TCS332 Fundamental of Information Security and Blockchain

Smart contract

B. Tech CSE III Semester

Instructor:

Dr Mohammad Wazid

Professor, Department of CSE

Graphic Era (Deemed to be University), Dehradun, India

Email: wazidkec2005@gmail.com

Homepage: https://sites.google.com/site/mwazidiiith/home

Smart contract

- **A contract can be considered as an agreement.**
- **A smart contract is a transaction protocol** that is intended to execute automatically and control legally relevant events as per the terms of a contract.
- The objectives of smart contracts are the **reduction of the need for trusted intermediates, enforcement costs, fraud losses, and the reduction of malicious acts.**

Smart contract

- Smart contracts are **self-executing** contracts with the terms of the agreement between buyer and seller being directly written into lines of code.
- **Nick Szabo**, an American computer scientist who invented a virtual currency called "**Bit Gold**" in **1998**, defined smart contracts as computerized transaction protocols that execute the terms of a contract.
- **Smart contracts render transactions traceable, transparent, and irreversible.**

Working of smart contract

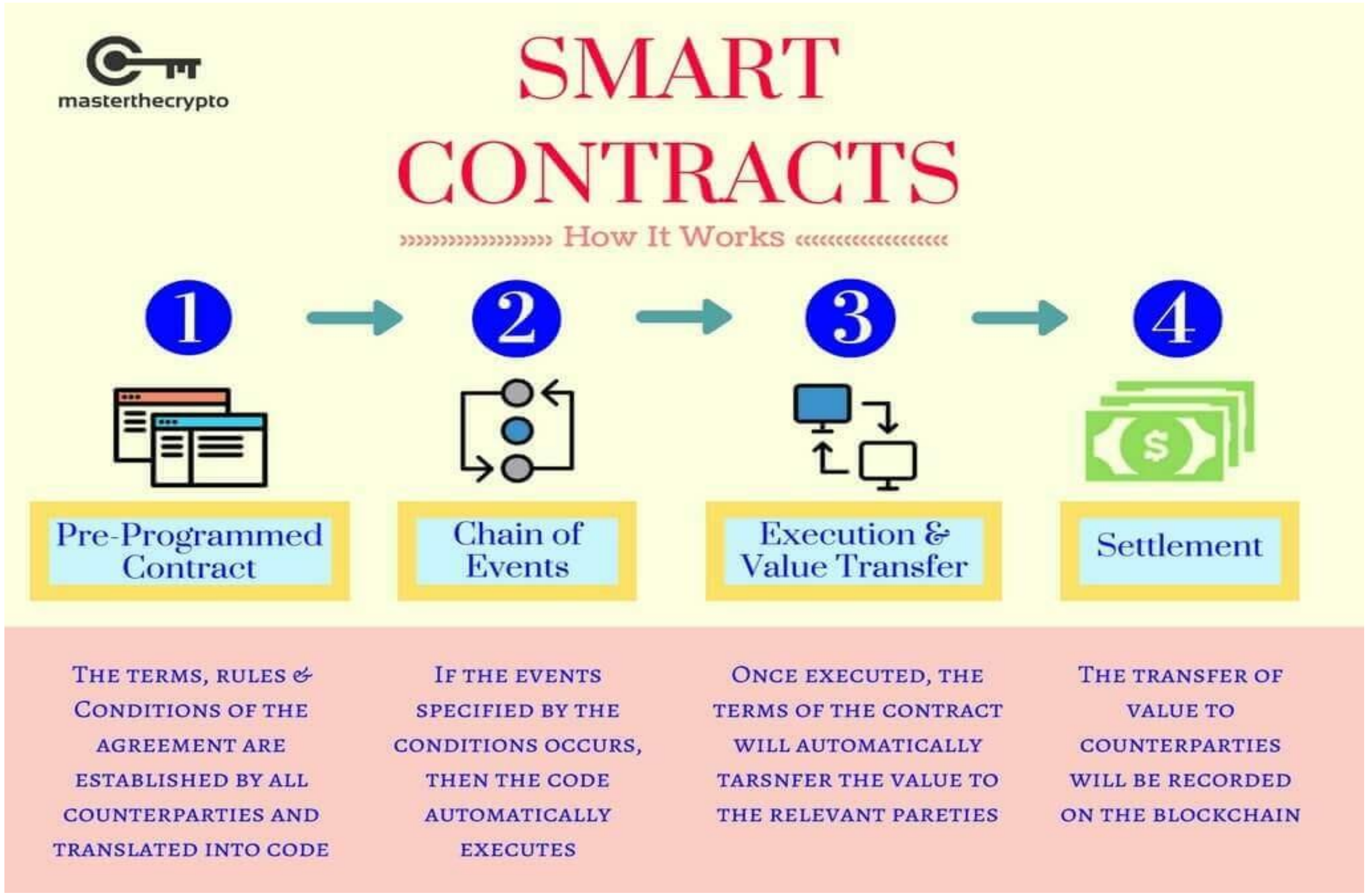
masterthecrypto
SMART CONTRACTS
How It Works
1
2
3
4
Pre-Programmed Contract
Chain of Events
Execution & Value Transfer
Settlement
THE TERMS, RULES & CONDITIONS OF THE AGREEMENT ARE ESTABLISHED BY ALL COUNTERPARTIES AND TRANSLATED INTO CODE
IF THE EVENTS SPECIFIED BY THE CONDITIONS OCCURS, THEN THE CODE AUTOMATICALLY EXECUTES
ONCE EXECUTED, THE TERMS OF THE CONTRACT WILL AUTOMATICALLY TARSNFER THE VALUE TO THE RELEVANT PARETIES
THE TRANSFER OF VALUE TO COUNTERPARTIES WILL BE RECORDED ON THE BLOCKCHAIN

Fig. Working of smart contract (image source: https://ipkitten.blogspot.com/2019/03/smart-contracts-pros-and-cons-of-new.html)

Traditional contract

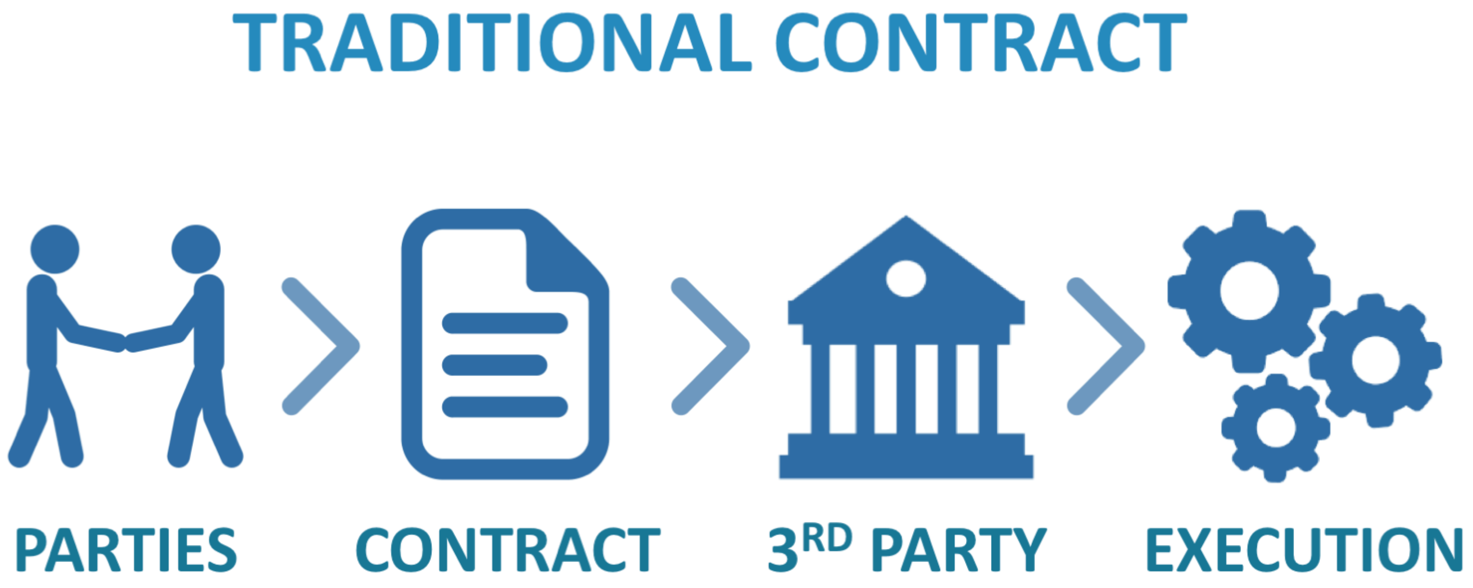

Fig. Activities under traditional contract (image source: https://dzone.com/articles/what-is-smart-contracts-blockchain-and-its-use-cas-1)

Programming

- Open online platform for solidity programming (https://remix.ethereum.org/).
- Solidity is used for the creation of smart contracts.
- Always save your solidity code as “filename.sol”

Programming

- **Program-1 (first.sol)**
- This program is for the creation of a smart contract. Note, we are not executing any task inside this contract. We have just created the smart contract with name “first”.

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >= 0.4.16 < 0.9.0;
/*this is our pragma statement, version of solidity which are supported here*/
contract first {}
/*name of the contract that you are going to create */
```

Programming

- **Program-2(helloworld.sol)**

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >= 0.4.16 < 0.9.0;
contract helloworld {
    function hello() pure public returns(string memory){
        return 'hello world';
    }
}
```

/*declare a function hello() inside the contract helloword. Make its return type. It should be declared with public visibility keyword to make it callable from the outside of the function. Returns keyword should be used for the return type. Here it is string type. Memory is the data location. In solidity variables can be saved in different data locations. "memory" means some temporary variable, we do not require any specific location for this to store. We do not want to modify this function therefore; We use pure keyword here. */

Programming

- **Program-3(SimpleStorage.sol)**

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >= 0.4.16 < 0.9.0;
contract SimpleStorage {
string public data;
 function set(string memory _data)
public{
   data=_data;
    }
}
```

Explanation of the program:

- Program to set some data variable and then supply some value to it. function set() is a setter function, which can be accessible to all smart contracts. Note data and _data both are different variable. We cannot pass data as a variable to function set() because this will shadow our data variable. Therefore, both should be different. Setter function accepts an argument.
- When we press the 'data' button we just call a data function on the blockchain. Orange colour button means we are setting some value to some variable. Blue colour button means we are calling a function on the blockchain. That's not a transaction.
- setter is a method that updates value of a variable. And a getter is a method that reads value of a variable.

Programming

- Transaction is not all about sending or transferring money. It means, we have sent a data package that certainly modify the state of the blockchain. In remix console we have “transaction receipt”.
- Use “view” to view the records of smart contract
- Environment: javascript VM means Local Ethereum blockchain. It is only associated with your computer. Not connected with the external world.

References

- Fries, Martin; P. Paal, Boris (2019). Smart Contracts (in German). Mohr Siebeck. ISBN 978-3-16-156911-1. JSTOR j.ctvn96h9r.
- Savelyev, Alexander (14 December 2016). "Contract Law 2.0: "Smart" Contracts As the Beginning of the End of Classic Contract Law". Social Science Research Network. SSRN 2885241.