Smart contracts basics

Contents

Why smart contracts ?

The concept of the smart contract was there well before the advent of the Bitcoin. Computer scientist **Nick Szabo** detailed his idea of cryptocurrency Bitgold as a sort of a precursor for Bitcoin. He also outlined the concept of smart contract in his 1996 publication.

Smart contract is a centerpiece and main thrust of Ethereum blockchain. A significant contribution of Ethereum is a working smart contract layer that supports any arbitrary code execution over the blockchain. Smart contract allows for user-defined operations of arbitrary complexity. This feature enhances the capability of Ethereum blockchain to be a powerful decentralized computing system.

Some situations introduce conditions, rules, policies beyond that of which a simple money transfer cryptocurrency protocols can handle. Smart contract addresses this need for application specific validation for blockchain applications. Smart contract has some advantages including :

- facilitates transaction for transfer of assets other than value or cryptocurrency.
- specification of rules for an operation on the blockchain
- facilitates implementation of policies for transfer of assets in a decentralized network.
- adds programmability and intelligence to the blockchain.
- represents a business logic layer, with the actual logic coded in a special high-level language.
- embeds function that can be invoked by messages that are like function calls. These messages and the input parameters for a single message are specified in a transaction.

Smart contracts defined

Since a smart contract is deployed in the blockchain, it is an immutable piece of code, and once deployed, it cannot be changed. We will have to redeploy the code as a new smart contract, or somehow redirect the calls from an old contract to the new one.

Smart contract can store variables in it called state variables. We could retrieve how these variables change over the blocks. Contract in the Ethereum blockchain has :

- pragma directive
- name of the contract
- data or the state variable that define the state of the contract
- collection of function to carry out the intent of a smart contract.

In order to test your own smart contract in Solidity : https://remix.ethereum.org/

Processing smart contracts

A smart contract can be created, on behalf of an externally owned account, by application programmatically from the command-line interface and by a script of commands from high level applications and user interface or UI. It can also be created from inside a smart contract.

We need an address for the smart contract to deploy it and invoke its functions. The address is computed by hashing the account number of externally owned account UI and the nonce.

Here are some of the artifacts generated by the Remix smart contract compile process and their use :

- ABI, Application Binary Interface, the interface schema for a transaction to invoke functions on the smart contract instance bytecode.
- Contract bytecode, this is the bytecode that is executed for instantiating a smart contract on the EVM. Think of it like executing a constructor of a smart contract to create an object.
- WebDeploy script, this as two items; json script to web application to invoke smart contract function, script for programmatically deploying a smart contract from a web application.
- Gas estimates, this provides a gas estimates for deploying the smart contract and for the function invocation.
- Function hashes, first four bytes of the function signatures to facilitate function invocation by a transaction.
- Instance bytecode, the bytecode of the smart contract instance.

Deploying smart contracts

A smart contract solution is written in high-level language and compiled bytecode. An ABI is also generated for high-level language application. Example, Web Apps to interact with the binary smart contract.

The smart contract requires an address for itself so that transaction can target it for invocation of its function.

Smart contracts can be deployed from Remix IDE, another smart contract, a command line interface, another high-level language application or Web application.

Tips

In order to become better in Solidity (beginner) : https://cryptozombies.io/fr/lesson/1