

Dive Lane

Ultimate iOS bridge to distributed web

How did we come to this?

TONS OF DAPPS!!!


Maybe use some EIP...
Oh yeah!!!

eip	title	author	type	category	status	created	requires
681	URL Format for Transaction Requests	Daniel A. Nagy <daniel@ethereum.org>	Standards Track	ERC	Draft	2017-08-01	20,137,831


Simple Summary

A standard way of representing various transactions, especially payment requests in Ethers and ERC #20 tokens as URLs.


Wallets and keys
are filling my brain...

 destinatis commented 4 days ago



When is this gonna get implemented?

 ligi commented 4 days ago Member

@destinatis It is implemented - e.g. in [WalleTH](#)

 destinatis commented 4 days ago

@ligi IOS required.

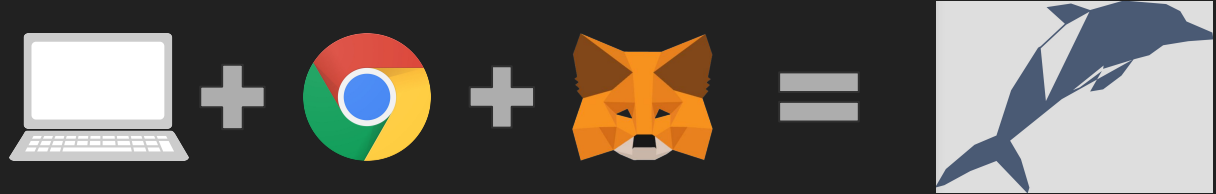
 1  1

iOS???

IOS NEEDS YOU!



So what?



Killer features:

- Cold Wallet for your private keys
- EIP#681 support
- **Signing and conducting transactions from ANYWHERE**
- No MetaMask + Chrome needed



But how?

- Open-source iOS wallet written on pure vanilla Swift
- Using Web3Swift open-source lib to implement Web3 features
- Using deep links based on EIP#681 - payment request URL specification

such blockchain

wow

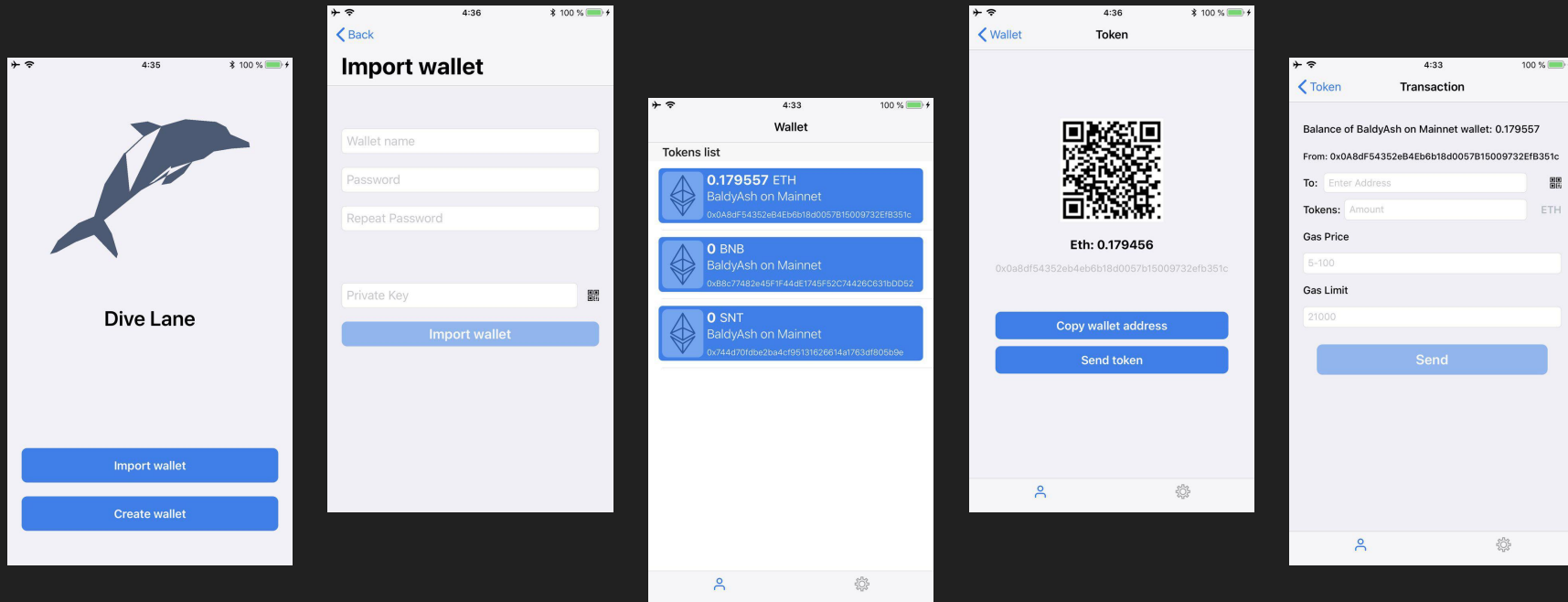
many EIPs

much web3



What we managed to make

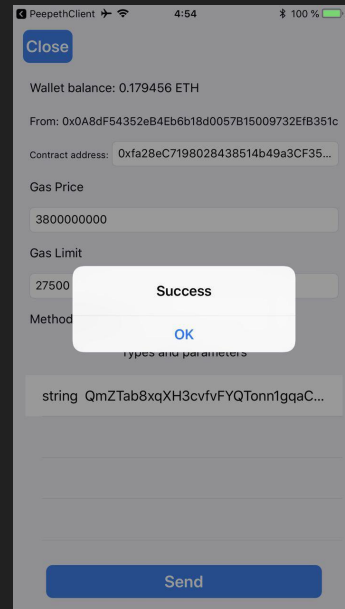
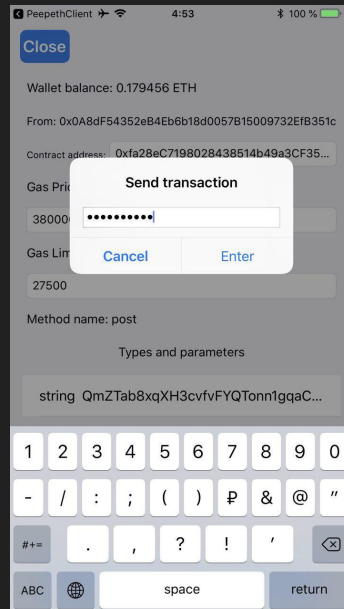
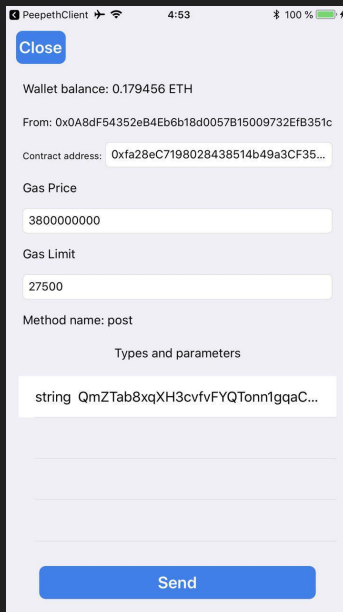
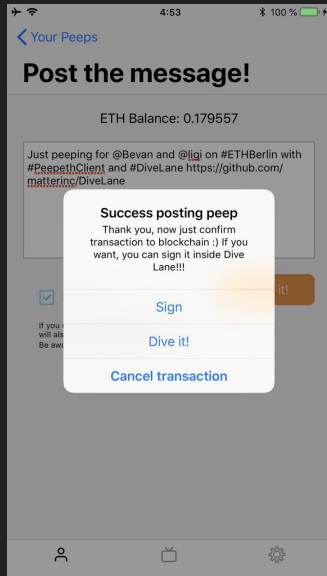
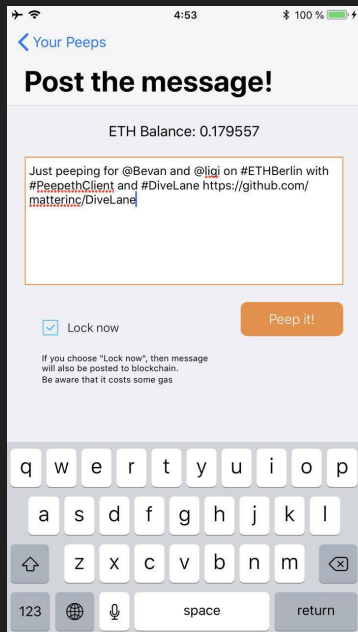
Just a simple wallet on the first glimpse, but...



Sign transactions from anywhere???

YES, we BUIDL it!

Easy and safe keys storage for DApp!!!



Just deeplinking

Easy and fast

In yours best Swift DApp client:

```
//following by EIP-681:  
let urlString = "ethereum:0xaaf3A96b8f5E663Fc47bCc19f14e10A3FD9c414B/pay?uint256=1000000&value=1000"  
UIApplication.shared.open(URL(string: urlString)!, options: [:], completionHandler: nil)
```

Also look at PR for Peepeth Client:

<https://github.com/matterinc/PeepethClient/pull/8>:



What's next?

- Android (TBD)
- JS(PR is during development already)
- IoT libraries + NFC support



Thank you for your attention!