CSE512: Introduction to Smart Contracts and Solidity

Anurag Jain

August 29, 2020

Overview: In this tutorial, we will introduce *Smart Contracts* and get started with building them on Ethereum.

1 Smart Contract

Definition 1.1 (Smart Contract)**.** A smart contract is an agreement between two or more parties, encoded in such a way that the correct execution is guaranteed by the blockchain.

1.1 What is a Smart Contract?

A contract is an agreement between two or more parties. Legal contracts are enforced by the judiciary and "executed" by lawyers, that check whether the certain conditions are met or not and then take action accordingly. In an electronic ledger, we can allow the computer to check these conditions and take the corresponding actions. The program that serves this purpose is known as a "Smart" Contract.

1.1.1 Motivating Example

Say you wish to borrow 1,000 rupees from someone willing to lend you and you have 1 Ether to keep as a collateral. (Remember, there are no centralized banks in a distributed ledger) However, you need to provide a guarantee that you will repay that amount with interest (say 10%). So, you and your friend can sign a smart contract that says `if (1100 rupees paid by next year){ return 1 Ether back to the borrower }else{ send the 1 Ether to the lender}`

This can be done automatically without needing a trusted intermediary like the bank or government (even anonymously). Since anyone can offer a loan you can imagine the interest rates would be much lower due to more competition among lenders (in fact, for a *flash loan* that is taken for a single transaction you only need to pay a flat fee of 10^{-18} Ether which costs roughly the same as 5×10^5 atoms of gold, less than the amount dissolved in 1 drop of seawater).

2 Ethereum

Ethereum is a public blockchain just like Bitcoin, but it can run Smart Contracts[1]. You can learn about it here: https://ethereum.org/en/learn/

3 Solidity

Ethereum Contracts are written in Ethereum Virtual Machine (EVM) bytecode. However, we don't directly write bytecode (usually) but use a higher level language like Solidity or Vyper and then compile the contracts before deploying them.

[1] Bitcoin allows only a few simple scripts

3.1 Getting Started

Before getting started to writing smart contracts, we would need to set up a local environment for compiling and locally deploying our smart contracts. An easy way to do this is using the Truffle Suite[2] (https://www.trufflesuite.com/). Perform the following steps before the tutorial:

- Install NodeJS (Highly recommended to use NVM (https://github.com/nvm-sh/nvm)
- Install truffle (`npm install -g truffle`)

Rest would be covered in the tutorial itself.

[2] It is highly recommended that you use Linux for this