How to use a Capsule signer with permissionless.js

Capsule offers a signing solution enabling the creation of secure, embedded MPC wallets accessible via email or social login. These wallets, compatible across different applications, offer portability, recoverability, and programmability, eliminating the need for users to establish separate signers or contract accounts for each application.

Setup

To use Capsule with permissionless.js, first create an application that integrates with Capsule.

- Refer to theCapsule documentation site
- for instructions on setting up an application with the Capsule.
- For a quick start, Capsule provides an example hub, availablehere
- .

Integration

Integrating permissionless.js with Capsule is straightforward after setting up the project. Capsule provides an Externally Owned Account (EOA) wallet to use as a signer with permissionless.js accounts.

Create the Capsule signer

After following the Capsule documentation, you will have access to aCapsuleWeb3Provider object that you can use to create aSmartAccountSigner object:

```

importCapsule, { createCapsuleViemClient }from"@usecapsule/web-sdk" import{ walletClientToSmartAccountSigner }from"permissionless" import{ http }from"viem" import{ sepolia }from"viem/chains"

// Param options here will be specific to your project. See the Capsule docs for more info. constcapsule=newCapsule(env, apiKey)

// Convert a Capsule viem client to a SmartAccountSigner // Follow the Capsule docs for more instructions on creating the Viem client https://docs.usecapsule.com/integration-guide/signing-transactions constviemClient=createCapsuleViemClient(capsule, { chain: sepolia, transport:http("https://rpc.ankr.com/eth_sepolia"), })

constsmartAccountSigner=walletClientToSmartAccountSigner(viemClient)

```

Use with permissionless.js

SimpleAccount Safe Account Kernel Account Biconomy Account ```

SimpleAccount import{ signerToSimpleSmartAccount }from"permissionless/accounts" import{ createPublicClient, http }from"viem" import{ generatePrivateKey, privateKeyToAccount }from"viem/accounts" import{ sepolia }from"viem/chains"

exportconstpublicClient=createPublicClient({ transport:http("https://rpc.ankr.com/eth_sepolia"), chain: sepolia, })

constsmartAccount=awaitsignerToSimpleSmartAccount(publicClient, { signer: smartAccountSigner, factoryAddress:"0x9406Cc6185a346906296840746125a0E44976454", entryPoint:ENTRYPOINT_ADDRESS_V06, })

```
```