Keyless Wallet

yaneshcode

Problem

New User Onboarding

Multiple steps required for users to be able to use a Dapp or trade crypto

The average non-tech savvy users are turned off by the current process

We need to support simple pathways and new user flow for mass adoption

LEARN HOW TO BUY BITCOIN

Ingredients:

First install a bitcoin wallet

Oh SHIT! You forgot step one for a few months and now the price has risen :(

Shit! now its been a year and you still haven't installed a bitcoin wallet. The price is now x3 and you cant afford a whole bitcoin anymore.

FUCK! Shitcoins are up 2000% and you still haven't installed a bitcoin wallet, and your stupid friends are now rich. You don't understand what an altcoin is

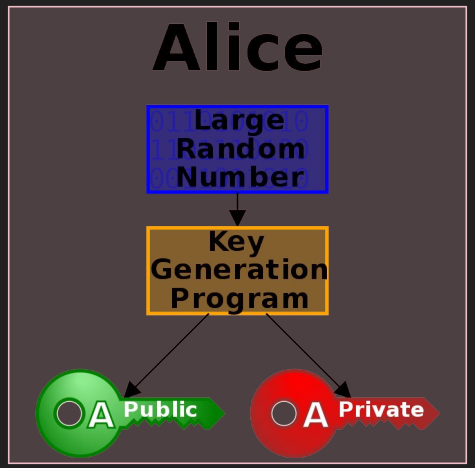
you finally installed a bitcoin wallet and bought but only enough to cover the cost of the wannacry ransom on you laptop

you think you can do things but you can't. prices is millions but you still think its a ponzi

Some terms

Account

Externally Owned Account (EOA)
Private/Public keypair



Wallet

Contract or Software
(Multi-sig wallet, Wallet Apps)

User Experience

- User does not need to hold an account (private/public keypair)

- User has immediate access to a wallet and can receive funds

- User can take full control of their wallet when they have an account

Counterfactual Instantiation of a Contract

CREATE2 opcode (**EIP-1014** ⟶Constantinople)

- Address
- Init code
- 32-bytes Salt

We can know beforehand what the address of the deployed contract will be.

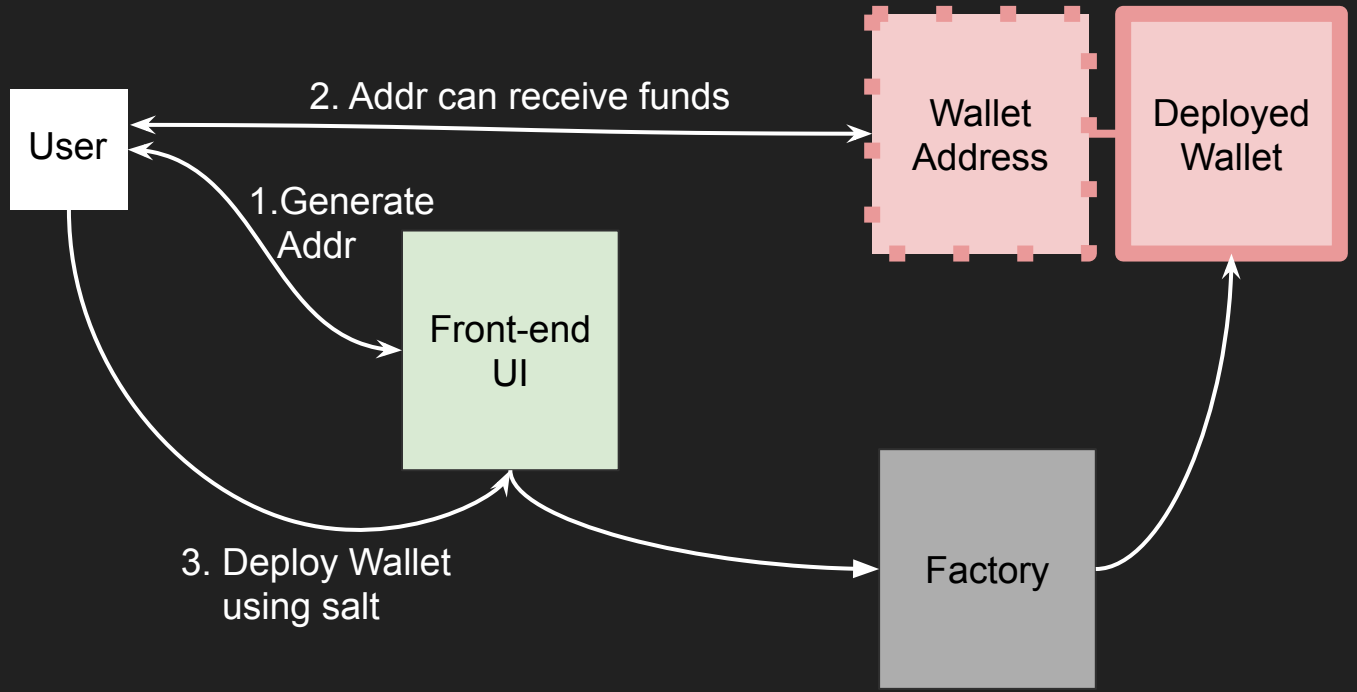
Creating a Keyless Wallet

1. User inputs password (salt) and generates wallet contract address locally.
   - At this point the user can already start receiving funds to this address.

2. To deploy the contract the user inputs their salt, the Dapp owner can deploy the wallet on the user’s behalf.

3. The Dapp owner initially has access to the wallet and can sign transactions on the user’s behalf. User can take control when they are no longer novice.

Creating a Keyless Wallet



Pros

- 1 step process to create a wallet. User does not need an account.

- No lost funds as Dapp owner has initial control on wallets.

- User can take ownership of their wallet in the future.

Pros

- 1 step process to create a wallet. User does not need an account.

- No lost funds as Dapp owner has initial control on wallets.

- User can take ownership of their wallet in the future.

- Huge save on gas costs (No deployment needed if user went inactive or never received funds)

Cons

- Lose some trustlessness.

- Upgrading is hard, but possible.

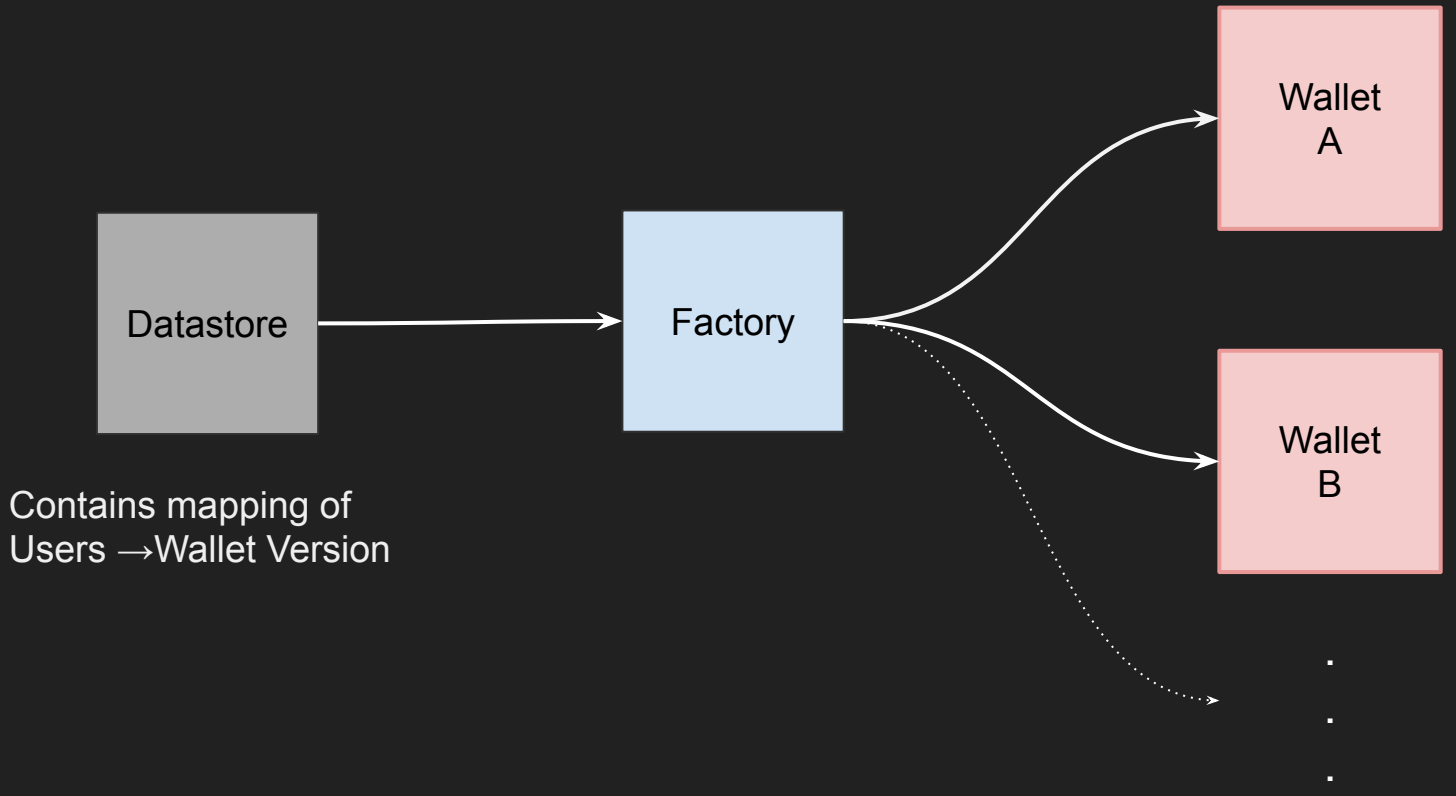
Upgrade strategy

Datastore pattern → init code of the wallet contract is versioned.

Keep track of which version users have pre-generated an address and use that version for deployment.

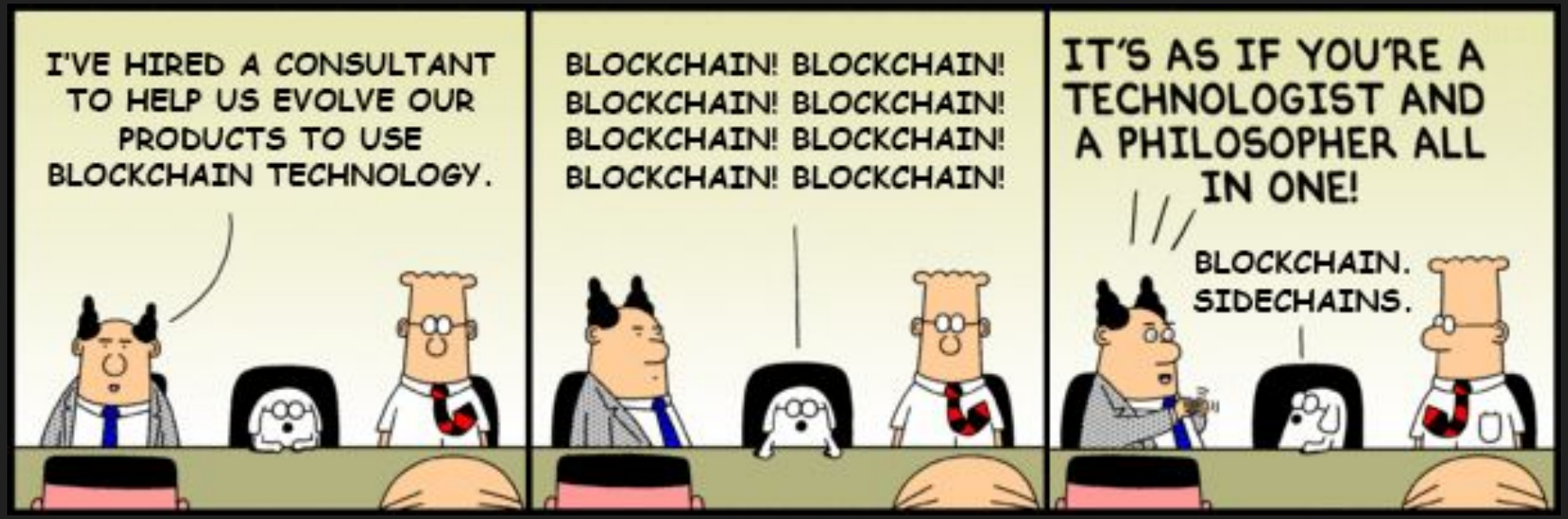
Users can enter an upgrade stage, deploy a wallet with a newer version, and migrate their funds to the new upgraded wallet.

Contract Architecture



DEMO

Q & A


I'VE HIRED A CONSULTANT TO HELP US EVOLVE OUR PRODUCTS TO USE BLOCKCHAIN TECHNOLOGY.
BLOCKCHAIN! BLOCKCHAIN!
BLOCKCHAIN! BLOCKCHAIN!
BLOCKCHAIN! BLOCKCHAIN!
BLOCKCHAIN! BLOCKCHAIN!
IT'S AS IF YOU'RE A TECHNOLOGIST AND A PHILOSOPHER ALL IN ONE!
BLOCKCHAIN.
SIDECHAINS.

Resources

https://eips.ethereum.org/EIPS/eip-1014

https://github.com/miguelmota/solidity-create2-example

https://medium.com/gitcoin/counterfactual-loan-repayment-828a59d9b730

https://hackernoon.com/create2-a-tale-of-two-optcodes-1e9b813418f8

https://hackernoon.com/the-create2-opcode-and-dapp-onboarding-in-ethereum-e2178e6c20cb

https://blog.goodaudience.com/one-weird-trick-to-fix-user-on-boarding-d54b7ff9d711