Glossary

This glossary contains terms and definitions used throughout the Safe documentation.

Account Abstraction

Account Abstraction is a new paradigm that focuses on improving the user experience when interacting with the blockchain by replacing the use ofexternally-owned accounts withsmart accounts .

Some Account Abstraction features are:

- Elimination of seed phrase reliance
- Ease of multichain interactions
- Account recovery
- Gasless transactions
- Transaction batching

See also:

- Account Abstraction(opens in a new tab)
- on ethereum.org
- ERC-4337: Account Abstraction(opens in a new tab)
- on erc4337.io

Bundler

Bundlers are nodes that participate in theERC-4337 standard who bundleuser operations from an alternative mempool and submit them to the blockchain. Bundlers pay for the associated transaction fees in advance, which are later refunded by the user account who proposed the user operation or by aPaymaster .

See also:

- Bundling(opens in a new tab)
- process on ethereum.org
- Bundlers(opens in a new tab)
- on erc4337.io

EIP-1271

TheEIP-1271(opens in a new tab) is anEthereum Improvement Proposal(opens in a new tab) that proposes a standard way for any contract to verify whether a signature on behalf of a given contract is valid. This is possible via the implementation of aisValidSignature(hash, signature) function on the signing contract, which can be called to validate a signature.

EIP-712

TheEIP-712(opens in a new tab) is an Ethereum Improvement Proposal that proposes a standard for hashing and signing of typed structured data instead of just bytestrings.

EntryPoint

According to theERC-4337 , the EntryPoint is the singleton smart contract that processes bundles ofUserOperation objects sent by theBundlers . It verifies and executes them by calling the target smart accounts according to predefined rules.

See also:

- EntryPoint(opens in a new tab)
- on ethereum.org
- Bundlers(opens in a new tab)
- on erc4337.io

ERC-4337

TheERC-4337(opens in a new tab) is anEthereum Request for Comments(opens in a new tab) that introduces a higher-layer pseudo-transaction object calledUserOperation . Users sendUserOperation objects into a separate mempool. A special class of actor calledBundlers package up a set of these objects into a transaction making ahandleOps call to a special contract, and that transaction then gets included in a block.

See also:

- ERC-4337 Documentation(opens in a new tab)
- on erc4337.io

Externally-Owned Account

An externally-owned account (also known as EOA) is one of the two types of Ethereum accounts. A private key controls it; it has no code, and users can send messages by creating and signing Ethereum transactions.

See also:

- Ethereum Accounts(opens in a new tab)
- on ethereum.org
- Ethereum Whitepaper(opens in a new tab)
- on ethereum.org

Gasless Transaction

Gasless transactions (also known as meta-transactions) are Ethereum transactions that are executed by a third party calledrelayer on behalf of asmart account to abstract the use of gas. Users must sign a message (instead of the transaction itself) with information about the transaction they want to execute. A relayer will create the Ethereum transaction, sign and execute it, and pay for the gas costs. The main benefit is that users can interact with the blockchain without holding the native token in their account.

See also:

- Relay Kit documentation
- on docs.safe.global

Multi-signature

A multi-signature account is asmart account that allows you to customize ownership and control structures according to your needs. Multipleexternally-owned accounts can be designated as owners, and you can specify how many of those owners must approve a transaction before it is executed.

Possible configurations:

- 0/0 Safe
- : An account with no owners, controlled entirely bySafe Modules
- . This configuration is typically used for automation or executing conditional functions within a protocol's architecture.
- 1/1 Safe
- : An account with a single owner who has full control and ownership. Ideal for setting up a smart account with a single EOA that can take advantage of all smart account functionalities. Warning: If the owner loses access to their private key, there is no way to recover the account. A recovery plan or an emergency mechanism to handle key loss should be set.
- N/N Safe
- : An account with multiple owners, all of whom must approve a transaction before it is executed. This setup is perfect for scenarios where equal ownership and responsibility among all participants are required. Warning: If any owner loses their private key, the Safe may become locked and unable to process transactions. A recovery plan or an emergency mechanism to handle key loss should be set.
- N/M Safe
- : An account with multiple owners, but only a subset of them is required to approve a transaction. This configuration is useful when you want to distribute responsibility while maintaining flexibility in decision-making.

How does it work?

- Multi-signature is a function native to Safe smart contract.
- The contract stores the owners' addresses and the threshold needed to execute a transaction in the smart contract storage.
- When the user wants to perform a transaction, they send a payload containing the transaction details and the owners' signatures to the Safe account.
- The safe account iterates through the signatures to verify that the payload has been signed correctly and by the correct owners.

Benefits:

- Enhanced Security
- : Reduces the risk associated with single points of failure, protecting your assets even if a key is lost or compromised.
- Customizable
- : Tailor each smart account to fit your specific needs, allowing you to set up configurations that work best for your particular use case.
- Interoperable
- : Flexibly assign a variety of signers as owners, including:* Hardware wallets such asLeger(opens in a new tab)
-
    - orTrezor(opens in a new tab)
-
    - .
-
    - Software wallets such asTrust(opens in a new tab)
-
    - orMetamask(opens in a new tab)
-
    - .
-
    - MPC wallets such asFireblocks(opens in a new tab)
-
    - orZengo(opens in a new tab)
-
    - .
-
    - Another smart contract account, such as Safe.
-
    - Wallets generated via Social Logins or Passkeys.
- Upgradable
- : Easily adjust the number of owners and the signing threshold for your account whenever
- Key Rotation
- : Rotate ownership of any accounts at any time, maintaining security while adapting to changes.
- Shared Control
- : Grant shared access and control of your account to multiple individuals, ensuring collaborative management.
- Auditability
- : Maintain a transparent, auditable record of who signed each transaction and when providing clear accountability for all account activities.

Network

A blockchain network is a collection of interconnected computers that utilize a blockchain protocol for communication. Decentralized networks allow users to send transactions, that are processed on a distributed ledger with a consensus mechanism ensuring the batching, verification, and acceptance of data into blocks. This structure enables the development of applications without the need for a central authority or server.

See also:

- Networks(opens in a new tab)
- on ethereum.org

Owner

A Safe owner is one of the accounts that control a given Safe. Only owners can manage the configuration of a Safe and approve transactions. They can be eitherexternally-owned accounts orsmart accounts . Thethreshold of a Safe defines how many owners need to approve a Safe transaction to make it executable.

See also:

- OwnerManager.sol(opens in a new tab)
- on github.com

Paymaster

Paymasters are smart contracts that allow an account to pay for the gas fees of other users. This feature abstracts away the concept of gas fees by subsidizing them for users, allowing them to pay with ERC-20 tokens, and enables many other use cases.

See also:

- Paymasters(opens in a new tab)
- on ethereum.org
- Paymasters(opens in a new tab)
- on erc4337.io

Relayer

A relayer is a third-party service acting as an intermediary between users' accounts andblockchain networks . It executes transactions on behalf of users and covers the associated execution costs, which may or may not be claimed.

See also:

- What's Relaying?(opens in a new tab)
- on docs.gelato.network

Safe{DAO}

The Safe{DAO} is theDecentralized Autonomous Organization(opens in a new tab) (DAO) that aims to foster a vibrant ecosystem of applications and wallets leveraging Safe accounts. This will be achieved through data-backed discussions, grants, and ecosystem investments, as well as providing developer tools and infrastructure.

See also:

- Safe{DAO} Forum(opens in a new tab)
- Safe{DAO} Governance process(opens in a new tab)
- on forum.safe.global
- Safe{DAO} Proposals(opens in a new tab)
- on snapshot.org

Safe{Wallet}

Safe{Wallet}(opens in a new tab) is the official user interface to manage Safe accounts.

See also:

- Getting Started with Safe{Wallet}(opens in a new tab)
- on help.safe.global

Safe Apps

Safe Apps are web applications that run in the Safe Apps marketplace. They support Safe, use the Safe Apps SDK to interact with it, and aren't owned, controlled, maintained, or audited by Safe.

See also:

- Safe Apps SDK(opens in a new tab)
- on GitHub

Safe Guard

A Safe Guard is a smart contract that adds restrictions on top of the n-out-of-m scheme that Safe accounts offer. They make checks before and after the execution of a Safe transaction.

See also:

- Safe Guards documentation
- on docs.safe.global
- Zodiac Guards(opens in a new tab)
- on zodiac.wiki
- Get the enabled Safe Guard
- andenable a Safe Guard
- with the Safe{Core} SDK on docs.safe.global

Safe Module

A Safe Module is a smart contract that adds functionality to Safe while separating module logic from Safe core contracts.

See also:

- Safe Modules documentation
- on docs.safe.global
- Safe Modules repository(opens in a new tab)
- on github.com
- Zodiac Modules(opens in a new tab)
- on zodiac.wiki
- Get the enabled Safe Modules
- andenable a Safe Module
- with the Safe{Core} SDK on docs.safe.global

Smart Account

A smart account (also known as a smart contract account) leverages the programmability of smart contracts to extend its functionality and improve its security in comparison withexternally-owned accounts . Smart accounts are controlled by one or multiple externally-owned accounts or other smart accounts, and all transactions have to be initiated by one of those.

Some common features that smart accounts offer to their users are:

- Multi-signature
- scheme
- Transaction batching
- Account recovery
- Gasless transactions

Safe is one of the most trusted implementations of a smart account.

Transaction

A transaction is an action initiated by anexternally-owned account to update the state of the EVM network. Transaction objects must be signed using the sender's private key, require a fee, and be included in a validated block.

A Safe transaction is a transaction sent to a Safe Proxy contract calling theexecTransaction(opens in a new tab) method.

See also:

- Transactions(opens in a new tab)
- on ethereum.org

Threshold

The threshold of a Safe account is a crucial configuration element that enables using Safe as a multi-signature smart account. It defines the number of required confirmations from the Safe owners a (Safe) transaction must have to be executable.

See also:

- Get the threshold
- andchange the threshold
- of a Safe with the Safe{Core} SDK on docs.safe.global

UserOperation

UserOperation objects are pseudo-transaction objects introduced by theERC-4337 that users send to theUserOperation mempool. They wrap the users' transactions, and are sent to theEntryPoint contract byBundlers .

See also:

- UserOperations(opens in a new tab)
- on ethereum.org
- UserOperation mempool(opens in a new tab)
- on erc4337.io

Wallet

A wallet is an interface or application that gives users control over their blockchain account. Wallets allow users to sign in to applications, read their account balance, send transactions, and verify their identity.

See also:

- Ethereum Wallets(opens in a new tab)
- on ethereum.org