

[app.guidetranslator.com](http://app.guidetranslator.com)

Update\_V.8\_konzept: real offline mode: 27: Februar,  
03:01

# Offline P2P: Implementiert ✓ — GL.iNet Router + WebSocket

Eine Transport-Abstraktionsschicht wird gebaut. Die App kann zwischen Cloud (Supabase) und lokalem WiFi (WebSocket) umschalten – automatisch, per QR-Code, ohne manuelles Setup.

 Ulrichs Erklärung: Was ist ein 'portabler Router'?

Ein GL.iNet Beryl AX ist ein kleiner WLAN-Router, so groß wie eine Zigarettenzschachtel (~200g). Der Guide steckt ihn ein, schaltet ihn an – und alle Touristen verbinden sich mit dem WLAN 'GuideTour'. Kein Internet nötig. Der Router ist das lokale 'Postamt' für die Gruppe. Die App läuft auf dem Router als kleines Programm (Relay-Server) und verteilt die Übersetzungen direkt an alle Handys.

# Transport- Abstraktionsschicht

- 4 neue Dateien in `src/lib/transport/`
- Supabase und lokaler WebSocket als austauschbare Transporte.
- Hooks akzeptieren jetzt optionalen externen Transport.

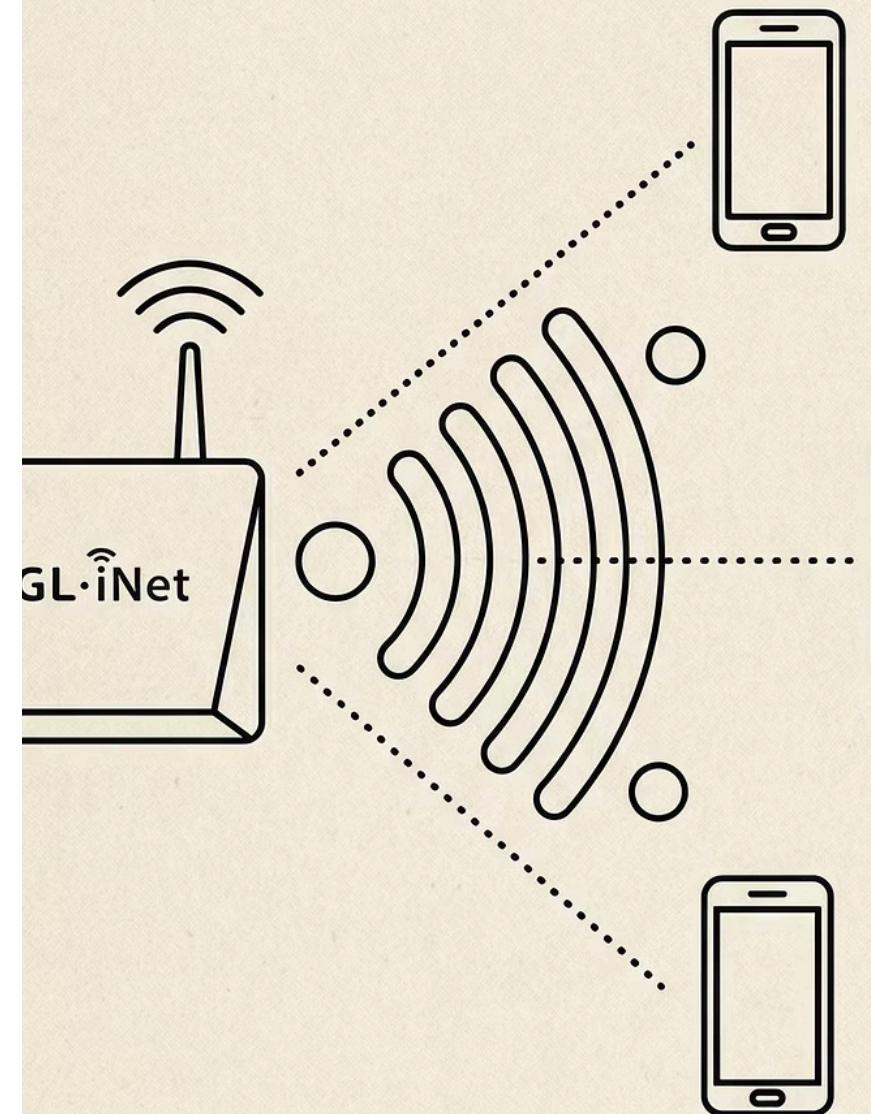
## Relay-Server (GL.iNet Router)

~200 Zeilen Node.js, keine heavy dependencies. Session-Management, Presence-Tracking, Health-Endpoint. opkg install node && node server.js

## QR-Code mit ?ws= Parameter

QR-Code enthält Server-URL. Listener scannt → App erkennt ?ws= → verbindet automatisch lokal. Kein manuelles Setup.

Kein Internet nötig. 50+ Listener. <100ms Latenz. Läuft auf einem 200g Router für ~80€.



# Was wir entschieden haben:

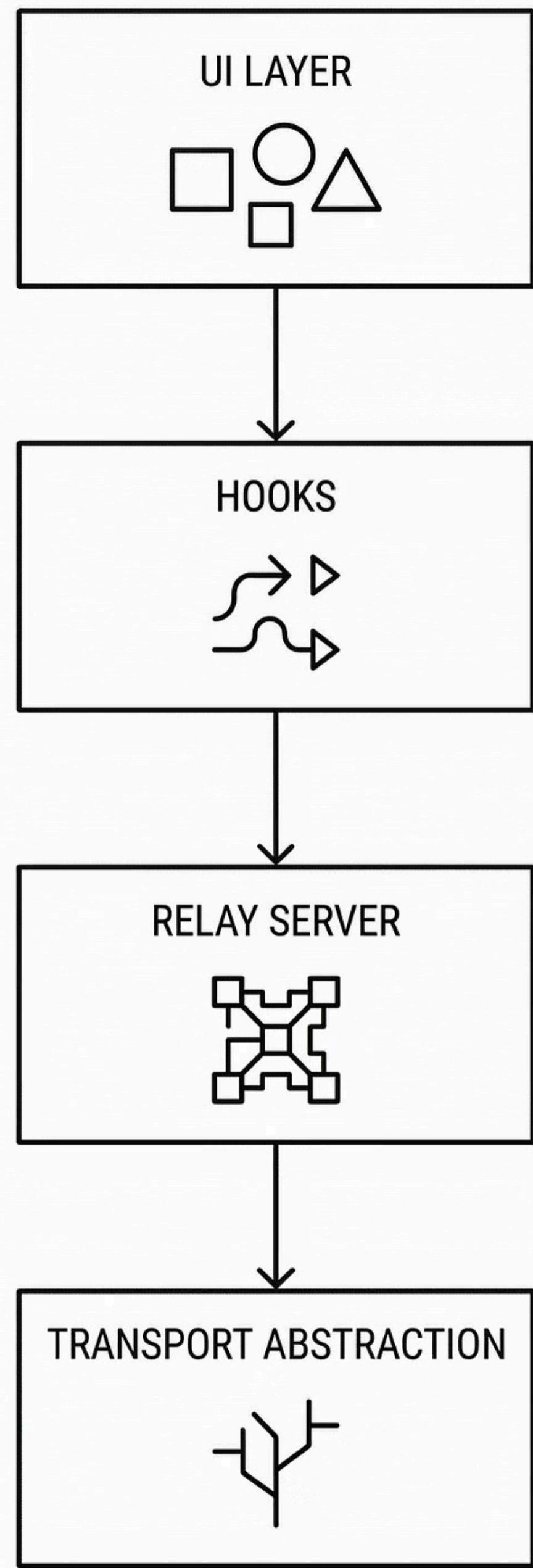
## 4 Schichten

Die Implementierung ist in 4 saubere Schichten aufgeteilt – Transport-Abstraktion, Relay-Server, Hook-Refactoring und UI-Updates.

### 💡 Ulrichs Erklärung: Was ist eine 'Transport-Abstraktionsschicht'?

Stell dir vor, du willst einen Brief schicken. Früher gab es nur einen Weg: Post (= Supabase Cloud). Jetzt gibt es zwei Wege: Post (Cloud) oder Bote (lokaler WebSocket). Die 'Abstraktionsschicht' ist wie ein Schalter: Die App weiß nicht mehr, welchen Weg sie nimmt – sie sagt nur 'schick das' und die Schicht entscheidet: Cloud oder lokal? Das macht den Code sauber und austauschbar.

Schicht	Dateien	Was macht sie
Transport-Abstraktion	src/lib/transport/types.ts, supabase-transport.ts, local-ws-transport.ts, connection-manager.ts	Interfaces + Supabase/WebSocket als austauschbare Klassen. Auto-Erkennung lokaler Server (GL.iNet, iOS/Android Hotspot IPs)
Relay-Server	relay-server/server.js	~200 Zeilen Node.js WebSocket-Server für den Router, Session-Management, Presence-Tracking, Health-Endpoint
Hook-Refactoring	useBroadcast.ts, usePresence.ts, useConnectionMode.ts, useLiveSession.ts	Hooks akzeptieren optionalen Transport. Neuer useConnectionMode Hook für Cloud/Lokal-Umschaltung
UI-Updates	LiveLandingPage, SpeakerView, ListenerView, SessionQRCode, LiveSessionPage	Cloud/Lokal-Toggle, Connection-Status-Anzeige, QR-Code mit ?ws= Parameter, Auto-Verbindung



Supabase bleibt vollständig erhalten – Cloud-Modus funktioniert wie bisher. Lokal ist ein zusätzlicher Modus, kein Ersatz.

# So funktioniert's: Guide + Touristen in 3 Schritten

Für den Guide und die Touristen ist der Ablauf denkbar einfach – Router einschalten, QR-Code scannen, fertig. Kein IT-Wissen nötig.

 Ulrichs Erklärung: Was ist der GL.iNet Beryl AX?

Der GL.iNet Beryl AX ist ein Mini-WLAN-Router – so groß wie eine Zigarettenzschachtel, ~200g, läuft auf Akku oder USB. Der Guide trägt ihn in der Tasche. Alle Touristen verbinden sich mit dem WLAN 'GuideTour'. Der Router verteilt die Übersetzungen direkt – ohne Internet, ohne Mobilfunk, auch im Keller eines Museums oder auf dem Kreuzfahrtschiff. Kostet ca. 80€.

## Guide – 3 Schritte

1. Router einschalten (GL.iNet Berry AX, ~200g)
  2. App öffnen → "Lokal (Offline)" wählen → Session starten
  3. QR-Code den Touristen zeigen → fertig

## Tourist — 3 Schritte

1. WiFi "GuideTour" verbinden  
(einmalig)
  2. QR-Code scannen → App erkennt ?  
ws= → verbindet automatisch lokal
  3. Sprache wählen → Übersetzung läuft

## Kein Internet nötig

Alles läuft über das lokale Router-WLAN – Kreuzfahrtschiff, Museum, Outdoor-Tour

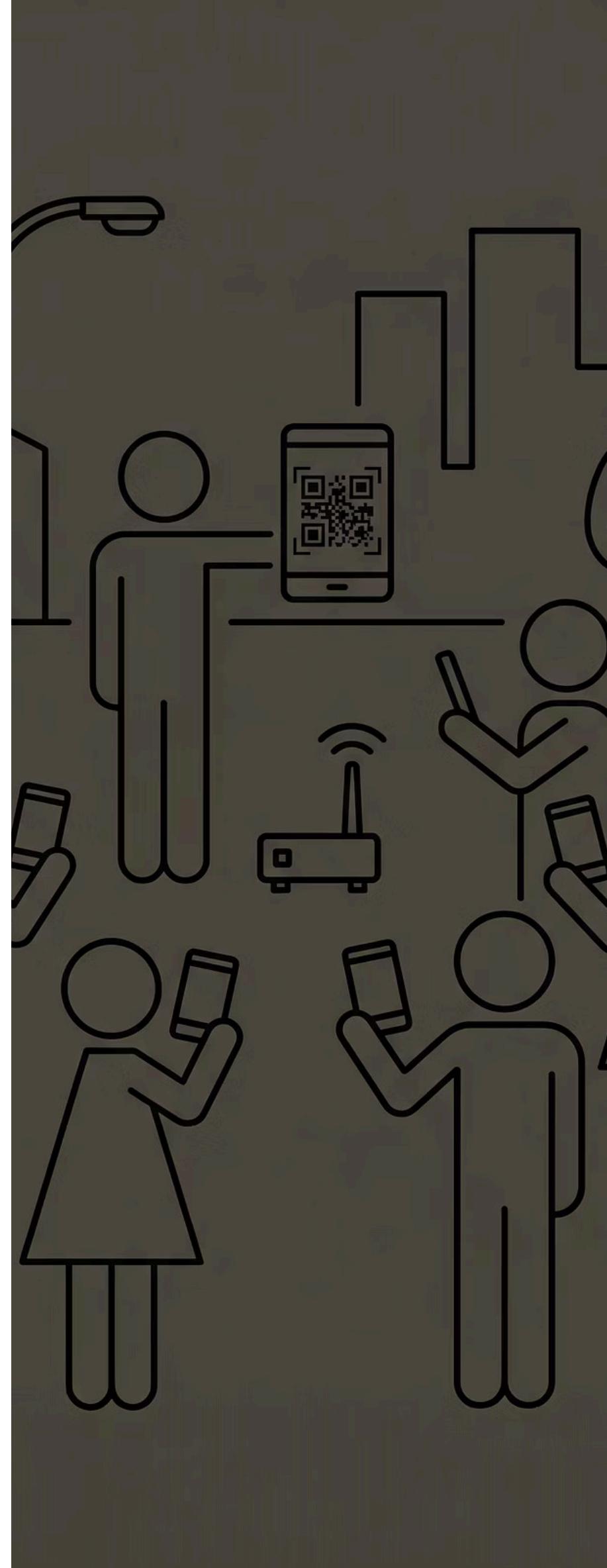
## 50+ Listener

WebSocket-Server auf dem Router  
skaliert problemlos auf 50+  
gleichzeitige Verbindungen

## <100ms Latenz

Direktverbindung ohne Cloud-Umweg – Übersetzung kommt in unter 100ms an

Der QR-Code enthält die Server-URL (`?ws=192.168.x.x:8080`) – Touristen müssen nichts manuell eingeben.



# Vollständige Offline-Analyse: 6 Optionen, 5 Szenarien

Eine vollständige Analyse aller 6 Technologie-Optionen für den Offline-Betrieb – bewertet gegen 5 reale Einsatzszenarien. Ergebnis: 3-Tier Hybrid-Architektur.

## 💡 Ulrichs Erklärung: Was sind die 5 Szenarien?

Die App wird in sehr unterschiedlichen Situationen eingesetzt: (A) Kreuzfahrt-Exkursion mit 20-50 Personen im Hafen – kein Internet, aber lokales WiFi möglich. (B) Kleiner Stadtführer mit 5-10 Personen in der Altstadt – kein WiFi, kein Internet. (C) Museum oder Messe mit 10-30 Personen – oft lokales WiFi vorhanden. (D) Behördengang oder Arztbesuch mit 3-8 Personen – unzuverlässiges Netz. (E) Kreuzfahrtschiff mit 50+ Personen – Schiffs-WiFi ohne Internet.

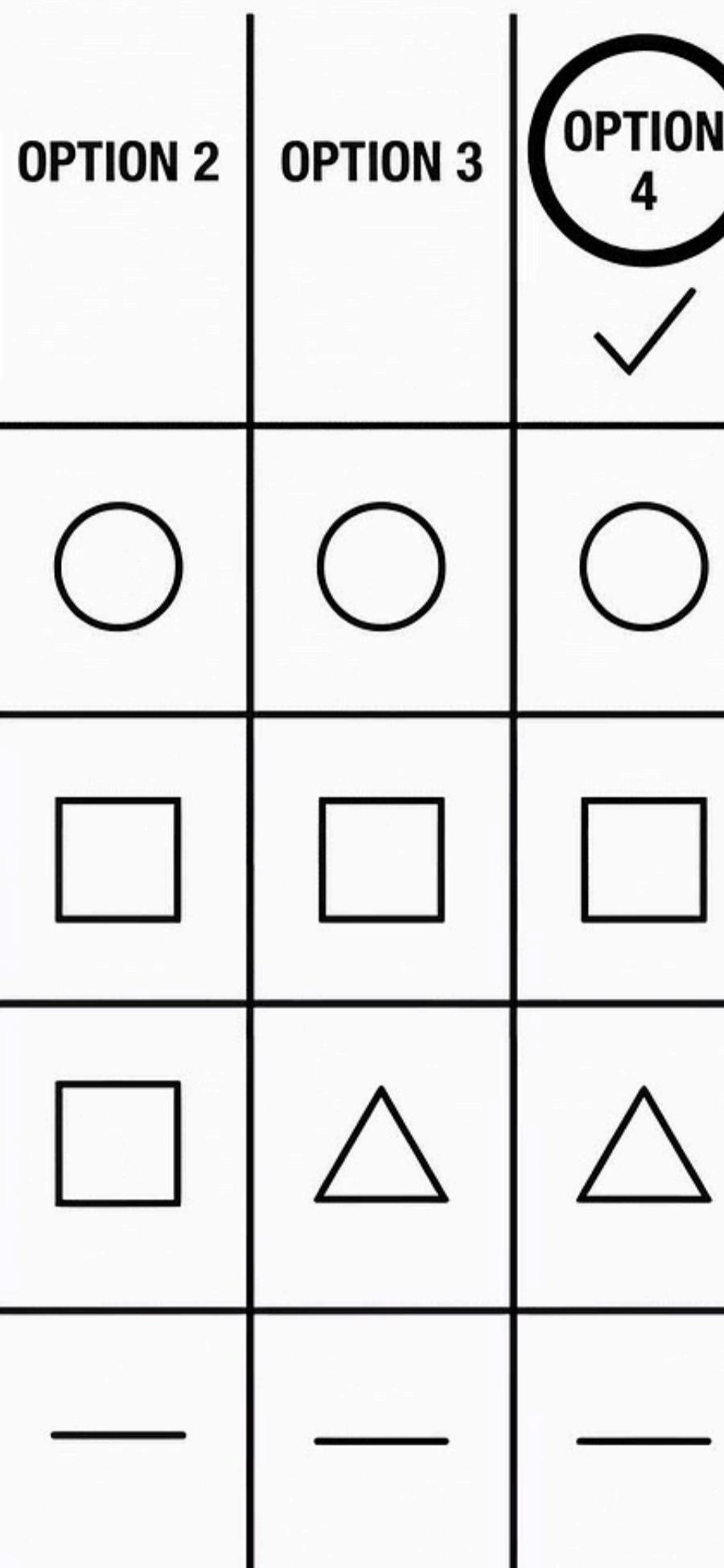
Szenario	Personen	Netz?	Beispiel	Beste Option
A. Kreuzfahrt- Exkursion	20–50	Kein Internet, lokales WiFi möglich	Hafen- Rundgang, Busfahrt	Tier 2 (WiFi)
B. Kleiner Stadtführer	5–10	Kein Internet, kein WiFi	Altstadt- Tour, Wanderung	Tier 3 (BLE)
C. Museum/M esse	10–30	Oft lokales WiFi	Ausstellung , Konferenz	Tier 2 (WiFi)
D. Behördenga ng/Arzt	3–8	Unzuverläss ig	Behörde, Arztbesuch	Tier 3 (BLE)
E. Kreuzfahrts chiff	50+	Schiffs-WiFi (kein Internet)	Bordprogra mm	Tier 2 (WiFi)

Fazit: Zwei Technologien decken alle 5 Szenarien ab – lokales WiFi + WebSocket (bereits implementiert) und BLE Direct (neu zu implementieren).



# 6 Optionen im Vergleich — Die Wahrheit

Sechs Technologien wurden analysiert. Zwei fallen sofort raus (kein iOS), eine ist unpraktikabel (QR-Aufwand), eine ist zu limitiert. Übrig bleiben zwei.



Option	Ohne Netz?	Mit Netz?	Cross-Platform	Keine Hardware	Autodiscovery	Bewertung
WiFi + WebSocket	⚠️ Wi-Fi nötig	50+	✓	✗ Router	✓ mDNS	⭐ Bereits implementiert
Phone Hotspot	✓	5-15	✓	✓	✗ manuell	✓ Fallback für kleine Gruppe n
WebRTC QR	✓	5-10	✓	✓	✗ manuell	⚠️ 14 QR-Scans für 7 Listener — unpraktikabel
BLE (GATT)	✓	7	✓	✓	✓ auto	⭐ Empfohlen für Tier 3
Multipeer/Nearby	✓	8-15	✗ NEIN	✓	✓ auto	✗ Kein Cross-Platform
WiFi Direct	✗ iOS	10+	✗ NEIN	✓	✓	✗ Kein iOS-Support

WebRTC Problem: Für 7 Listener = 14 QR-Scans = ~3-4 Minuten Setup. WiFi + WebSocket: 2 Interaktionen pro Listener = ~1 Minute für alle.

# Deep-Research: Architektur korrigiert — 4 Tiers statt 3

Nach tieferer Recherche wurden zwei wichtige Korrekturen gemacht: Phone-Hotspot ist besser als BLE für kleine Gruppen – und BLE taugt nur als Discovery-Layer, nicht als Textkanal.

## 💡 Ulrichs Erklärung: Was hat sich geändert?

Die erste Analyse war zu optimistisch bei Bluetooth. Die Wahrheit: iOS kann nur 31 Bytes per Bluetooth senden – ein übersetzter Satz hat 50-200 Bytes. Das passt nicht. Dafür ist der Phone-Hotspot viel besser als gedacht: Android kann den Hotspot vollautomatisch erstellen – kein Benutzer muss in die Einstellungen. Und es gibt eine Zukunfts-Option: WiFi Aware (iOS 26) wird in 1-2 Jahren alles vereinen.

### Tier 1: Cloud ✓

Supabase Realtime – unverändert. Wenn Internet da ist.

### Tier 2: Router ✓

GL.iNet + WebSocket – unverändert. Für große Gruppen (20-50+).

### Tier 3: Phone-Hotspot ★ NEU

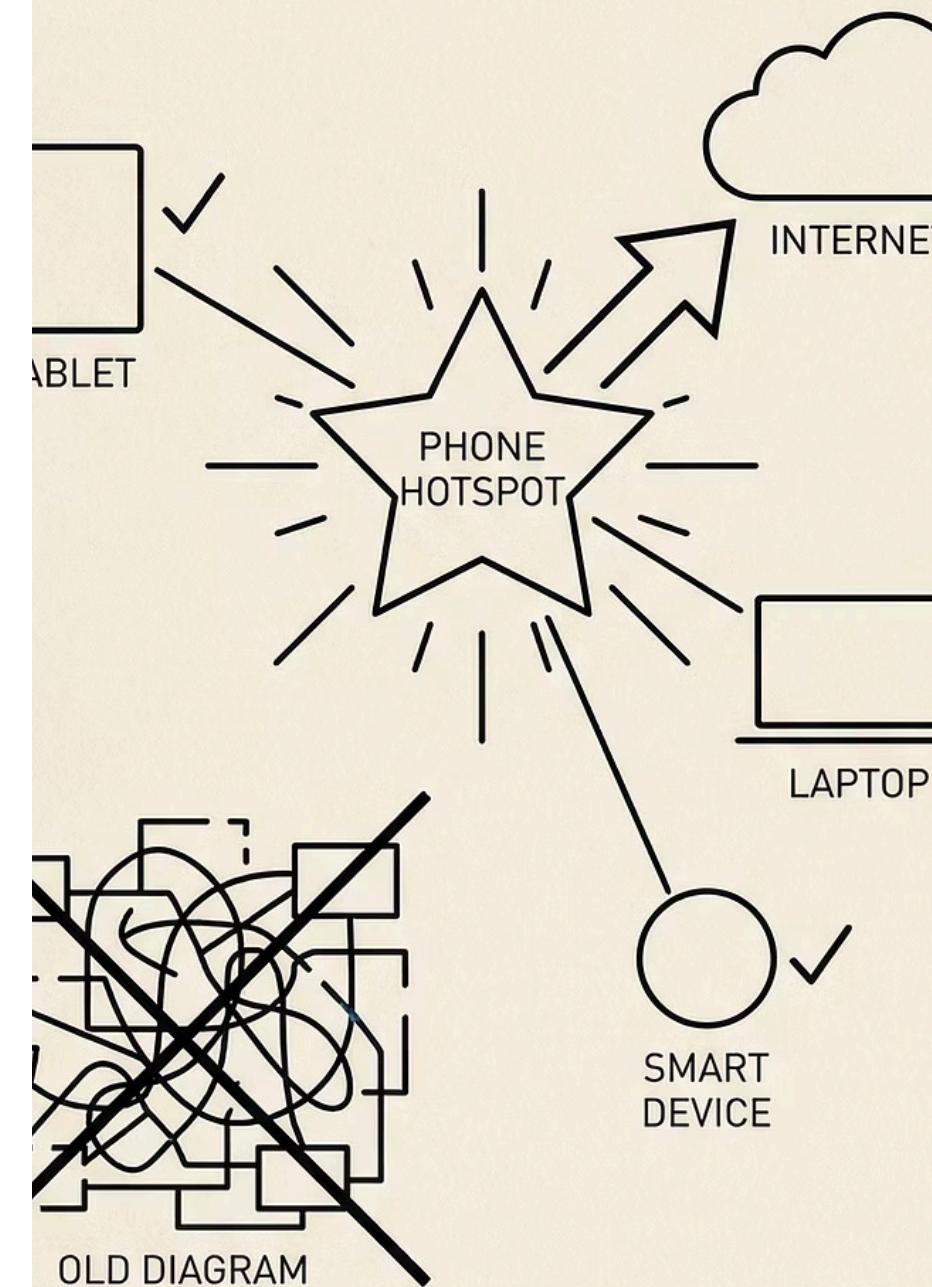
Guide's Handy als Router. Android: automatisch. iOS: manuell. Kein extra Gerät nötig!

### Tier 4: BLE ● Korrigiert

Nur für 2-5 Personen + als Discovery-Layer. Nicht als Haupt-Textkanal geeignet.

Bonus-Entdeckung: WiFi Aware (iOS 26, WWDC 2025) wird in 1-2 Jahren Tiers 2-4 vereinen – cross-platform, kein Hotspot nötig, WiFi-Geschwindigkeit.

# CORRECTED ARCHITECTURE



# Tier 3: Phone-Hotspot — Der Star der Show ⭐

Der Phone-Hotspot mit bestehendem WebSocket-Relay ist die beste Lösung für den kleinen Stadtführer – kein extra Gerät, kein manuelles Setup auf Android, sofort einsatzbereit.

## 💡 Ulrichs Erklärung: Was ist startLocalOnlyHotspot()?

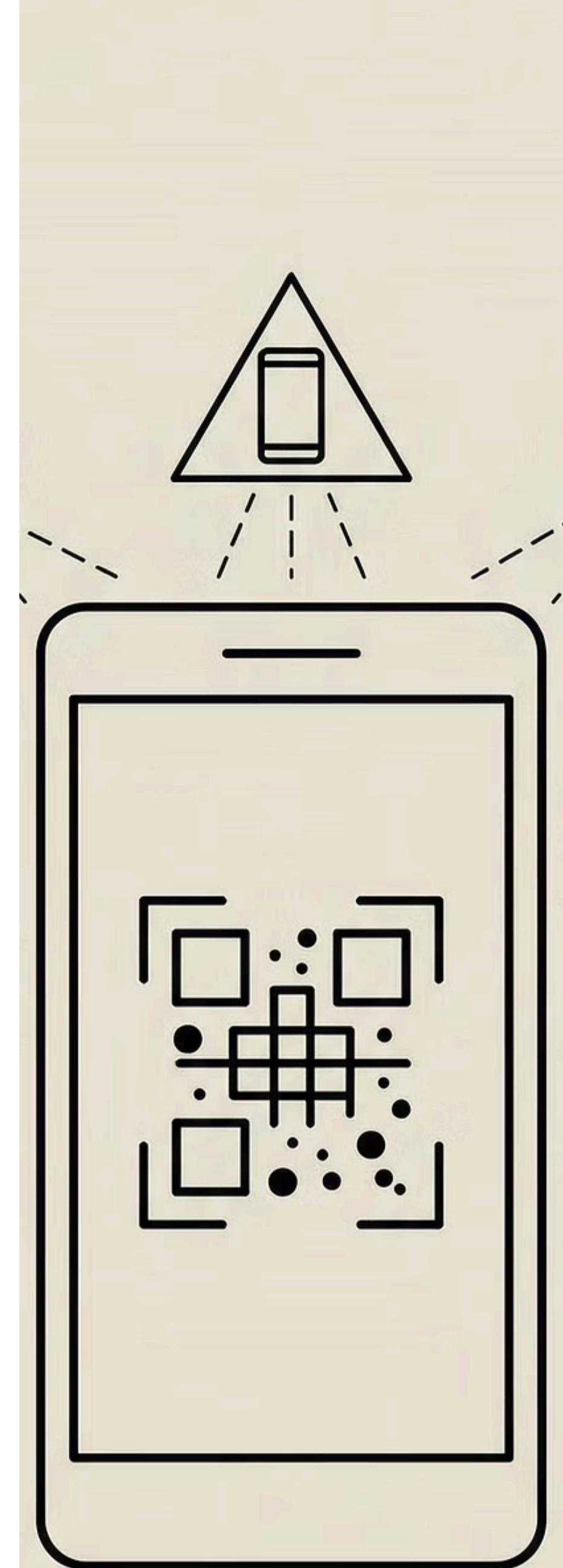
Auf Android kann die App den WLAN-Hotspot vollautomatisch erstellen – ohne dass der Guide in die Einstellungen gehen muss. Die App sagt dem Handy: 'Erstelle jetzt einen Hotspot namens TourGuide-a7b3 mit Passwort x9k2mNpQ!' Das Handy macht es sofort. Kein Internet wird geteilt. Das Guide-Handy bleibt selbst im WLAN. Dann zeigt die App einen QR-Code – Touristen scannen ihn und sind automatisch verbunden.

### 🤖 Android Guide — Vollautomatisch

1. Guide öffnet App → "Tour starten" → "Offline-Modus"
2. App ruft startLocalOnlyHotspot() auf – automatisch
3. SSID + Passwort werden automatisch generiert
4. Kein Internet wird geteilt, WiFi des Guides bleibt aktiv (Android 12+)
5. App startet WebSocket-Server auf 192.168.43.1:8765
6. App zeigt WiFi-QR-Code: WIFI:T:WPA;S:TourGuide-a7b3;P:x9k2mNpQ;;
7. Listener scannen → auto-connect → App öffnen → Sprache wählen → fertig

### 🍎 iOS Guide — Manuell (10 Sekunden)

1. App zeigt: "Bitte aktiviere den Persönlichen Hotspot"
2. Guide aktiviert in Einstellungen oder Control Center (~10 Sekunden)
3. App erkennt Hotspot automatisch → startet Server auf 172.20.10.1
4. App generiert WiFi-QR für Listener
5. Gleicher Flow wie Android – aber max 5 Listener (iOS Hard-Limit)



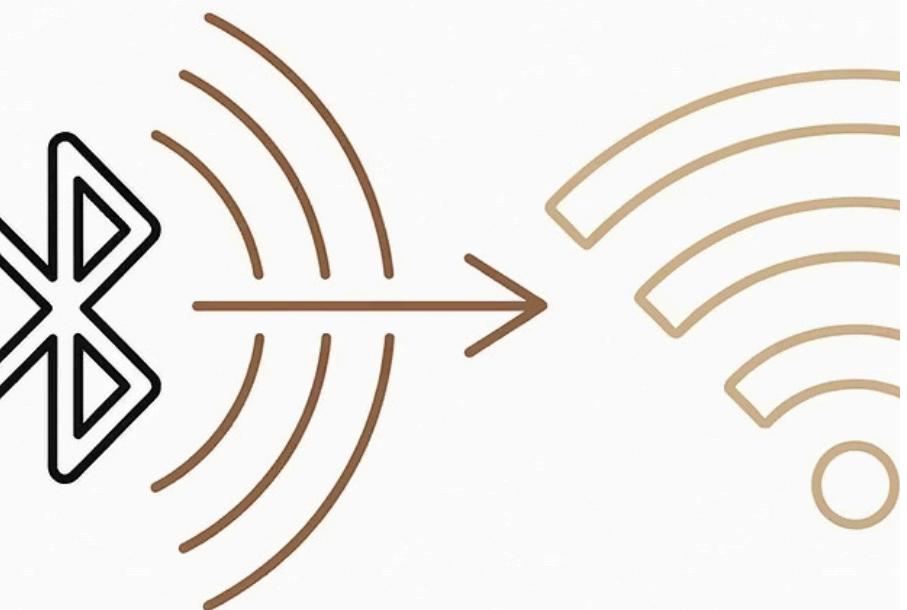
Eigenschaft	Android	iOS
Programmatisch	✓ Ja (startLocalOnlyHotspot)	✗ Manuell
Max Listener	8–10	5
WiFi bleibt aktiv	✓ Ja (Android 12+)	✗ Nein
Internet nötig	✗ Nein	✗ Nein
Akku (2-3h Tour)	~40–60%	~40–60%
Reichweite outdoor	20–30m	20–30m
WiFi QR auto-join	✓ Ja	✓ Ja (seit iOS 11)
Bestehender WebSocket-Code	✓ Funktioniert sofort	✓ Funktioniert sofort

# BILE HOTS

Der bestehende WebSocket-Relay-Code funktioniert sofort – der Hotspot erstellt ein lokales WiFi, unser relay-server läuft darüber. Kein neuer Code für den Datentransport nötig.

# BLE: Korrigiert — Discovery-Layer, nicht Textkanal

Die erste Einschätzung war zu optimistisch. BLE ist NICHT als Haupt-Textkanal geeignet — aber perfekt als automatischer Discovery-Layer und Ultra-Fallback für 2-5 Personen.



## 💡 Ulrichs Erklärung: Warum taugt BLE nicht für Text?

iOS kann per Bluetooth nur 31 Bytes auf einmal senden — das ist weniger als dieser Satz hier. Ein übersetzter Satz hat 50-200 Bytes. Das passt nicht rein. Kein einziges Produktions-App nutzt Bluetooth allein für Gruppentext. Alle nutzen Bluetooth nur zum 'Finden' (Discovery) — und dann WiFi für den eigentlichen Text. Wie ein Wegweiser: Bluetooth zeigt wo der Guide ist, WiFi liefert die Übersetzung.

## BLE als Textkanal — Probleme

- iOS: Nur 31-Byte Legacy-Advertisements (kein Extended Advertising)
- Ein Satz (50-200 Bytes) passt nicht rein
- iOS Background-Peripheral stark eingeschränkt
- Advertising-Frequenz sinkt im Background
- Kein Produktions-App nutzt BLE allein für Gruppentext
- Connected Mode (GATT): MTU nur 185 Bytes auf iOS, Chunking nötig

## BLE als Discovery-Layer — Perfekt

- Guide's Phone advertisiert: 31 Bytes = Service UUID + Session-Code
- Listener findet Guide automatisch — kein QR-Scan nötig
- Listener kennt Session-Code → verbindet sich zum WiFi-Hotspot
- WiFi liefert den Text
- = BLE für Discovery, WiFi für Daten

BLE als Ultra-Fallback für 2-5 Personen (Arzt, Behörde): GATT Connected Mode mit Notifications, MTU 185 Bytes, 100-300ms Latenz — akzeptabel für kurze Sätze.

# Implementierungs-Prioritäten + WiFi Aware Ausblick

Was als nächstes gebaut wird – und ein Blick in die Zukunft: WiFi Aware (iOS 26) wird in 1-2 Jahren alle Offline-Tiers vereinen.

## 💡 Ulrichs Erklärung: Was ist WiFi Aware?

WiFi Aware ist eine neue Technologie, die Apple bei der WWDC 2025 für iOS 26 angekündigt hat. Sie erlaubt direkte Gerät-zu-Gerät-Kommunikation mit WiFi-Geschwindigkeit – ohne Hotspot, ohne Router, ohne Internet. Und das Beste: Es ist der gleiche Standard wie auf Android (seit 2017). In 1-2 Jahren, wenn iOS 26 weit verbreitet ist, könnte die App einfach sagen: 'Verbinde dich mit dem Guide' – und alle Handys verbinden sich direkt, egal ob iPhone oder Android.

Pr io	Was	Aufwand	Impact
1	Phone-Hotspot (Tier 3) – Android programmatisch + iOS manuell	Mittel	⭐ HOCH – deckt Stadtführer-Case, kein extra Gerät
2	BLE Discovery (Addon zu Tier 3) – automatische Speaker-Erkennung	Mittel	Mittel – kein QR-Scan nötig
3	BLE GATT Fallback (Tier 4) – für 2-5 Personen	Hoch	Niedrig – Nischenszenario Arzt/Behörde
4	WiFi Aware (Zukunft) – wenn iOS 26 verbreitet	Hoch	⭐ HOCH – vereint alle Tiers



## FUTURE CONNECTIVITY

### 🌐 WiFi Aware — Die Zukunft (iOS 26, WWDC 2025)

- Cross-Platform: gleicher Standard auf iOS und Android
- Direkte Gerät-zu-Gerät-Kommunikation ohne Hotspot oder Router
- WiFi-Geschwindigkeit, niedrige Latenz
- Bestehende WiFi-Verbindung bleibt gleichzeitig aktiv
- Stand Feb 2026: iOS 26 noch nicht weit verbreitet, Cross-Platform-Interoperabilität noch buggy
- Kein Capacitor-Plugin existiert noch
- In 1-2 Jahren: wird Tiers 2-4 vereinen

Heute: Phone-Hotspot implementieren. Morgen: WiFi Aware beobachten. In 2 Jahren: alles vereinfachen.

## Der Plan:

**Android Plugin:** startLocalOnlyHotspot() + eingebetteter WebSocket-Relay-Server (Java-WebSocket lib)

**iOS Plugin:** Eingebetteter WebSocket-Relay-Server (NWListener) + manuelle Hotspot-Anweisung

**TypeScript-Layer:** Plugin-Wrapper + WiFi-QR-Code + UI-Updates