On Public and Private Blockchains

Posted by Vitalik Buterin on August 6, 2015

Over the last year the concept of "private blockchains" has become very popular in the broader blockchain technology discussion. Essentially, instead of having a *fully* public and uncontrolled network and state machine secured by cryptoeconomics (eg. proof of work, proof of stake), it is also possible to create a system where access permissions are more tightly controlled, with rights to modify or even read the blockchain state restricted to a few users, while still maintaining many kinds of partial guarantees of authenticity and decentralization that blockchains provide. Such systems have been a primary focus of interest from financial institutions, and have in part led to a backlash from those who see such developments as either compromising the whole point of decentralization or being a desperate act of dinosaurish middlemen trying to stay relevant (or simply committing the crime of using a blockchain other than Bitcoin (https://www.reddit.com/r/Bitcoin/comments/3bg1af/i_knew_it_would_never_work_the_moment_she_told_me/)). However, for those who are in this fight simply because they want to figure out how to best serve humanity, or even pursue the more modest goal of serving their customers, what are the practical differences between the two styles?

First, what exactly are the options at hand? To summarize, there are generally three categories of blockchain-like database applications:

- **Public blockchains**: a public blockchain is a blockchain that anyone in the world can read, anyone in the world can send transactions to and expect to see them included if they are valid, and anyone in the world can participate in the *consensus process* - the process for determining what blocks get added to the chain and what the current state is. As a substitute for centralized or quasi-centralized trust, public blockchains are secured by cryptoeconomics - the combination of economic incentives and cryptographic verification using mechanisms such as proof of work or proof of stake, following a general principle that the degree to which someone can have an influence in the consensus process is proportional to the quantity of economic resources that they can bring to bear. These blockchains are generally considered to be "fully decentralized".
- **Consortium blockchains**: a consortium blockchain is a blockchain where the consensus process is controlled by a pre-selected set of nodes; for example, one might imagine a consortium of 15 financial institutions, each of which operates a node and of which 10 must sign every block in order for the block to be valid. The right to read the blockchain may be public, or restricted to the participants, and there are also hybrid routes such as the root hashes of the blocks being public together with an API that allows members of the public to make a limited number of queries and get back cryptographic proofs of some parts of the blockchain state. These blockchains may be considered "partially decentralized".
- **Fully private blockchains**: a fully private blockchain is a blockchain where write permissions are kept centralized to one organization. Read permissions may be public or restricted to an arbitrary extent. Likely applications include database management, auditing, etc internal to a single company, and so public readability may not be necessary in many cases at all, though in other cases public auditability is desired.

In general, so far there has been little emphasis on the distinction between consortium blockchains and fully private blockchains, although it is important: the former provides a hybrid between the "low-trust" provided by public blockchains and the "single highly-trusted entity" model of private blockchains, whereas the latter can be more accurately described as a traditional centralized system with a degree of cryptographic auditability attached. However, to some degree there is good reason for the focus on consortium over private: the fundamental value of blockchains in a fully private context, aside from the replicated state machine functionality, is cryptographic authentication, and there is no reason to believe that the optimal format of such authentication provision should consist of a series of hash-linked data packets containing Merkle tree roots; generalized zero knowledge proof technology (https://github.com/scipr-lab/libsnark) provides a much broader array of exciting possibilities about the kinds of cryptographic assurances that applications can provide their users. In general, I would even argue that generalized zero-knowledge-proofs are, in the corporate financial world, greatly *underhyped* compared to private blockchains.

For now, I will thus focus on the simpler “private versus public” blockchain discussion. In general, the idea that there is “one true way” to be blockchaining is completely wrong headed, and both categories have their own advantages and disadvantages.

First, private blockchains. Compared to public blockchains, they have a number of advantages:

1. The consortium or company running a private blockchain can easily, if desired, change the rules of a blockchain, revert transactions, modify balances, etc. In some cases, eg. national land registries, this functionality is necessary; there is no way a system would be allowed to exist where Dread Pirate Roberts can have legal ownership rights over a plainly visible piece of land, and so an attempt to create a government-uncontrollable land registry would in practice quickly devolve into one that is not recognized by the government itself. Of course, one can argue that one can do this on a public blockchain by giving the government a backdoor key to a contract; the counter-argument to that is that such an approach is essentially a Rube Goldbergian alternative to the more efficient route of having a private blockchain, although there is in turn a partial counter-argument to that that I will describe later.
2. The validators are known, so any risk of a 51% attack arising from some miner collusion in China does not apply.
3. Transactions are cheaper, since they only need to be verified by a few nodes that can be trusted to have very high processing power, and do not need to be verified by ten thousand laptops. This is a hugely important concern right now, as public blockchains tend to have transaction fees exceeding $0.01 per tx, but it is important to note that it may change in the long term with scalable blockchain technology (https://github.com/vbuterin/scalability_paper/raw/master/scalability.pdf) that promises to bring public-blockchain costs down to within one or two orders of magnitude of an optimally efficient private blockchain system
4. Nodes can be trusted to be very well-connected, and faults can quickly be fixed by manual intervention, allowing the use of consensus algorithms which offer finality after much shorter block times. Improvements in public blockchain technology, such as Ethereum 1.0's uncle concept and later proof of stake, can bring public blockchains much closer to the "instant confirmation" ideal (eg. offering total finality after 15 seconds, rather than 99.9999% finality after two hours as does Bitcoin), but even still private blockchains will always be faster and the latency difference will never disappear as unfortunately the speed of light does not increase by 2x every two years by Moore's law.
5. If read permissions are restricted, private blockchains can provide a greater level of, well, privacy.

Given all of this, it may seem like private blockchains are unquestionably a better choice for institutions. However, even in an institutional context, public blockchains still have a lot of value, and in fact this value lies to a substantial degree in the philosophical virtues that advocates of public blockchains have been promoting all along, among the chief of which are freedom, neutrality and openness. The advantages of public blockchains generally fall into two major categories:

1. Public blockchains provide a way to protect the users of an application from the developers, establishing that there are certain things that even the developers of an application have no authority to do. From a naive standpoint, it may be hard to understand why an application developer would want to voluntarily give up power and hamstring themselves. However, more advanced economic analysis provides two reasons why, in Thomas Schelling's words, weakness can be a strength. First, if you explicitly make it harder or impossible for yourself to do certain things, then others will be more likely to trust you and engage in interactions with you, as they are confident that those things are less likely to happen to them. Second, if you personally are being coerced or pressured by another entity, then saying "I have no power to do this even if I wanted to" is an important bargaining chip, as it discourages that entity from trying to compel you to do it. A major category of pressure or coercion that application developers are at risk of is that by governments, so "censorship resistance" ties strongly into this kind of argument.
2. Public blockchains are open, and therefore are likely to be used by very many entities and gain some network effects. To give a particular example, consider the case of domain name escrow. Currently, if A wants to sell a domain to B, there is the standard counterparty risk problem that needs to be resolved: if A sends first, B may not send the money, and if B sends first then A might not send the domain. To solve this problem, we have centralized escrow intermediaries (https://escrow.com/services/domain-name-holding-escrow.aspx), but these charge fees of three to six percent (https://escrow.com/support/fee-calculator.aspx). However, if we have a domain name system on a blockchain, and a currency on *the same blockchain*, then we can cut costs to near-zero with a smart contract: A can send the domain to a program which immediately sends it to the first person to send the program money, and the program is trusted because it runs on a public blockchain. Note that in order for this to work efficiently, two completely heterogeneous asset classes from completely different industries must be on the same database - not a situation which can easily happen with private ledgers. Another similar example in this category is land registries and

title insurance, although it is important to note that another route to interoperability is to have a private chain that the public chain can verify, btcrelay-style (https://github.com/ethereum/btcrelay), and perform transactions cross-chain.

In some cases, these advantages are unneeded, but in others they are quite powerful - powerful enough to be worth 3x longer confirmation times and paying $0.03 for a transaction (or, once scalability technology comes into play, $0.0003 for a transaction). Note that by creating privately administered smart contracts on public blockchains, or cross-chain exchange layers between public and private blockchains, one can achieve many kinds of hybrid combinations of these properties. The solution that is optimal for a particular industry depends very heavily on what your exact industry is. In some cases, public is clearly better; in others, some degree of private control is simply necessary. As is often the case in the real world, it depends.

← **PREVIOUS POST (/2015/08/04/ETHEREUM-PROTOCOL-UPDATE-1/)**

NEXT POST → (/2015/08/07/SECURITY-ALERT-1-WINDOWSALETHZERO-2/)