WFD Verbindung folgenden Schritten:

1. **WFD Device Discovery**: Ein WFD Sender und Empfänger erkennen sich gegenseitig bereits bevor ein Verbindungsversuch unternommen wird.
2. **WFD Service Discovery**: Optionaler Schritt, ermöglicht es einem Sender oder Empfänger zu erkennen, welche Services vom Gegenüber unterstützt werden.
3. **Device Selection**: Der Sender oder Empfänger wählt das Gerät zu dem er sich verbinden möchte. Eingaben des Benutzers oder lokale Regeln können verwendet werden um die Geräteauswahl zu steuern.
4. **WFD Connection Setup**: Dieser Schritt wählt die Verbindungsmethode (Wi-Fi P2P oder TDLS) und richtet einen WPA2 gesicherten single hop link.
5. **WFD Capability Negotiation**: Sender und Empfänger übertragen eine Reihe von Real Time Streaming Protocol (RTSP) Nachrichten um festzustellen, welche Größe die Audio/Video Nutzlast während einer WFD Session maximal erreichen darf.
6. **WFD Session Establishment**: Der WFD Sender wählt das Audio/Video Format für die WFD Session und informiert den Empfänger darüber.
7. **User Input Back Channel Setup**: Optionaler Schritt, ermöglicht es Kontroll- und Informationsdaten via Userinput vom Empfänger an den Sender zu übermitteln.
8. **Link Content Protection Setup**: Optionaler Schritt, ermöglicht das Übertragen von geschützten Inhalten.
9. **Payload Control**: Kontrolliert den Fluss der Payload um Verbindungsprobleme zu erkennen.
10. **WFD Source and WFD Sink standby**: Optionaler Schritt, erlaubt es WFD Sender und WFD Empfänger Stromsparmodi zu aktivieren oder deaktivieren während eine WFD Session aktiv ist.
11. **WFD Session Teardown**: Terminiert die WFD Session.

##### WFD Capability Negotiation

Die Einigung auf die maximale Größe der Audio/Video Payload stellt einen der zentralen Punkte des gesamten Verbindungsaufbaus und Managements dar. Aus diesem Grund wird im Folgenden diese Einigung genauer erklärt.

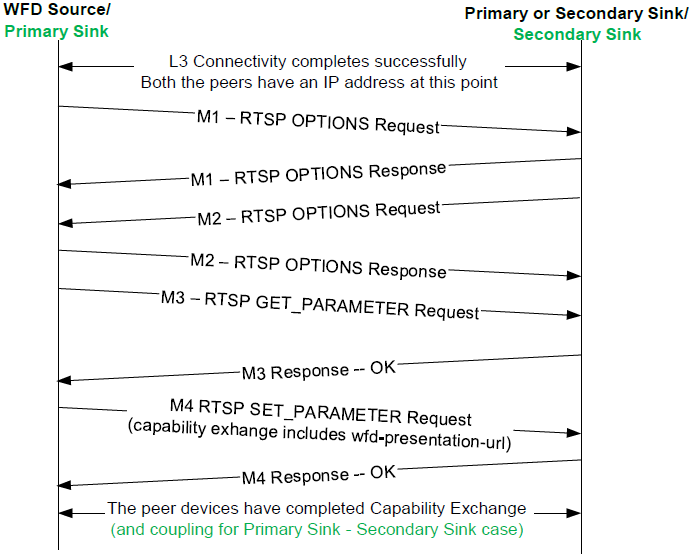


Abbildung 6.4 WFD Capability Negotiation Flow using RTSP (Wi-Fi Alliance)

Der 5. Schritt im Verbindungsaufbau ist die WFD Capability Negotiation. Dabei werden 8 RTSP Nachrichten gruppiert jeweils zu Anfrage und Antwort zwischen WFD Sender und WFD Empfänger ausgetauscht. Anhand von Abbildung 6.4 wird die WFD Capability Negotiation näher erklärt.

* **RTSP M1 Messages**: Der WFD Sender sendet eine RTSP OPTIONS Anfrage um die unterstützten RTSP Methoden des Empfängers zu bestimmen. Bei Erhalt eines RTSP M1 Paketes sendet der WFD Empfänger eine Liste aller unterstützten RTSP Methoden an den WFD Sender.
* **RTSP M2 Messages**: Im Grunde das exakt selbe Verhalten wie bei einem RTSP M1 Paket, jedoch sendet dieses Mal der WFD Empfänger die RTSP OPTIONS Anfrage an den Sender und erhält eine Liste mit allen verfügbaren Methoden als Antwort.
* **RTSP M3 Messages**
  + **RTSP M3 request:** Der WFD Sender sendet eine RTSP GET\_PARAMETER Anfrage an den WFD Empfänger. Diese beinhaltet eine Liste aller WFD Möglichkeiten an denen der Sender interessiert ist.
  + **RTSP M3 response:** Der WFD Empfänger antwortet mit einem RTSP GET\_PARAMETER Antwortpaket.
* **RTSP M4 Messages**
  + **RTSP M4 request:** Aufgrund der RTSP M3 response entscheidet sich der WFD Sender für die optimalen[[2]](#footnote-2) Parameter für die WFD Session zwischen WFD Sender und WFD Empfänger.
  + **RTSP M4 response:** Der WFD Empfänger antwortet mit einem RTSP M4 response Antwortpaket.

# Chromecast, Firetv

Nicht direkt screensharing

Mit entsprechenden geräten werden diese „erweitert“

Handy als fernbedienung, etc

# Fazit

Problem: technologie noch immer unausgereift

Weiterhin implementieren unterschiedliche devices miracast auf unterschiedliche art, was zu inkompatibilität führt (siehe tests)

Für nicht smarte geräte werden zusatzdevices benötigt

Weiterhin probleme mit qualität, latenz, etc bei schwachen netzwerken

Für beste ergebnisse: try and error oder kabel verwenden :P

**Miracast is da größte Scheiß, Finger weg davon, nehmts AirPlay, Chromecast und co, owa jo net Miracast Devices!!!!111einseinself**

Literaturverzeichnis

Apple (2016) *Mit AirPlay drahtlos Inhalte von Ihrem iPhone, iPad oder iPod touch streamen* [Online]. Verfügbar auf https://​support.apple.com​/​de-at/​HT204289.

Betters, E. (2015) *Miracast explained: How is it different from Chromecast and AirPlay?* [Online]. Verfügbar auf http://​www.pocket-lint.com​/​news/​133437-miracast-explained-how-is-it-different-from-chromecast-and-airplay.

Chris Hoffman (2014) *What is Miracast and Why Should I Care?* [Online]. Verfügbar auf http://​www.howtogeek.com​/​200796/​what-is-miracast-and-why-should-i-care/​ [Stand 06.02.2016].

Czerulla, H. A. and Hansen, S. (2013) ‘Miracast-Check: Was die AirPlay-Alternative derzeit leistet’, *c't*, no. 17.

GRAVIS (2011) *AirPlay: 10 Fragen, 10 Antworten* [Online]. Verfügbar auf http://​www.gravis.de​/​blog/​airplay-10-fragen-10-antworten/​.

Hoffman, C. (2013) *Wireless Display Standards Explained: AirPlay, Miracast, WiDi, Chromecast, and DLNA* [Online]. Verfügbar auf http://​www.howtogeek.com​/​177145/​wireless-display-standards-explained-airplay-miracast-widi-chromecast-and-dlna/​.

Länger, K. (2015) *HDMI ohne Kabel: Miracast & Co* [Online]. Verfügbar auf http://​www.tecchannel.de​/​pc\_mobile/​peripherie/​3199755/​hdmi\_ohne\_kabel\_miracast\_co/​index2.html.

Müssig, F. (2010) *Hands-On: Intel Wireless Display* [Online]. Verfügbar auf http://​www.heise.de​/​newsticker/​meldung/​Hands-On-Intel-Wireless-Display-900035.html.

Paine, S. (2014) *WiDi – Wireless Display Overview, Specifications, Testing and Demos* [Online]. Verfügbar auf http://​www.umpcportal.com​/​2014/​02/​widi-wireless-display-overview-specifications-testing-and-demos/​.

Wi-Fi Alliance *Wi-Fi Display Specification v1.1* [Online]. Verfügbar auf https://​www.wi-fi.org​/​downloads-registered/​Wi-Fi\_Display\_Specification\_v1.1.zip [Stand 07.02.2016].

Ziesecke, D. (o.J.) *Intel WiDi* [Online]. Verfügbar auf http://​www.voip-information.de​/​intel-widi.php.

1. <http://www.intel.de/content/www/de/de/support/emerging-technologies/000014931.html?wapkw=widi&_ga=1.144481589.1454502059.1454616028> [↑](#footnote-ref-1)
2. [↑](#footnote-ref-2)