The Technique Detail of Ethereum and The Comparison Between Bitcoin, Ethereum and Tether

ABSTRACT

Ethereum is a kind of application platform based on block-chain and Ethereum virtual machine. This article will focus on the comparison between Bitcoin and Ethereum, Ethereum and Tether. It will mainly discuss about the technique side of Ethereum and Tether.

KEYWORDS

Bitcoin, Ethereum, Tether

1. The Technique Detail of Ethereum

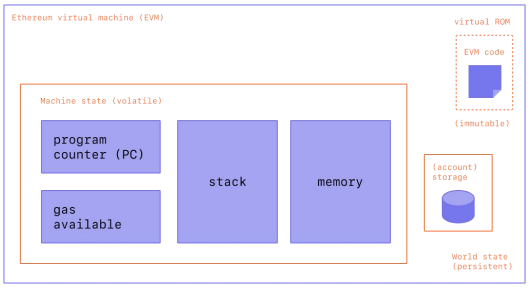
Ethereum is a kind of block-chain technique which has two function. The first one is allowing the user send cryptocurrency to anyone for a small fee, just like Bitcoin. The second one is powering decentralized application using Ethereum virtual machine and smart contract.

First we will talk about the Ethereum virtual machine.

Normal virtual machine creating a level of abstraction between the executing code and the executing machine. For example, we could using Oracle VM Virtual box to run windows operation system on Apple's Mac. The virtual machine technique improve the portability of software, just like we could run windows application on mac OS.

Instead of distributed ledger, Ethereum is a distributed state machine. Ethereum state is a large data structure which holds not only all accounts and balances, but a machine state.

Based on the yellow paper of Ethereum, the Ethereum Virtual Machine is a quasi-Turing complete machine. The quasi qualification comes from the face that the computation is intrinsically bounded through a parameter, 'gas'. Also the yellow paper talks about the EVM stack based architecture and execution overview.



Based on the yellow paper, there are some implementations of EVM, for example, "EthereumJS" using JavaScript, the "Geth" using Go, "Aleth" using C++, etc. They are all lived on the Ethereum main network. There are also other implementations that used for testing like SputnikVM using Rust, Mantis using Scala, etc. These kind of EVM implementation makes different kinds of smart contracts which are written on different language running on the Ethereum network possible. The detail of these EVM will not be further elaborated.

Next, we will take about smart contract.

About the smart contract, they provide open API services for Decentralized application and could run 24/7 and can't be taken down. Based on HTTP protocol, normal API usually contains the server side and database, they could response to the client side request and send response. API define the rules that what kinds of data the client side should send. Also, the client side needs to know what format of data it will receive and makes preparation. The database will changed though different request and is the core of the API function.

The smart contract of Ethereum is more or less like the normal API. They are deployed to the Ethereum network. User accounts can interact with them by submitting transactions that execute a function defined on the smart contract. They could define rules via the code. The "rules" means that what kind of format of date they could receive and what kind of data they will send. Also, anyone can write a smart contract and deploy it to the network, just like normal server-side API. But different from the normal API, we have to pay money to the AWS services or Microsoft Azure, we need to pay Gas to deploy the contract to the Ethereum network. Moreover, you could also call other smart contracts in your own smart contract like normal API.

In order to write smart contracts, developer need to use specific "smart contract languages", like "Solidity" or "Vyper". They also need to be compiled before they can be deployed on the Ethereum virtual machine. For example, if you want to run your smart contract on EthereumJS virtual machine, you need to compile your code into Javascript so the virtual machine could execute your code.

Here is one example of smart contract written in Solidity from the official document of Solidity mainly:

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;
contract Coin {
    // The keyword "public" makes variables
    // accessible from other contracts
    address public minter;
    mapping (address => uint) public balances;

    // Events allow clients to react to specific
    // contract changes you declare
    event Sent(address from, address to, uint amount);

    event getBalance(address someone);
```

```
    // Constructor code is only run when the contract
    // is created
    constructor() {
        minter = msg.sender;
    }

    // Sends an amount of newly created coins to an address
    // Can only be called by the contract creator
    function mint(address receiver, uint amount) public {
        require(msg.sender == minter);
        require(amount < 1e60);
        balances[receiver] += amount;
    }

    // Sends an amount of existing coins
    // from any caller to an address
    function send(address receiver, uint amount) public {
        require(amount   <=   balances[msg.sender],   "Insufficient
balance.");
        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        emit Sent(msg.sender, receiver, amount);
    }}

    // return the balances of specific address
    function getBalance(address addr) public view returns(uint) {
        return balances[addr];
    }
```

The language is very similar to typescript. The example is a simple form of cryptocurrency with three function.

Firstly, it covers the creation of the coins using mint function. Only the creator could create the new coins. Secondly, anyone could send coins to each other using Ethereum address if your amount of your existing coins is not lower than the amount of the coins that you want to send. Thirdly, the line "`event Sent(address from, address to, uint amount)`" shows the function of "event". It means that the kind of smart contract could listen for the events emitted on the block-chain and make the change base on its rules, just like normal API. Finally, the function of "`getBalance`" could get the coin of the specific address of the coin owners.

Finally, we will talk about decentralized application.

Decentralized application is a kind of application based on smart contracts and front-end user interface, just like normal web application. The only difference is that the back-end service is based on the smart contracts and running on the Ethereum network using Ethernet virtual machine. The details of the smart contracts have been elaborated previously.

In order to make the development of decentralized application easier, the idea of software frameworks will be included. Software frameworks help developers build their application faster because frameworks provide some unique function that doing a lot of work for developers. And developers do not need to care about the detail on the framework implementation. For example, the Vuejs framework. It helps the web application developer manipulate DOM in a new way and improves the efficiency.

In the area of Ethereum decentralized application frameworks, they focus mainly on the connection of block-chain instance, the testing of contracts and the configuration and deployment. There are a lot of available frameworks, such as Truffle, Embark, Ether-lime, etc.

2. The Comparison Between Bitcoin and Ethereum

Bitcoin is the most famous application of block-chain focusing on the cryptocurrency. The inner technique of Bitcoin is hash function and digital signatures.

The hash function and the idea of block-chain provide the authority of the Bitcoin cryptocurrency. The miners need to put a lot of calculation energy to meet the specific value of hash and broadcast to the whole net. And after six confirmation, the block will be confirmed and the miner will get the block reward.

The digital signature provide the function of transition cryptocurrency. The signature contains the information of who create the coin and the coin transfer routes. Because of the block-chain, anyone could track any coin transfer routing and get to know weather the transition is legal or not.

Ethereum, comparing with Bitcoin, is more complicated. The function of cryptocurrency is only a part of the decentralized application of Ethereum. Just like part I says, anyone could build their own cryptocurrency based on the smart contract using Solidity or Vyper. Users could interact with the smart contract (API) by submitting transactions and execute its function. The storage state of the smart contract will change like normal back-end service.

The proof of work currencies are different. The Bitcoin uses the SHA-256 hash function. The Ethereum uses the Ethash, belonging to the SHA-3 hash function family.

3. The Comparison Between Ethereum and Tether

There are a lot of cryptocurrency based on Ethereum smart contract, like Tether, Chainlink, USD Coin, Dai, etc. They are called "Token Contracts", belong to the family of "ERC-20 tokens", "ERC-721 tokens", "ERC-1504" etc.

The "ERC" means that they follow a list of standards so that they could be shared or exchanged with each other. The standards are created by Ethereum community. And the ERC-20 and ERC-721 just a name of the token standards. The standards acts like protocols and makes different API could communicate with each other. The tokens likes the per defined object, contains specific information, data and function, such as the total supply and balance of the owner. The detail of these standards could be found on Ethereum document website: https://eips.ethereum.org/all

In the family of tokens contracts based on Ethereum, based on coinmarketcap.com, Tether has the most volume:

Rank	Name	Platform	Market Cap	Price	Volume (24h)
1	Tether	Ethereum	$20,035,449,911	$0.999831	$99,387,898,987
2	Chainlink	Ethereum	$5,488,518,841	$13.84	$1,892,091,491
3	USD Coin	Ethereum	$3,275,725,877	$0.999950	$1,302,234,580
10	Uniswap	Ethereum	$967,992,343	$3.81	$1,185,891,683
7	Dai	Ethereum	$1,127,876,362	$1.00	$650,139,594

Tether(also called USDT) is designed as a kind of stable coin, originally designed to be worth 1.00 dollar. In the cryptocurrency market, it's usually used for exchange with other cryptocurrency because of the function of stable price, quick transition time and low transition fee. It's an ideal bridge between fiat currencies and cryptocurrency.

REFERENCES

1. Ethereum Virtual Machine:
   https://medium.com/mycrypto/the-ethereum-virtual-machine-how-does-it-work-9abac2b7c9e
2. Ethereum Yellow paper:
   https://ethereum.github.io/yellowpaper/paper.pdf
3. EthereumJS Documentation:
   https://ethereumjs.readthedocs.io/en/latest/
4. Token Contracts Lists:
   https://coinmarketcap.com/tokens/views/all/
5. ERC-20 Documents:
   https://eips.ethereum.org/EIPS/eip-20