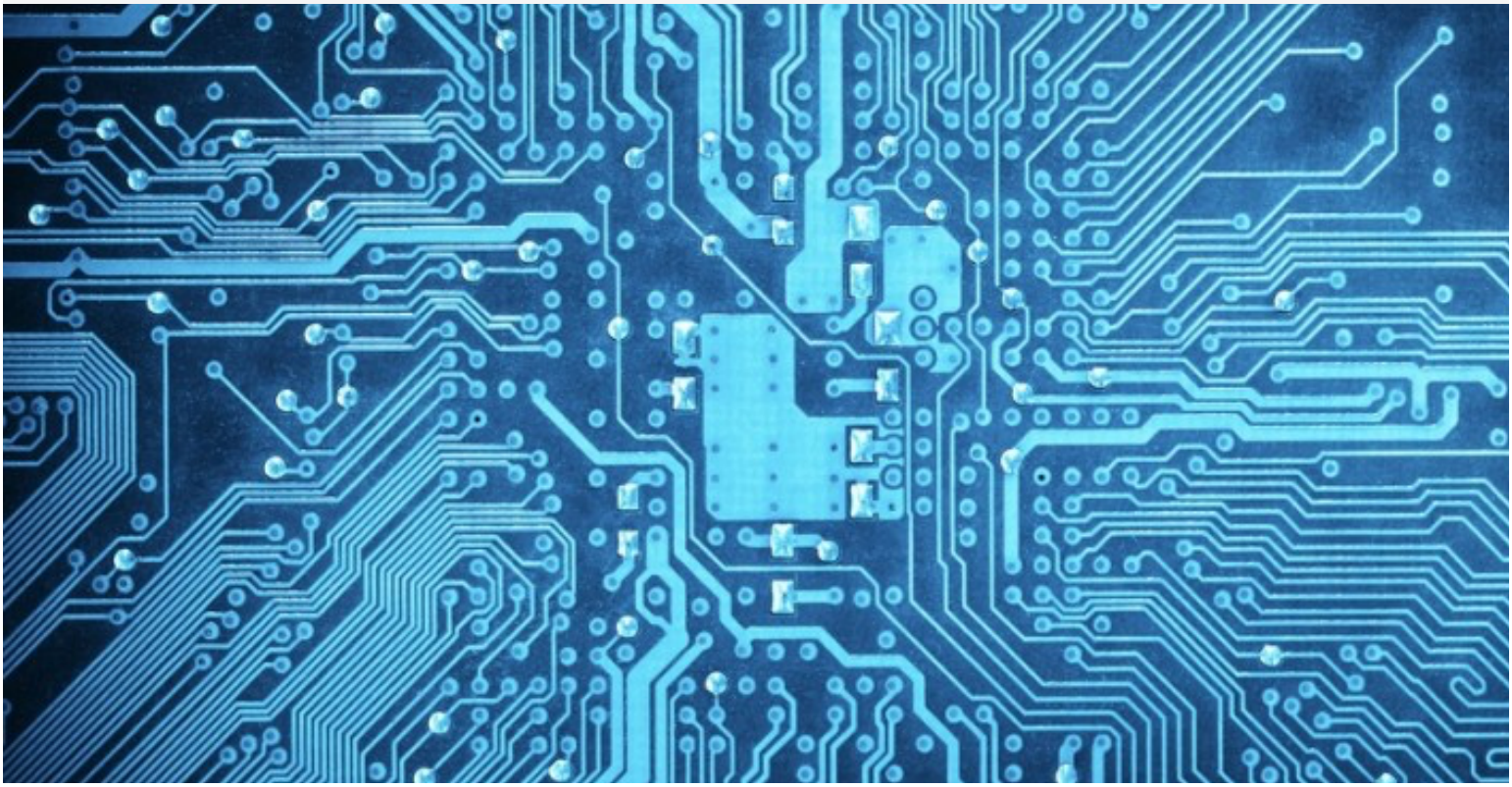

Models For Scaling Trustless Computation

Published on February 26, 2018

Kyle Samani | ✓ Following
Managing Partner, Multicoin Capital - I do not check LinkedIn Messag...
35 articles

Each of the major smart contract platforms is making a unique set of trade-offs. These trade-offs are not simply the presence or absence of specific features, but rather represent fundamentally different views of what trustless computation means.

This essay aims to provide a coherent framework through which one can understand these trade-offs and how they impact some of the predominant narratives in crypto such as digital gold, programmable money, censorship resistance, and permissionless-ness. Some definitions:

Censorship resistance — complete freedom of expression. In more technical terms, the ability to commit any arbitrary record to the blockchain.

Permissionless-ness — the ability to access the network and verify the integrity of the chain without asking a 3rd party for permission.

I will not explore layer 2 scaling solutions such as Lightning, Raiden, and Plasma.

But first, we need to establish context for the term "trustless." Nick Szabo frames trustlessness as an inverse function of technical efficiency. Basically, the less efficient the computer, the more difficult it is to manipulate. The more difficult it is to manipulate, the more you can trust it, therefore making it trustless.

In other words, to paraphrase Szabo, blockchains trade technical efficiency for social scalability.

The ultimate manifestation of crypto is not just a trustless database (or blockchain) in which everyone agrees on its current state — it's trustless general purpose computation, which is a superset of a database. This can be a little bit challenging to grasp. Blockchains enable everyone on the planet to come to consensus about the state of the world. Trustless general purpose computation is a step beyond that. It's the ability to not just know the state of the world but to *prove* that a specific computation was run *correctly*.

Although Bitcoin is technically programmable — enabling trustless general purpose computation through its Script language — in practice Bitcoin is just a trustless database. Although there are some efforts to make Bitcoin more expressive (such as MAST, taproot, scriptless scripts, and RSK) none of these are in production today. To put it bluntly, developers have left Bitcoin for greener pastures.

Ethereum provided the first accessible platform on which developers could run *any* arbitrary computation trustlessly. Today, Ethereum owns the vast majority of the developer mindshare in crypto.

Many have called Ethereum a world computer. While this is technically true, in practice it's not because of two related factors: throughput and cost. Running a computation on Ethereum is on the order of 100,000,000x more expensive than running the same computation on Amazon Web Services (AWS).

The Scalability Trilemma

The challenge of scaling trustless computation can be thought of as a trilemma. The scalability trilemma posits that blockchains in which every node processes every computation and in which every node comes to consensus about the order of those computations can have two of three properties: safety, scalability, and decentralization of block production (DBP).

- DBP can be quantified as the number of block producers.
- Scalability can be quantified as the number of transactions per unit of time that the system can process.
- Safety can be quantified as the cost of mounting a Byzantine attack that affects liveness or transaction ordering. Note that safety does not refer to the integrity of cryptographic signatures, or the ability of a 3rd party to derive a set of private keys from public keys.

What causes a system to achieve one set of trade-offs vs another? A combination of consensus scheme and system architecture. The remainder of this essay will explore these concepts, and lastly, loop in some interesting work in off-chain computation.

Each of the systems below makes a different set of trade-offs in pursuit of scalable trustless computation. Throughout this essay, I'll refer to each leg of the triangle using these numbers:

A Fourth Dimension: Time To Finality (AKA Latency)

Although not explicitly part of the scalability trilemma, there's a fourth dimension to consider in scaling systems for trustless computation: time to finality (TTF), which directly impacts latency. Some systems never guarantee finality, but rather probabilistically approach finality (e.g. Bitcoin). Others offer a finality guarantee after some period of time. Finality is important not only to avoid double-spend attacks, but also because finality guarantees are necessary to enable cross-chain communications. The slower the TTF, the higher the latency in cross-chain communications.

There's not a visually clean way to represent TTF on a two-dimensional triangle. As such, I'll use dark background colors to indicate fast TTF and light background colors to represent slower TTF.

Leg 1: Permissionless Proof-of-Work (Bitcoin, Ethereum 1.0, others)

Prior to Bitcoin, all digital cash systems suffered from the same fundamental flaw: users had to trust a 3rd party to administer the system. This 3rd party could censor transactions. Designing a system in which anyone could verify the integrity of the chain and in which no single 3rd party could censor transactions was the primary design goal of Bitcoin. Proof-of-work (PoW) consensus made a censorship-resistant, permissionless ledger possible.

The downside of optimizing so strongly in favor of censorship resistance is that modern PoW systems don't scale without centralizing block production. This fundamental tradeoff ultimately resulted in the Bitcoin Cash fork, which, relative to Bitcoin, will centralize block production (though there are debates about this practice).

Of the consensus models presented in this essay, PoW is the most permissionless. Literally anyone with a computer and an Internet connection can begin validating transactions and mining. This theoretically enables maximal DBP.

In practice, all *permissionless* PoW systems centralize block production. We know this empirically. Mining for all major PoW-based blockchains centralizes due to economies of scale in mining operations. Today, no more than 20 organizations/pools control the vast majority of mining power in both the Bitcoin and Ethereum blockchains. Given our admittedly limited history, it seems likely this centralization of block production will always occur for both ASIC and GPU-based mining algorithms.

PoW systems suffer from slow TTF. They do not, by design, ever guarantee finality. Instead, as new blocks are added to PoW-based chains, older transactions become exponentially more likely to be final. This is why many don't consider Bitcoin transactions "final" until there have been six additional blocks confirming the transaction. Given 10-minute blocks, it can take an hour or even longer to finalize a Bitcoin transaction. At that point, the probability of a chain reorganization is close enough to 0 to consider the transaction final.

In practice, because of the economies of scale of PoW mining, PoW chains fall in the lower right-hand corner of the triangle:

Braided PoW (Kadena)

Kadena is the only system I'm aware of that's attempting to solve the scalability trilemma using a PoW scheme. Kadena accomplishes this by creating a "braid" of chains that it calls Chainweb. In Chainweb, each chain must, in addition to validating transactions in its own chain, validate block headers of some number of pre-specified chains in order to produce a new block.

To relay messages and value across chains, users need to submit Merkle proofs asserting the state of one chain to other chains in the Chainweb. Because not all chains are directly connected, users may have to "hop" a few times to relay a message from one chain to another.

At first glance, this looks like Ethereum's proposal for sharding (more on that below). However, whereas Ethereum's sharding implementation separates transaction collation and validation into spoke chains and a single hub chain respectively, Kadena doesn't unbundle transaction verification and consensus. In Chainweb, every chain maintains its own consensus. Kadena presents a fundamentally unique approach.

Kadena visualizes ChainWeb as follows:

One of unique traits of ChainWeb is that scaling the system explicitly *increases* security. Why? Because it becomes harder to mount 51% attacks across an increasing number of chains, each of which references blocks from every other chain. In this model, undoing a transaction on one chain requires undoing transactions on every other chain in the system. Given how naturally intertwined the chains are, this becomes exponentially more difficult as the number of chains grows. This stands in contrast to blockchains on leg 3 of the triangle, in which the safety of the system in aggregate doesn't substantially increase as the number of chains grows.

This design basically solves the scalability trilemma. But there is one major trade off: TTF and cross-chain latency.

Individual chains could be many hops from one another. Kadena aims to offer <1 minute TTF in a PoW scheme. Sending messages across the entire Chainweb will take a few hops, requiring a few minutes.

Despite this limitation, Kadena presents a real solution to solving the scalability trilemma that's built on a *proven* consensus model: PoW. Although braiding adds a new dynamics to the system as a whole, each chain is a PoW chain. PoW has undergone much more scrutiny than proof-of-stake (PoS). It takes years to demonstrate the safety of a consensus scheme in the real world. Because of this, there is real value and opportunity for Kadena in building a scalable system on PoW consensus.

On the triangle, Kadena falls here:

Proof-of-Stake

The schemes presented above were PoW-based. The schemes below are PoS schemes.

In practice, *all* PoS systems centralize block production relative to purely permissionless PoW schemes *by design*. This is due to the intrinsic trade-off between number of block producers and throughput. (This excellent essay by Vitalik Buterin details the tradeoffs.)

However, this doesn't mean that all PoS schemes centralize block production equally. As we'll see, there's a wide range of DBP that's achievable using PoS schemes.

PoS schemes are far less battle-tested than PoW schemes in real-world settings. For example, the first PoS implementation, Peercoin, faced nothing-at-stake attacks, among others. As such, PoS schemes should be considered fundamentally riskier.

Leg 2: Centralizing Block Production (EOS, Cardano, NEO, others)

Delegated proof-of-stake (DPoS) schemes embrace centralization by recognizing that PoW schemes naturally centralize due to economies of scale in mining. Given this practical reality, Dan Larimerinvented DPoS, which embraces the fact that blockchains naturally centralize anyways, and uses that to achieve scalability.

For example, EOS, Larimer's most recent effort, publicly boasts that the system will only have 21 block producers at a time. It's expected that, in time, it will only be possible to run an EOS node in a datacenter.

By limiting the number of block producers, it can be assumed that each block producer has more resources. Additionally, reducing the number of block producers reduces latency in Byzantine Fault Tolerant (BFT) algorithms, which typically require n^2 messages to achieve consensus. Reducing DBP explicitly increases scalability

DPoS based chains not only offer scalability but fast TTF, and therefore, low latency. EOS specifically is aiming for .5-second block times, which is simply not feasible in other consensus schemes. This is a major advantage for DPoS systems. Many applications require low latency and high throughput.

Take, for example, decentralized exchanges. This is becoming painfully obvious as the 0x ecosystemmatures on Ethereum. One of the largest problems in 0x today is the rapidly growing number of order collisions, which are a direct result of slow block times and high latency in Ethereum. The 0x team has proposed solutions. It's unclear how effective these will be given the intrinsic limits of the underlying Ethereum blockchain. This wouldn't be a problem at all given the fast blocks, high throughput, and low latency that DPoS systems offer.

Overall, a bet on DPoS is a bet on a few things:

1. There are certain applications that require high throughput and low latency on a *neutral database*.
2. Not all distributed systems need to be *that* distributed so as to able to withstand full-frontal government.

On the triangle, DPoS systems will cluster along the bottom of the triangle. Note that, relative to the prior two triangles, which illustrated systems with slow TTF, DPoS systems offer fast TTF and are therefore presented using a darker background color.

Leg 3: A Universe Of Many Chains (Cosmos, AION, ICON, Ark)

The Cosmos, AION, ICON, and Ark teams believe that there will be hundreds of thousands or even millions of chains. Rather than monolithic chains, such as a Ethereum or EOS, the teams building systems on leg 3 of the triangle believe that different applications should not necessarily share a single set of validators. Instead, they should have unique validator sets.

In the context of the scalability trilemma, each chain on systems in leg 3 comprise less value. In both PoW and PoS schemes, safety increases with value. However, in exchange for that additional risk, each chain gains sovereignty (which may not be necessary or even valuable), scalability, and fast TTF.

Why is sovereignty valuable? This is easiest way to understand is to walk through some examples. As countries adopt chains for administrative purposes, they will want their own validator sets that enforce their social values and not those of some other country or a global commune. Ethereum and Ethereum Classic, for example, should not share validators.

The obvious counter to the case for sovereignty is Ethereum itself. In 2013, Vitalik set out to build Ethereum because he recognized that every app developer in crypto was dealing with the same challenges around mining and consensus. He recognized the opportunity to abstract all of that complexity so that developers can focus on the application layer.

There are some interesting parallels to draw in the history of cloud computing. In the early 2000s, many web-hosts hosted multiple websites on a single server. Whenever one site received too much traffic, the server would crash, taking down other websites that were hosted on the same server.

In many ways, Ethereum is like the early web-hosts. It simply bundled too many things together, creating a system that in aggregate was unstable. When one application broke the system, it broke the system for everyone.

The solution to the web-hosting problem was virtual machines (VM). By isolating every application into a separate VM, a single server could run multiple applications, maximizing hardware utilization and driving down cost while still preserving integrity. In the event of an influx of traffic, a single VM would crash, rather than all the VMs on the server. In time, VMs became portable across physical servers, increasing redundancy and security even further. Combined with systems enabling massive horizontal scalability, VMs became one of the key components of cloud computing. More recently, Docker containers have come to replace VMs, but Docker containers don't fundamentally change this analogy.

If distributed apps must live on separate chains for the reasons outlined above, then there will be a great opportunity for the systems on leg 3 of the triangle.

To support this future, systems on leg 3 need to make it trivially easy to spin up new chains, and for chains to interoperate with one another. Cosmos is making this vision a reality with Ethermint. Ethermint is a 100% open and free. It's a blank-slate template chain that runs the Ethereum Virtual Machine (EVM) on top of Tendermint, the semi-centralized, high throughput, low TTF consensus algorithm pioneered by the Cosmos team. By making it trivially easy for developers to spin up new chains, Cosmos hopes that developers will do just that. AION, ICON, and Ark share the same general vision and are working to provide template-ized ways for developers to spin up chains quickly and easily. (Note that although Wanchain is often listed as a "interoperable chain" solution, it is substantially different than everything else in this section and does not provide the same functionality.)

Note that these systems offer fast TTF. That's because each chain is relatively centralized, allowing for low latency. Despite each chain being relatively centralized, the system in aggregate is rather decentralized because there are so many chains with independent validator sets, each of which may leverage novel consensus mechanisms.

On the triangle, vision of many smaller, interoperable chains can be visualized as follows:

Sharding (Ethereum 2.0, Polkadot)

Vitalik and Gavin Wood have openly discussed sharding Ethereum for years (see here and here).

Each shard is basically a unique chain. The difference between a shard in Ethereum and an independent chain in the Cosmos ecosystem is that in Cosmos, each chain must manage its own consensus (and therefore safety), whereas Ethereum shards do not. In sharding, consensus and therefore safety are pooled across all shards and managed by a Validator Manager Contract in a master shard. In most cases, pooled safety should be better than many chains with lower safety.

No one has put a sharded blockchain into production — yet (Zilliqa claims to; however, their implementation is not full state-sharding).

Sharding solves the scalability trilemma, albeit at the expense of chain sovereignty. Additionally, cross-shard communications are subject to latency. Latency is primarily a function of the consensus algorithm per shard.

Ethereum is going to use Casper to finalize each shard. Although the specific parameters are not yet set, it seems likely that Casper will offer TTF measured in minutes, making sharded Ethereum a high-latency system.

Using a novel consensus algorithm called threshold relay, Dfinity and Algorand aim to offer DBP, safety, more scalability than Ethereum within a single shard (though far less than the scalability offered by DPoS), and fast TTF. If threshold relay works in practice as well it has in test environments, we can expect to see Dfinity and Alogrand shard their chains and leverage the fast TTF and low latency to provide for efficient cross-shard communications.

Polkadot will launch as a sharded network that uses a TBA BFT consensus algorithm that offers fast finality at the expense of DBP (with a similar set of trade-offs as Tendermint). However, Polkadot is likely to launch a sharded network before Ethereum or Dfinity. Unlike Ethereum and Dfinity, Polkadot does not require shards to use a particular state machine, such as the EVM or WebAssembly (WASM). Rather, Polkadot allows each chain to define its own state machine. This would allow protocols such as Stellar, which focus on issuing arbitrary digital assets without any smart contract functionality, or Zcash, which requires a highly optimized state-machine to process SNARK proofs, to move its self-contained consensus systems onto Polkadot.

To summarize the above:

Full state sharding is an unsolved computer science problem. It may be that no one ever pulls it off at scale in a production environment. However, if it's possible, it presents perhaps

the best path to solving the scalability trilemma while offering fast TTF and low latency.

On the triangle, sharded PoS can be visualized here:

Verifiable Off-Chain Computation

What if, instead of introducing inefficiency to achieve trustlessness, a user just asks a single computer to perform a computation, trustlessly? Is there a way to prove that the computation was performed correctly without asking dozens or hundreds of computers to perform the same computation? Can we achieve some guarantee of correctness without the massive technical inefficiency that's intrinsic to blockchains?

Truebit is attempting to fulfill this vision through a prover-verifier game. This will be accomplished through an interactive, prover-response protocol. In the expected case of an unchallenged result, each computation is completed locally with no overhead, a single solver and a handful of verifiers. In the unlikely event of a challenge, both the solver and verifier will need to re-run a computationally intense WASM-based virtual machine to determine the malicious actor.

The interactive verification protocol combines the transparency, security, and immutability of the base chain with the efficiency of offchain compute. As Truebit takes place in interactive rounds, it is also probabilistic, so it will not be effective in environments that require low TTF. Truebit runs on the first "unanimous" consensus mechanism, which requires a minimum of one rational verifier for each task to guarantee safety.

In time, we may see Truebit or a competitor use SNARKs and STARKs to verify the accuracy of any arbitrary computation using non-interactive proofs instead of interactive proofs. If this is possible, Truebit will decrease TTF, increasing the design space for trustless offchain computation. However, SNARKs and STARKs as general purpose, zero-knowledge proofs of arbitrary computation are still highly speculative, unproven, technically inefficient, and may ultimately fail to fulfill their potential.

Open Source, Copying, and Politics

On a long enough time scale, the dominant chains are likely to incorporate the best technologies from smaller, less dominant chains. For example, Ethereum is adopting zkSNARKs, the primary technology enabling private transactions in Zcash. Moreover, Vitalik has stated that he wants move to Ethereum to WASM, which will be

adopted by EOS and Dfinity before Ethereum. Ethereum maximalists offer Plasma as a solution to all of Ethereum's deficiencies because it allows for alternative consensus algorithms and state machines within the Ethereum ecosystem.

Given this and the Smart Contract Network Effect Fallacy, how will chains differentiate?

Politics. Ideologies. Beliefs.

Assuming that no one ever solves the scalability trilemma without any compromises, then different people and businesses will require unique chains that are most suited for the task at hand.

Eventually, the media will come to frame this as a religious debate. Blockchains will become like religions because their believers are missionaries, motivated to spread their gospel and convert others.

Will users choose the chain that's maximally decentralized and therefore maximally censorship-resistant, even at the expenses of performance and higher network fees? Or will they accept lower thresholds for censorship resistance? If so, where's that line?

Censorship-Resistant Digital Gold

Through the framework presented in this essay, we can make an objective assessment on how and why digital gold and programmable money are likely to be independent for the foreseeable future.

The Bitcoin core team has made it a priority to maximize DBP and transaction verification at all costs, even at the expense of scalability and other forms of utility. By making Bitcoin highly inefficient, it will be more resilient. There appears to be no limit to which the Bitcoin core team will trade technical efficiency for social scalability.

You can argue that, even given their beliefs, the Bitcoin core team is misguided. For example, PoS advocates argue that PoS is more secure than PoW because if a malevolent attacker is identified, she can be immediately slashed. This stands in contrast to PoW schemes, in which a malevolent party can continue to attack the network in perpetuity.

We have no idea if, in the long run, PoS schemes will be more resistant to government assault than PoW schemes. There are good arguments to be made both ways. However, we do know that PoW is battle-tested. Given that Bitcoin core's singular priority is maximal censorship resistance, the Bitcoin core developers are making the right decision given what we know empirically.

There are valid arguments that the Bitcoin core view is too extreme, that Bitcoin sacrifices real utility for superfluous censorship resistance, and that the lack of utility will ultimately cause Bitcoin to fall into irrelevance.

The question at hand is, how much DBP is enough DBP? If you assume that the design constraint is *not* "withstand full-frontal, coordinated assault by the U.S., Chinese, and Russian governments" but some lower bar, then the design space for trustless computation broadens.

In the '90s, many thought that the Internet would be the ultimate democratizing force that would dismantle legacy media companies and oppressive governments by connecting people in ways that were never before possible. As it turns out, for the most part, big companies and governments leveraged the Internet to aggregate power and control.

Since all of the systems for trustless computation are permissionless, that means governments can leverage them to their advantage. Perhaps, rather than full-frontal assault, we'll see governments adopt crypto in novel and unforeseen ways that strengthen their grip on society rather than loosen it.

Heterogenous Trustless Computation

For the foreseeable future, all of the models for scaling trustless computation described above will coexist and blossom as the crypto ecosystem undergoes a Cambrian Explosion of experimentation.

There will not be a clear, linear path between here and some steady future state. Rather, things are about to become more heterogeneous and intertwined before they become more homogenous.

For example: Developers may build on Polkadot, only to find that an individual Polkadot relay chain reaches limits. That Polkadot relay chain may connect to other Polkadot relay chains, which may in turn connect to EOS, Ethereum, and Kadena chains through the Cosmos Hub. Each of those chains may in turn be sharded. Various Ethereum shards may contain Plasma chains secured using DPoS and proof of authority (PoA) consensus.

Contracts in each of these major systems will be able to call Truebit for off-chain computation, making everything above even less clear. How much work will be able to be off-loaded to Truebit, and how will that integrate intra and inter-chains?

It's also not clear where and how value will accrue across these systems. There's a case to be made that interoperability chains such as Cosmos will not accrue that much value (perhaps billions vs trillions) if Cosmos's ATOM tokens don't become Menger Goods. Polkadot's

DOT tokens face the same risk.

In time, I expect to see substantial, though not complete convergence. Given the fundamental value of sovereignty for some chains (e.g. those controlled by governments), I expect systems like Cosmos will always have a place, even if another chain is the biggest winner.

Nothing has been decided, and everything is up for grabs.

If you enjoyed this post and think others may as well, please click the "Like" button to help others discover it more easily!

Thanks to Trent McConaghy (Ocean); Peter Czaban (Web3/Polkadot), Jesse Walden (a16z); Will Martino (Kadena), Matt Luongo (Keep), James Prestwich (Integral), Robbie Bent (Truebit), and Zaki Manian (Cosmos) for their input on this essay.

Note: After writing a first draft of this essay, I discovered that Trent McConaghy proposed a similar framework with a different naming scheme a full 20 months before me. Special thanks to Trent for providing input on essay.

Disclosure: Multicoin Capital has long positions on all 3 legs of the triangle.

Kyle Samani

Managing Partner, Multicoin Capital - I do not check LinkedIn Messages

✓ Following

More from Kyle Samani See all 35 articles

New Models For Utility Tokens
Kyle Samani on LinkedIn

Blockchains: A New Social Order
Kyle Samani on LinkedIn

Announcing Multicoin Capital
Kyle Samani on LinkedIn

Tesla Is Not A Car
Kyle Samani on Linke

Messaging