Platform Alternatives

**Introduction:**

I want to build my bike-sharing system based on the blockchain system. The function I want to implement is quite straightforward. The bike renters only need to know two numbers - the deposit and the rental cost. The user of the bike scans It and sees what he has to pay in terms of deposit. When the deposit is received by the bike, control is granted. I will compare between Ethereum and IOTA these two implementation platforms.

**Keywords: Bike-sharing system, smart contract, deposit, IOTA, Ethereum**

**IOTA whitepaper**

Tangle (a directed acyclic graph) for storing transactions.

DAG (directed acyclic graph)

MCMC algorithms

It fixes the bitcoin problem which is bitcoin cannot make micro-payment

The procedures IOTA used to build a new node.

1. Approve two transactions (directed edge), “genesis” node will make transaction of tokens to “founder” address

(There is no mining in IOTA system and it will not imposes any rule for choosing the transaction to approve)

1. Make sure two transaction not conflicting
2. Solve a cryptographic puzzle

The specialty of IOTA network is it is an asynchronous network, it does not have to achieve consensus. It will decide which will become orphaned using tip selection algorithms.

The weight of a transaction is proportional to the amount of work that the issuing node invested into it. The term height means longest oriented path, depth means reverse-oriented path, score is the cumulative weight.

**Keywords: Tangle, DAG, MCMC, procedures**

**ETH Whitepaper**

Ethereum is a blockchain that allows us to run programs in its trusted environment.  Ethereum has a virtual machine, called the Ethereum Virtual Machine (EVM). The EVM allows code to be verified and executed on the blockchain, providing guarantees it will be run the same way on everyone's machine. This code is contained in "smart contracts" . Beyond just tracking account balances, Ethereum maintains the state of the EVM on the blockchain. All nodes process smart contracts to verify the integrity of the contracts and their outputs.

A smart contract is code that runs on the EVM. Smart contracts can accept and store ether, data, or a combination of both. Then, using the logic programmed into the contract, it can distribute that ether to other accounts or even other smart contracts.

As we know, there is heavy support behind Ethereum’s technology in what is called The Enterprise Ethereum Alliance. This is a supergroup of Fortune 500 companies that have all agreed to work together to learn and build upon Ethereum’s blockchain technology like smart contract. In my opoinion, I am very excited about Ethereum’s technology is its potentioal to impact IoT projects and processes.

Business effectiveness

**Keywords: Ethereum, EVM, smart contract, TEEA**

**Advantages and Disadvantages**

Ethereum advantage:

Most applications are powered by ethereum. It built to support and encourage more applications to be developed and are much easier for people to use. Bitcoin is like DOS, It was the first to be launched and played a big part in the success of the computer. But DOS poses some difficulty for beginners, it is very tricky to program, and only a few applications are available to run on it. Ethereum is like Windows and Mac Os, with developers allowed to create thousands of applications on their platform. Next, I listed several advantages and disadvantages of Ethereum.

1. Ethereum uses turing language on the blockchain which allows the exchange of simple or complicated contracts on the network. This makes Ethereum into a killer app on the blockchain.
2. Ethereum has the backing from corporate clients, only bitcoin can match this level of corporate buy in. No other cryptocoins have this much corporate clout.
3. Ethereum is a platform with a clear vision of where it wants to go and they are implementing it with a strict rigor of which the community is proud of.
4. Its ecosystem is robust which make it easier for developers to enter and contribute.

Ethereum disadvantage:

1. The speed of transactions for its cryptocurrency ETH. I remembered around last December, a digital pet application which based on the Ethereum blockchain named CryptoKitties had clogged the Ethereum transportation for many days, and I myself have had some laggy delays in transactions.
2. The platform still need a important upgrades from moving from proof of work to proof of stake which is more efficient.

**Keyword: applications, turing language, cprporate clients, clear vision, ecosystem. Speed of transactions, PoW, PoS**

Which is better and why for the targeted use case

We know IOTA is using DAGs, DAGs are fundamentally unable to generalized smart contracts. Even though you can do some types of smart contracts, you only can do anything in which transaction order does not matter. But smart contracts in which order doesn’t matter is a pretty limited subset of all smart contracts. In my program, I prefer using Ethereum plartform which is more scalable and efficient. The most important point is IOTA can not support the function I want, especially about the deposit payment and the rent fee which indeed need a smart contracts which need the order function.

**Keywords: DAGs, smart contracts, order**

**Problem and challenges**

Ethereum

The problem in ethereum is the security of smart contract. As we know, smart contracts may run exactly as programmed but this does not mean that they will run as the creators intended. We crowdsourced a list of the major bugs with smart contracts on Ethereum so far. We can categorize the list by categories of bugs:

* Variable/function naming mixups: FirePonzi, Rubixi
* Pblic data that should not have been public : the public PNG seed casino, chearable RPS
* Re-entrancy(A calling B calling A): the DAO, Maker’s ETH-backed token
* Sends failing due to 2300 gas limit: King of the Ether
* Array/loops and gas limits Governmental
* Much more subtle game-theoretic weaknesses where at the limit people even debate whether or not they’re bugs: the DAO

Actually, people already started to develop some tools to fix the problem in smart contract, ranging from better development environments to better programming languages to formal verification and symbolic execution (Solidity, makes it really easy to make this sort of mistake

). In Vitalik Buterin’s opinion, progress in smart contract safety is necessarily going to be layered, incremental, and necessarily dependent on defense-in-depth.

**Keywords: Security of smart contract, better development environments, better programming languages, Solidity**

IOTA

IOTA have some problems about its promise of being a fee-less system. From my point of view, the PoW doesn’t get “harder” as there isn’t a specific difficulty you need to hit. Rather, you can do more proof of work in order to give your transaction a higher own weight. We know IOTA use MCMC algorithm for tip selection, it will walk towards transactions with a higher score. So essentially, the “difficulty” of mining kind of exists as your new transactions have to have an own weight to roughly match the own weights of the rest of the transactions being published on the network. So in a system where the total hashing power is dominated by mining pool, all the mining pools can just use higher average own weights in order to effectively lock out any transactions with lower own weights from being verified. This will force everyone else on the network to process their transactions through them and they can charge them fees.

The entire idea of a fee-less system assumes that everyone does the proof of work for their own transactions aka everyone provides computational power to the network proportional to their usage of the network. However, If we take that assumption, blockchains essentially havr zero transaction fees too.

**Keywords: fee-less system, PoW, MCMC, weight, blockchain, transaction fee**