Account Types

Suggest Edits

In addition to the wallet infrastructure model, the account type is crucial in determining the wallet's functionality and capabilities. Circle Programmable Wallets offer two account types:Externally Owned Accounts (EOA) andSmart Contract Accounts (SCA) .

Externally Owned Accounts (EOA)

EOA is the most commonly used account type in Ethereum. It consists of a private and public key, with the public key representing the account address. EOAs require secure management of the private key and always require gas tokens for initiating transactions. While EOAs offer the flexibility to initiate any transaction, they have certain limitations, such as the need for a private key and the inability to control them programmatically.

Benefits:

- Zero Cost Creation:
- Creating an Externally Owned Account (EOA) does not incur any fees, making it an accessible option for users. This is especially important for Networks like Ethereum Mainnet, where gas costs are higher.
- Versatile Transaction Initiation:
- EOAs can initiate any transaction on the Ethereum network, allowing users to interact with various decentralized applications and perform a wide range of operations.

Limitations:

- Private Key Dependency:
- EOAs require the secure management of a private key. Users must protect their private key to ensure the security and control of their EOA, as it grants access to account functions and funds.
- Native Token Dependency for Gas:
- To perform transactions or execute operations on the Ethereum network, EOAs must hold native tokens such as Ether (ETH) to cover gas fees. These tokens must pay for the computational resources utilized during transaction validation and execution.

Smart Contract Accounts (SCA)

SCA is another type of Ethereum that replaces the EOA with a smart contract. SCAs do not have private keys but are often controlled by an EOA that interacts with the deployed smart contract. SCAs allow developers to write customized logic within the smart contract, enabling advanced functionalities.

SCAs have been in the Ethereum ecosystem for many years but recently gained popularity thanks to a new standard called ERC-4337 (account abstraction). ERC 4337 went live on Mainnet in March 2023, enabling a trustless standard the community could build SCA around. ERC-4337 also introduced the concept of paymaster - which enables simplified gas abstraction, enabling anyone to sponsor gas fees for the end user or letting the end user pay for gas in any ERC-20 token.

When considering wallet infrastructure models, factoring in the infrastructure model (user-controlled or developer-controlled) and the account type (EOA or SCA) is essential. User-controlled wallets offer autonomy and security while providing a seamless user experience, while developer-controlled wallets simplify user interactions with the blockchain. Choosing the most suitable combination of infrastructure model and account type empowers developers to create powerful and user-friendly applications tailored to their needs.

Benefits:

- Custom Logic and Functionality:
- Smart Contract Accounts (SCAs) allow developers to build custom logic and functionality directly into the account. By deploying smart contracts to control SCAs, developers can create innovative and customized features, enabling advanced capabilities and automated operations within the wallet.
- Gas Abstraction and Fee Sponsorship:
- SCAs offer gas abstraction, simplifying gas fee management for end-users. Developers can implement mechanisms, such as the paymaster pattern, to sponsor gas fees on behalf of users. This reduces the burden on users to hold and manage native tokens for gas and enhances the overall user experience by providing a seamless gas fee payment process.

Limitations:

- Gas Tokens Required for Creation:
- Gas tokens are required when creating SCA wallets. However, emerging solutions like the one implemented in Programmable Wallets address this limitation by introducing a lazy deployment mechanism. This process postpones

the gas fee payment until the first outbound transaction, easing the initial cost burden of SCA wallet creation.

Choosing between EOAs and SCAs

An EOA or an SCA can be chosen by specifying the account type parameter during wallet creation, as shown below.

Applicable APIs:

- Create a user-controlled challenge for PIN setup and create wallet(s)POST /user/initialize
- .
- Create a user-controlled wallet(s):POST /user/wallets
- .
- Create a developer-controlled wallet(s)POST /developer/wallets
- .

You can click the linked API references to see them in action.

Account Type Parameter // EXAMPLE REQUEST BODY IN JSON // For deveveloper-controlled wallets // entityentitySecretCiphertext and walletSetId are also required in the request.

// SCA { "idempotencyKey": "a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11", "accountType": "SCA", // HERE "blockchains": ["MATIC-MUMBAI"] }

// EOA { "idempotencyKey": "a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11", "accountType": "EOA", // HERE "blockchains": ["MATIC-MUMBAI"] }

Selection guidance

Based on our experience, SCAs generally provide a more straightforward user experience for most use cases. They allow developers to focus on creating innovative functionalities within the smart contract. However, the specific needs of each project should be carefully considered when choosing between EOA and SCA account types.

If you are building on Ethereum - creating an SCA wallet might become expensive, and we recommend using EOAs. If you are using any Polygon or any L2s, we strongly recommend using SCAs.

Key Considerations

Externally Owned Account (EOA) Smart Contract Account (SCA) Fees for account creation No, it is free Yes, creating a wallet incurs gas fees. Note: With our process of lazy deployment, you won't have to pay the gas fee at the time of wallet creation. Instead, the fee will be charged when you initiate your first outbound transaction. Gas abstraction No, this isn't feasible Yes, it is supported via Paymasters. Modular plugins No Yes, modules/plugins are coming soon that extend the functionality of the base wallet. Where it makes sense If you are building on Ethereum (since gas costs on Ethereum can be expensive) If you want a web2-type user experience that makes blockchain invisible or wants advanced control of your wallets e.g. embedded wallets, payments wallets, developer wallets, and vault type wallet.

Supported Blockchains

Blockchain Network Externally Owned Account (EOA) Smart Contract Account (SCA) Avalanche Fuji Testnet Avalanche Mainnet Ethereum Sepolia Testnet Ethereum Mainnet

(Coming Soon) Polygon Mumbai Testnet Polygon Mainnet Updated16 days ago *Table of Contents * * Externally Owned Accounts (EOA) * * Smart Contract Accounts (SCA) * * Choosing between EOAs and SCAs * * * Selection guidance * * * Key Considerations * * * Supported Blockchains