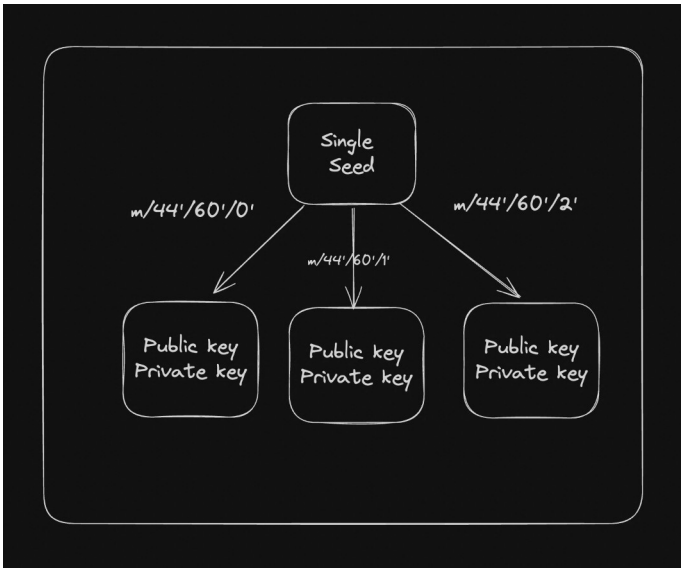
EARLIER WHAT WAS THE ISSUE WAS YOU HAVE TO KEEP PUBLIC AND PRIVATE KEYS FOR ALL THE WALLETS WHICH WERE KNOWN AS HD WALLETS FOR THAT WE HAVE TO REMEMBER SO MANY CODES

Hierarchical Deterministic (HD) Wallet

Hierarchical Deterministic (HD) wallets are a type of wallet that can generate a tree of key pairs from a single seed. This allows for the generation of multiple addresses from a single root seed, providing both security and convenience.

Problem

You have to maintain/store multiple public private keys if you want to have multiple wallets.



THIS WAS SOLVED BY BIP32 WHICH BROUGHT LIKE A THING OF MAINTAINING LIKE ONE MASTER SEED FROM WHICH THE OTHER KEYS CAN BE DERIVED

Solution - BIP-32

Bitcoin Improvement Proposal 32 (BIP-32) provided the solution to this problem in 2012. It was proposed by Pieter Wuilla, a Bitcoin Core developer, to simplify the recovery process of crypto wallets. BIP-32 introduced a hierarchical tree-like structure for wallets that allowed you to manage multiple accounts much more easily than was previously possible. It's essentially a standardized way to derive private and public keys from a master seed.

ETH and its innovation

Bitcoin had one limitation – it only solves for one usecase – a new currency (BTC) that hopefully has some value (digital gold/inflation hedge).

It deos not support use cases like –

1. Buying stocks
2. Stablecoins
3. Custom tokens
4. Lending/Borrowing

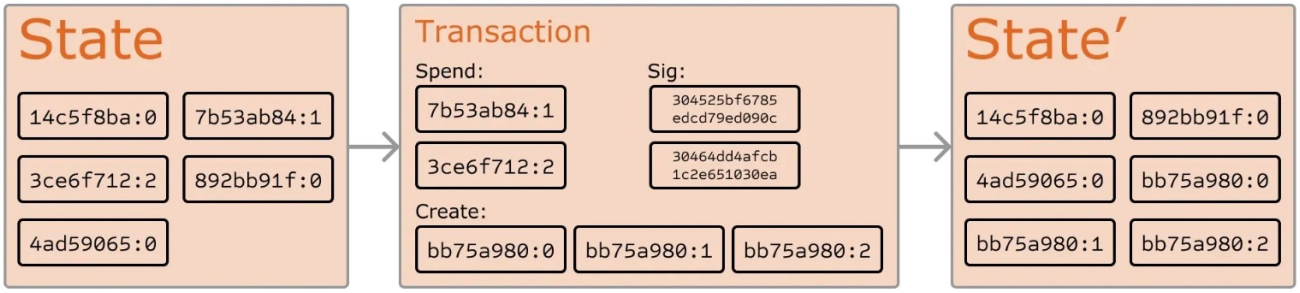
From 2011-2014, a bunch of usecase-specific blockchains emerged, which solved one of these usecases.

Then came ETH (~2014)

ETH whitepaper

https://ethereum.org/whitepaper/

Bitcoin As A State Transition System



From a technical standpoint, the ledger of a cryptocurrency such as Bitcoin can be thought of as a state transition system, where there is a "state" consisting of the ownership status of all existing bitcoins and a "state transition function" that takes a state and a transaction and outputs a new state which is the result. In a standard banking system, for example, the state is a balance sheet, a transaction is a request to move $X from A to B, and the state transition function reduces the value in A's account by $X and increases the value in B's account by $X. If A's account has less than $X in the first place, the state transition function returns an error. Hence, one can formally define:

Ethereum

The intent of Ethereum is to create an alternative protocol for building decentralized applications, providing a different set of tradeoffs that we believe will be very useful for a large class of decentralized applications, with particular emphasis on situations where rapid development time, security for small and rarely used applications, and the ability of different applications to very efficiently interact, are important. Ethereum does this by building what is essentially the ultimate abstract foundational layer: a blockchain with a built-in Turing-complete programming language, allowing anyone to write smart contracts and decentralized applications where they can create their own arbitrary rules for ownership, transaction formats and state transition functions. A bare-bones version of Namecoin can be written in two lines of code, and other protocols like currencies and reputation systems can be built in under twenty. Smart contracts, cryptographic "boxes" that contain value and only unlock it if certain conditions are met, can also be built on top of the platform, with vastly more power than that offered by Bitcoin scripting because of the added powers of Turing-completeness, value-awareness, blockchain-awareness and state.

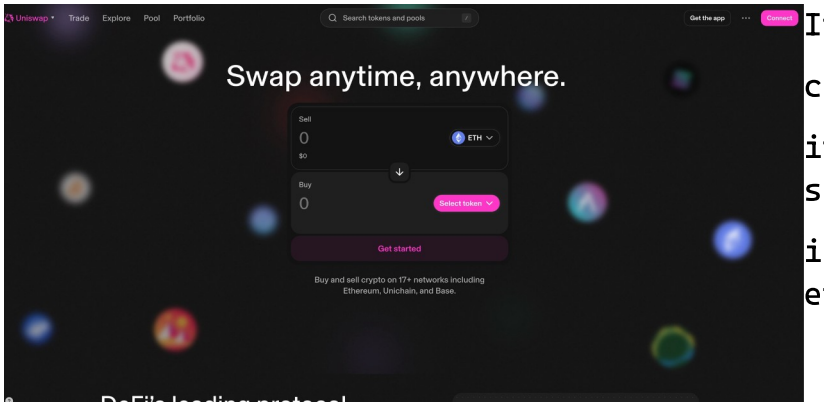
Smart contracts

Smart contracts are executable code that is stored on chain and can execute on-chain (machines of the miners).

The code is written in a language called solidity and the it executes on a virtual machine on the miner called an Ethereum virtual machine (EVM).

So the ETH blockchain stores every users ETH balance as one use case, but other than that 100s of use-cases can be written and deployed on the ETH blockchain.

For example, https://app.uniswap.org/ is a DEX that lets you swap one token for another.



Its a dapp on ethereum

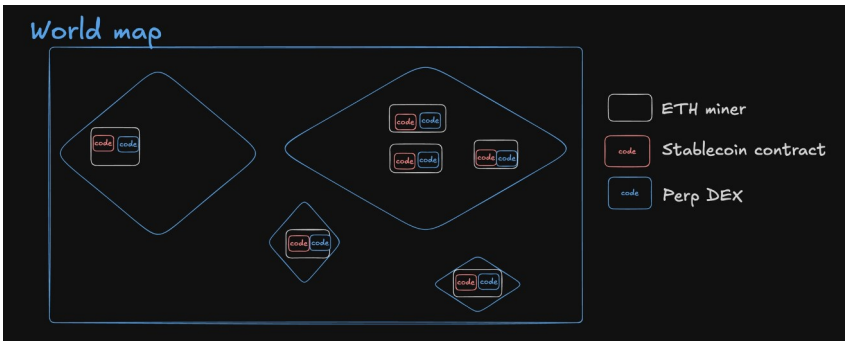
convert the like eth to usdc

its like a smart contract written in solidity which is deployed on the eth

in binary code and stored on the all eth mining machine

uniswap etherscan contract == name of the code

Smart contract on ETH -



https://github.com/jai123singh/AfterLife-Protocol/tree/main/smartContract/src

smart contracts are codes which we can make and that code will run on the all the miners machines

for eth == solidity

for solana == rust, c

pumpfun, jupyter

What are dapps

Dapps (decentralized applications) on Solana are applications built on the Solana blockchain that run on a distributed network rather than centralized servers.

How they work on Solana

Solana dapps consist of two main components: on-chain programs (smart contracts) written typically in Rust, and a frontend that interacts with those programs. The blockchain handles the backend logic—storing data, processing transactions, and enforcing rules—while users interact through familiar web or mobile interfaces.

What makes Solana different

Solana's architecture prioritizes speed and low costs. It uses a proof-of-stake consensus combined with a mechanism called proof-of-history, which timestamps transactions to enable parallel processing. This allows Solana to handle thousands of transactions per second with fees typically under a cent, making it attractive for applications requiring high throughput.

Common categories

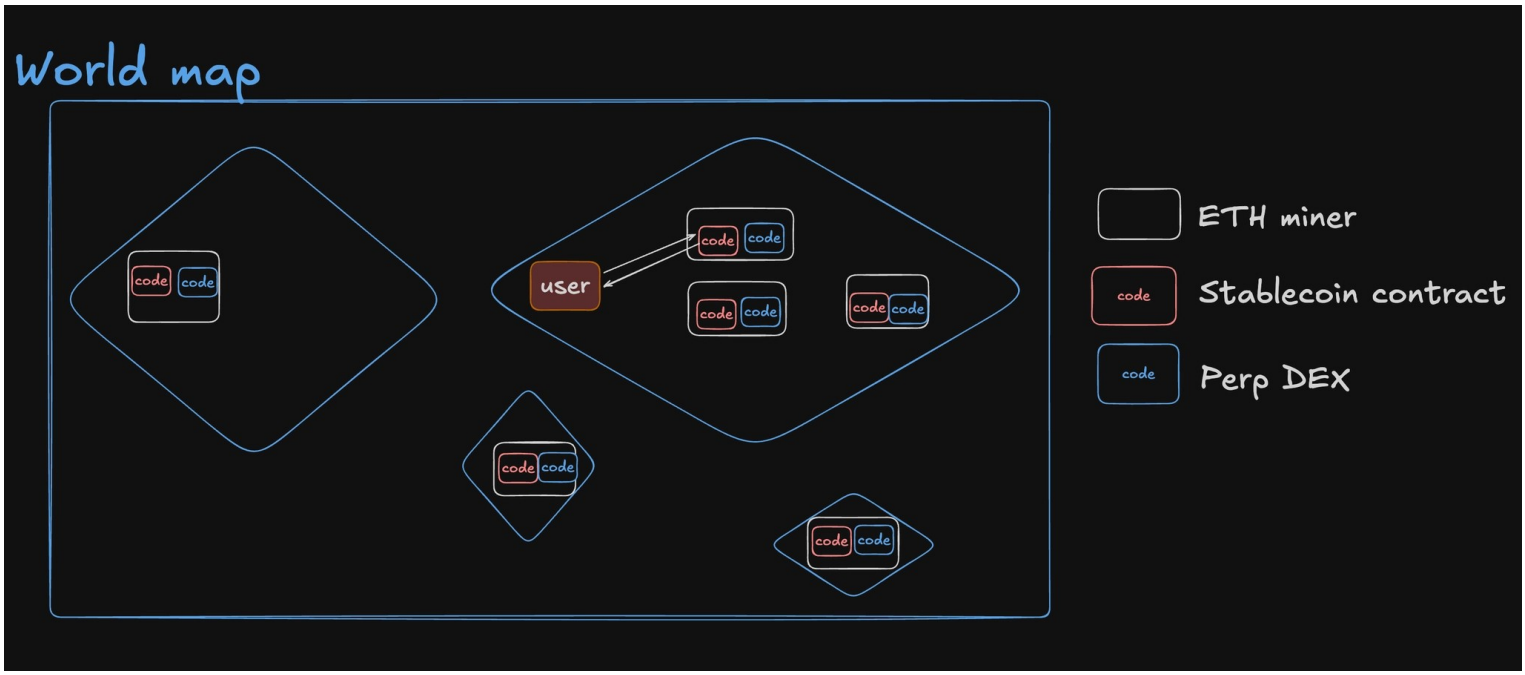
https://raydium.io/swap/ == coin swapper

https://jup.ag/

https://magiceden.io/

https://www.drift.trade/

https://sanctum.so/

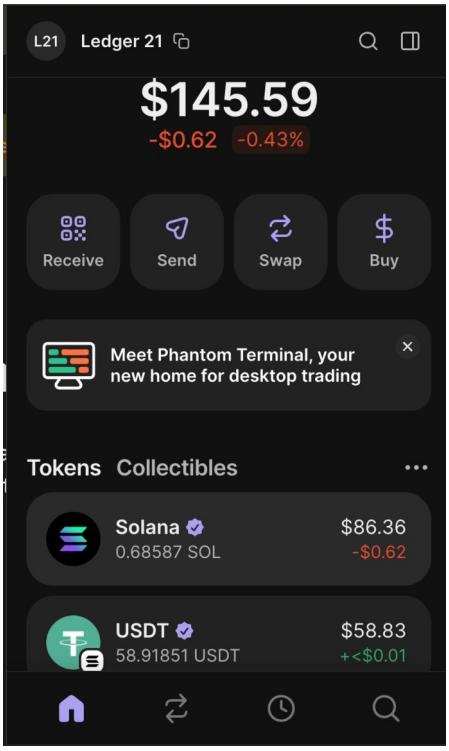


How this decentralized app work is like?
Suppose you want to convert 1 sol into usdc you enter that before that we connect the wallet to the dapp and which means just a connection is established between the system and then when you proceed with the transaction it shows a popup in which phantom will show what transaction is happeing(it like simulate the transaction and show what will actually happen) if you confirm that it will happen as it will sign that using your private key

WE CAN TRUST THE EXTENSION LIKE BACKPACK, PHANTOM BUT NOT ALL THE DAPPS WEBSITE WHICH ARE RUNNING SO TO DO THAT THESE EXTENSION EXIST WHICH CAN WORK ON ANY WEBSITE

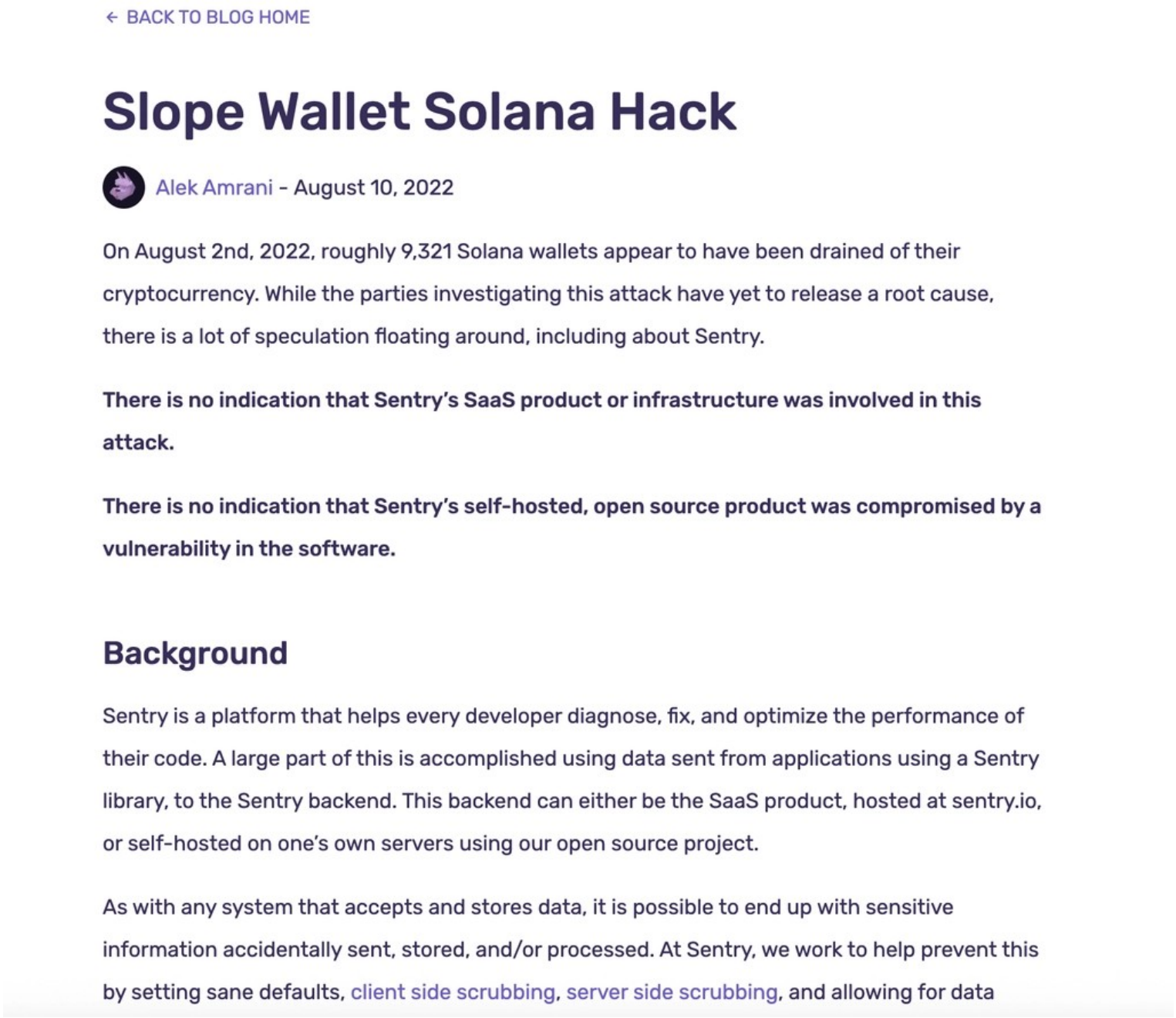
we never provide our pvt keys to the dapps website it just the wallet establish a connection nothing else

What are wallets extensions?
Wallet extensions are browser add-ons (like MetaMask, Phantom, Backpack) that let you interact with blockchains—storing your private keys, signing transactions, and connecting to decentralized applications (dApps).



Why are they extensions and not websites?
The core reason is security and trust. A browser extension runs locally on your device and can securely store your private keys in a way that a website fundamentally cannot.
Here's the problem with a website approach: if you entered your private key into a website, you'd be trusting that site's servers with complete control over your funds. The site could be compromised, the operators could be malicious, or the connection could be intercepted. Your keys would leave your device.
An extension solves this by keeping your keys local. When you visit a dApp and it needs you to sign a transaction, the dApp doesn't get your private key—instead, it sends a signing request to the extension. The extension shows you what you're approving, you confirm it, and the extension signs the transaction locally before sending only the signed result back. Your key never leaves your machine.
Extensions also have persistent access to inject code into web pages, which lets them provide a standardized interface (like window.solana) that any dApp can use to request connections and transactions.

Can phantom read my private key?
Yes and no. You cant really trust a wallet extension completely. It is totally possible a wallet could be logging your private key somewhere.
Ref - Slope wallet in 2022
https://blog.sentry.io/slope-wallet-solana-hack/



← BACK TO BLOG HOME

Slope Wallet Solana Hack

Alek Amrani - August 10, 2022

On August 2nd, 2022, roughly 9,321 Solana wallets appear to have been drained of their cryptocurrency. While the parties investigating this attack have yet to release a root cause, there is a lot of speculation floating around, including about Sentry.

There is no indication that Sentry's SaaS product or infrastructure was involved in this attack.

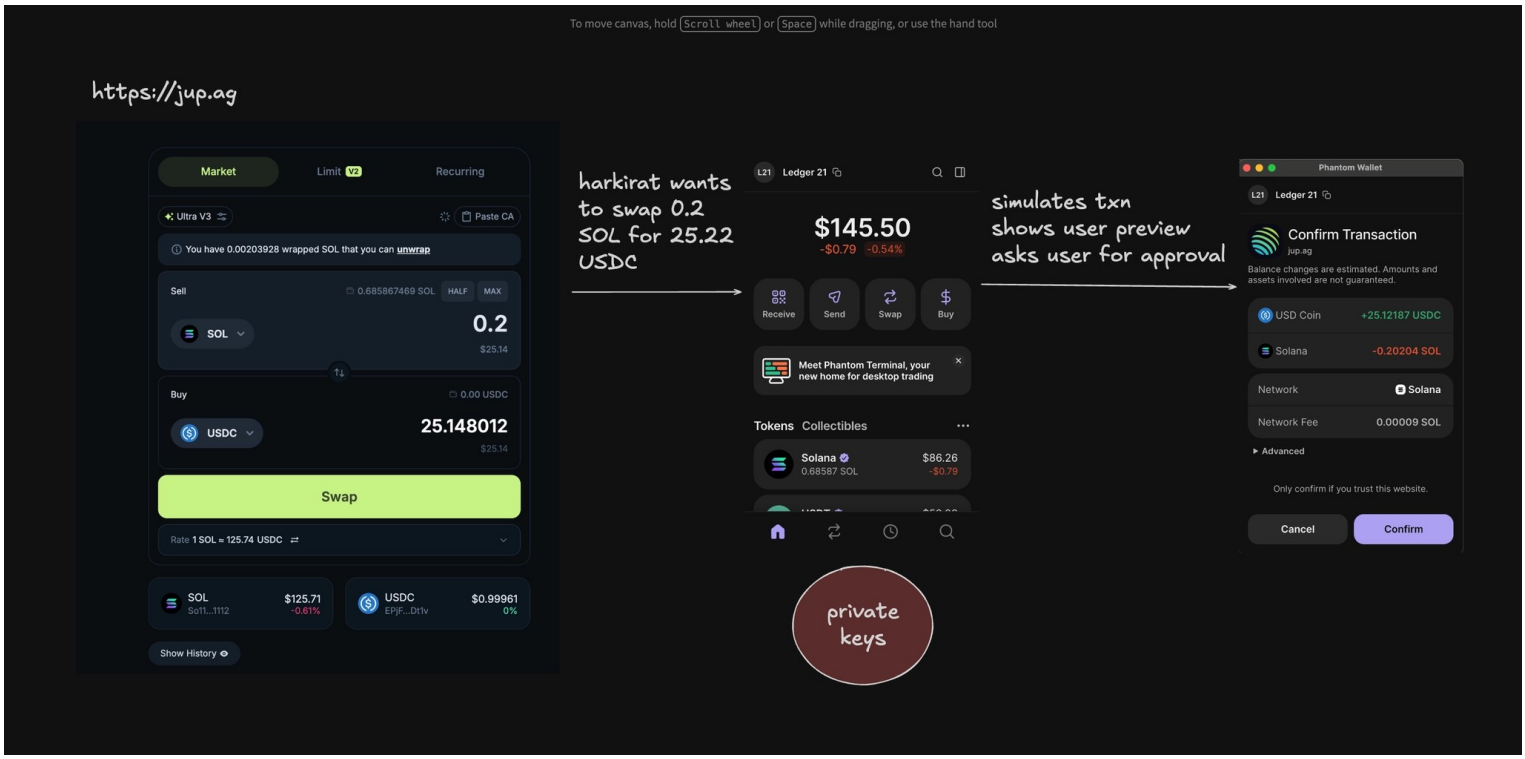
There is no indication that Sentry's self-hosted, open source product was compromised by a vulnerability in the software.

Background

Sentry is a platform that helps every developer diagnose, fix, and optimize the performance of their code. A large part of this is accomplished using data sent from applications using a Sentry library, to the Sentry backend. This backend can either be the SaaS product, hosted at sentry.io, or self-hosted on one's own servers using our open source project.

As with any system that accepts and stores data, it is possible to end up with sensitive information accidentally sent, stored, and/or processed. At Sentry, we work to help prevent this by setting sane defaults, client side scrubbing, server side scrubbing, and allowing for data

Then why do people use wallets?
People use wallet extensions because they want to interact with decentralized applications—and those dApps need a way to request transaction signatures from you.
The typical flow looks like this: you visit a dApp (say, a decentralized exchange like Uniswap), click "Connect Wallet," and your extension pops up asking if you want to connect. Once connected, when you want to swap tokens, the dApp constructs the transaction details and asks your extension to sign it. You review and approve in the extension, it signs locally, and broadcasts the transaction to the blockchain.
Without an extension (or some equivalent like a mobile wallet with WalletConnect), there's no secure bridge between the dApp in your browser and your private keys. The dApp can't sign transactions on its own—it needs your authorization, and the extension is the gatekeeper that makes that possible while keeping your keys safe.
So people use them because they're essentially the standard interface for using web-based crypto applications. If you want to trade on decentralized exchanges, mint NFTs, use DeFi lending protocols, or interact with any on-chain application through your browser, you need something to manage your keys and sign transactions. Extensions are the most convenient option for desktop users.



What is a ledger/hardware wallet?

Ledger is a physical wallet whose job is to store your seed phrase/private keys.

If you use a hardware wallet, your private keys are NEVER stored in phantom but always stored in a physical device.

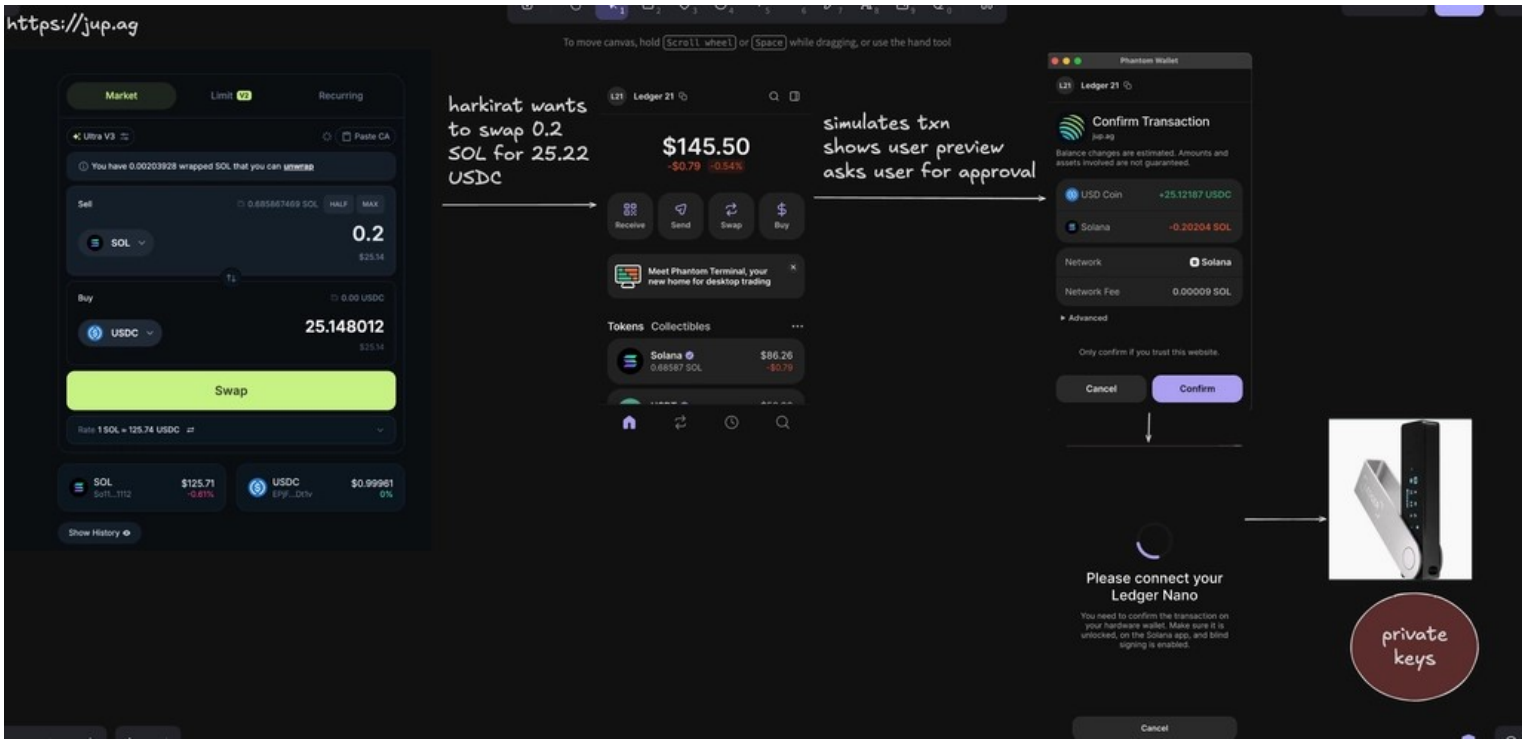
In this case though, you are trusting the creators of the ledger hardware wallet to not do anything shady.

So whats the ideal solution , what if even ledger as a company is compromised —

Unfortunately there isnt one.

You can create your private keys in your machine (terminal etc) directly, but then you’re trusting apple that they are not logging your storage somewhere.

The only other solution is multisigs



BLOWFISH == website maintains the list of good and bad website