



CoinDesk

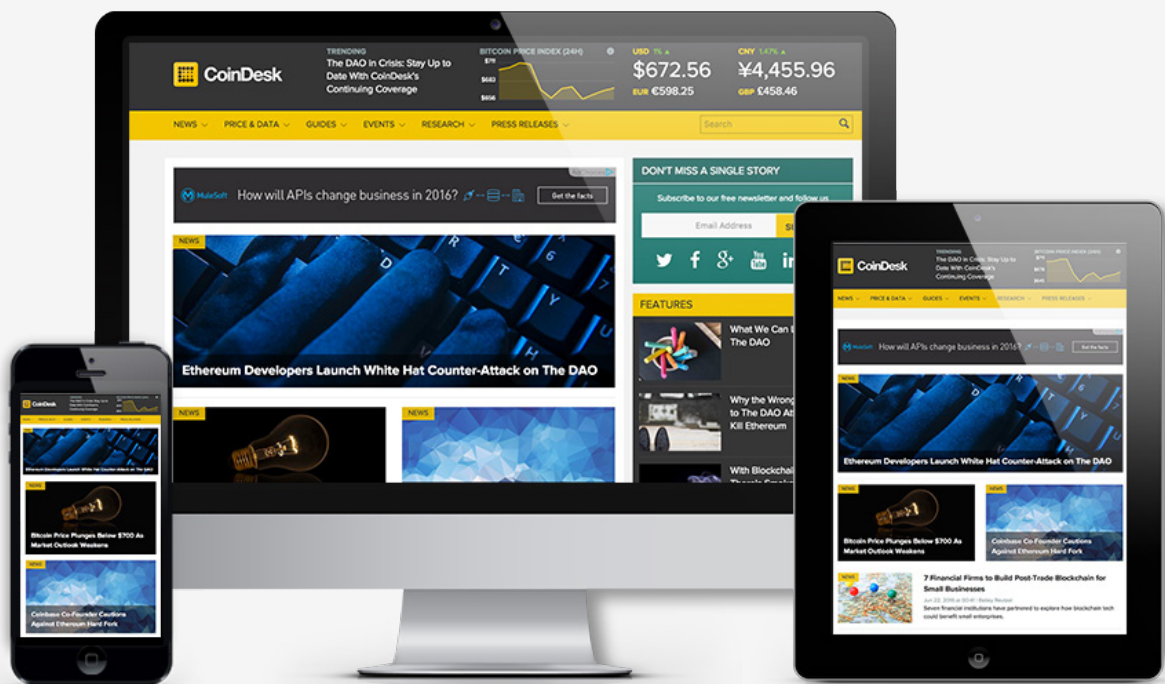
# UNDERSTANDING ETHEREUM

One of the most prominent next-generation blockchain platforms, Ethereum has energized a global development community and reimagined how decentralized technology can be applied to new digital challenges. This report features an overview of its expanding community and evolving open-source technology.

REPORT

[www.coindesk.com](http://www.coindesk.com)

**CoinDesk** is recognized as the world leader in news, analysis and information on digital currencies and blockchain technology. Our platform and resources allow us the opportunity to provide industry-leading research and in-depth analysis on the pressing issues that surround one of the most exciting emerging technologies of our time.



# Table of Contents

<b>1. What is Ethereum?</b>	04	<b>4. Platform Functions &amp; Use Cases</b>	23
Vision	05	Smart Contracts	23
Project Origins	06	DAOs	23
Key Milestones	07	Dapps	24
<b>2. Building Blocks</b>	08	<b>5. Technical Improvements (Challenges)</b>	25
Linguistics & Scripting	08	Scripting	25
Transactions	09	The Price of Gas	25
The Ethereum Blockchain	09	Mining Centralization	26
Block Size	10	Turing Completeness	27
Blockchain Size	10	<b>6. Ethereum 2.0 (Solutions)</b>	28
Block Times	10	Proof-of-Stake	28
Consensus Algorithm	11	Casper	29
Transaction Validators	12	State Channels	29
The Ethereum Virtual Machine	12	Sharding	30
Solidity	13	Development Timeline	30
Supporting Protocols	13	<b>7. Technical Infrastructure &amp; Key Players</b>	32
Whisper	13	Developer Leads	32
Swarm	14	The Ethereum Foundation	33
Oracles	14	Decentralized Projects	35
Mist	14	The DAO	35
<b>3. Supply, Trading &amp; Availability</b>	15	Other DAOs	36
Overview	15	Augur	36
Inflation Rate	15	Startups	38
Gas	15	<b>Conclusion</b>	45
Economic Structures	16	<b>Appendix: Getting Started With Ethereum</b>	46
Trading	16	Writing Solidity	47
Price	16		
Market Dynamics	17		
China	18		
Adoption	20		
Node Distribution	22		
Developer Activity	22		

# What is Ethereum?

## “The next Internet.”

It's a phrase that's often used when discussing bitcoin, the decentralized digital currency, and the blockchain, its distributed global ledger. Yet, the phrase is perhaps misleading in its simplicity.

While commonly referred to as a singular construct, “the Internet” is rather a web of protocols and rule sets that combine to power complex communications, collaboration and business processes.

When viewed similarly, “the blockchain”, or the public, permissionless blockchain protocols, could be seen as a more primitive version of what could become a mature “Internet of Value”. Such a public utility could one day provide a similarly layered architecture to expand the Internet of Information, or the Internet as we know it today, to deliver all manner of financial and non-financial transactional services.

If the Internet decentralized access to information, thereby increasing access to communication tools, the vision for the blockchain is that it would decentralize, and reduce barriers to establishing trust and transacting in the digital world.

First introduced in 2014, ethereum can be seen as both a realization of this future, and a recognition of the limitations of the bitcoin network, the first widely used public blockchain.

In his keynote announcement for the project, creator and inventor Vitalik Buterin described ethereum in such terms, arguing that bitcoin was not designed to serve as the blockchain's answer to the Transmission Control Protocol (TCP) or Internet Protocol (IP), the code that forms the basic communication language of the Internet.

Buterin wrote:

**“Bitcoin was designed to be a [Simple Mail Transfer Protocol] SMTP. It's a protocol that is very good at one particular task. It is good for transferring money, but it was not designed as a foundational layer for any kind of protocols to be built on top.”**

In remarks, Buterin spoke of the need for a technology that was more expansive, and that replicated the functionality of Turing-complete programming languages in a way that would be so powerful as to describe any blockchain application that could possibly be built.

**“Ethereum does not have features, it just has a programming language,” he said.**

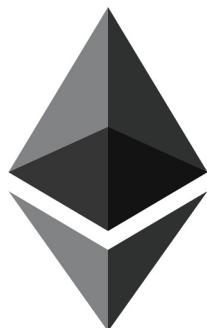
## Vision

Ethereum is perhaps best viewed as an attempt to apply learnings from bitcoin's decentralized, global cryptographic network to challenges beyond value exchange. Rather than disintermediate third parties in e-commerce, it envisioned how removing other traditional arbitrators of trust could enable a new wave of application development.

Ethereum's "business problem", as articulated by former ethereum CCO Stephan Tual, is that most Internet services are centralized.

**"You trust your bank to keep your money safe. The same is true of Facebook when you upload a picture of your kids or you push a document to Dropbox. As a developer you need to submit your application to an app store and risk having it removed."**  
Tual explains.

Ethereum seeks to enable the creation of similar Internet services while restoring the control of personal data and funds to users.



ethereum

**Ambitious in scope at a time when many blockchain platforms were slightly modified versions of bitcoin, ethereum would seek to enable innovations in four areas:**

- **Currency issuance.** Buterin positioned ethereum as a platform that would enable thousands of digital currencies to operate on the same network, with the goal being an "economic democracy" that would enable more efficient funding of philanthropic and other difficult-to-finance societal goods.
- **Decentralized autonomous organizations.** Buterin envisioned how new forms of digital entities could be built to manage shared resources under a set of terms and conditions enshrined in code and empowered by the collective decisions of stakeholders.
- **Smart contracts.** New contracts, he said, could be built that instead of being enforceable through a legal system, would programmatically enforce themselves.
- **Smart property.** The definition of property would expand with the idea that cryptographic, blockchain-based tokens could serve as representations of real-world assets, like museum passes or tickets.

## Project Origins

On a more anthropological level, ethereum can be seen as an outgrowth of an ideological subset of the bitcoin community that sought to build additional functionality onto the network without creating a wholly new blockchain.

Given that the bitcoin blockchain can securely arrange and record transactions of bitcoins, they posited, there should be no reason that these bitcoins can't be modified or otherwise made to represent other assets.

Rather than simply sending and receiving money, this community wanted to use bitcoins to represent commodities, derivatives or even deeds to real estate, in a sense, anything for which a secure, fixed unit of code could function as a digital asset. But, the problem was that certain beneficial features that worked with bitcoin proved difficult to translate to bitcoins that represented other assets. With a native blockchain currency like bitcoin, a user could control his or her assets

simply by controlling the associated private keys. However, with a colored bitcoin representing an asset such as an ounce of gold, a user might control the private key to that asset without controlling the gold itself.

With this in mind, the ethereum team set out to build its own blockchain and a new programming language, designing it from scratch to create a “world computer”, the computational power of which could be accessed in real time by an open market of users.

Programs could be run on the ethereum blockchain, with transactions serving to mediate interactions between these programs. As CTO Gavin Wood's yellow paper explained, anything that can be represented by a computer would be admissible on ethereum.

While the ethereum project would come under fire from critics throughout its early development lifecycle, as of the time of writing, the technology has arguably provided ample evidence that it is moving toward its ambitious aims.



## Key Milestones



**JANUARY 2014** – Ethereum inventor Vitalik Buterin announces the project at The North American Bitcoin Conference.

**JULY 2014** – The Ethereum Foundation begins selling ether tokens in a 42-day public sale. In total, it **sells** 60,102,216 ETH for 31,591 BTC, worth \$18,439,086 at that time.

**JULY 2015** – Ethereum launches Frontier, a command-line version of the platform for developer testing.

**AUGUST 2015** – Kraken becomes the first major digital currency exchange to list ethers for sale. Major exchanges including Coinbase and Gemini follow suit.

**JANUARY 2016** – Eleven major banks – Barclays, BMO Financial Group, Credit Suisse, Commonwealth Bank of Australia, HSBC, Natixis, Royal Bank of Scotland, TD Bank, UBS, UniCredit and Wells Fargo – announce a trial of a permissioned version of the platform.

**JANUARY 2016** – The first ethereum startups begin to raise funding for projects as diverse as a decentralized stock market and developer tool suites.

**MARCH 2016** – Ethereum releases Homestead, the first “production-ready” version of its blockchain platform.

**MARCH 2016** – The total value of all ethers on the ethereum network passes \$1bn.

**MAY 2016** – The DAO becomes the largest decentralized autonomous organization, collecting more than \$160m worth of ethers to be invested in other projects.

**JUNE 2016** – The DAO collapses after an unknown attacker exploits a flaw in the project’s code. The event forces ethereum’s development community to consider protocol-level code changes to rescue customer funds.

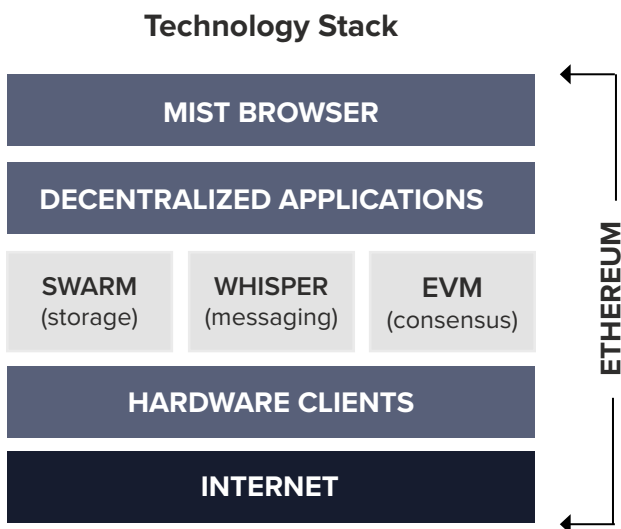


# Building Blocks

**As with the Internet, Ethereum similarly isn't just one thing, but rather a sum of many different parts.**

A non-exhaustive list of components includes a cryptographic token and address system, a network of validators (miners), a consensus algorithm, a blockchain ledger, the ethereum Virtual Machine, a set of programming languages and complex economic structures.

The following section attempts to highlight these specific components and illuminate the functionality each provides to the larger ethereum network.



*Ethereum Technology stack: Courtesy of [Stephan Tual](#)*

## Linguistics & Scripting

In computer science, a scripting language is a programming language that supports scripts, or programs designed for run-time environments that execute tasks and reduce the need for human operators. Because of this, scripting languages tend to be best utilized for experiments and rapid prototyping.

Bitcoin has an intentionally rudimentary scripting language, and there's a reason for this. From its inception, bitcoin's developers have prioritized the ability to "push" transfers of bitcoin via the bitcoin network over all other applications.

While there have been discussions about adding a more powerful scripting language to facilitate easier application development, the view of the bitcoin development community has largely been that it is more important to prioritize censorship resistance and network security over adding functionalities to the code.

Ethereum, by contrast, aims to be "Turing-complete." This means that, if a system has unlimited resources, memory, computational power and storage, then infinite "loops" can be executed. In other words, the logic and functionality that may be embedded in ethereum transactions is only practically limited by the availability of the protocol's native currency.



However, this functionality comes at the cost of security. Powerful scripting allows for greater function, but the additional tools also create new potential attack vectors. (See “Challenges.”)

## Transactions

The most notable difference between the two blockchains is that ethereum blocks contain both a transaction list and the most recent “state” of the ledger of these transactions.

This is a necessary feature to manage two types of accounts:

- **Externally owned accounts (EOAs).** Defined as the basic form of account, EOAs interact with and generate updates on the ethereum blockchain.
- **Contracts.** Contracts programmatically execute when they receive instructions in the form of a transaction from an EOA. Contracts can push or pull funds, and request these actions from other contracts, calling on the code to perform dynamic actions.

Ethereum notably does not use transaction inputs or outputs, which deviates from the unspent transaction outputs (UTXO) model bitcoin popularized.

In bitcoin’s model, each newly minted bitcoin becomes an unspent transaction output with an owner who retains the right to consume that bitcoin at a later time. During a bitcoin transaction, these UTXOs become the inputs that are “consumed” in the transaction. When these bitcoins are spent, or pushed, to another user, a brand new UTXO is created.

Ethereum, by contrast, uses a more familiar method. It stores the current “state” of its network, including a full list of accounts and their associated balances. Rather than confirming that UTXOs used

in a transaction are valid, ethereum determines whether the sender has a sufficient balance, much like a bank verifying whether a check will clear.

This design feature becomes important when transactions include contracts as recipients. If the transaction recipient is a contract, then that contract’s code will execute, changing both the state of that contract and potentially triggering other contracts to execute code as well.

## The Ethereum Blockchain

Both ethereum and bitcoin operate global transaction ledgers that today achieve remote and distributed validation through the use of a Proof-of-Work (PoW) protocol, a design in which participants expend significant energy to identify unique pieces of data that can then be easily verified by the wider network.

This data is used to generate blocks, or certain finite quantities of transaction data, which serve as a reference for all other network participants. The resulting blockchain is able to provide a history of the network at each of these intervals, creating a shared truth as it relates to events.

Blocks in both bitcoin and ethereum are today similar, containing information such as the block number (denoting how many blocks have passed since the initial block) and the difficulty (a metric that denotes how challenging it is to complete the work needed to create a block).

On the bitcoin network, the transaction script is “stateless”, meaning there is no state prior to execution of the script, and an update to this state is not saved after its execution. Contracts on ethereum are considered “stateful”, meaning that they are aware about past information stored on the network and, if instructed via smart contract, can be programmed to take actions in the future. When peers, or members of the ethereum

network, receive a block of data, they then run all transactions to verify a mathematical figure representing the system state at that time. If the nodes can validate this data, they accept the block for inclusion on the blockchain.

### Block Size

On the bitcoin blockchain, blocks are limited in size to 1 MB. This not only creates a cap on the amount of transactions that can be processed per second (currently it's seven), but it also has turned into a major point of contention within the bitcoin community as it seeks to increase this limitation.

Ethereum has no such limit on the size of its blocks. Because ethereum executes scripts and contracts, this is a necessity, as limiting the size of a block would not only stunt the concept of Turing-completeness, but limit the amount of storage a contract could use to execute.

Rather than limit the size of its blocks, ethereum employs a mechanism which makes contracts more and more expensive to execute the larger they are in size. [See "Gas" section]

### Blockchain Size

As on the bitcoin network, the more transactions that are executed on ethereum, the more information all the peers on the network need to store.

The need to track and store all these transactions, in turn, requires resources from the network of computers running the blockchain. As of May 2016, the size of the ethereum blockchain has grown to approximately 17 GB.

While this is still dwarfed by the bitcoin network's blockchain size, which stood at a little under 69 GB

at that time, it's worth pointing out that the bitcoin network is over eight years old, while at 17 GB, ethereum has been operating for just nine months.

At an average growth rate of around 1 GB per month, ethereum's blockchain is growing more slowly than bitcoin's, which is expanding in size at approximately 3 GB per month. However, ethereum has gained significant traction since its genesis block, and as the network becomes more popular, that monthly growth rate could accelerate.

While this could become a concern, ethereum is currently seeking to migrate to a new consensus algorithm that aims to alleviate this issue [See "Proof-of-Stake"].

### Block Times

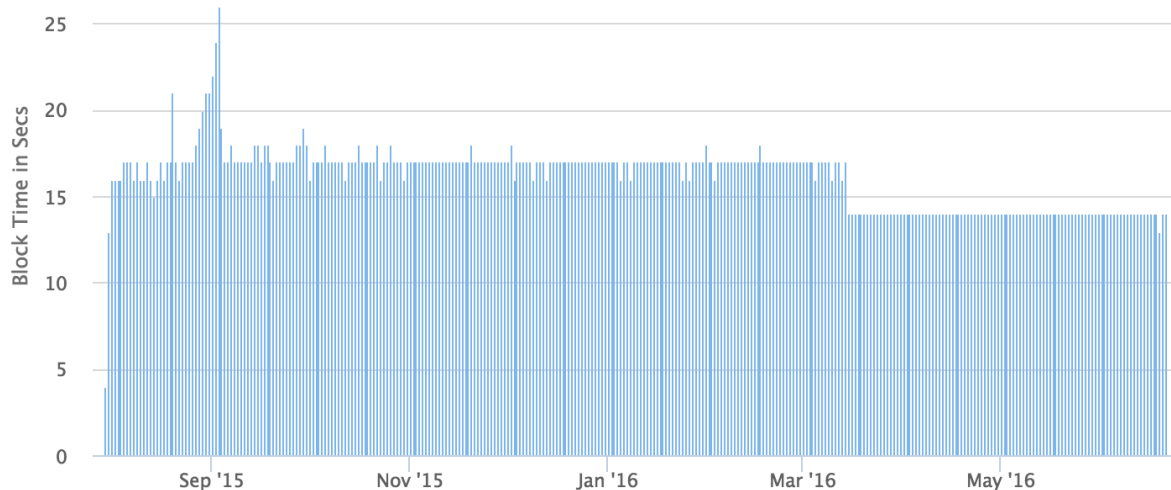
During its design phase, the ethereum team was also keen to address what they perceived as issues or limitations in the operation of the bitcoin blockchain. One issue that attracted attention was the time it takes transactions on the network to settle against the blockchain.

Bitcoin's blockchain adds new blocks roughly every 10 minutes, which means that a transaction is generally not confirmed on the ledger until this time. In practice, actual confirmations may take longer, as those who use the protocol generally wait for six confirmations, or six blocks, before considering a transaction settled.

Bitcoin and ethereum are not the only blockchains, or ledger structures, that use consensus methods. Various protocols have various block times. The Ripple protocol, developed by San Francisco startup Ripple, is designed to update its state every three to six seconds. Ethereum has set its target on 12-second block times, though current block times are closer to 14 seconds.

### Ethereum Average BlockTime Chart

source: [etherscan.io](https://etherscan.io)



## Consensus Algorithm

For any distributed computing system to properly function, there needs to be a mechanism by which the entire network can come to agreement on its state, or how its token supply is divided among registered addresses on the network.

Bitcoin uses what is referred to as “Nakamoto consensus”. Truly the pivotal innovation behind bitcoin, Nakamoto’s invention solved a long-standing computer science problem known as Byzantine Fault Tolerance, or the Byzantine Generals’ Problem, the idea that one cannot trust someone who has the potential motivation to lie, and one cannot trust the integrity of a given communication if it has first passed through an intermediary.

Bitcoin solves this problem through creating a chain of proof of work. The miners on the blockchain expend energy to solve a complicated mathematical equation in a bid to receive rewards when they find the next “block”.

Since the next block always follows from the previous block (meaning you start the equation from the point of the block), miners rush to verify that the block is valid so they can quickly turn to finding the next block and claim the reward.

It is the incentive compatibility, and also immutable nature of entries in the blockchain, that provides a solution to the Byzantine Generals’ problem.

Although there are plans in the future to migrate the network to a different protocol in the years ahead, at the time of writing, ethereum uses a similar PoW protocol known as Ethash. Ethash differs from bitcoin’s “Nakamoto-style” algorithm in a number of ways, the most familiar of which is that it uses different cryptographic primitive for its hashing function, known as SHA-3, rather than SHA-256.

While the differences are nuanced, Ethash is designed to make ethereum both resistant to the high-powered mining chips that currently dominate the bitcoin industry, and more accessible to “light” client implementations that allow users to use ethereum without needing to first download the ethereum blockchain to their device.

## Transaction Validators

Ethereum also differs from bitcoin in its transaction validation, both as it stands today, and as the network intends to function as it implements key changes in the future.

In 2009, the first bitcoin users were able to run mining software on home computers, using CPU power. As bitcoins became more valuable, a race for hashing power began, leading innovators to develop more and more powerful mining equipment.

Today, the majority of bitcoin mining is done in data centers, largely by VC-backed companies that control the production cycle of equipment and collaborative collections of individual miners known as mining pools.

To mitigate this consolidation, ethereum mining was set up so that it could only be conducted with graphics processing units (GPUs). The network is permissionless, meaning that anyone who purchases a graphics card and elects to run an ethereum client can begin processing transactions.

However, if the intended switch to a new “proof-of-stake” consensus protocol is completed – mining may no longer be needed in the near future. [See “Challenges” section]

## The Ethereum Virtual Machine

The ethereum protocol is designed to do far more than process peer-to-peer transactions. It is designed to execute complex code, where the functionality is only limited by the imagination of its developers and available resources.

As such, a system is needed to interpret instructions, and on ethereum, this task is handled by ethereum virtual machines (EVMs). Smart contracts are facilitated and enforced through EVMs, which implement and execute instructions written in any of a variety of languages via a bytecode.

A bytecode, also called a portable code, is a type of instruction set created to be executed by a software interpreter. Just like the example of an “if-then” argument in a Microsoft Excel spreadsheet (albeit with often much greater degrees of complexity), EVMs interpret the bytecode, evaluate the transaction states and execute the code to deliver predictable outcomes.

EVMs in particular deliver this through a “Turing-complete” scripting language – allowing at least in theory, for infinitely complex contracts.



### Solidity

Ethereum would be incomplete without a native programming language – and that language is Solidity. Solidity is the code that makes it possible to run contracts or programs in a distributed manner.

To describe Solidity crudely, it closely resembles the browser-based JavaScript language, but for executing ethereum contracts. In contrast to an “object-oriented” language like JavaScript (which combines variables, functions and data to run certain human-operated commands), Solidity is “contract-oriented”. Its run-time environment tasks are automated, and its objects are bundled together to avoid the need for manual commands.

Solidity is often described as ethereum’s scripting language, but it is actually a compiled language, not a scripting language. It compiles instructions into bytecode so that they can then be read by the network.

This is a critical feature given that contracts are not wholly compiled and independent programs, but rather partially compiled programs that depend on EVMs to run.

Solidity is also designed to express agreements that encode relationships and arguments that exist in real life. It therefore includes more concepts than an object-oriented language. Identity, ownership and protections form a core part of the programming grammar, which doesn’t have a parallel in JavaScript.

As the language matures and adds more libraries and users, it has the potential to create massive and powerful constructs that could end up having real-world applications.

For example, the Internet of Things (IoT), the vision for connecting devices and appliances to

the Internet, will require a massive amount of machine-to-machine communication, infrastructure and contract execution. A language like Solidity could play a key role in enabling these devices to talk to one another.

### Supporting Technologies

In addition to the main ethereum blockchain protocol, there are also supporting technologies in development that seek to help the network, and components built on the network, run more efficiently.

For example, whole new protocols are being constructed that aim to increase the functionality of distributed applications, while tools are evolving to allow these programs to harness data from multiple blockchains.

While there may be little that unites the following concepts on the surface, all are aimed at making ethereum more flexible for developers and users.

### Whisper

A communications protocol and tool set that allows applications built on the ethereum protocol to talk to each other, Whisper combines aspects of a distributed hash table and a point-to-point communications system.

Whisper is best explained in practice as it can be used to help facilitate exchange by recording buy or sell offers, allow for the creation of general chat room-like apps or even provide “dark” communications between parties who don’t know anything about each other.

With Whisper, you can imagine an ethereum application for whistleblowers who want to communicate to a journalist where they’ve stored a trove of data, but don’t want their identity to be linked to that data.

### Swarm

SwarmHash or Swarm is a peer-to-peer file sharing system designed to efficiently store and retrieve data needed for use in ethereum applications and contracts. The easiest analogy to draw would be that Swarm is essentially BitTorrent for ethereum.

As we will discuss later, storing data directly on the ethereum blockchain is expensive [See “Price of Gas”]. While contract code will have to be stored on the chain, reference data needed for contract execution should not. For instance, if a simple contract were to say deliver an e-card with pictures, the photos would take up a lot of space.

Perhaps a school would want to send out an album with photos of its latest graduating class. Such an application, if run on ethereum, might require a contract that is 1 KB, but be designed to deliver 1 GB of data. Storing and transacting that 1 KB of code might cost users a few cents, whereas storing the album itself could cost more.

By instead storing the album remotely, and accessing the file via a BitTorrent-like system, this would allow ethereum applications to deliver the instructions, with the files to be transferred via Swarm, not the ethereum blockchain directly.



### Oracles

For smart contracts to execute properly, they need not just be a well designed series of “if then” statements – they also need to know how to ascertain the accuracy of given inputs to those “if-then statements”.

If it’s raining in New York City, and there are multiple reliable sources that can confirm it is raining, how does ethereum weed through potentially fraudulent sources to identify the veracity of the input?

Here, there is a need for a construct that communicates outside realities to smart contracts.

In ethereum, these are called ‘oracles’. While a number of projects are building their own private oracle systems [See “Augur”], there have been some attempts to create platform-agnostic systems for verifying inputs to multiple blockchains.

Though there are currently a limited number of data sources that can be cryptographically proven – it isn’t hard to imagine a future in which smart technologies and the Internet of Things could allow all sorts of external data to be incorporated into contracts.

### Mist

If ethereum is to be the new TCP/IP, the project needed a new version of ‘browser’, a usable front-end technology with which users explore the applications and offerings that utilize ethereum.

Styled as a decentralized application discovery tool, Mist is meant to serve as a wallet for smart contracts that features a graphical interface and allows users to dynamically set transaction fees and manage custom tokens.

At the time of writing, Mist is still in beta and is under heavy development.



# Supply, Trading & Availability

**Another important component of the ethereum network, and one that has attracted the interest of investors is ether (ETH). ETH is a unit of account and store of value on the ethereum blockchain, equivalent to bitcoins (BTC) on the bitcoin network.**

Ether, while having an economic value as a scarce commodity, is not meant to serve as an alternative currency like bitcoin. Rather, it has been positioned as a system resource that powers the creations of those seeking to use the platform.

If bitcoin's value is derived from the security of the network and its scarcity, ether has value because it is needed to execute scripts and contracts on the ethereum network.

For that reason, ether has been called the "digital oil" to bitcoin's "digital gold".

## Inflation Rate

The ethereum network includes a mechanism for releasing new ethers into the system over time. Of note for investors familiar with bitcoin and other digital currencies, is that there is a difference of approach in ethereum.

For example, in bitcoin, the limit of all bitcoins that will ever exist is currently set at 21m BTC, a cap that would require a consensus

of participants to change. Ethereum, by comparison, has no hard limit on how much of its token will exist in the future.

Rather, its development team sought to use its token system in a way that would encourage access by introducing 18m ethers per year through mining. This steady rate of inflation, they reasoned, would then decrease over time as the overall token supply increased.

"New participants in the system will be able to purchase new ETH or mine for new ETH whether they are living in the year 2015 or 2115," developer Joseph Lubin wrote in his introduction of the issuance model.

## Gas

If ether served as a way to enable access to ethereum's world computer and ensure its functionality, an economic structure was also needed to limit access.

In order to complement ether and better explain its function of its token, ethereum introduced the concept of "Gas," a throttling mechanism that determines, in real time, how much ether each contract costs.

Gas has a fixed value, currently set at 10 "szabo", with one ether being made up of 1m szabo. The longer it takes for the contract to run, and the



more systematic resources it requires, the more fuel is needed to execute the contract.

Running contracts based on the Gas throttle, or ether limit, is a market-based solution that simultaneously limits the potential for hackers to spam the network and eliminates the need to set a fixed size for new transaction blocks.

### Economic Structures

Still, ether was created for more than executing transactions.

While bitcoin had succeeded in proliferating naturally over time through mining, the ethereum community sought to find a way to jumpstart this process and incentivize a base of evangelists who could help the network grow. To reach a critical mass of developers, ethereum's team used ether as an incentivization method to bring the project to life.

In July 2014, ethers became directly available for purchase on Ethereum.org, and more than \$18m was raised through the effort.

A point of contention that has emerged centers on the legality of the initial sale. At time of publication, no action has been taken against any of the individuals or groups involved; nor has any action been taken against other blockchain development teams that have used this approach to community building.

Nonetheless, the legal complexities involved have been acknowledged by its developers.

As stated above, this issue is by no means unique to ethereum given that a finite piece of data that can be exchanged via a blockchain has no natural legal equivalent or definition. While global regulators have sought to label all

cryptographic, blockchain-based tokens “virtual currencies”, the term doesn't quite capture how innovators in the ecosystem perhaps want their technology to be used or understood.

### Trading

So, what does the ethereum market look like in practice? Given that ethereum is a public utility, answers to this question are readily available through data analysis.

In the following section, we'll examine the current state of the ethereum project, how its marketplace is developing and the progress being made by the core development team.

### Price

While there may be no true value of any digital asset, the ethereum market provides clarity as to what users and traders believe is the value of ether, a metric that could also be argued is indicative of overall confidence in the project.

As an investment, ether has shown similar growth as bitcoin the digital currency. At the time of ethereum's initial crowdsale, users were able to purchase 2,000 ETH with 1 BTC, which was trading for just over \$600 at the time.

ETH has since seen its price rise and fall. Of particular note is that speculators seem to be attracted to coordinating action around major project releases [See “Development Timeline”].

Still, such downward movements have been slight compared to ETH's overall price appreciation. At the time of the crowdsale, the price of 1 ETH was roughly \$0.30. Compared to its value of \$14.30 at time of publication, this represents a 4,666% increase in value.

## Ethereum Charts



As the above graph shows, enthusiasm for ether is reflected in its recent price, and it has arguably been on an upward trajectory.

## Market Dynamics

An analysis of the network's blockchain shows that trading is today driving the majority of volume, though how much could be defined as speculative is uncertain.

Data from CoinDesk Research shows approximately 750,000 ETH (about \$10.8m at then prices) was being traded on digital currency exchanges daily in May, with this activity representing 50% of daily ether transactions. At time of publication this figure increased to 5.8m ETH (or \$81.2m), with this activity representing 66% of daily ether transactions.

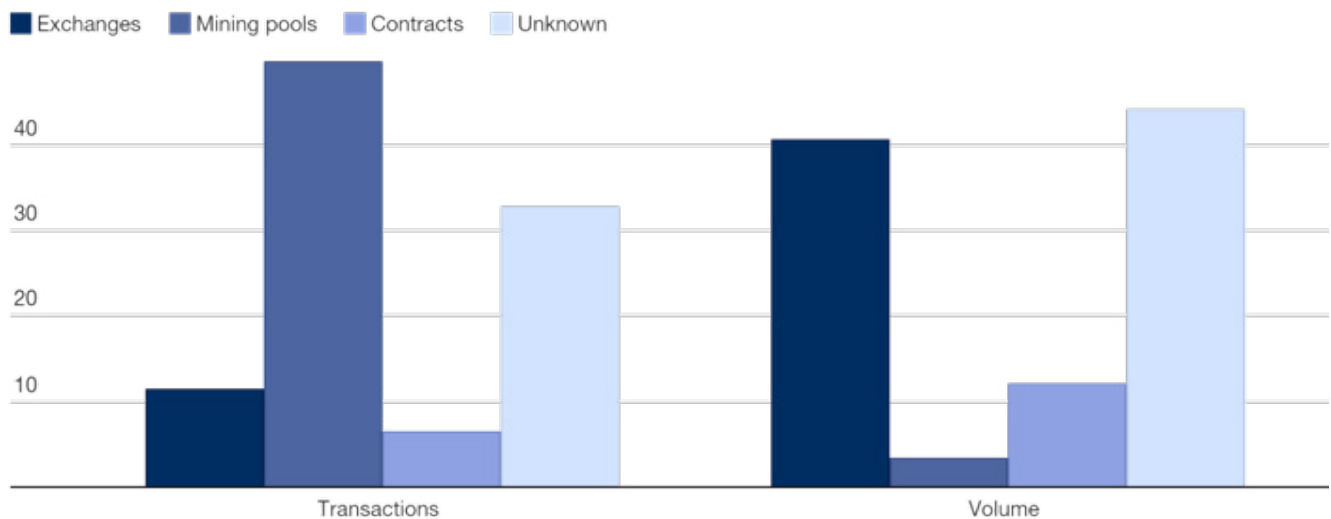
A deep dive into the data from blocks 1,468,000 to 1,568,000 on the ethereum blockchain shows which entities are the most active in transacting ether.

During this observation period, 699,900 transactions (representing 14m ETH) took place. In total, 14% of these transactions were conducted on exchanges.

By comparison, transactions sent between contracts (including those that are part of decentralized applications) accounted for 6.39% of the transaction total and 12% of volume.

The remainder were conducted by mining pools and other unknown entities.

### Analysis of Ethereum Transactions: April to May 2016



The data shows that trading is still the dominant use of ether, and that decentralized applications, while beginning to come online, still account for only a small part of the network's activity today.

## China

Analysis of the outflows of this capital reveals just under 80% of ether is traded for BTC, with the remaining trades denominated in USD, EUR and CNY. This figure was down from 90% just one month earlier.

So far, trading is concentrating on a small number of exchanges, with Poloniex and Kraken emerging as the market leaders.

### Ethereum Exchanges by USD Volume: June 2016

1. Poloniex (ETH/BTC) \$50,769,621
2. Kraken (ETH/BTC) \$14,425,770
3. Bitfinex (ETH/BTC) \$10,656,583
4. Bitfinex (ETH/USD) \$7,326,959
5. Yunbi (ETH/CNY) \$2,086,474
6. BTC-e (ETH/BTC) \$1,814,588
7. GDAX (ETH/BTC) \$1,426,082
8. BTC-e (ETH/USD) \$1,181,723
9. GDAX (ETH/USD) \$911,669
10. Bittrex (ETH/BTC) \$709,726

Perhaps most notable aspect of the ethereum market, however, is how it has developed comparatively to bitcoin's.

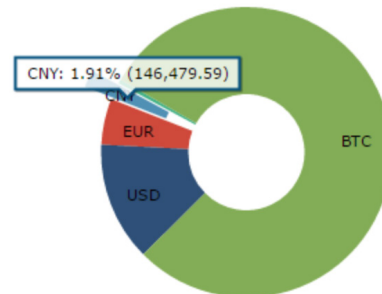
As of June 2016, CoinDesk Research data indicates 94.35% of bitcoin trades are for the BTC/CNY trading pair, with fee-free, China-based exchanges OKCoin and Huobi holding a more than 90% market share based on the strength of this market demand.

By contrast, ETH/CNY trades account for just roughly 2% of the overall ether market. This near 90% differential indicates the price of ether could grow should ethereum attract the attention of this already active part of the global blockchain community.

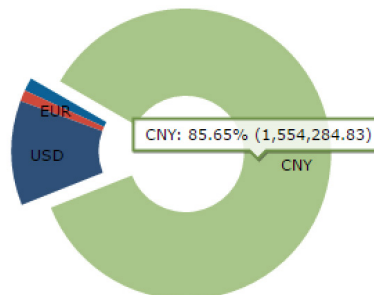
At present, more than 90% of bitcoin volume is being driven by Huobi and OKCoin, and at press time, neither had voiced plans to support ether.

Representatives from both exchanges told CoinDesk Research they are watching ethereum's development with interest.

### ETH Volume by Currency



### BTC Volume by Currency



## Adoption

Beyond the speculative use of Ethereum's token, there are metrics that suggest the platform is being adopted by an increasing number of application creators and users.

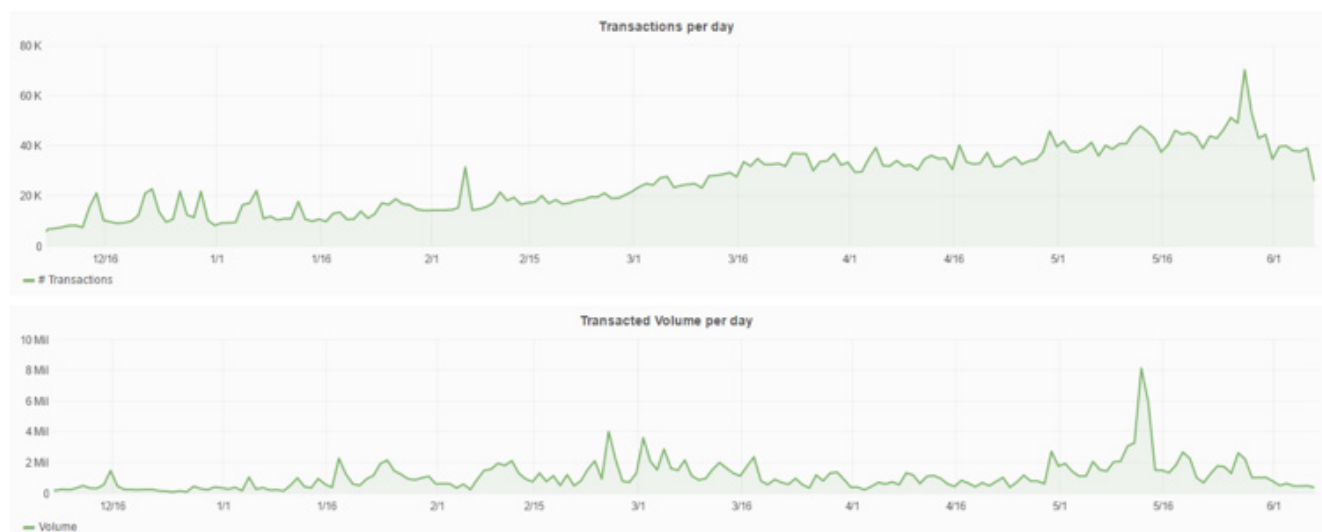
The number of Ethereum transactions, for example, has been steadily rising, hitting roughly 40,000 transactions per day as of June 2016.

Overall, transaction figures have roughly doubled since January of the same year.

A chart on the number of transactions per day also displays steady movement to the upper-right hand corner of the graph, even while volume remains more inconsistent and closely tied to intervals of high price volatility.

Other positive indicators include the rising number of unique addresses and the increasing network difficulty, which indicate more users are joining the network and more miners are securing the network.

Transaction Chart



Unique Address Growth Chart



Block Difficulty Growth Chart



## Node Distribution

Perhaps one of the strongest indicators of support for the ethereum network is the number of computers running versions of the ethereum client and its full blockchain history.

As of mid-2016, ethereum had 5,384 nodes connected to its network, a figure that was just shy of the 5,757 observed on the older bitcoin network.

There is also an observable relationship between the geographical distribution of both networks, with the majority of nodes being hosted in the US and Germany.

## Developer Activity

To date, one of ethereum's biggest success stories has been the perceived strength of its developer community. But, as for assessing the state of this community in practice, some quantitative metrics are available.

For example, ethereum and bitcoin's official GitHub pages shed light on developer activity by listing figures such as the number of commits, or changes to software files, and the number of contributors enrolled in the project through the platform.

Overall activity in the two development communities also appears comparable via GitHub data, though ethereum perhaps experiences more downtime due to its smaller number of contributors. Yet another useful metric is the number of forks, or copies of the repository developers can use to experiment with potential changes.

Here, bitcoin displays a clear advantage, perhaps due to the length of the project's development. Over 6,000 forks have been added to the bitcoin GitHub since 2009, while 525 forks have been made to the ethereum network since 2014.

### Developer Activity: Ethereum vs Bitcoin

	Commits	Releases	Contributors
Ethereum	7,387	88	75
Bitcoin	10,000	157	371

Source: GitHub



# Platform Functions and Use Cases

**While the architecture of the network is certainly impressive, it's what's built on these intricate components that truly illustrates ethereum's potential.**

Far from simple wallet and exchange constructions, ethereum has so far enabled the creation of a wholly unique lexicon of concepts. These range from simple programs that aim to replace traditional financial contracts to more complex constructions that could come to challenge investment firms and corporations.

## Smart Contracts

The basic building blocks of programs written for the ethereum platform are called “smart contracts”, and more complex structures on the network are best considered elaborate collections of these tools.

As an example of the type of functionality smart contracts achieve, it may be best to consider it as a kind of modern elevator management system. You could almost think of the elevators as running a “smart contracting system” on a “private blockchain”. Individuals who approach the elevator bank will each push a button for a different floor, and the management system organizes an efficient ordering of which elevators will go to which different floors.

Smart contracts can be thought of as “if-then” formulas in an Excel spreadsheet. They don't function similarly to contracts between people, but rather “fire and forget” predetermined outcomes, waiting for the inputs to be presented in order to deliver those outcomes.

To call a contract and modify the state of the address, a payment of ether must be sent along with the call, which then executes a method to adjust the required fields.

Ethereum's blockchain lets users rely on the inputs. While blockchains don't guarantee that all entries will be true, the immutable and permanent nature of the entries into the data structure forces those entering data into the environment to put their reputation on the line with each entry. Users can never take back what is “said” or entered.

In this way, ethereum can be viewed as an incentive structure that users can lean on to assume truth. With this assumption of truth, they can start to build these contracts or situationally dependant outcomes on top of the network's data.

## DAOs

While smart contracts on their own are interesting, it is the idea of large numbers of contracts working in unison that showcases the breadth and potential impact of ethereum's technology.

Combined, smart contracts can be made to form what have been called DAAs or DAOs, acronyms for distributed autonomous agents and distributed autonomous organizations.

## Dapps

Here again, it's important to note that the larger purpose of the ethereum network is to serve as a platform for the creation of distributed applications (dapps).

Dapps can be comprised of single DAO or even a series of DAOs that work together to create an application. This could result in something close to an application you may already be familiar with, like a Microsoft Outlook or Angry Birds, but the point is not that these apps be made to provide a certain functionality.

Rather, what makes a dapp is that it is implemented on the ethereum network, not locally on a computer or phone, or even on a single company's server.

At time of publication, CoinDesk Research has found there are more than 230 dapps in various stages of development, a figure that spans the spectrum to includes those at the concept, demo and live stages.

Further, analysis shows that dapps can currently be broken down into four categories:

1. **Smart contract services, utilities & analytics**
2. **Gambling and games**
3. **Information validation & oracle services**
4. **Registry and corporate governance.**

**Vitalik Buterin has summed up the differences between these concepts in the following terms:**

- **Smart contracts.**  
Versions of the technology that are single-purpose and ephemeral, so they are created for a specific task and can disappear at the end. A financial contract is a good example here.
- **Autonomous agents.**  
More long-term focused smart contracts, Buterin envisions a series of contracts forming an internal AI that can be charged with decision-making.
- **DAOs.**  
Described as a long-term contract between many people, DAOs are closer to historical business structures, allowing users to join, exercise voting power and even eventually exit such collaborations. DAOs are designed to hold onto assets and use a kind of voting system to manage their distribution.

There can be many different types of DAOs. The more basic ones live entirely on the blockchain, but more advanced ones might have some of their data stored on other decentralized networks or across a number of servers.

# Technical Improvements (Challenges)

**While the work that has been done to date is without a doubt impressive, there is still much to be done to improve ethereum.**

In the following sections, we review some of the planned improvements and larger challenges facing the network's development team ahead of this goal.

## Scripting

Ethereum's programming language remains a work in progress.

Solidity is a brand-new concept in computer programming, and script-based systems remain largely untested. Further, the language's compiler is buggy, and there aren't repositories and public libraries yet.

This makes creating functional smart contracts on ethereum difficult. Each module has to be as perfectly crafted as each gear in a Rolex. If the modules don't interact exactly as designed, the system breaks down.

One independent review of the ethereum code exposed the extent of what is becoming a more widely acknowledged problem outside the network's development community, estimating

there are potentially 100 bugs per 1,000 lines of code. Compare that to Microsoft's one bug per 2,000 lines of code, and you have an idea of the extent to which the project may need to make improvements long term.

While not all contracts will be as buggy as the one that was reviewed, the state of the solidity compiler is something that will need to be addressed before ethereum can scale.

Imagine gears in a Rolex only working right with each other 90% of time. You'd spend a lot of time readjusting the time as it slipped out of sync.

Such an issue could develop with ethereum's smart contract modules, except they may not just fail to keep proper time, they may stop working, suffer from security issues or potentially execute improper contract outcomes.

## The Price of Gas

The economics of the platform are also in early stages. To borrow a phrase from Western politics, on ethereum, the cost of "gas is too damn high."

As an example, it cost \$250,000 to process 1 GB of ethereum transaction data in May. At that time, the contract would cost 640,000,000,000 gas, or about 17,500 ETH, at \$14 per ETH.

To be fair, most contracts will be far smaller than 1 GB, and users would likely not want to store 1 GB of reference data on the ethereum blockchain when they could use a protocol like Swarm Hash. But storage and resources are still very expensive.

## Mining Centralization

As discussed above, ethereum also sought to implement an architecture that would alleviate issues that have contributed to the centralization of mining power on the bitcoin network, enabling a wider variety of users to be incentivized to boost the platform as a whole.

As recently as March 2016, however, one mining entity, dwarfpool, had amassed 48% of the network's hashrate, leading to concerns about

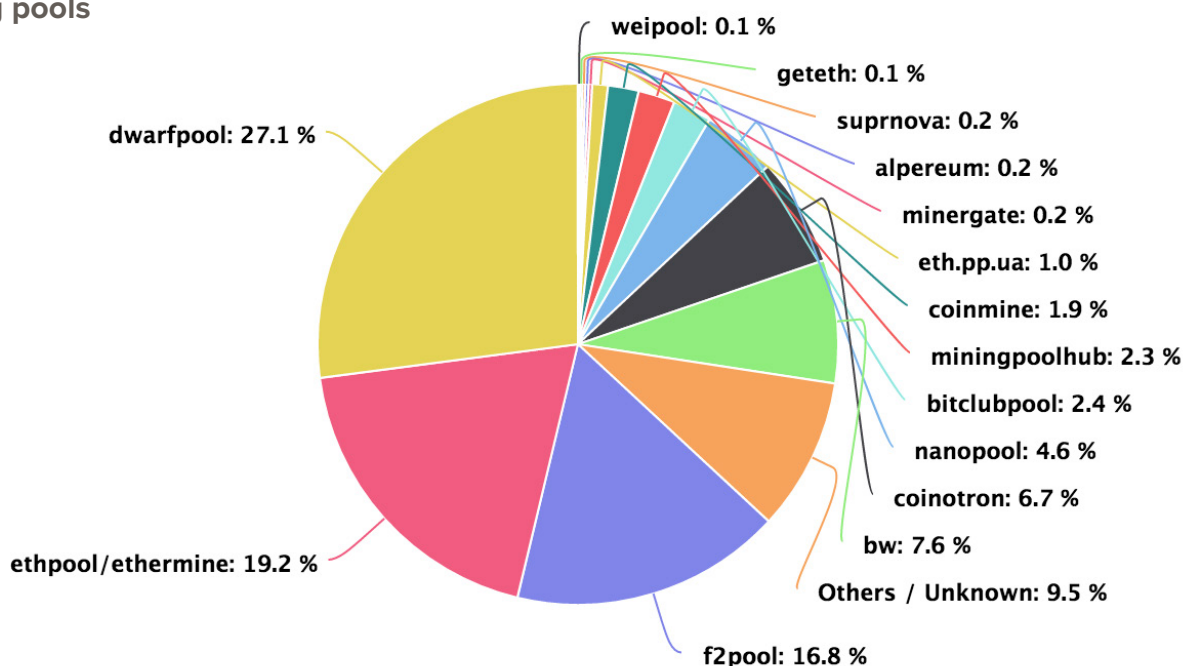
centralization and the possibility that one entity could gain control of the network.

Such an attack would find the entity changing the ethereum ledger at will and forcing its version of the blockchain to be considered valid, thereby undermining trust in the network.

A look at the network shows that its transaction validators have consolidated into a small number of entities and pools. However, this is due to the functionality of its existing PoW protocol, which as we covered previously, is designed to be replaced.

Ultimately, it is a move toward PoS [See “Proof-of-Stake”] that the developers see as a critical way to restore what was an original value proposition of decentralized blockchain networks, that anyone could participate simply by running a program on a computer.

### Hashrate distribution of mining pools



## Turing Completeness

As discussed above, ethereum is purportedly “Turing-complete”, but in reality, the system is limited by memory, computation power, storage on the network and economic costs.

The more complex the instruction set, the more messages that have to be passed back and forth within the system, the more delegates and code calls required by the contract, the higher the cost. The gas system ensures this.

Ethereum, however, has an accompanying economic system of ether and gas that makes it, at least at the moment, prohibitively expen-

sive to use. It creates an economic limit on the Turing-completeness simply by making storage space so expensive.

In some ways, ethereum can never really be a true Turing machine – at some point, a limit to computational power is hit, even if it grows to where the limiting factor or bottleneck is available electricity.

But for ethereum to achieve its vision, it only needs to reach a point of economic equilibrium where it is “practically” Turing complete, and limited by the economics of how much it costs to use.



# Ethereum 2.0 (Solutions)

**Blockchain technology has ushered in a new age in distributed computing. There is a powerful belief that distributing formerly centralized systems will be massively beneficial, both in removing the potential abuse and making them more fault tolerant.**

But, distributed systems are inherently less efficient than centralized systems. They are generally also slower, more costly and more complicated.

This must be the case, as when data is centrally stored, controlled systems do not need consensus layers. There is no computational power that needs to be spent to align the state of a centralized database across a broad system.

This challenge is one that faces all public blockchains, and ethereum offers no specific or special solution to this dilemma, at least today.

Yet, there are ideas being developed to attack this issue. From sharding and state channels to changes in the consensus algorithm, serious efforts are underway to find solutions that could allow ethereum to massively scale.

## Proof-of-Stake Transition

One of the proposed improvements to ethereum's current design involves a unique technical feat that would find the network turning off its Proof-of-Work (PoW) transaction validation mechanism and replacing it with one based on Proof-of-Stake (PoS).

PoW is a powerful consensus algorithm because it allows the system to prove that work was actually done to mine a block. PoS validation on the other hand, doesn't use a mining process. Holders of the network's tokens own stakes in the network based on percentage of ownership, and vote to validate and include blocks in the blockchain.

But, there are problems with PoS systems today. Should powerful forces gain the majority of ethers on the network, PoS could ensure these actors continue to have an outsized influence on the network. This would create a new upper class reminiscent of the landed gentry, a term that refers to a British social class able to support its lavish lifestyle purely from rental income.

But, there are benefits as well. If joining the network can be simplified, requiring only that the user download a program and hold a balance of ether, barriers to entry in the form of costly equipment can be all but eliminated.



One of the architects working on this migration is Vlad Zamfir, and he is candid about the technical challenges of the current Ethash protocol, which he said “doesn’t scale”.

Put more simply, he said: “Everything about ethereum is going to have to change.”

## Casper

Zamfir has so far spent 11 months researching, studying and testing out concepts to enable the eventual transition to PoS consensus. In August 2015, he made public a proposal for a new consensus algorithm that would be known as ‘Casper’, the name a nod to the fact that it is an adaptation of its existing GHOST mechanism, which replaces miners with ‘validators’.

These nodes estimate (based on what they can observe of the network) how the network state should look were they to verify all contracts, transactions and changes in the ledger that have occurred since the last point of consensus. They then broadcast that guess to each other and evaluate what other nodes are broadcasting to them.

As nodes recognize each other’s guesses, or votes, they begin to coalesce around a single network state. When the nodes are in agreement to some mathematical level, the network reaches consensus, and then records are updated in all nodes, including those that are not validators, and those validators who have not yet reached the same conclusion.

One dilemma that has emerged has been termed the “nothing at stake” problem, whereby PoS validators have nothing to lose by voting for more than one blockchain history, which in turn precludes consensus. Since there is no mining, and little resource is used to validate transactions, it becomes comparably easier to try and solve several versions of the blockchain at the same time.

Casper’s solution to this involves bonding. Validators must post value in the form of ETH into a smart contract that monitors their validation process. By putting value on the line, the incentive to “cheat” and validate multiple chains is eliminated by making it more costly to lose the bonded value pledged than it would be to gain a reward through cheating.

Casper is being built to monitor the nodes and detect “dishonest” actions. When Casper recognizes a “cheater”, it executes the contract to permanently confiscate the posted bond, and bans the node from becoming a validator in the future.

There will be several key benefits to this system, according to Zamfir:

- **A focus on CPU power rather than GPU power, making the network more egalitarian**
- **Better support for lightweight clients**
- **The capacity for more transactions per second**
- **The possibility of even faster block times.**

## State Channels

One partial solution, which doesn’t actually scale the core protocol but does effectively arrive at an improvement, involves state channels. Put simply, state channels are a method of conducting transactions that could occur off of the main blockchain. This is a critical component that would be needed to scale the ethereum protocol.

If state changes can be moved off of the ethereum blockchain, significant scaling becomes possible. It does, however, have to be done carefully to ensure that it doesn’t add risk to the network’s participants.



This requires some system that would lock the blockchain state by form of contract. In other words, in order to protect the participants in the off-chain transaction, both parties must be able to sign off on the validity of the transaction itself. The participants then must submit back the state created in the channel to the main blockchain, and the main blockchain must accept it as an update that necessarily amends and overrides the previously reported state from the channel.

This would unlock the value that is being kept off-blockchain and allow it to move back on blockchain, with the computational requirement for the state change having taken place off-chain and without creating a systemic burden.

State channels could become a powerful solution to scaling, and have benefits in other areas as well. For example, it could be seen as a way to provide heightened privacy. In the case of disputes, parties can end contracts without revealing what might have taken place.

## Sharding

Still, there is another solution being developed known as “sharding” that has, at the time of the report, yet to be introduced in a public blockchain.

In a sense, sharding attempts to leverage the insights of traditional database sharding, wherein portions of the full database are held on separate servers as a way to spread out the load and improve performance.

When applied to a public blockchain environment, implementing this architecture becomes more difficult, albeit comparably beneficial.

The successful sharding of the ethereum database would allow for multiple blockchains

to exist within the same network so that businesses, individuals or entities could run the equivalent of a public or private blockchain (with distinct transaction validators), but on a platform that leverages the security and functionality of a public platform.

By sharding the network into smaller chunks, the network state can be split, too. Each account will be it's own shard, which will only be able to send or call transactions within the limitations of this environment.

At the top level of the protocol, there won't be any major change, but underneath there could be a world of difference. Instead of the top layer of the network having to process each transaction and each contract, the smaller shards can be processed and then sent back to the top layer of the protocol. There, the state of the entire ledger would be updated with the processed information.

Until this takes place, ethereum truly cannot be a practical platform because it is extremely inefficient. But, by distributing the computational load among the shards, ethereum may yet become suitable for enterprise-level applications.

## Development Timeline

Ethereum has differed from other open-source blockchain projects in that it presented a detailed overview of its long-term roadmap early on in its development cycle.

First unveiled in March 2015, ethereum's timeline included four release steps, each with its own outline for what development changes would be needed to implement that vision.

In the following section, we review those steps:

### Frontier

Described as the ethereum network in its “barest form”, it was 19 months after the project’s initial debut that the genesis block in Frontier was generated on 30th July, 2015.

Frontier was the first version of ethereum, one described by the organization as a beta release aimed at developers who wanted to experiment with the project’s tools.

It offered basic command-line capabilities, and provided users the ability to mine ether and upload and execute contracts. This was the tool to stand up key components of the ecosystem such as exchanges and dapp development projects.

### Homestead

At time of writing, the most recent milestone cleared by the ethereum team, Homestead was described as the first “production version” of the network.

Released on 14th March, 2016, Homestead still features a command-line interface, but was framed as the first commercial iteration of the technology. Homestead was automatically introduced at block number 1,150,000 on the ethereum blockchain.

Perhaps most notable about the launch was that it required the ethereum community to undergo the hard fork, a process by which a change was made to the network’s consensus algorithm that invalidated a past rule, rendering nodes incompatible unless they upgraded.

The feat further came at a time of deep contention within the bitcoin community about its ability to make such a shift, and was widely seen as a validation of ethereum’s development team and its decision-making abilities.

### Metropolis

At time of writing, the next major release of ethereum will be Metropolis. Though no set date for the transition has been announced, ethereum has always been a developer-led effort, and developer-led efforts don’t necessarily stick to timelines.

Metropolis will be the fully-featured version of the product, aimed at non-technical users, and will be the first official non-beta version. It will also include the first fully functional version of the Mist browser, providing a graphical user interface atop the client.

This version is expected to bring fundamental back-end improvements and upgrades to Solidity. In many ways, Metropolis will represent ethereum version 1.0.

### Serenity

It won’t be until Serenity that we reach what the community is calling ‘ethereum 2.0’, a version of the platform that’s ready to scale.

Serenity will see major and fundamental changes in the way that ethereum functions as a platform and protocol. The first of these changes will be a migration away from the consensus algorithm currently underlying the ethereum blockchain. Ethereum will fork from a bitcoin-like PoW mining process to one whereby holders of ethers validate the state of the network through a voting mechanism.

In addition to the switch to PoS consensus, Serenity also plans to introduce scaling solutions including ‘sharding’ and ‘state channels’ to the ethereum protocol.

# Technical Infrastructure & Key Players

**The infrastructure of the ethereum project in many ways mirrors a core ethos of the project itself by being broadly distributed.**

Ethereum is a massive undertaking, led primarily by its developers, but relying on distributed efforts of a diverse community.

## Developer Leads

While ethereum may not have an anonymous creator at the heart of its origin story, there is certainly no shortage of mystery about the project.

Members of the ethereum community tend to be elusive when discussing its early history, though it's an open secret that there have been changes to its membership. The original thread introducing the project on the Bitcoin Talk online forum, for example, has been modified since its **original publishing**, with the full list including the names of developers and architects who have since moved on to other, unrelated projects.

Some of the prominent members, it should be noted, have stayed in the ecosystem after leaving positions with the Ethereum Foundation, a development that is fueling growth in its nascent startup community.

A major difference is that while bitcoin creator Satoshi Nakamoto abandoned the project at an

early stage, ethereum has arguably been fueled by the active involvement of its creator, Vitalik Buterin, and the development team can be seen as having a comparably more defined structure.

Two of the more often cited core developers include Gavin Wood (formerly the project's lead C++ developer) and Jeffrey Wilcke (its lead Go developer). Other prominent developers include several employed by the Ethereum Foundation. These are director of technology Taylor Gerring; JS client developer Martin Becze; lead dapp developer Fabian Vogelsteller; Mist developer Alex Van de Sande; and Swarm developer Viktor Tron.

In some cases, the Ethereum Foundation provides a mediating role in managing development needs. For instance, when Wood **departed in early 2016** to focus on a new startup project, Christian Reitwiessner, the creator of the Solidity language, was appointed by the organization as his successor in managing C++ development.

Development of ethereum's main code, however, remains a community driven effort, and as such, gaining a sense of the composition of the project's development community is difficult.

All that said, there are clear leaders in the community – like Buterin himself, who while holding no official power over a number of implementations, clearly have influence, not assigned, but earned through thought leadership and continued effort.



## The Ethereum Foundation

The leading organization behind the ethereum project, the Ethereum Foundation (Stiftung Ethereum) was established as a non-for profit company in June 2014 in Zug, Switzerland. Zug, a small city of 24,000 people, has made a concerted effort to help drive its local economy by taking a **progressive stance** toward such projects.

The Ethereum Foundation is the entity which issued the initial ether sale, and it was created to oversee this process, manage funding for development and pay back debt the effort had incurred through legal bills in the run-up to its launch.

The non-profit has an ongoing effort to organize and coordinate the community, managing accounts on **Meetup**, **YouTube**, **Twitter**, Q&A forum **Stack Exchange** and **Facebook**.

While the Ethereum Foundation has very little actual formal influence on many of the projects in the space, there is a high degree of cross-over between those who work at the Foundation and those involved in other prominent projects.

Today, the Ethereum Foundation is led by a governing and an advisory board and special advisors, and is organized as follows:



### EXECUTIVE DIRECTOR

#### Ming Chan

**An alumna of Massachusetts Institute of Technology,**

Chan has a background in enterprise IT and management consulting projects, founding and growing businesses, and working with top educators, scientists, and inventors to bring inspiring research innovation to life. Her work include legal and regulatory matters related to blockchain technology.

## GOVERNING BOARD



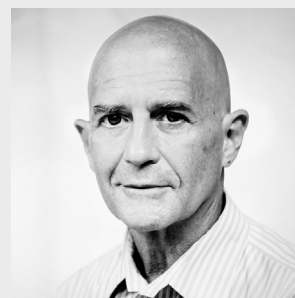
### Vitalik Buterin

**The creator of Ethereum**, Buterin co-founded Bitcoin Magazine in September 2011. Eventually moving on to development, he sold the platform and wrote the ethereum white paper in November 2013. He now leads ethereum's research team.



### Jeffrey Wilcke

**One of the founders of Ethereum**, Wilcke started the first implementation of ethereum using the Go programming language in 2013, and was the Go team lead at the time of the release of the genesis block and ethereum platform.



### David Ben Kay

**A lawyer** specialized in creating innovative intellectual property solutions for emerging markets in Asian markets, Kay was formerly General Counsel of Microsoft China.

**ADVISORY BOARD MEMBERS** include Bernd Lapp, a former head of sales at German mobile app Centralway; Stefano Bertolo, a scientific project officer at the European Commission and Yessin Schiegg, CFO of Zurich-based consulting firm Alpha Associates.

**SPECIAL ADVISORS** include entrepreneur and author William Mougayar; Thomas Greco, a special adviser to Asian FinTech company Omise; and Vladislav Martynov, CEO and co-founder of Yota Devices.

## Decentralized Projects

While just halfway through its planned rollout, ethereum has already seen a number of projects emerge that are seeking to bring its core concepts to life.

Far from just theory, ethereum-based projects are inspiring developers, overcoming challenges in the wild, inspiring research papers, grabbing global headlines and operating without the backing of a conventional corporate structure.

In the following section, we explore some early and notable examples.

### The DAO

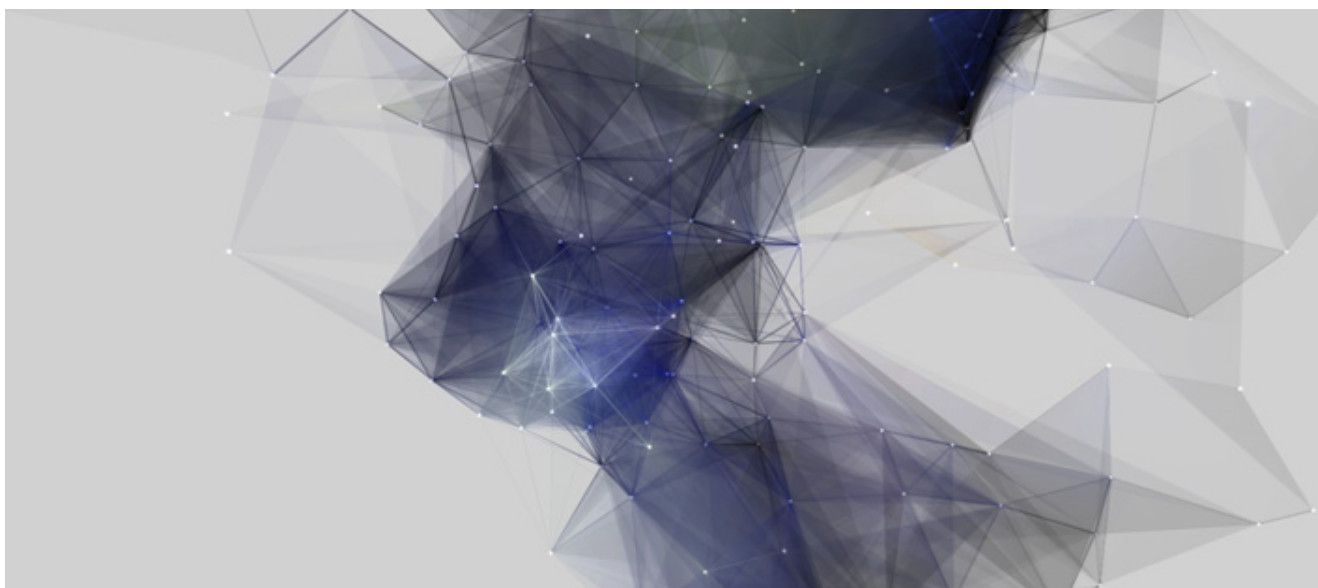
The most prominent ethereum project yet launched, The DAO was a DAO designed to collect ether investments and distribute those funds to projects voted on by an open community of donors and members.

In its short lifespan, The DAO amassed upwards of \$160m denominated in ether, and saw a number of proposals put forth for voting, though none were passed. The DAO quickly emerged as a magnet for academic criticism about how DAOs should be designed and their participants incentivized.

At time of publication, however, the project had effectively collapsed following an incident in which an attacker was able to exploit a functionality in The DAO's code. Called a "recursive call exploit", the attacker effectively requested funds from The DAO repeatedly, and the contract approved these fund requests without first checking the balance.

At time of publication, ethereum developers were considering a number of possible solutions to the loss of customer funds. These included a hard fork, or alterations to ethereum's code that would effectively reverse the hack, and a soft fork, which would enact code preventing the stolen funds from being redeemed.

While live, approximately 10m DAO tokens changed hands daily on the ethereum network.



## Other DAOs

At time of publication, a number of smaller DAOs have raised funds in ether or are in early stages of development, and trends in this market were beginning to take shape.

Digix, a DAO meant to create a gold-tracking asset for ethereum, raised \$5.5m in March in a crowdsale. MakerDAO, likewise, intends to launch a “stablecoin” with a fixed value that can enable a credit-based monetary system on the network.

This formation, in which a team of developers raises money to deliver code that can then be managed by a diversity of participants, seems most common among entrepreneurs seeking to launch products or exchanges centered on ether trading or investing.

Other notable projects that don’t quite fit into this framework include Golem Project, which is building technology that would allow users to trade the idle time of their computers, and Augur, a decentralized prediction market.



## Augur

Positioned as the first open-source, decentralized prediction market, Augur seeks to enable its global users to bet on the outcome of future events, with the goal of encouraging collective forecasting.

The development team aims to use the decentralized nature of blockchains to avoid issues that have historically plagued predictions markets, with centralized management seen as a point of failure that allowed earlier efforts to be shuttered by global governments.

Augur is rare among ethereum projects as it intends to leverage multiple blockchain technologies as well as the bitcoin currency to facilitate its operations and thus provides a compelling example of how future projects could leverage similar designs.


### How Augur Uses the Ethereum Blockchain

Augur uses ethereum to remove the need for users to trust counterparties, reduce costs and make the platform resilient against central points of failure.

The platform automates the custodianship of funds as well as the trading and settlement of these funds through smart contracts on ethereum, and by virtue of its design, allows all of it to be done with minimal trust.



## How Augur Works



Augur requires a lot of ethereum calls. It's not uncommon for a user to automatically make hundreds of RPC calls; or messages sent between the user and ethereum nodes, during an instance of using Augur. These calls are "free", although they consume bandwidth and time.

Let's take the following example:

*An Augur user wants to bet that a Republican nomination for the US presidency will be named the party's nominee. On 11th April, the candidate had received 30% of the vote necessary for the nomination and ETH was valued at \$15.*

*A user would make a bet on this future outcome, paying \$0.30 per share in ETH, that it will come to pass. If this user bought 1,000 contracts, and his nominee was victorious, he or she would receive a payout of \$300.*

### How does this work?

#### STEP 1:

An Augur user submits a bid or ask order. Orders are executed if another trader will match or offer better terms. Augur's middleware handles the various serialization, networking and formatting tasks required to communicate the order from the web application to Augur's ethereum smart contract, and sends a success/failure confirmation back to the user interface.

#### STEP 2:

The market reaches its end date. Augur users collectively report what happened to the blockchain using a "commit and reveal" encryption scheme that keeps reporters from knowing how others voted.

#### STEP 3:

Traders in possession of prediction market shares receive (or don't receive) automated payouts according to the outcome determined by Augur's reporting system. This functionality is facilitated through Augur's middleware and smart contract system on the ethereum blockchain.

## Startups

Though in its early stages, the first wave of ethereum projects is being observed with interest by venture capital firms with an expertise in the blockchain domain, with some receiving seed-level investments to develop more mature business strategies.

While a positive indicator, it remains to be seen what role venture firms will play in the development of the ethereum startup ecosystem as whole, given that the platform was meant to encourage the launch of communal projects incorporating the new governance structures its design and technology make possible.

One early trend is that more traditional startups in the ecosystem are seeking to position their platforms as enablers of ethereum DAOs and decentralized projects either through ancillary services or specialized technologies. This differs from the historic rollout of the bitcoin network, as many startups sought to develop key infrastructure (exchanges and wallets) intended to be used directly by consumers.

Here are some of the more notable companies to yet emerge:

<b>Akasha</b>	>
<b>Backfeed</b>	>
<b>BlockApps</b>	>
<b>Colony</b>	>
<b>ConsenSys</b>	>
<b>Ether.camp</b>	>
<b>Ethcore</b>	>
<b>Otonomous</b>	>
<b>Plex.ai</b>	>
<b>Provenance</b>	>
<b>Slock.it</b>	>
<b>String</b>	>



## Akasha

**Headquarters:** Zug, Switzerland

**Venture Funding:** N/A

**Investors:** N/A

**Number of Employees:** 3

Founded by Bitcoin Magazine and ethereum co-founder Mihai Alisie, the Akasha Project is using ethereum and the Inter-Planetary File System (IPFS) to explore how blockchain-based systems could play a role in eliminating censorship on the Internet.

The project is envisioned as a decentralized version of the blogging platform in which users would publish, vote for and share content distributed on a large, distributed network of servers. A beta version is expected to be launched in Q3 or Q4 2016.

### **Notable milestones:**

MAY 2016 – Akasha unveils its product on World Press Freedom Day, opening signups for its alpha release.

## Backfeed

**Headquarters:** Tel Aviv, Israel

**Venture Funding:** N/A

**Investors:** N/A

**Number of Employees:** 8

Backfeed seeks to launch a “social operating system for decentralized organizations” on top of the ethereum network. Its core product is a unique consensus protocol that envisions how bitcoin’s mining system might be reimaged to empower and incentivize a new wave of collaborative, digital projects.

### **Notable milestones:**

MAY 2015 – Backfeed is founded by Matan Field, the founder of decentralized, blockchain-based ridesharing startup La’Zooz.

JANUARY 2016 – Backfeed Magazine officially launches. The proof-of-concept is designed to implement aspects of the startup’s governance and incentivization tools for content curation.

## BlockApps

**Headquarters:** New York, USA

**Venture Funding:** Seed (Undisclosed)

**Investors:** Undisclosed

**Number of Employees:** 12

BlockApps aims to enable enterprise businesses to launch private, consortium or public blockchain applications through a full-stack blockchain infrastructure solution.

The company's signature offerings are STRATO, a single-node blockchain instance that uses a RESTful API to serve as a developer sandbox for ethereum applications, and Bloc, a web application software development kit that supports ethereum smart contracts.

### Notable milestones:

FEBRUARY 2016 – BlockApps is named the first public partner of open-source tech giant Red Hat, joining its OpenShift Blockchain Initiative.

MARCH 2016 – BlockApps becomes the first “certified offering” on Microsoft's Blockchain-as-a-Service (BaaS) offering in its Azure cloud computing platform.

## Colony

**Headquarters:** London, UK

**Venture Funding:** £150k seed

**Investors:** Undisclosed

**Number of Employees:** 8

Positioned as a platform for online freelancers, Colony intends to disrupt the \$4.4bn freelancer market currently occupied by talent marketplaces such as Upwork Global.

Colony foresees a future wherein more skilled professionals operate as freelancers without full-time benefits, and it seeks to help these individuals better leverage their reputation as a means to build value in their work and earn more frequent employment.

### Notable milestones:

MAY 2016 – Colony is awarded the \$10,000 grand prize in a startup competition at CoinDesk's flagship conference, Consensus.

## ConsenSys

**Headquarters:** New York, USA

**Venture Funding:** N/A

**Investors:** N/A

**Number of Employees:** 80+

Founded by ethereum co-founder Joseph Lubin, Consensus Systems (ConsenSys) is a “decentralized applications studio” that offers developer tools for users seeking to launch applications on the Ethereum network.

Employing a unique hub-and-spoke business model, ConsenSys functions as a collaborative collective of entrepreneurs seeking to design and commercialize their works.

The organization has also created core technology used by the community. This includes the Haskell and Java ethereum clients, the Truffle JavaScript-based development framework for ethereum, a lightweight wallet and a persona management system.

### Notable milestones:

OCTOBER 2015 – ConsenSys is the first technology to be included as part of Microsoft’s BaaS offering, a development sandbox for enterprise businesses.

APRIL 2016 – The startup partners with green energy startup LO3 on an effort called TransActive Grid that facilitates the sale of renewable energy.

APRIL 2016 – Insurance giant John Hancock begins working with ConsenSys on blockchain proofs-of-concept.

## Ether.camp

**Headquarters:** New York, USA

**Venture Funding:** Seed (Undisclosed)

**Investors:** Undisclosed

**Number of Employees:** 1-10

A company that grew out of ConsenSys’ hub-and-spoke development model, Ether.camp primarily offers technology tools, including an integrated development environment that serves as a sandbox for developers.

On the technology front, Ether.camp provides a studio for smart contract prototyping as well as a Java implementation of the ethereum protocol.

Elsewhere, Ether.camp provides network transparency tools similar to that of bitcoin industry startup Blockchain, enabling users to both gain insight into publicly available data about ethereum and track contracts on the network.

### Notable milestones:

OCTOBER 2015 – Ether.camp is among the first projects to have its technology made available in Microsoft Azure.

## Ethcore

**Headquarters:** Mittweida, Germany

**Venture Funding:** \$750k

**Investors:** Blockchain Capital,  
Fenbushi Capital

**Number of Employees:** 11-50

Led by ethereum co-founder and former project CTO Gavin Wood, Ethcore develops software solutions for enterprise businesses and financial institutions that want to leverage the network's technology as well as the firm's subject expertise.

The startup offers a premium ethereum client called Parity, which processes blocks on the network performing tasks including database population, EVM code execution, proof-of-work verification, receipt verification and transaction signature checking.

It further intends to embark on the creation of application-level libraries for developers, while adding IoT features to its Parity roadmap.

### Notable milestones:

JANUARY 2016 – Ethcore begins working with French bank BNP Paribas to explore use cases of blockchain technology.

APRIL 2016 – Ethcore releases version 1.0 of its Parity client, the first component of its blockchain technology suite.

## Otonomous

**Headquarters:** Singapore

**Venture Funding:** Undisclosed

**Investors:** Undisclosed

**Number of Employees:** 12+

Otonomous is focused on enabling startups to form and manage their firms on the blockchain in ways that model traditional corporate structures.

Startups will be able to use Otonomous' tools to set up a company, allocate tokens representing ownership, create vesting schedules and track and manage the distribution of those shares via cap table management software connected to the blockchain.

### Notable milestones:

N/A – Otonomous is still in stealth mode with plans to launch publicly in the coming months.



## Plex.ai

**Headquarters:** Ontario, Canada

**Venture Funding:** N/A

**Investors:** N/A

**Number of Employees:** 4

Co-founded by a former Deloitte consultant, Plex.ai aims to use artificial intelligence, ethereum and machine learning to create a platform that would enable auto insurance providers to obtain remote driving data from customers.

A stealth-mode company, Plex.ai has been accepted into Velocity, a startup incubator led by the University of Waterloo.

### **Notable milestones:**

N/A – Plex.ai is still in stealth mode.

## Provenance

**Headquarters:** London, UK

**Venture Funding:** \$65k+

**Investors:** Everledger

**Number of Employees:** 5

Supply chain startup Provenance is leveraging the ethereum blockchain to deliver services that aim to provide transparency and visibility into the global shipment of goods.

Provenance uses ethereum as a way to authenticate data in instances where trust in that information could be a key driver of value. For example, it is working with the local fishing industry in Indonesia to create ways for these professionals to authenticate catches.

### **Notable milestones:**

SEPTEMBER 2015 – Everledger wins €30,000 at a FinTech competition hosted by BBVA.

DECEMBER 2015 – Allianz France, a subsidiary of insurance giant Allianz, begins working with Everledger on blockchain proofs-of-concept and use cases.

## Slock.it

**Headquarters:** Mittweida, Germany

**Venture Funding:** N/A

**Investors:** N/A

**Number of Employees:** 1-10

Focused on using ethereum's technology for applications in the Internet of Things (IoT) and founded by Stephan Tual, Slock.it has created a compact computer optimized to participate on the ethereum network, which can interact with a hardware lock.

The aim is to secure physical assets with the lock, ranging from apartments to bikes.

These will then be rentable by their owners to others via the ethereum blockchain. The computer will register the unlocking and locking of the device on the blockchain, triggering payments between participants.

### Notable milestones:

MARCH 2016 – Slock.it partnered with German power company RWE to envision how ethereum smart contracts could be used to both authenticate and manage the payment process for users of electric vehicle charging stations.

MAY 2016 – Slock.it played a role in authoring the code for the fast-growing distributed autonomous organization, TheDAO.

## String (formerly Koinify)

**Headquarters:** Mountain View, California, USA

**Venture Funding:** \$1.4m

**Investors:** Amino Capital, FBS Capital, IDG Capital Partners, Zhen Fund

**Number of Employees:** 11-50

String is seeking to lead the development of an “alternative financial environment” for ethereum application developers that functions as an alternative to major stock exchanges.

Formerly a startup centered on helping decentralized applications raise money through token sales, String pivoted in 2015 to focus on developing code for a synthetic asset market called the Mirror Asset System, which will use smart contracts and blockchain technology to replicate existing financial assets.

### Notable milestones:

SEPTEMBER 2014 – Koinify raises \$1.2m for decentralized application crowdfunding platform, with investors including IDG and zPark Ventures.

NOVEMBER 2015 – Koinify relaunches as String, detailing its vision and announcing its advisory board.

# Conclusion

**So, how do we parse this idea of a world computer? Is this the future of contracts, digital applications, value exchange and more?**

Bitcoin exposed the world to the idea of trustless data structures, and gave us a glimpse of the future. Surely, ethereum is giving us a better view. Iteratively, it is a fantastic step forward, but it isn't there yet.

The switch from PoW to PoS could come to greatly reduce the computational load of the network, and distributed storage solutions and state channels could greatly expand its capabilities. But right now, these solutions are still in development, and as such, they may not come to their intended fruition.

This realization should not be seen as a fatal issue, as there is no need to have every interaction in the world take place on the ethereum protocol. But, in its current state, ethereum is not a scalable platform.

Ethereum today may be best viewed as a proof-of-concept that was designed to serve as a stepping stone for what would become an ethereum 2.0 with the release of Serenity. At that point, we may finally see a scalable iteration of the network.

This release remains, in all likelihood, a ways off. In the meantime, there is a significant risk of fatigue in the community as there is already a great deal of focus, and a seeming desire, for ethereum to be used as a fully deployable and scalable network today.

At time of publication, ethereum's development community is effectively at a crossroads. With millions in investor funds compromised in The DAO attack, all eyes remain on its core developers.

The team faces a difficult choice. Whether they decide to negate the valid transaction in a bid to rescue investor funds or choose not to intervene on philosophical grounds, the decision will have ramifications. Ethereum is now writing blockchain case law in real time.

Can ethereum navigate this potential fork in the road and the challenges ahead? While the future is always uncertain, the real question is whether, in ethereum's case, it will be truly ethereal, or if its technology can adapt to the needs of the real world.

# Appendix: Getting Started With Ethereum

To begin using ethereum, users need a piece of software – called a **client** – that can run contracts and communicate with other computers using those protocols.

There are multiple clients written in different languages, which helps to broaden support for the network. Having multiple teams implement the protocols also helps to make them more reliable and robust.

There are several clients that run on top of the wallet, offering additional features, the more notable of which are outlined below:

- **Cpp ethereum.** Led by Christian Reitwiessner, cpp ethereum is a C++ client.
- **Ethereumjs-lib.** An implementation in javascript.
- **Ethereum (J).** A Java version.
- **EthereumH.** A version written in the Haskell programming language.
- **Go-ethereum.** Written in Google's Go language, this is currently the most popular ethereum client. Commonly called "geth", it includes a mining component while allowing users the ability to create contracts and transfer funds between addresses
- **Parity.** A low-footprint version written in a language called Rust, spawned by Mozilla.
- **Pyethapp.** A Python implementation that includes mining and virtual machine capabilities. This has been subcontracted to a team at Brainbot, led by Heiko Hees.
- **Ruby-Ethereum.** A version written in the Ruby web application programming language.



### Writing Solidity

Most contracts written in Solidity will be intended (like gears in an engine) to interact with other contracts, meaning coders must be very careful in their constructions so that objects, or scripts, interact the correct way.

In a sense, ethereum contracts can be perceived as needing to operate as automated production facilities. Components need to move along assembly lines linearly, and lots of other functions can be employed to cut, bend, fill, punch, paint, label and otherwise work on building the intended final product.

Every part of the assembly line has to be in sync, or the final product will not work as intended.

If you are not comfortable working in command line, there is a simple tutorial on <http://ethereum.org/token> that will help you create a token. There are also step-by-step instructions in how to implement a contract via that token.

Once you're more familiar, Christian Reitwiessner has elaborated on the developing social ethics and best practices of smart contract development in [public presentations](#).

### How to Learn Solidity

To begin with, having a solid basis in JavaScript will be extremely helpful in learning Solidity.

But, whether you know JavaScript or not, here is a list of resources you can use to learn more about coding in Solidity:

**Solidity Documentation** – The most comprehensive resource for Solidity, this tutorial is geared toward people familiar with programming, but who may not have experience with ethereum or blockchain technology in general.

**Ether.fund** – This online resource maintains a list of example Solidity contracts that can be a useful resource for developing your own contracts or understanding how different methods of creating contracts work.

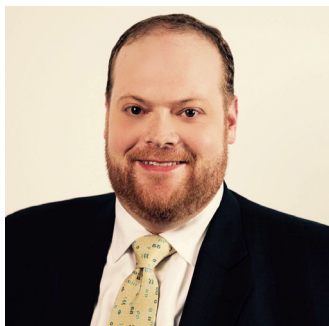
**Ethereum Github Wiki** – A community-maintained wiki for the technology, this resource contains a list of resources for dapp developers that will be most useful for those with some programming background. These include tools, code examples, development environments and technical references.

**ConsenSys** – If you are new to programming and the ethereum blockchain, you might find this “Intro to Programming Smart Contracts” by ethereum startup ConsenSys useful. It introduces basic concepts in dapp development, and walks the reader through one possible dapp development workflow.

**Ledger Labs** – Another “Intro to Dapp Development” tutorial is available from Canada-based blockchain consultancy Ledger Labs. While a work in progress, it currently walks the reader through installing Geth, running a local node, a basic contract design and a more advanced auction contract example.

If you are completely new to programming, you might find that you need to first learn the basic concepts involved in any coding.

Online interactive platform Codecademy has **free interactive tutorials** that will teach you the basics of JavaScript, the language on which Solidity is based. While the details and syntax will be different, many of the basic concepts you will learn are applicable in Solidity.



**Authored by Jacob Dienelt.** Dienelt is a founding partner of Immutable Data Partners, a blockchain technology and product concept consultancy. He also served as the blockchain architect at bitcoin exchange itBit and is an eight-year veteran of Morgan Stanley.



**Edited by Pete Rizzo.** Editor for CoinDesk, Rizzo oversees production of both the daily news website, CoinDesk.com, and CoinDesk Research, its monthly report offerings. He was previously an Editor at payments news source PYMNTS.com.



**Data by Adam Hayes.** Hayes is co-founder and CEO of ChainLink, a blockchain-based startup conferring tamper-proof certificates of title and authenticity to property and luxury items. He has written a number of research papers on bitcoin, cryptocurrency valuation and blockchain technology.





CoinDesk, LLC  
636 Avenue of the Americas  
New York, NY  
10011  
[www.coindesk.com](http://www.coindesk.com)