ABOUT |

ADVERTISE | STORE | CONTACT | BTC CHANNELS

BITCOIN PRICE
$705.20

BITCOIN MAGAZINE

SUBSCRIBE

MEMPOOL: #core-dev #blockchain #technical #wall-street

Ethereum: A Next-Generation Cryptocurrency and Decentralized Application Platform

Jan 24, 2014 1:50 AM by Vitalik Buterin

The author of this article, Vitalik Buterin, is also the founder of Ethereum, and this article is intended as an expository piece and not a review.

Over the past year, there has been an increasingly large amount of discussion around so-called “Bitcoin 2.0″ protocols – alternative cryptographic networks that are inspired by Bitcoin, but which intend to make the underlying technology usable for far more than just currency. The earliest implementation of this idea was Namecoin, a Bitcoin-like currency created in 2010 which would be used for decentralized domain name registration. More recently, we have seen the emergence of colored coins, allowing users to create their own currencies on the Bitcoin network, and more advanced protocols like Mastercoin, Bitshares and Counterparty which intend to provide features such as financial derivatives, savings wallets and decentralized exchange. However, up until these point all of the protocols that have been invented have been specialized, attempting to offer specific and rich feature sets targeted toward specific industries or applications usually financial in nature. Now, a group of developers including myself have come up with a project that takes the opposite track: a cryptocurrency network that intends to be as generalized as possible, allowing anyone to create specialized applications on top for almost any purpose imaginable. The project: Ethereum.

Cryptocurrency Protocols Are Like Onions…

One common design philosophy among many cryptocurrency 2.0 protocols is the idea that, just like the internet, cryptocurrency design would work best if protocols split off into different layers. Under this strain of thought, Bitcoin is to be thought of as a sort of TCP/IP of the cryptocurrency ecosystem, and other next-generation protocols can be built on top of Bitcoin much like we have SMTP for email, HTTP for webpages and XMPP for chat all on top of TCP as a common underlying data layer.

So far, the three main protocols that have followed this model are colored coins, Mastercoin and Counterparty. The way the colored coins protocol works is simple. First, in order to create colored coins, a user tags specific bitcoins as having a special meaning; for example, if Bob is a gold issuer, he may wish to tag some set of bitcoins and say that each satoshi represents 0.1 grams of gold redeemable from him. The protocol then tracks those bitcoins through the blockchain, and in that way it is possible to calculate who owns them at any time.

Mastercoin and Counterparty are somewhat more abstract; they use the Bitcoin blockchain to store data, so a Mastercoin or Counterparty transaction is a Bitcoin transaction, but the protocols interpret the transactions in a completely different way. One can have two Mastercoin transactions, one sending 1 MSC and the other 100000 MSC, but from the point of view of a Bitcoin user that does not know how that Mastercoin protocol works they both look like small transactions sending 0.0006 BTC each; the Mastercoin-specific metadata is encoded in the transaction outputs. A Mastercoin client then needs to search the Bitcoin blockchain for Mastercoin transactions in order to determine the current Mastercoin balance sheet.

I personally have had the privilege of talking directly to many of the originators of the colored coins and Mastercoin protocol, and have participated considerably in the development of both projects. However, over about two months of research and particpation, what I eventually came to realize is that, while the underlying idea of having such high-level protocols on top of low-level protocols is laudable, there are fundamental flaws in the implementations, as they stand today, that may well prevent the projects from ever gaining anything more than a small amount of traction.

The reason is not that the ideas behind the protocols themselves are bad; the ideas are excellent, and the response of the community alone is proof that they are trying to do something that is very much needed. Rather, the reason is that the low-level protocol that they are trying to build their high-level protocols on top of, Bitcoin, is simply not cut out for the task. This is not to say that Bitcoin is bad, or is not a revolutionary invention; as a protocol for storing and transferring value, Bitcoin is excellent. However, as far as being an effective low-level protocol is concerned, Bitcoin is less effective; rather than being like a TCP on top of which one can build HTTP, Bitcoin is like SMTP: a protocol that is good at its intended task (in SMTP's case email, in Bitcoin's case money), but not particularly good as a foundation for anything else.

The specific failure of Bitcoin is particularly concentrated in one place: scalability. Bitcoin itself is as scalable as a cryptocurrency can be; even if the blockchain balloons to over a terabyte, there is a protocol called "simplified payment verification", described in the Bitcoin whitepaper that allows "light clients" with only a few megabytes of bandwidth and storage to securely determine whether or not they have received transactions. With colored coins and Mastercoin, however, this possibility disappears. The reason is this. In order to determine what color a colored coin is, you need to not just use Bitcoin simplified payment verification to prove that it exists; you also need to trace it all the way back to its genesis, and do an SPV check each step of the way. Sometimes, the backward scan is exponential; and with metacoin protocols there is no way to know anything at all without verifying every single transaction.

And this is what Ethereum intends to fix. Ethereum does not intend to be a Swiss Army knife protocol with hundreds of features to suit every need; instead, Ethereum aims to be a superior foundational protocol, and allow other decentralized applications to build on top of it instead of Bitcoin, giving them more tools to work with and allowing them to gain the full benefits of Ethereum's scalability and efficiency.

Contacts, Not Just For Difference

At the time that Ethereum was being

developed, there was a large amount of interest in allowing financial contracts on top of cryptocurrencies; the basic type of contract being a "contract for difference". In a contract for difference, two parties agree to put in some amount of money, and then get money out in a proportion that depends on the value of some underlying asset. For example, a CFD might have Alice put in $1000, Bob put in $1000, and then after 30 days the blockchain would automatically return to Alice $1000 plus $100 for every dollar that the LTC/USD price went up during that time period and send Bob the rest. These contracts allow people to speculate on assets at high leverage, or alternatively protect themselves from cryptocurrency volatility by canceling out their exposure, without any centralized exchange.

At this point, however, it is clear that contracts for difference are really only one special case of a much more general concept: contracts for formula. Instead of having the contract take in $x for Alice, $y from Bob, and return to Alice $x plus an additional $z for every dollar that some given ticker went up, a contract should be able to return to A an amount of funds based on any mathematical formula, allowing contracts of arbitrary complexity. If the formula allows random data as inputs, these generalized CFDs can even be used to implement a sort of peer-to-peer gambling.

Ethereum takes this idea and pushes it one step further. Instead of contracts being agreements between two parties that start and end, contracts in Ethereum are like a sort of autonomous agent simulated by the blockchain. Each Ethereum contract has its own internal scripting code, and the scripting code is activated every time a transaction is sent to it. The scripting language has access to the transaction's value, sender and optional data fields, as well some block data and its own internal memory, as inputs, and can send transactions. To make a CFD, Alice would create a contract and seed it with $1000 worth of cryptocurrency, and then wait for Bob to accept the contract by sending a transction containing $1000 as well. The contract would then be programmed to start a timer, and after 30 days Alice or Bob would be able to send a small transaction to the contract to activate it again and release the funds.

Aside from this narrow contract-for-difference model, however, the whitepaper outlines many other transaction types that will become possible with Ethereum scripting, of which a few include:

Multisignature escrows, of a

Code example of an Ethereum

currency contract, written in a high-level language.

```
if tx.value < 100 * block.basefee:
```

similar spirit to the Bitcoin arbitration service Bitrated, but with more complex rules than Bitcoin. For example, there will be no need for the signers to pass around partially signed transactions manually; people can authorize a withdrawal asynchronously over the blockchain one at a time and then have the transaction finalized automatically once enough people make their authorizations.

Savings accounts - one interesting setup works as follows. Suppose that Alice wants to store a large amount of money, but does not want to risk losing everything if her private key is lose or stolen. She makes a contract with Bob, a semi-trustworthy bank, with the following rules: Alice is allowed to withdraw up to 1% per day, Alice with Bob approval can withdrawn any amount, and Bob alone can withdraw up to 0.05% per day. Normally, Alice will only need small amounts at a time, and if Alice wants more she can prove her identity to Bob and make the withdrawal. If Alice's private key gets stolen, she can run to Bob and move the funds into another contract before the thief gets away with more than 1% of the funds. If Alice loses her private key, Bob will eventually be able to recover her funds. And if Bob turns out to be evil, Alice can withdraw her own funds twenty times faster than he can. In short, all of the security of traditional banking, but with almost none of the trust.

Peer-to-peer gambling - any kind of peer-to-peer gambling protocol can be implemented on top of Ethereum. A very basic protocol would simply be a contract for difference on random data such as a block hash.

Creating your own currency - using Ethereum's internal memory store, you can create an entire new currency inside of Ethereum. These new currencies can be constructed to interact with each other, have a decentralized exchange, or any other kind of advanced features.

This is the advantage of Ethereum code: because the scripting language is designed to have no restrictions except for a fee system, essentially any kind of rules can be encoded inside of it. One can even have an entire company manage its savings on the blockchain, with a contract saying that, for example, 60% of the current shareholders of a company are needed to agree to move any funds (and perhapps 30% can move a maximum of 1% per day). Other, less traditionally capitalistic, structures are also possible; one idea is for a democratic organization with the only rule being that two thirds of the existing members of a group must agree to invite another member.

Beyond the Financial

The financial applications, however, only scratch the surface of what Ethereum, and cryptographic protocols on top of Ethereum, can do. While Ethereum's financial applications may be what initially excites many people in the cryptocurrency community, the long-term promise is arguably in the ways that Ethereum can work together with other, non-financial, peer-to-peer protocols. One of the main problems that non-financial peer-to-peer protocols have faced so far is the lack of incentive - that is to say, unlike centralized for-profit platforms, there is no financial reason to participate. In some cases, participation is in some sense its own reward; it is for this reason that people continue to write open source software, contribute to Wikipedia, and make comments on forums and write blog posts. In the context of peer-to-peer protocols, however, participation is often not a "fun" activity in any meaningful sense; rather, it consists of putting in a large quantity of resources, letting a daemon run in the background potentially hogging CPU and battery power, and forgetting about it.

For example, there have already for a long time been data protocols such as Freenet that essentially provide everyone with decentralized uncensorable static content hosting; in practice, however, Freenet is very slow, and few people contribute resources. File sharing protocols all suffer from the same problem: although altruism is good enough for spreading popular commercial blockbusters around, it becomes markedly less effective for those with less mainstream preferences. Thus, perversely, the peer-to-peer nature of file sharing may actually be helping the centralization of entertainment and media production, not hindering it. All of these problems, however, can potentially be solved if we add incentivization - empowering people to build not just nonprofit side projects, but also businesses and livelihoods, around participating in the network.

Incentivized data storage - essentially, a decentralized Dropbox. The idea works as follows: if a user wants to have a 1GB file backed up by the network, they would construct a data structure known as a Merkle tree out of the data. They would then put the root of the tree, along with 10 ether, into a contract and upload the file onto another specialized network that nodes wishing to rent out their hard drive space would listen for messages on. Every day, the contract would automatically pick a random branch of the tree (eg. "left -> right -> left -> left -> left -> right -> left"), ending at a block of the file, and give 0.01 ether to the first node to provide that branch. Nodes would store the entire file to maximize their chance of getting the reward.

BitMessage and Tor - Bitmessage is a next-generation email protocol that is both fully decentralized and encrypted, allowing anyone to send messages to any other Bitmessage user securely without relying on any third parties except for the network. However, Bitmessage has one large usability flaw: instead of sending messages to human-friendly email addresses, like "[email protected]", you need to send to garbled 34-character Bitmessage addresses (eg. "BM-BcbRqcFFSQUUmXFKsPJgVQPSiFA3Xash"). Ethereum contracts offer a solution: people can register their names on a special Ethereum contract, and Bitmessage clients can query the Ethereum blockchain to get the 34-character Bitmessage address associated with any name behind the scenes. The online anonymizing network Tor suffers from the same problems, and thus can also benefit from this solution.

Identity and Reputation Systems - once you can register your name on the blockchain, the logical next step is obvious: have a web of trust on the blockchain. Webs of trust are a key part of an effective peer to peer communication infrastructure: you don't just want to know that a given public key refers to a given person; you also want to know that the person is trustworthy in the first place. The solution is to use social networks: if you trust A, A trusts B, and B trusts C, then there is a pretty good chance that you can trust C, at least to some extent. Ethereum can serve as the data layer for a fully decentralized reputation system - and potentially ultimately a fully decentralized marketplace.

Many of the above applications consist of actual peer-to-peer protocols and projects that are already well under development; in those cases, we intend to establish partnerships with as many of these projects as we can, and help fund them in exchange for bringing their value into the Ethereum ecosystem. We want to help not just the cryptocurrency community, but also the peer to peer community as a whole, including file sharing, torrents, data storage and mesh networking. We believe that there are many projects, especially in the non-financial area, that can potentially bring great value to the community, but for which development is underfunded precisely because they lack an opportunity to effectively introduce a financial component; perhaps Ethereum may be what ultimately pushes dozens of these projects to the next stage.

Why are all of these applications possible on top of Ethereum? The answer lies in the currency's internal programming language. An analogy here may be made with the internet. Back in 1996, the web was nothing but HTML, and all people could do with it was serve static web pages on sites like Geocities. Then, people decided that there was a great need for people to submit forms in HTML, so HTML added a forms feature. This was like a "colored coins" of web protocols: try to solve a specific problem, but do it on top of a weak protocol without looking at the larger picture. Soon, however, we came up with Javascript, a programming language inside the web browser. And it was Javascript that solved the problem: because Javascript is a universal, Turing-complete programming language, it can be used to build apps of arbitrary complexity; Gmail, Facebook and even Bitcoin wallets have all been made with the language. And this was not because the Javascript developers decided that they wanted people to build Gmail, Facebook, and Bitcoin wallets; they just wanted a programming language. What we can do with the language is up to our own imaginations. And that is the spirit that we want to bring to Ethereum. Ethereum does not intend to be the end of all cryptocurrency innovation; it intends to be the beginning.

Further Innovations

Along with its main feature of a Turing-complete, universal scripting language, Ethereum will also have a number of other improvements over existing cryptocurrency:

Fees - Ethereum contracts will regulate its Turing-complete functionality and prevent abusive transactions such as memory hogs and infinite loop scripts by instituting a transaction fee for each computational step of script execution. More expensive operations, such as storage accesses and cryptographic operations, will have higher fees, and there will also be a fee for every item of storage that a contract fills up. To encourage contracts to clean up after themselves, if a contract reduces the amount of storage that it uses a negative fee will be charged; in fact, there is a special `SUICIDE` opcode to clear a contract and send all funds and the hefty negative fee back to its creator.

Mining algorithms - there has been a lot of interest into making cryptocurrencies whose mining is resistant against specialized hardware, allowing ordinary users with commodity hardware to participate without any capital investment and helping to avoid centralization. So far, the main antidote has been Scrypt, a mining algorithm that requires a large amount of both computational power and memory to compute; however, Scrypt is not memory-hard enough, and there are companies building specialized devices for it. We have come up with Dagger, a prototype proof of work that is even more memory-hard than Scrypt, as well as prototype proof-of-stake algorithms such as Slasher that get around the issue of mining entirely. Ultimately, however, we intend to host a contest, similar to the contests that determined the standards for AES and SHA3, where we invite research groups from universities around the world to devise the best possible commodity-hardware-friendly possible mining algorithm.

GHOST - GHOST is a new block propagation protocol pioneered by Aviv Zohar and Yonatan Sompolinsky that allows blockchains to have much faster block confirmation times, ideally in the range of 3-30 seconds, without running into the issues of centralization and high stale rate that fast block confirmations normally bring. Ethereum is the first major currency to integrate a simplified single-level version of GHOST as part of its protocol.

The Plan

Ethereum is potentially a massive and wide-reaching undertaking, and will take months to develop. With that in mind, the currency will be released in multiple stages. The first stage, the release of the whitepaper, has already happened. Forums, a wiki and a blog have been set up, and anyone is free to visit them and set up an account and comment on the forums. On January 25, a 60-day fundraiser will launch at the confrence in Miami, during which anyone will be able to purchase ether, Ethereum's internal currency, for BTC much like the Mastercoin fundraiser; the price will be 1000 ether for 1 BTC, although early investors will get roughly a 2x benefit to compensate for the increased risk that they're taking for participating in the project earlier. The fundraiser participants will not just get ether; there will also be a number of additional rewards, likely including free tickets to conferences, a spot to put 32 bytes into the genesis block, and for the top donors even the ability to name three sub-units of the currency (eg. the equivalent of the "microbitcoin" in BTC).

The issuance of Ethereum will not be any single mechanism; instead, a compromise approach combining the benefits of multiple approaches will be used. The issuance model will work as follows:

Ether will be released in a fundraiser at the price of 1000-2000 ether per BTC, with earlier funders getting a better price to compensate for the increased uncertainty of participating at an earlier stage. The minimum funding amount will be 0.01 BTC. Suppose that X ether gets released in this way

0.225X ether will be allocated to the fiduciary members and early contributors who substantially participated in the project before the start of the fundraiser. This share will be stored in a time-lock contract; about 40% of it will be spendable after one year, 70% after two years and 100% after 3 years.

0.05X ether will be allocated to a fund to use to pay expenses and rewards in ether between the start of the fundraiser and the launch of the currency

0.225X ether will be allocated as a long-term reserve pool to pay expenses, salaries and rewards in ether after the launch of the currency

0.4X ether will be mined per year forever after that point

There is an important distinction compared to

Bitcoin and most other cryptocurrencies: here, the eventual supply is unlimited. The "permanent linear inflation" model is designed to make ether neither inflationary nor deflationary; the lack of a supply cap is intended to dampen some of the speculative and wealth inequality effects of existing currencies, but at the same time the linear, rather than traditionally exponential, inflation model will mean that the effective inflation rate tends to zero over time. Additionally, because the initial currency supply will not start from zero, the currency supply growth in the first eight years will actually be slower than Bitcoin, giving fundraiser participants and early adopters a chance to benefit substantially in the medium term.

At some point in February, we will release a centralized testnet - a server which anyone can use to send transactions and create contracts. Soon after that, the decentralized testnet will come, which we will use to test different mining algorithms and make sure that the peer to peer daemon works and is secure, and take measurements to look for optimizations to the scripting language. Finally, once we are sure that the protocol and the client is secure, we will release the genesis block, and allow mining to begin.

Looking Forward

Since Ethereum includes a Turing-complete scripting language, it can be mathematically proven that it can do essentially anything that a Bitcoin-like blockchain-based cryptocurrency potentially can do. But there are still problems that the protocol, as it stands today, leaves unresolved. For example, Ethereum offers no solution for the fundamental scalability problem in all blockchain-based cryptocurrencies - namely, the fact that every full node must store the entire balance sheet and verify every transaction. Ethereum's concept of a separate "state tree" and "transaction list", borrowed from Ripple, mitigates this to some extent, but nevertheless no fundamental breakthrough is mine. For that, technology like Eli ben Sasson's Secure Computational Integrity and Privacy (SCIP), now under development, will be required.

Additionally, Ethereum offers no improvements on traditional proof-of-work mining with all its flaws, and proof of excellence and Ripple-style consensus are left unexplored. If it turns out that proof of stake or some other proof of work algorithm is a better solution, then future cryptocurrencies may use proof of stake algorithms like MC2 and Slasher instead. If there is room for an Ethereum 2.0, it is in these areas that it the improvements will lie. And ultimately, Ethereum is an open-ended project; if the project gets enough funding, we may even be the ones to release Ethereum 2.0 ourselves, carrying over the original account balances onto an even further improved network. Ultimately, just as is our slogan for the currency itself, the only limit is our imaginaion.

by Vitalik Buterin

Vitalik Buterin is a co-founder of Bitcoin Magazine who has been involved in the Bitcoin community since 2011, and has contributed to Bitcoin both as a writer and the developer of a fork of bitcoinjs-lib, pybitcointools and multisig.info, as well as one of the developers behind Egora. Now, Vitalik's primary job is as the main developer of Ethereum, a project which intends to create a next-generation smart contract and decentralized application platform that allows people to create any kind of decentralized application on top of a blockchain that can be imagined.

KEYWORDS: #ethereum #bitcoin #protocols #data #alice

You Might Like:

AUG 01, 2016

Without Unified, Federal Regulations for Digital Currencies, the U.S. Risks Falling Behind

#REGULATION

Related Articles:

AUG 03, 2016

E.U. Representatives Clarify the Proposed Anti-Money Laundering Directive

#REGULATION

FEB 19, 2016

Insufficient Backups, Not Bitcoin, at Fault as Hollywood Hospital’s Data Held Ransom by Hackers

#SCAM

JUL 11, 2016

Deutsche Börse Launches Blockchain and Fintech Venture Capital Fund

JAN 25, 2016

PayPal: Unprecedented Disruption in Payments and Financial Services

#WALL-STREET

#NONE

BTC MEDIA

About Advertising Careers Contact Terms of Service yBitcoin Store Facebook Twitter

Reddit RSS