Ethereum: Platform Review

Opportunities and Challenges for Private and Consortium Blockchains

Executive Summary

Kathleen Breitman and Richard Gendal Brown

Executive Summary

Kathleen Breitman, Senior Strategy Associate

Ethereum is a blockchain with a built-in cryptocurrency proposed by Vitalik Buterin in 2013. Ethereum software powers a distributed network where users can transact in ethers but also enter into “smart contracts.” Smart contracts are short computer programs which encode business logic and are automatically enforced by the network’s consensus algorithm. Using such contracts, developers can even build distributed applications such as prediction markets, mutual insurance, or crowdfunding. Currently, Ethereum and related derivatives (Eris, Clearmatics) are popular with blockchain enthusiasts and developers. There are many attractive features of the platform: Ethereum natively supports Turing-complete smart contracts and the scripting language is user-friendly.

Given the widespread interest in Ethereum, R3 commissioned Vitalik Buterin to write an exploration of different trade-offs in the Ethereum design, upcoming technical developments, and its applications in private systems. For developers using Ethereum or Ethereum-derived solutions, Buterin’s commentary will serve as a useful reference. For business practitioners, the paper will provide further grounding in the complex nature of distributed systems.

In R3’s assessment of cryptographic currencies and ledgers, we evaluate each solution in terms of technical scalability, privacy, and the ability to introduce automation into business processes through smart contracts. In the next two pages, we summarize Buterin’s writing on these topics, though we encourage all interested parties interested in accessible technical explanations to read his paper in full.

Scalability

The specific kind of decentralization blockchains provide is unique. They don’t facilitate a distributed system in the sense that it is "split up between different parties," but are instead replicated: every single node on the network processes every transaction and maintains the entire state. While blockchains gain some of the benefits of decentralization, their scalability is much worse than traditional systems. In fact, no matter how many nodes a blockchain has, its transaction processing capacity can never exceed that of a single node.

Scalability is a key outstanding question for distributed ledger solutions in capital markets. The utilities and exchanges serving modern market infrastructure support systems with high throughput. To take extreme examples, the DTCC performs 100 million operations per day (~1200 per second) while the Shanghai Stock Exchange’s trading system has a peak order capacity of over 80,000 transactions per second. Prima facie, this limits the utility derived from blockchain-based solutions such as Ethereum.

To combat native deficiencies of blockchain architecture, Ethereum’s development team is currently exploring two solutions to enhance the system’s scalability: sharding and state channels. Sharding refers to a technique similar to database sharding, where different parts of the state are stored by

different nodes and transactions are directed to different nodes depending on which shards they affect so that they can be processed in parallel. State channels conduct most transactions off the blockchain between parties directly, using the blockchain only as a kind of final arbiter in case of disputes.

Privacy

Even though having transactions be processed on a system maintained and audited by many parties is great for authenticity, it poses high costs in terms of privacy. In capital markets, it's non-negotiable that the open interests of all parties are adequately concealed. Truly effective solutions to blockchain privacy generally come in two flavors: low-tech solutions relying simply on clever combinations of cryptographic hashes and signatures, and high-tech solutions using advanced cryptographic techniques to obfuscate the meaning of a transaction. Both strategies have their own limitations: low-tech approaches tend to be simpler to implement, but usually offer privacy gains that are only statistical in nature, whereas high-tech approaches have greater promise but may rely on less well-tested security assumptions.

While there's no perfect privacy solution, low or high-tech, that has been generally accepted by developers working on distributed ledgers, there are some tricks that allow for high degrees of privacy for certain use cases. The solutions here entail advanced cryptography such as ring signatures, additively homomorphic encryption, zero-knowledge proofs and secret sharing. Privacy solutions can also be highly customized (e.g., divulging specific information to specific parties under specific circumstances). The potential for widespread use of highly customized systems would satisfy rigorous business requirements while preserving a very high degree of privacy.

Buterin concludes: "It is on the Ethereum project's roadmap to implement these schemes and make it as easy as possible to develop privacy-preserving applications on Ethereum using these technologies, and in the case of state channels there are projects implementing them already. However, private-chain Ethereum users are free to race ahead and implement whatever schemes from the above list are desired as precompiled contracts on their own schedule."

Smart Contracts

Many companies leveraging Ethereum's code base for development are attracted to its native support of smart contracts and ease of use. Beyond accessibility, Ethereum's architecture lends itself to experimentation with smart contracts. For instance, if a private company recording their shares on a blockchain would like to incorporate complex voting rules for shareholders, an Ethereum-derived solution can add this business logic without changing the base layer or any other part of the system. Initially constructing simple applications (e.g., payments) on top of Ethereum allows for an "on-ramp" to more advanced applications (e.g., financial contracts) on top of the basic asset layer without compromising the base layer of code.

As with most new software, ensuring consistent execution is critical to actualizing widespread adoption. Programmer error and malice are two key threats to systems relying on computer code such as smart contracts. As a low-tech solution to these concerns, Buterin hopes there emerges a

market of professional firms that create standard-form contracts for a variety of use cases which receive a high level of scrutiny from multiple independent auditors. Auditing and standardization can protect against coder error, though in the specific case of error there has also been several decades of active research into very strongly typed programming languages that specifically try to make errors easier to avoid.

A high-tech approach to ensuring correctness in code execution entails formal verification, the science of using computer programs to automatically mathematically prove statements about other computer programs. Currently, Christian Reitwiessner, the lead developer of the Ethereum high-level programming language Solidity, is actively working on integrating the formal proving engine why3 into Solidity.

R3 POV

Richard Gendal Brown, Chief Technical Officer

Ethereum is trying to do something novel and ground-breaking in the distributed ledger space and it is a remarkable achievement. We view Ethereum as a guide for new ways of thinking about distributed systems, as well as an excellent prototyping and simulation platform.

Some potential solutions addressed in the paper warrant scrutiny from a capital markets perspective. For example, if sharding solutions are introduced, it is not obvious that they would mesh well with real-world finance use cases. For example, it's unclear how a sharding method could cleanly support atomic delivery-versus-payment trades with assets that sit in different shards. As with all new technologies, extensive analysis and simulation will be required.

More generally, we believe there is a problem with appropriating a technology designed for one set of requirements and trying to apply it to another. We question the fundamental fit of a technology which has to back-into our consortium's standards, as opposed to having appropriate levels of scalability, privacy and business logic from the start. This is not meant to be an antagonistic remark; it is not a thoroughbred racehorse's fault that it doesn't have an ample trunk and heated wing mirrors. At R3, we continue to explore which aspects of existing platforms can inspire solutions to financial services problems and which ones are not appropriate for the use of our members.