Smart Contract

Smart contracts are pieces ofexecutable code that live in a NEAR account. They canstore data ,perform transactions in the account's name, andexpose methods so other accounts can interact with them.

Developers can choose between using Javascript or Rust to write smart contracts in NEAR. Indistinctly from the language chosen, the contract will be compiled into WebAssembly, from which point it can be deployed and executed on the NEAR platform.

Want to build a smart contract? Check ourQuickstart Guide to build your first smart contract in NEAR.

What can Smart Contract Do?

Smart contracts have complete control over the account, and thus can perform any action on its behalf. For example, contracts can:

- Transfer NEAR Tokens
- Call methods on themselves orother contracts
- Create new accounts and deploy contracts on them
- Update their own code

Besides, smart contracts can store data in the account's storage. This allows contract's to create almost any type of application, from simple games to complex financial systems.

What contracts cannot do * Smart contracts cannotaccess the internet * , so they cannot make HTTP requests or access external data * Smart contracts cannotexecute automatically * , they need to be called by an external account

What are Contract's Used for?

Smart contracts are useful to createdecentralized applications . Some traditional examples include:

- Decentralized Autonomous Organizations
- , where users create and vote proposals
- Marketplaces
- , where users create and commercialize digital art pieces
- Decentralized exchanges
- , where users can trade different currencies
- And many more...

For instance, you can easily create a crowdfunding contract that accepts NEAR. If the goal is met in time, the creator can claim the funds. Otherwise, the backers are refunded.

Contract's Storage

Smart contract store they data in the account's state. The contract's storage is organized askey-value pairs encoded using eitherJSON orBorsh serialization. This allows to store any type of data efficiently, from simple numbers to complex objects.

Since the data occupies space in the network, smart contracts need topay for the storage they use . For this, accounts automatically lock a portion of their balance each time new data is stored in the contract. This means that:

- If data is added to the contract's storage the account'sbalance decreases ↓
- .
- If data is deleted the account'sbalance increases ↑
- .

Currently, it cost approximately1 Ⓝ to store100kb of data.

tip Serializing and deserializing the storage happens automatically through our SDK tools, so you can focus on coding the logic of the contract Edit this page Last updatedonMar 25, 2024 bygagdiez Was this page helpful? Yes No

Previous Access Keys Next Transactions