D4D4D4;--ch-t-background: #1E1E1E;--ch-t-lighter-inlineBackground: #1e1e1ee6;--ch-t-editor-background: #1E1E1E;--ch-t-editor-foreground: #D4D4D4;--ch-t-editor-rangeHighlightBackground: #ffffff0b;--ch-t-editor-infoForeground: #3794FF;--ch-t-editor-selectionBackground: #264F78;--ch-t-focusBorder: #007FD4;--ch-t-tab-activeBackground: #1E1E1E;--ch-t-tab-activeForeground: #ffffff;--ch-t-tab-inactiveBackground: #2D2D2D;--ch-t-tab-inactiveForeground: #ffffff80;--ch-t-tab-border: #252526;--ch-t-tab-activeBorder: #1E1E1E;--ch-t-editorGroup-border: #444444;--ch-t-editorGroupHeader-tabsBackground: #252526;--ch-t-editorLineNumber-foreground: #858585;--ch-t-input-background: #3C3C3C;--ch-t-input-foreground: #D4D4D4;--ch-t-icon-foreground: #C5C5C5;--ch-t-sideBar-background: #252526;--ch-t-sideBar-foreground: #D4D4D4;--ch-t-sideBar-border: #252526;--ch-t-list-activeSelectionBackground: #094771;--ch-t-list-activeSelectionForeground: #fffffe;--ch-t-list-hoverBackground: #2A2D2E; }

Multi-Agent Setup

In this guide, you'll learn how to set up and manage a Safe Smart Account with multiple agents. This setup ensures that every transaction proposed by one agent is approved by at least one other agent. To maintain full functionality, we recommend including human signers in addition to the AI agents.

Below, we demonstrate a 2-out-of-4 setup as an example.

Two Agents Propose, Check, and Execute Transactions

Setup Safe Smart Account with agent one

You will use the SafeProtocol Kit .

_17 import Safe from '@safe-global/protocol-kit' _17 _17 const AGENT_1_ADDRESS = // ... _17 const AGENT_1_PRIVATE_KEY = // ... _17 const AGENT_2_ADDRESS = // ... _17 const HUMAN_SIGNER_1_ADDRESS = // ... _17 const HUMAN_SIGNER_2_ADDRESS = // ... _17 const RPC_URL = 'https://rpc.ankr.com/eth_sepolia' _17 _17 const newSafe = await Safe.init({ _17 provider: RPC_URL, _17 signer: AGENT_1_PRIVATE_KEY, _17 safeOptions: { _17 owners: [AGENT_1_ADDRESS, AGENT_2_ADDRESS, HUMAN_SIGNER_1_ADDRESS, HUMAN_SIGNER_2_ADDRESS], _17 threshold: 2 _17 } _17 }) The Smart Account is now created with a fixed address. If the account has not been deployed yet, it will automatically deploy when the first transaction is executed.

Propose a Transaction with Agent One

Agent One can now propose transactions. We recommend sending these transactions to theSafe Transaction Service .

Using this service provides several benefits:

- It allows Agent Two to easily receive, sign, and execute the transaction.
- Transactions appear in the Safe Wallet interface, where human signers can review, approve, and execute them.

You can use theAPI Kit to propose transactions to the Safe Transaction Service.

Here's an example of how Agent One can propose a simple transaction to the zero address:

_29 import SafeApiKit from '@safe-global/api-kit' _29 _29 const apiKit = new SafeApiKit({ _29 chainId: 11155111n _29 }) _29 _29 const tx = await newSafe.createTransaction({ _29 transactions: [_29 { _29 to: '0x0000000000000000000000000000000000000000', _29 data: '0x', _29 value: '0' _29 } _29] _29 }) _29 _29 // Every transaction has a Safe (Smart Account) Transaction Hash different than the final transaction hash _29 const safeTxHash = await newSafe.getTransactionHash(tx) _29 // The AI agent signs this Safe (Smart Account) Transaction Hash _29 const signature = await newSafe.signHash(safeTxHash) _29 _29 // Now the transaction with the signature is sent to the Transaction Service with the Api Kit: _29 await apiKit.proposeTransaction({ _29 safeAddress: safeAddress, _29 safeTransactionData: tx.data, _29 safeTxHash, _29 senderSignature: signature.data, _29 senderAddress: AGENT_ADDRESS _29 })

Receive and sign transaction with agent two

In the next step, the second AI agent needs to receive the transaction and, after performing any necessary checks, sign and execute it.

The second AI agent will run on its own machine, so you would have to initialize the Safe instance with the Smart Account's address.

_18 const SAFE_ADDRESS = '0x...' // The address of the Smart Account from step one _18 _18 // Initialize the Safe object with the same address, but a different signer _18 const existingSafe = await Safe.init({ _18 provider: RPC_URL, _18 signer: AGENT_2_PRIVATE_KEY, _18 safeAddress: SAFE_ADDRESS _18 }) _18 _18 // Get pending transactions that need a signature _18 const pendingTransactions = await apiKit.getPendingTransactions(SAFE_ADDRESS) _18 // We assume there is only one pending transaction _18 const transaction = pendingTransactions.results[0] _18 _18 // Here, your AI agent could check this transaction. _18 _18 // As only one more signater is required, AI agent two can execute the transaction: _18 existingSafe.executeTransaction(transaction)

Next steps

Your AI agents can make autonomous decisions, and the human signers can do so, too. We are exited to see what you will build.

If you have a technical question, feel free to reach out onStack Exchange(opens in a new tab) with the safe-core tag.

Human approval for agent action Agent with spending limit Was this page helpful?

Report issue