Engineer Software

Eine dApp in mobilen Solana-Wallets öffnen

10. August 2026

Eine mit `@solana/wallet-adapter-react` entwickelte React-dApp verbindet Desktop-Wallets problemlos. Auf dem Telefon scheitert derselbe Ablauf: Die Wallet muss die dApp im eigenen integrierten Browser öffnen, doch die vorgesehenen Deep Links tun es stillschweigend nicht. Backpack landet auf einer Seite mit „download the app“; Solflare öffnet die App, aber nie die Website. Jede Variante scheint zu scheitern. Wir untersuchten das Problem und fanden vier getrennte Ursachen mit demselben Symptom.

Die technischen Erkenntnisse auf Ihr System anwenden

Die technischen Überlegungen hinter „Eine dApp in mobilen Solana-Wallets öffnen“ können Ihre nächste Änderung unterstützen. Wir können ähnliches Verhalten in Ihrer Software, Ihren Daten oder Ihrer Infrastruktur untersuchen. Oberflächen, die schnell laden, auf dem Smartphone funktionieren und sich in allen Browsern Ihrer Kunden gleich verhalten.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Argumentation und ihre Grenzen nachvollziehen

Lesen Sie die Argumentation und ihre Grenzen in dieser Notiz. Wir können prüfen, ob dieselbe Diagnose in Ihrer Umgebung zutrifft. Full-Stack-GIS-Kartenentwicklung geleitet — Full-Stack-Kartenentwicklung von der Datenbank bis zum Browser

Full-Stack-GIS-Kartenentwicklung geleitet Kontakt aufnehmen →

Die vier Probleme

- **Der Backpack-Link war falsch aufgebaut.** Das einzige dokumentierte Format ist `https://backpack.app/ul/v1/browse/<url>?ref=<ref>` — ein Universal Link mit der Ziel-URL im Pfad und einem erforderlichen `ref`. Ein geratenes eigenes Schema wie `backpack://ul/v1/browse?url=...` passt zu keiner von der App registrierten Route. Deshalb landet der Benutzer auf der Installationsseite der Wallet.

- **Auch Solflare benötigt seinen Universal Link:** `https://solflare.com/ul/v1/browse/<url>?ref=<ref>`, nicht das bloße Schema `solflare://`. Ein nacktes Schema kann die App ohne Weiterleitung öffnen — genau der Fall, in dem die App startet, der Website-Tab aber von Hand geöffnet werden muss.
- **Beide Parameter müssen kodiert sein.** `url` ist die vollständige absolute dApp-Adresse, `ref` die anfragende Origin; beide werden durch `encodeURIComponent` geführt. Ein nicht kodiertes `?` oder `&` im Ziel beschädigt die Auswertung. Die Wallet öffnet dann ihren Startbildschirm statt des Browsertabs.
- **Der Auslöser zählt genauso wie der Link.** Universal Links wechseln nur bei einer vom Betriebssystem als vertrauenswürdig eingestuften Navigation die App. Beim Einfügen in die Adressleiste tun sie bewusst nichts — so kann auch ein vollkommen korrekter Link beim Test scheinbar scheitern.

Die dokumentierten Formate

- Phantom: `https://phantom.app/ul/browse/<url>?ref=<ref>` — hier ohne `/v1`.
- Solflare: `https://solflare.com/ul/v1/browse/<url>?ref=<ref>`
- Backpack: `https://backpack.app/ul/v1/browse/<url>?ref=<ref>`

Ein Muster genügt für alle drei:

```
const WALLET_BROWSE = {
  phantom: (url, ref) =>
    `https://phantom.app/ul/browse/${url}?ref=${ref}`,
  solflare: (url, ref) =>
    `https://solflare.com/ul/v1/browse/${url}?ref=${ref}`,
  backpack: (url, ref) =>
    `https://backpack.app/ul/v1/browse/${url}?ref=${ref}`,
};

function walletBrowseLink(
  walletName,
  targetUrl = window.location.href,
) {
  const build = WALLET_BROWSE[walletName.toLowerCase()];
  if (!build) return null;
  return build(
    encodeURIComponent(targetUrl),
    encodeURIComponent(window.location.origin),
  );
}
```

Den Link so auslösen, dass iOS und Android ihn akzeptieren

- **Ein echtes, vorab berechnetes Linkelement darstellen.** Ein einfaches `<a href={walletBrowseLink('phantom')}>` ist auf beiden Plattformen der zuverlässigste Auslöser.
- **Wenn es programmatisch sein muss**, weisen Sie `window.location.href` synchron innerhalb des Tap-Handlers zu — vorher kein `await`, `fetch` oder `setTimeout`. Nach asynchroner Arbeit ist der Gestenkontext verloren, und iOS fällt auf die Website der Wallet zurück. Niemals `window.open` verwenden.
- **Nie durch Einfügen in die Adressleiste testen.** Universal Links werden dort absichtlich nicht ausgelöst. Testen Sie mit einem angetippten Link oder einem von der Kamera gescannten QR-Code.
- **Integrierte Messenger-Browser beachten.** In den In-App-Browsern von Telegram oder Instagram werden Universal Links häufig abgefangen; stattdessen lädt die gewöhnliche Website der Wallet. User-Agent-Erkennung bleibt bestenfalls eine Heuristik. Bieten Sie deshalb zusätzlich einen sichtbaren Ausweg an: „In Safari oder Chrome öffnen, dann verbinden“.

Die umfassendere Lösung für Android

Selbst gebaute Deep Links sind der Weg für iOS. Unter Android ermöglicht Solana Mobiles Mobile Wallet Adapter einer dApp im mobilen Browser die direkte Verbindung mit der installierten Wallet-App, ganz ohne Umweg über deren integrierten Browser. Neuere Versionen von `@solana/wallet-adapter-react` registrieren den mobilen Adapter automatisch. Ein Update der wallet-adapter-Pakete kann Android daher bereits für sich genommen korrigieren. Die Zielarchitektur: Mobile Wallet Adapter auf Android, Browse-Universal-Links auf iOS, wo Apple kein entsprechendes Verfahren erlaubt.

Auf einem Gerät überprüfen

- Echtes Gerät, installierte Wallet, Link aus dem Systembrowser geöffnet — nicht aus einem Messenger.
- Einen dargestellten Link antippen oder einen QR-Code scannen; niemals in die Adressleiste einfügen.

- Prüfen, dass die Wallet öffnet und die dApp in ihrem integrierten Browsertab lädt — der zweite Teil ist der fehleranfällige.
- Ohne installierte Wallet wiederholen: Der Universal Link sollte auf die Wallet-Website zurückfallen. Erscheint diese Seite bei installierter App, stimmt weiterhin der Link oder der Auslöser nicht.
- Danach den Messenger-Weg testen und bei Bedarf den Hinweis zum Öffnen im Browser ergänzen.

Quellen

- Phantom: Deep Links auf iOS und Android
- Solflare: der Browse-Deep-Link
- Backpack: der Browse-Deep-Link
- Solana Mobile: Mobile Wallet Adapter

Alle Blogbeiträge

Messen, bevor Sie ändern

Beschreiben Sie den Arbeitsablauf, die vorhandenen Schnittstellen und die Nutzer, die die Veränderung benötigen. Wir können eine prüfbare Lieferung definieren und die Übergabe so gestalten, dass Ihr Team die gelieferten Bestandteile warten kann.

Preise ansehen Kontakt aufnehmen →

Ihre Beobachtungen besprechen

Besprechen Sie „Eine dApp in mobilen Solana-Wallets öffnen“ im Kontext Ihres eigenen Projekts. Erzählen Sie uns, was Sie beobachtet haben und wie ein nützliches Ergebnis aussehen würde.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://software.engineer.company/de/notes/solana-mobile-wallet-deeplinks/