



# The Challenges and Learnings of Implementing Wallet Connection on Mobile

Bryan Moreno, Carolina Pinzón

Shopify

## Bryan Moreno

- Full-stack developer at Shopify
- From Belo Horizonte, Brazil 🇧🇷
- Now based in Berlin, Germany 🇩🇪



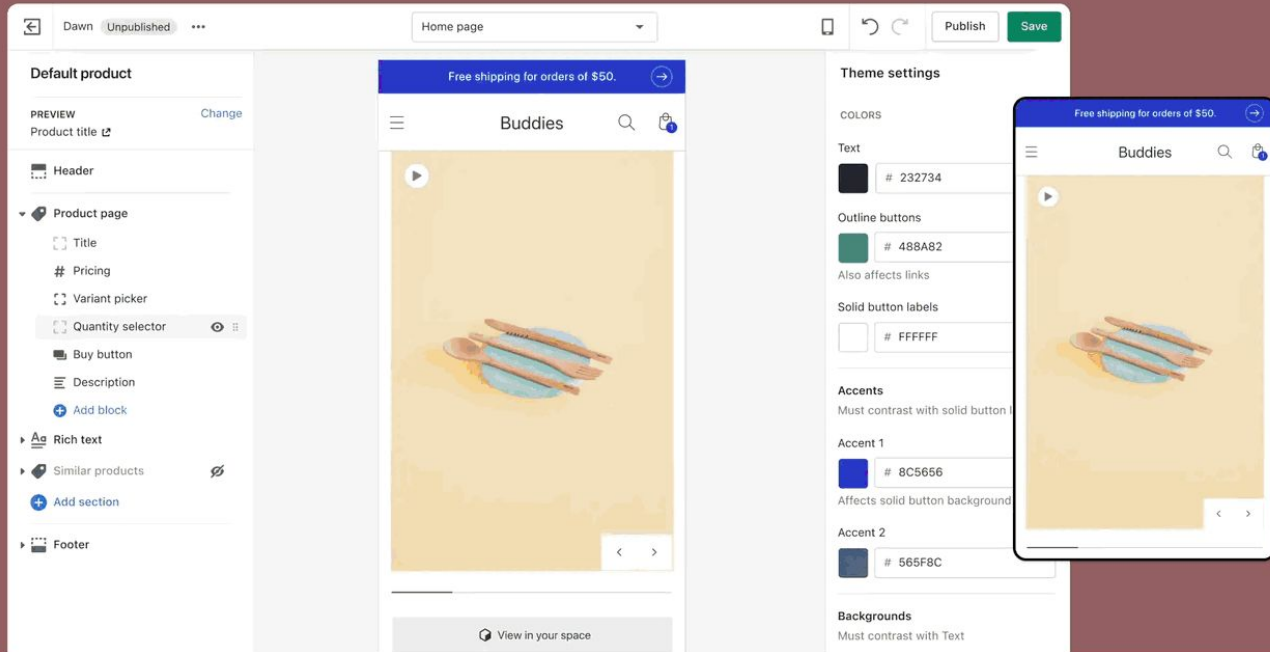




## Carolina Pinzon

- From Bogotá, Colombia 🇨🇴
- Now based in Vancouver, Canada 🇨🇦
- Frontend developer at Shopify

No.1 eCommerce Platform for All Businesses.  
Start, Run + Grow Your Business with  **shopify**



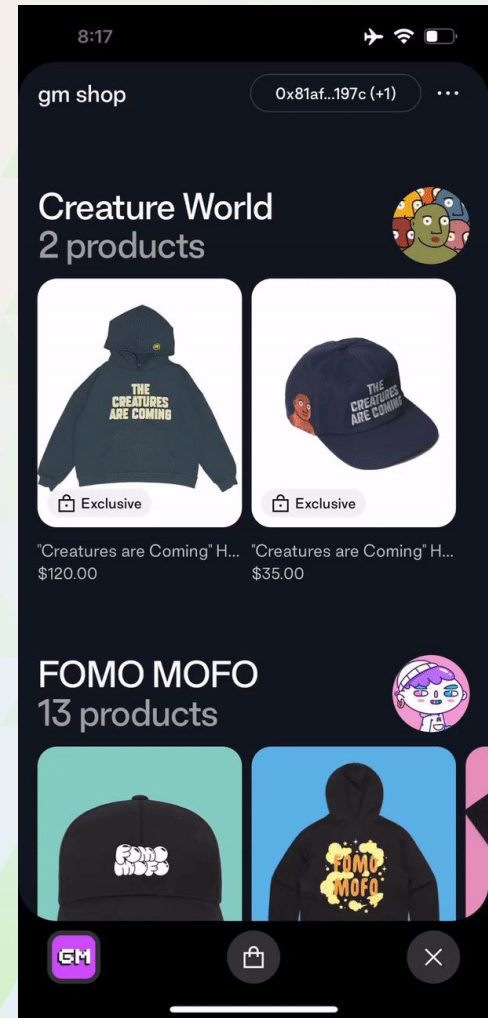


gm shop

What is gm shop?

# gm shop is our take on tokengated commerce

What if we could unite brands with their most loyal fans, granting those fans access to exclusive products, gated by the NFTs they already own?



To build that, we need  
to allow users to  
connect their wallets  
and know which NFTs  
they own

**proof of ownership** in as  
*few steps* as possible

In a seamless fashion.






WalletConnect

How did we support wallet connections?



# MVP - Metamask and Rainbow

 WalletConnect [Docs](#) [Specs](#) v1.0 ▾ [Website](#)

WalletConnect v1.0

Quick Start ▾

Dapps ▾

Standalone Client

NodeJS Client

React-Native

Web3 Provider

Wallets >

JSON-RPC API Methods >

Smart Contract Wallets

Mobile Linking

Cloud Registry [↗](#)

Legacy Clients

Client API Reference

Bridge Server API Reference

Push Server API Reference

## React-Native

### Quick Start For Dapps (React-Native)

A drop-in library which helps easily connect your [React Native](#) dapps to [Ethereum](#) Wallets on [Android](#), [iOS](#) and the [Web](#).

**Notice:** This library assumes you have already enabled prerequisite support for [Web3](#) inside your application. This can be done by creating a new project using `npx create-react-native-dapp`, or by introducing support for Web3 in an existing project by using `npx rn-nodeify --install --hack`.

For more details, check out the [documentation](#).

### Installing

To get started, install `@walletconnect/react-native-dapp`:

```
yarn add @walletconnect/react-native-dapp
```

If you haven't already, you may also need to install `react-native-svg` alongside a persistent storage provider such as `@react-native-async-storage/async-storage`:

# Metamask + Coinbase Wallet

## Deeplinking

### Tip

[Click here to create deeplinks for your application.](#)

Deeplinks enable instant invocation of the user's preferred wallet application with correctly parameterized transactions.

Only the (authenticated) user can confirm the transaction, and the wallet can be a web, mobile or desktop app.

URLs embedded in QR codes, hyperlinks in web pages, emails, or chat messages enable robust, cross-application signaling between otherwise loosely coupled applications.

You can use deeplinks for things like:

- Creating a URL so your users can open your app directly in MetaMask Mobile to interact with your application with their Ethereum account.
- Providing a one-click experience such that users can easily make payments to another account (with pre-filled parameters like recipient address, amount, network, etc.)
- Let your users make gasless and instant transactions with Connex payment channel requests
  - This requires that the user opts in for the InstaPay experimental feature.



Bring your own connector

We were not seeing the success rate we  
wanted

We still had failures  
for no apparent reason

## Our hypothesis:

WalletConnect's library assumes a persistent connection, and that was getting in the way





After a connection and signature:

Our backend could check whether the wallet had the NFT to unlock the products, and the client could drop the session.

Remember: our use case was to establish **proof of ownership** in *as few steps* as possible

**+2,173** **-3,935** ■■■■□

## We rewrote our wallet connector code

Getting rid of WalletConnect's React Native dapps library and using their primitives to build our own connector.

The results were **good!**

User feedback, internal feedback and our numbers were significantly better! This version shipped to our users.

Wallet error rate relative to successful wallet connects



# Users with really valuable NFTs prefer them

- They are way more security-conscious than the average user.
- We wanted to incentivize good behavior
- So we implemented support for Ledger wallets





But some users wanted  
support for more  
hardware wallets

Unfortunately, not all of them have mobile  
support.





So as a *last resort*, let  
the user connect via  
desktop\*

And propagate that connection to the app.

\*Still not implemented, so take with a grain  
of salt

WalletConnect on the  
**desktop** is built as a  
first-class citizen

The background is a complex, abstract geometric pattern. It features a variety of triangles and lines in warm, muted colors including shades of orange, yellow, light blue, and pale green. The pattern is dense and layered, with some elements appearing more prominent than others. The overall effect is a textured, modern aesthetic.

There's still a long way to go.

## What we learned

- Rainbow - [WalletConnect Core](#)
- Ledger - [WalletConnect Core](#)
- Coinbase Wallet - [Deeplinking](#)
- Metamask
  - Android: [WalletConnect Core](#)
  - iOS: [Deeplinking](#)
- Desktop fallback as a last resort for all other wallets

We needed a robust solution. However, your mileage could vary depending on **the use case, the size of your team and the expected reliability.**



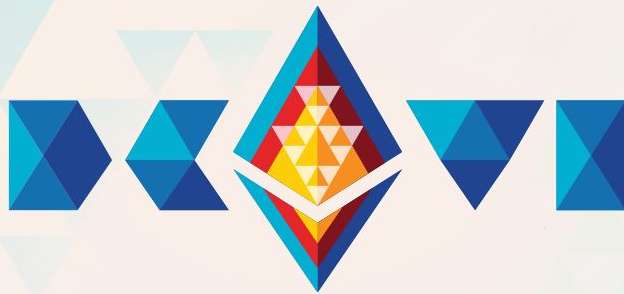


“We shape our tools and thereafter our  
tools shape us”

John Culkin



If we want Ethereum to reach the next billion, it needs to be easily accessible on the platforms people use



# Thank you!

Bryan Moreno, Carolina Pinzón

Software Engineers, Shopify

[bryan.moreno@shopify.com](mailto:bryan.moreno@shopify.com)

[carolina.pinzon@shopify.com](mailto:carolina.pinzon@shopify.com)

Bounty up for grabs!

Check **gitcoin** for  
more details

