BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2025
http://excel.fit.vutbr.cz

Design Options for Web3 Non-Custodial Cryptocurrency Wallet based on Web2 Authentication

Bc. Adam Maczkó*

Abstract
Cryptocurrency newcomers still face a stark choice: either master confusing seed phrases or hand their money to custodial exchanges. This paper aims to close that gap by designing a wallet that feels like a familiar Web 2.0 login yet never gives up user control. We review leading key-splitting methods and wallet types, then combine their best ideas into an embedded ”2-of-3” scheme: one key on paper, one encrypted in the browser, one locked in a secure server enclave. Routine micro-payments need only one tap, while larger transfers require a signature by the ”master keys”. All enforced by a smart-contract account. Early tests show the prototype is as easy to use as a password manager but as safe as full self-custody, offering a practical path to wider blockchain adoption.

*xmaczk00@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Decentralized finance promises direct asset ownership, but practical self-custody remains difficult for newcomers [1]. Users used to Web 2.0 logins often choose custodial exchanges that hold their private keys. The 2022 FTX failure shows the systemic risk of that choice [2]. A wallet that combines familiar authentication with verifiable self-custody would reduce this risk and lower the barrier to responsible cryptocurrency use.

We investigate how a Web 2.0 authentication flow can be combined with non-custodial key ownership. A satisfactory design should meet five criteria:

- **Self-custody** – no single party can spend funds alone.
- **Basic usability** – initial setup should not rely on expert key management.
- **Recoverability** – mechanisms for recovery of the private key.(*e.g.*, guardians in Smart Contract Wallets).
- **Liveness** – funds remain accessible when the vendor server is offline.
- **Cost** – on-chain operations are acceptable in terms of gas fees on Ethereum.

Existing solutions. Custodial wallets offer familiar log-ins but sacrifice self-custody (FTX example). Browser extension wallets such as MetaMask return control to the user but demand seed-phrase handling. MPC wallets (*e.g.*, ZenGo) distribute key shares but still depend on a vendor-controlled component [3]. Embedded wallets (*e.g.*, Thirdweb) simplify onboarding by reducing the burden of manual key management, but rely on continuous server availability [4]. Shamir social-recovery schemes (poster Fig. 1) eliminate vendor custody but impose off-chain key reconstruction before every signature. ERC-4337 smart-contract accounts allow for programmable policies but leave key management to the integrator. None of these approaches fully satisfies all five criteria.

We outline an embedded wallet that combines an ERC-4337 account with a 2-of-3 multi-signature policy (poster Fig. 2).

Authentication uses OpenID Connect with optional passkey MFA. For low-value transactions, the wallet issues a capped session key. Attempts above the cap are rejected and must be resubmitted with signatures from any two of the three master keys. This approach costs more gas than Shamir recovery but avoids explicit key reconstruction and maintains operation if the server becomes unavailable. Unsigned transaction data are sent over TLS to the Trusted Execution Environment (TEE), which returns a detached signature.

2. Key management alternatives

Early work focused on three established techniques for splitting or sharing a private key. Shamir's Secret Sharing (poster Fig. 1) cuts the secret into n shares so that any m can rebuild it. Fault tolerance is strong, but every signature requires an explicit reconstruction step, which poses a threat where an attacker can steal the private key. Multiparty computation (MPC) eliminates that overhead by letting fragments cooperate on a joint signature; vendors such as ZenGo show that the model works in production, but the price is an always-online, vendor-controlled share and a more complex recovery path [3]. Multisignature schemes (poster Fig. 2) keep each key independent and allow the blockchain itself to verify that a threshold of signatures is present. Because no key ever leaves its owner and no reconstruction is required, multisig aligns well with transparent auditing and with partial availability: if one signer is offline, the others can still complete the threshold. This analysis identified multisignature as the most practical foundation for a consumer wallet, balancing resilience, simplicity, and on-chain costs.

3. Wallet architecture alternatives

With multisignature chosen, several wallet options were assessed. A browser-embedded wallet keeps users inside the dApp and can benefit from Web 2.0 authentication services such as Thirdweb's embedded wallets, which generate self-custodial accounts after an email or social-media login and even offer passkey-based 2FA [5]. A smartphone "hardware" wallet that relies on the Trusted Platform Module initially looked attractive, but real-world TPM 2.0 chips typically support the *secp256r1* curve and not the *secp256k1* [6]. Because Ethereum relies on *secp256k1*, verifying an r1 signature on-chain would require an expensive precompile or a costly custom verifier [7]. Traditional USB hardware wallets provide strong isolation, but interrupt the seamless web flow and impose additional hardware on novices. Finally, ERC-4337 smart contract wallets enforce policy on the chain and allow features such as spending limits and social recovery, but still depend on a reliable signing back-end and a clear distribution of key shares [8].

4. Proposed architecture overview

The wallet is delivered as an embedded browser component (poster Fig. 3) that greets users with a standard OpenID Connect flow, so signing in with an e-mail address or a social account feels no different from conventional Web 2.0 sites. After successful login, an optional FIDO2 passkey can raise the bar against phishing, and the client decrypts a locally stored private key that had been encrypted with AES using a key derived from the user password. No seed phrase is exposed, which keeps the initial experience familiar.

Key custody relies on a 2-of-3 ECDSA multi-signature set (poster Fig. 2). One private key is printed as a mnemonic that the user keeps offline. The second, just unlocked, resides in the browser in encrypted form. The third is held inside a server-side TEE that never releases raw key material. Because each signer is independent, no explicit reconstruction is needed, and the wallet can function as long as any two keys are available.

When the user wishes to transfer funds the client, assembles an unsigned UserOperation, transmits it over TLS to the server, receives the signature of UserOp and adds its own. If the payment amount falls below a preset spending cap, the operation may instead be signed by a session key that was authorised earlier by two master keys. Any attempt to exceed that limit is rejected and must be resubmitted with fresh signatures from two of the three master keys. This rule avoids Shamir-style key reconstruction while accepting only a modest gas premium relative to single-signature accounts. The design also tolerates server outages: the mnemonic plus the browser key still satisfy the threshold, whereas the TEE key alone cannot drain funds.

5. Conclusion

This design shows that a Web 2.0 log-in can be integrated with a non-custodial wallet. A 2-of-3 multisig guarded by a TEE and smart-contract limits delivers practical security without seed-phrase hassle, pointing toward safer mainstream crypto use.

6. Acknowledgements

I would like to thank my supervisor, doc. Ing. Ivan Homoliak, Ph.D., for his valuable insights.

References

[1] Tanusree Sharma, Vivek C Nair, Henry Wang, Yang Wang, and Dawn Song. "i can't believe it's not custodial!". In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–16, New York, NY, USA, 2024-05-11. ACM.

[2] Shange Fu, Qin Wang, Jiangshan Yu, and Shiping Chen. Ftx collapse: a ponzi story. In *International*

Conference on Financial Cryptography and Data Security, pages 208–215. Springer, 2023.

[3] Nikolaos Makriyannis, Oren Yomtov, and Arik Galansky. Practical key-extraction attacks in leading mpc wallets. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 3053–3064, New York, NY, USA, 2024-12-02. ACM.

[4] Ivan Homoliak and Martin Perešíni. Sok: Cryptocurrency wallets – a security review and classification based on authentication factors, 2024.

[5] Thirdweb. Thirdweb wallet security.

[6] Wei-Yang Chiu, Weizhi Meng, and Wenjuan Li. Tpmwallet: Towards blockchain hardware wallet using trusted platform module in iot. In *2023 International Conference on Computing, Networking and Communications (ICNC)*, pages 336–342, 2023.

[7] Dogan Alpaslan and Noam Hurwitz. What is rip-7212? precompile for secp256r1 curve support.

[8] Xinxin Fan and Xueping Yang. Enabling web2-based user authentication for account abstraction. In *2024 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 9–10, 2024.