**Raspberry Pi gegen Android als Server**

Am Anfang des Projektes wurde das Ziel gesetzt, einen handelsüblichen Raspberry Pi als Server zu verwenden und die Datenbrille direkt über den Video-Ausgang an die Brille anzuschließen. Der größte Vorteil dabei liegt darin, dass die veraltete Steuerung der Brille wegfällt und keine weitere Applikation für die Steuerung der Brille mehr nötig wäre. Dadurch sinkt die Komplexität des Projektes, da die Brille als normalen Bildschirm fungiert und keinerlei Kommunikation über das Netzwerk zwischen Server und Client mehr nötig wäre.  
Nach längerer Recherche wurde jedoch klar, dass zu dem beiliegendem Adapter der Brille keinerlei Funktionsbeschreibungen gefunden werden konnten und die direkte Ansteuerung nicht möglich war. Den Adapter wegzulassen und stattdessen die Schnittstelle direkt über die IO-Ports des Raspberry Pis anzusteuern war ebenfalls nicht möglich, da die Pin-Belegung nicht veröffentlicht wurde und eine Anfrage an den Support von Epson brachte keinen Erfolg.

|  |  |
| --- | --- |
| Übersicht Pro und Contra | |
| Pro Raspberry Pi | **Contra Raspberry Pi** |
| Wegfall der veralteten Steuerung | Keine Kamera welche die benötigte Auflösung hat, um QR-Codes zu scannen |
| Senken der Komplexität der Applikation (Wegfall von Server + Client) | NFC, GPS und Bluetooth nur über Erweiterungen (Wegfall von IO-Ports) |
|  | Externe Stromversorgung nötig  (Zusätzlicher Hardwareaufwand von z.B. Spannungsreglern) |
|  | Durch Erweiterungen unhandlich zu transportieren |
|  | Keine vordefinierte Eingabemöglichkeit (Nur Maus + Tastatur -> Transport ) |
| Pro Android | **Contra Android** |
| Eingabe über Touch | Erhöhte Komplexität der Applikation (Wegfall von Server + Client) |
| Sehr leicht zu transportieren |  |
| QR, NFC, GPS, Beschleunigungssensor und Kamera in nahezu jedem Gerät vorhanden |  |

**Lokal gegen nicht lokale Daten der Website**