Smart Contract Fundamentals

A smart contract is a self-executing contract with the terms of the agreement between buyer and seller being directly written into lines of code.

Smart contracts are like a workflow process diagram or task flow. Once deployed, a smart contract can store states and execute computations. The code controls the execution, and transactions are trackable and irreversible.

Benefits of Smart Contracts

Smart contracts reduce coordination costs and enforcement of agreements.

The most common language for Smart Contracts is Solidity

OpenZeppelin comes with a wide array of smart contracts for various important functions. We can use and extend these contracts to create more secure dapps in less time.

On the developing team, it requires a good frontend designer (React) and some good smart contract engineer. Web3.js can be used to connect those two components together.

Typical Workflow

1. Setting up the development environment
2. 2. Creating a Truffle project / Hardhat project
3. 3. Writing the smart contract
4. 4. Compiling and migrating the smart contract
5. 5. Testing the smart contract
6. 6. Creating a user interface to interact with the smart contract
7. 7. Interacting with the dapp in a browser