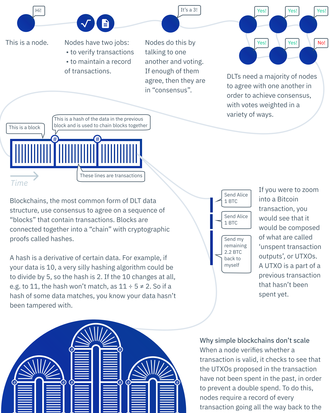
RADIX

Cerberus - How Radix achieves infinite linear scalability while preserving atomic composability

Why Blockchains Can't Scale

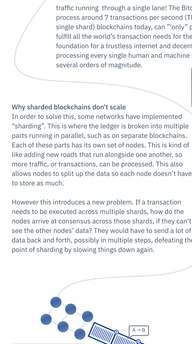


Blockchains, the most common form of DLT data structure, use consensus to agree on a sequence of "blocks" that contain transactions. Blocks are connected together into a "chain" with cryptographic proofs called hashes.

A hash is a derivative of certain data. For example, if your data is 10, a very silly hashing algorithm could be to divide by 5, so the hash is 2. If the 10 changes at all, e.g. to 11, the hash won't match, as 11 ÷ 5 ≠ 2. So if a hash of some data matches, you know your data hasn't been tampered with.



If you were to zoom into a Bitcoin transaction, you would see that it would be composed of what are called 'unspent transaction outputs', or UTXOs. A UTXO is a part of a previous transaction that hasn't been spent yet.



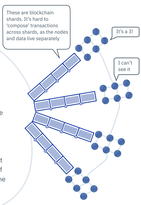
Why *simple* blockchains don't scale
When a node verifies whether a transaction is valid, it checks to see that the UTXOs proposed in the transaction have not been spent in the past, in order to prevent a double spend. To do this, nodes require a record of every transaction going all the way back to the very first transaction – the entire blockchain – otherwise they wouldn't be able to perform a complete search. Over time, this increases the burden on nodes, as they have to store more and more data.

Additionally, every node also has to process every transaction, as there is only a single global ledger state that every node agrees on. For example, if you want to execute an Ethereum smart contract, to come to consensus on that contract execution, every single Ethereum node in the world is asked to process that transaction. This means a blockchain designed in this way cannot process more transactions than a single node can.

This is already causing major bottlenecks. Imagine the entire world's traffic running through a single lane! The Bitcoin blockchain for example can only process around 7 transactions per second (TPS) in total. Even the fastest single lane (or single shard) blockchains today, can "only" process ~65,000 TPS. For a single DLT to fulfill all the world's transaction needs for the entire 21st century: serving as the foundation for a trustless internet and decentralized global financial system by processing every single human and machine transaction alike; 65,000 TPS is too low by several orders of magnitude.

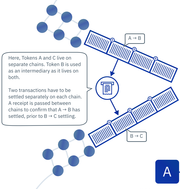
Why sharded blockchains don't scale
In order to solve this, some networks have implemented "sharding". This is where the ledger is broken into multiple parts running in parallel, such as on separate blockchains. Each of these parts has its own set of nodes. This is kind of like adding new roads that run alongside one another, so more traffic, or transactions, can be processed. This also allows nodes to split up the data so each node doesn't have to store as much.

However this introduces a new problem. If a transaction needs to be executed across multiple shards, how do the nodes arrive at consensus across those shards, if they can't see the other nodes' data? They would have to send a lot of data back and forth, possibly in multiple steps, defeating the point of sharding by slowing things down again.



Sharding blockchain breaks atomic composability
To expand on this problem further, sharding the ledger creates the issue of "non-composability".

On a single shard, "composing" a transaction across multiple tokens or decentralized applications (dApps) is easy, as all the nodes have all the data, and can easily come to consensus. Across multiple shards however, nodes lose the ability to come to consensus as they lose sight of each other's data. What was a single transaction on one shard has to be broken into multiple separate transactions across multiple shards.





Without atomic composability, this transaction failed half way through, "trapping" the user with Token B.

This causes major problems - particularly so in decentralized finance ("DeFi"). In DeFi, dApps and assets like tokens have the most utility when they can be connected together – composed – in single transactions. dApps or tokens living on separate shards makes DeFi slow and cumbersome, and opportunities for efficient capital allocation may be lost.

For example, if you want to sell Token A and buy Token C, and the only way to do so is to buy Token B as an intermediary step, you wouldn't want the A → B transaction to settle, unless you knew that the B → C transaction would also settle, otherwise you might get stuck owning B and you can no longer get C.

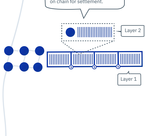
Ideally, you would want to be able to compose a single transaction that settles once across all shards, e.g. A → B → C all in one go, guaranteeing you get C. This is what "cross-shard atomic composability" enables.

Some networks have implemented what's called "state locking" to solve this, which locks states across shards prior to settlement. This would allow you to back out of the A → B transaction if the B → C transaction were to fail. However, this coordination takes time, and allows high frequency arbitrageurs to easily front-run any trades. It's therefore only a partial solution.

How 'Layer 2' solutions solve scalability, but sacrifice decentralization and atomic composability
Another solution is to execute transactions "off-chain", such as on the Bitcoin Lightning Network; or through "rollups" or "sidechains", which are now being used on Ethereum. These are collectively referred to as Layer 2 (L2) scaling solutions because transaction processing is taken off the primary network, the Layer 1 (L1).

Assets, such as tokens, can then be ported between the L1 and L2 networks; either individually, or in batches, which is more efficient.

But L2 solutions create silos of transaction processing which are often more centralized and less secure than the primary network. Additionally, with many different types of L2 solution being implemented, composing a transaction across many different L2s quickly makes things very complicated. L2 solutions thus reduce interoperability and the ability to settle transactions atomically. They thus also break "atomic composability".



Tradeoffs in blockchain system design

As we can see, there appear to be tradeoffs when designing blockchain systems. Those tradeoffs are commonly referred to as the "Blockchain Trilemma", which posits that there is a three-way compromise between:

- scalability;
- decentralization; and
- security.

Improving one tends to come at the expense of the others.

However, it's a bit more nuanced than that:

- decentralization isn't useful at all without security, and so security is an uncompromisable feature; and
- scalability on its own also isn't very useful.

  Imagine "spinning up" 1m Bitcoin blockchains. You'd have a set of systems that could process 7m transactions per second, but they wouldn't be very useful, as those 1m blockchains couldn't interoperate particularly well and you'd have silos of transaction processing and liquidity.

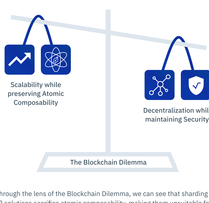
  The opposite of such fracturing is for a system to exhibit "atomic composability", which is the ability to structure a single transaction across any section of the ledger and for all parts of the transaction to settle atomically - either all together or not at all.

  For DeFi to seriously differentiate itself and outcompete the rest of finance, it will require unlimited atomic composability.

  Scalability is therefore only useful when it comes with composability.

The Blockchain Trilemma could therefore be reimagined as the "Blockchain Dilemma", with a more refined two-way compromise between:

- scalability while preserving atomic composability; and
- decentralization while maintaining security.



Through the lens of the Blockchain Dilemma, we can see that sharding or L2 solutions sacrifice atomic composability, making them unsuitable for DeFi.

Current Solutions	Achieves	Sacrifices
Central control of nodes	Increase in scalability	Decentralization
Sharding or L2	Increase in scalability	Atomic Composability

Given the Blockchain Dilemma, a radically different paradigm is needed if DeFi is to ever scale to fulfill the future needs of billions of people.

The version of Cerberus described in this infographic series is scheduled to launch as part of the fully sharded Radix Xi'an release. Please visit www.radixdlt.com for details on the Radix roadmap.

© Copyright Radix Tokens (Jersey) Limited

v1.0 - June 2021