Engineer Software

Opening a dApp inside Solana mobile wallets

August 10, 2026

A React dApp built on `@solana/wallet-adapter-react` connects desktop wallets without trouble, but on a phone the same flow falls apart: the wallet has to open the dApp inside its own in-app browser, and the deep links that should make that happen quietly do not. Backpack lands on a “download the app” page; Solflare opens the app but never the site; every variant seems to fail. We took the problem apart, and it turned out to be four separate problems wearing one symptom.

Apply the engineering to your system

The engineering behind “Opening a dApp inside Solana mobile wallets” can inform your next change. We can investigate similar behaviour in your software, data or infrastructure. Interfaces that load quickly, work on a phone and behave the same in every browser your customers happen to use.

Explore our services Get in touch →

Follow the reasoning and its limits

Read the reasoning and its limits in this note. We can test whether the same diagnosis holds in your environment. Directed full-stack GIS map development — full-stack mapping, from database to browser

Directed full-stack GIS map development Get in touch →

The four problems

- **The Backpack link was malformed.** The only documented format is `https://backpack.app/ul/v1/browse/<url>?ref=<ref>` — a universal link with the target URL in the path and a required `ref`. A custom-scheme guess like `backpack://ul/v1/browse?url=...` matches no route the app registers, so the user ends on the wallet’s install page.
- **Solflare needs its universal link too:** `https://solflare.com/ul/v1/browse/<url>?ref=<ref>`, not the bare `solflare://` scheme. A bare scheme can launch the app without routing it — which is exactly “the app opens, but the site tab has to be opened by hand”.

- **Both parameters must be encoded.** `url` is the full absolute dApp address and `ref` is the requesting origin, each passed through `encodeURIComponent`. An unencoded `?` or `&` in the target corrupts the parse, and the wallet opens on its home screen instead of the browser tab.
- **The trigger matters as much as the link.** Universal links only switch apps on a navigation the operating system trusts — and they deliberately do nothing when pasted into the address bar, which is also how a perfectly correct link "fails" during testing.

The documented formats

- Phantom: `https://phantom.app/ul/browse/<url>?ref=<ref>` — no `/v1` in this one.
- Solflare: `https://solflare.com/ul/v1/browse/<url>?ref=<ref>`
- Backpack: `https://backpack.app/ul/v1/browse/<url>?ref=<ref>`

One pattern serves all three:

```
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),
  );
}
```

Triggering the link so iOS and Android accept it

- **Render a real anchor, precomputed.** A plain `<a href={walletBrowseLink('phantom')}>` is the most reliable trigger on both platforms.
- **If it must be programmatic**, assign `window.location.href` synchronously inside the tap handler — no `await`, no `fetch`, no `setTimeout` first. After asynchronous work the gesture context is gone, and iOS falls back to the wallet's website. Never `window.open`.
- **Never test by pasting into the address bar.** Universal links deliberately do not fire there; test with a tapped link or a QR code scanned by the camera.
- **Mind the messenger webviews.** Opened inside Telegram's or Instagram's in-app browser, universal links are frequently swallowed and the wallet's plain website loads instead. User-agent detection is heuristic at best, so also give users a visible escape hatch: "open in Safari or Chrome, then connect".

The bigger fix on Android

Hand-rolled deep links are the iOS story. On Android, Solana Mobile's Mobile Wallet Adapter lets a dApp running in the mobile browser connect straight to the installed wallet app, with no in-app-browser detour at all. Recent versions of `@solana/wallet-adapter-react` register the mobile adapter automatically, so upgrading the wallet-adapter packages can fix Android by itself. The target architecture: Mobile Wallet Adapter on Android, browse universal links on iOS, where Apple allows no equivalent.

Verifying on a device

- Real device, wallet installed, link opened from the system browser — not from a messenger.
- Tap a rendered link or scan a QR code; never paste into the address bar.
- Confirm the wallet opens and the dApp loads in its in-app browser tab — the second half is the part that fails.
- Repeat without the wallet installed: the universal link should degrade to the wallet's website. That page appearing while the app is installed means the link or the trigger is still wrong.
- Then test the messenger path, and add the "open in browser" hint if it fails there.

Sources

- Phantom: deep links on iOS and Android
- Solflare: the Browse deep link
- Backpack: the Browse deep link
- Solana Mobile: Mobile Wallet Adapter

All blog posts

Measure before you change

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Discuss your observations

Discuss “Opening a dApp inside Solana mobile wallets” in the context of your own project. Tell us what you observed and what a useful result would look like.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

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

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://software.engineer.company/notes/solana-mobile-wallet-deeplinks/