
THE ESSENTIAL GUIDE TO BLOCKCHAIN PLATFORMS
From Bitcoin to Solana to Kaspa
A comparative view of different blockchains from those emerged to emerging
Ross P. Green

The Essential Guide to Blockchain Platforms

From Bitcoin to Solana to Kaspa

Ross P. Green

The Essential Guide to Blockchain Platforms

by Ross P. Green

Dedicated to my daughter, Belle.

Thank you for trying to understand blockchain.

Contents

Preface

Due to the dynamic evolution of blockchain technology and its relatively youthful status, there is a noticeable dearth of literature on emerging blockchains, distinct from the well-documented cases of Bitcoin and Ethereum. Faced with the absence of comprehensive resources on platforms like Solana, Bittensor, or Kaspa, I took it upon myself to compile a singular, cohesive book to disseminate knowledge on these groundbreaking technologies. While YouTube videos can offer valuable insights, the dispersed nature of information often necessitates sifting through numerous sources. This book consolidates data across various blockchain platforms, eliminating the need to navigate disparate videos or peruse individual volumes burdened with extraneous details that may be neither necessary nor memorable.

Acknowledging the vast landscape of over 200 existing blockchains, it is impractical to encompass all of them within the confines of this book. Therefore, I have sought a balance between well-established blockchains and those in the nascent stages of development. The terms "emerged" and "emerging" are used to distinguish between those with significant adoption and those still in the early phases, not widely recognized globally within the blockchain community.

The decision to delve into "emerged" chains is not intended to replicate existing documentation but aims to shed light on their existence and the problems they aim to address. This, in turn, provides a context for understanding the raison d'être of emerging chains.

For readers contemplating investments in the native cryptocurrencies of these chains, comprehensive information is provided throughout the book including the tokenomics. I maintain impartiality regarding investment and technological preferences for any blockchain, offering information for the reader's consideration. Therefore, there is no correlation for the blockchains described to what I’m invested in either financially or technologically.

It's essential to clarify that while I am personally invested in some of the primary focus blockchains in this book, such investments do not extend to every blockchain featured. The selection criteria involve a blend of personal interest and the broader fascination of the blockchain community.

Given the vast scope of blockchain technologies, the book does not delve into highly technical details, development levels, or algorithms extensively which may be the case for a book dedicated to a single blockchain. Instead, it strikes a balance between readability and detail, aiming to provide enough information for a foundational understanding without overwhelming the reader. The intention is to give the reader enough detail, but not to completely bend their mind!

As the blockchain space evolves rapidly, future editions or additional books may explore other blockchains. Recognizing the challenge of capturing real-time developments, the References section at the end of chapters directs readers to further details and the latest news.

To maintain accuracy, some experts and CEOs from the blockchains covered have reviewed relevant chapters. I trust you will find this book both enlightening and thought-provoking, encouraging further exploration and inquiry.

Introduction

The objective of this book is to cater to individuals with a foundational understanding of blockchain technology, offering them an opportunity to deepen their knowledge by delving into newer emerging blockchains and comparing them with established counterparts. A basic proficiency in blockchain concepts, including a high-level understanding of how blockchains operate, familiarity with terms such as forks, and comprehension of consensus mechanisms, is assumed. Additionally, readers are encouraged to possess some knowledge of Bitcoin and Ethereum, along with an acquaintance with cryptographic concepts like hashing and public and private key encryption for digital signatures.

While the book accommodates raw beginners, a smoother experience is anticipated for those with a grasp of these basics. Although certain areas are explained and in layman's terms and supported by simple analogies, prior familiarity with these concepts enhances reader engagement.

The narrative of this book unfolds the evolution from well-established blockchains to emerging ones, each attempting to enhance aspects such as security, scalability, or decentralization. Notably, certain emerging chains, like Bittensor for AI and Sei for Decentralized Finance, introduce diverse functionalities.

The surge in blockchain and cryptocurrency developments has led to a multitude of platforms seeking to elevate scalability, security, decentralization, or create foundations for varied applications such as Decentralized Finance (DeFi), AI, or NFTs. After the inception of Bitcoin, Ethereum emerged with a mission to amplify scalability while laying the groundwork for smart contracts and decentralized applications (DApps). The inherent tradeoff between scalability, security, and decentralization, commonly known as The Trilemma, became apparent, posing a challenge for many blockchains.

Addressing this challenge, various emerging blockchains employ innovative approaches. Kaspa, for instance, tackles the scalability limit by allowing parallel block creation, demonstrating comparable security and decentralization to Bitcoin. Kadena claims to have resolved The Trilemma by enabling parallel chain creation through Proof of Work mining.

A newly emerging blockchain, Saito, challenges the conventional notion of The Trilemma, proposing it as an economic limitation rather than a technological one. According to Saito, the predominant incentive problem in most blockchains, where budgets are primarily allocated to security through mining or staking, hinders scalability and decentralization. Saito's unique solution involves rewarding nodes beyond mining and staking, promoting a self-sustaining network.

Saito claims they have solved the Trilemma by basically suggesting that there is no Trilemma! The details on Saito are covered much later in this book, but in many blockchains only mining nodes and staking nodes get paid, which is great for the security of the network but what about scalability and decentralization?! Nobody is paid for this, and rather, most blockchains are challenged with enhancing these two facets of the Trilemma. As an analogy for incentives, what if a painter gets paid to pick up the paint, but not to paint the walls? If there is no incentive to paint the walls, then how will this turn out? Likewise, if there is no economic incentive to scale or decentralize because nodes are not rewarded, this yields an incentive problem.

How about if scalability and decentralization are paid for? How about nodes other than mining or staking nodes get rewarded for this contribution? This is what Saito attempts to solve, a network that pays for itself rather than people (in the form of volunteers or pure altruism) paying electricity and infrastructure costs to run their own nodes with no economic incentive.

Other blockchains such as Sei aims to optimize a specific sector such as DeFi and trading applications and as a result can finalize transactions very rapidly. Also, the Algorand blockchain can finalize transactions instantly and there is no possibility of a fork!

Another interesting approach has been taken by Nano blockchain which is focused on peer-to-peer payments. After all, what was cryptocurrency invented for? The Bitcoin whitepaper outlined how the blockchain could serve as a foundation for peer-to-peer cash with no middleman or central entity having control, essentially a solution born from the 2008 Global Financial Crisis. Surprisingly though, very few projects have pursued peer-to-peer cash except for Nano. The blockchain and cryptocurrency sphere has been thus far, mainly used as a trading playground where many scams have taken place. People have made and lost lots of money from such wild speculation and also nodes run on the blockchain itself can profit hugely from fee and block rewards. How about rather than letting nodes profit from these rewards and users wildly speculating, there is a system where no rewards are paid so nodes and users can transact for free when making payments? This is a practical approach to solving real world problems where businesses and users can save on credit card fees when making purchases, all peer-to-peer in a decentralized fashion. This is a very different and refreshing approach compared to other blockchains.

The ongoing evolution in solving The Trilemma and creating application blockchains, such as AI for Bittensor, presents a captivating and continually evolving landscape showcased in this book. By combining technological insights with details on the tokenomics of each chain, readers may find valuable perspectives to inform their investment decisions.

The decision to focus this book on Layer 1 blockchains was not only to provide a foundation for understanding different blockchain technologies, but to consider why they have become so valuable. If one looks at the top 100 cryptocurrencies by market capitalization, there is a pattern in that about 80% are layer 1s and the remaining are layer 2s and other application tokens and meme coins. This may tell you something. It tells you that the layer 1 coins are seen

by the market to have utility and therefore, value. After all, layer 1 coins are used for transaction fees, securing the network and governance. This is immediate and obvious utility. The same cannot be said so easily for other tokens and although some have utility, this requires extra intensive research to discover the real use for the crypto token. Of course, it depends if you are a long term or short-term investor and what your goals are, but this is a pattern that is likely not a coincidence.

Throughout the book, I aim to spark thought-provoking discussions without strong biases. The various claims, criticisms, and insights presented are attributed to blockchain experts and founders, fostering an open and balanced exploration of diverse perspectives. I want to clarify that these are not my criticisms or claims, but that of others.

This book strategically poses numerous open-ended questions to provoke contemplation among readers and foster a receptive mindset. This deliberate approach aims to encourage reader engagement, acknowledging the nascent stage of the technology discussed. It is important to note that certain questions presented may currently lack definitive answers, owing to the evolving nature of the field. Over the passage of several years, these inquiries have the potential to find resolution.

Attempting to provide conclusive responses to these questions in the present moment may be deemed unrealistic, given the dynamic and transformative landscape. A pertinent illustration of this uncertainty is evident in Ethereum's recent transition to Proof of Stake and the forthcoming implementation of sharding. Given the recency of these developments, predicting the precise outcome remains elusive. In contrast, Bitcoin, particularly at its foundational level, benefits from the accrued wisdom of time, offering a more solid foundation for concrete answers to emerge.

As a final note, this book primarily focuses on Layer 1 blockchains, foundational platforms where DApps can potentially run. Interoperability chains (Layer 0), such as Cosmos and Polkadot, and other concepts like Layer 2 chains and applications or sidechains, fall beyond the scope of this book.

The Trilemma

One main aspect of this book regarding blockchain platforms is the Trilemma. There are three critical components for a blockchain platform to function and those are security, decentralization and scalability. The challenge with all blockchains is to balance these three components so that they are all enhanced as much as possible without any trade-off.

The Three Aspects of the Blockchain Trilemma

The reality is the three aspects of the Trilemma are intertwined such that maximizing one component usually results in diminishing another. This creates a predicament for software developers who must then trade-off one of these components to enhance the other two. The Trilemma is such that there can only be two pillars optimized out of the three and so the third one will not be optimal like the other two. Essentially, optimizing one aspect results in the expense of another, making it challenging to achieve all three simultaneously. Let's take a look at these aspects a little more to understand why this is.

Security

The first cornerstone of the blockchain Trilemma is security and is essentially the most important, especially given that the largest application for blockchains at present is cryptocurrency where large amounts of funds are stored. However, even if the use case is not cryptocurrency, it's still crucial to ensure the integrity and protection of data, for example in the case of supply chains or ID verification.

Decentralization

The next cornerstone is decentralization which relates to the distribution of making decisions, control and data across a network of many participants (or nodes) without needing to rely on a central authority. Each node has a local copy of the blockchain ledger where any change to the ledger, such as new transactions added, requires **consensus** across those nodes. As a result, there is no need to trust a centralized entity and no single point of failure.

There is much discussion in the community on whether decentralization is a spectrum. Clearly one node is too centralized, hundreds is more decentralized and many thousands is very decentralized. This is an over simplification because as you will find, there is more to decentralization than purely the number of nodes. After all, if a network has 10,000 nodes where most are controlled by a single entity (a cloud-based service), that's already a single point of failure.

There have been issues with different blockchains in the past where an action was required due to a vulnerability exploit, for example, in a decentralized exchange (DEX) or liquidity pool. The exploit was resolved in many cases with a consensus vote by validator or miner nodes. However, this raises concerns regarding centralization because it indicates a level of control amongst a group of nodes. If a network has 100 independent validator nodes and they come to consensus to resolve an issue via a vote, is this centralization or not? Some say yes because

although 2/3 are required to agree that's only about 66 nodes, which raises the question of whether that same majority could act maliciously. It doesn't sound unthinkable that 66 nodes could collude in some way. However, others say it's an example of decentralization because 100 independent validators came to consensus, which is what decentralization is about, rather than a single node or entity deciding. If all 100 nodes are truly independent (not controlled fully or partly by a single entity), then this could be seen as decentralization. If many are not independent nodes, then this is likely the opposite. In addition, if a certain action can be disabled or triggered by an administration key then this smacks of centralization.

The key question here being, is decentralization a spectrum? After all, if there are many validator nodes, but not hundreds of thousands, so long as they're independent it allows a situation to be quickly resolved in a decentralized manner, should an exploit occur. We don't live in a perfect world with zero exploits. The reality is it can and does happen. Is there a sweet spot to decentralization? Hundreds of thousands of independent validators, although very decentralized, could cause coordination issues in general and efficiency issues in the event of an emergency. There are enhancements and scalability improvements being carried out in many projects to address this at scale, as you will find in this book, but it's not without its challenges as there are generally trade-offs, hence the Trilemma.

The final key to all this is the incentive structure. If there is a solid incentive structure to reward nodes for their services, this keeps the network honest. Therefore, even with a moderate level of decentralization, it's unlikely that collusion (think the 66 nodes out of 100 example) could occur. This could be the perfect balance to performance, decentralization and preventing collusion while still being able to act swiftly in an emergency. However, it may not be as simple as this, as you will find throughout this book, because some projects aim for much higher decentralization without a trade-off in scalability. Has any blockchain truly achieved this yet? Do all blockchain projects agree that there is a sweet spot? How can decentralization be measured? Is there even some confusion on the definition of decentralization? Is it an over-used term? Is the incentive structure more important? Hopefully, in some of the later chapters you can make up your mind.

Scalability

The final cornerstone is the ability for a blockchain to process an increasing number of transactions without sacrificing speed and at a low cost, even as the network grows. As more users join the ecosystem and therefore initiating more transactions, the blockchain needs to handle this without huge delays, which is important for high adoption. Scalability helps to ensure that transaction fees remain low for users, otherwise the fees can increase significantly when there is congestion. This again is crucial for adoption if one wants to make the blockchain

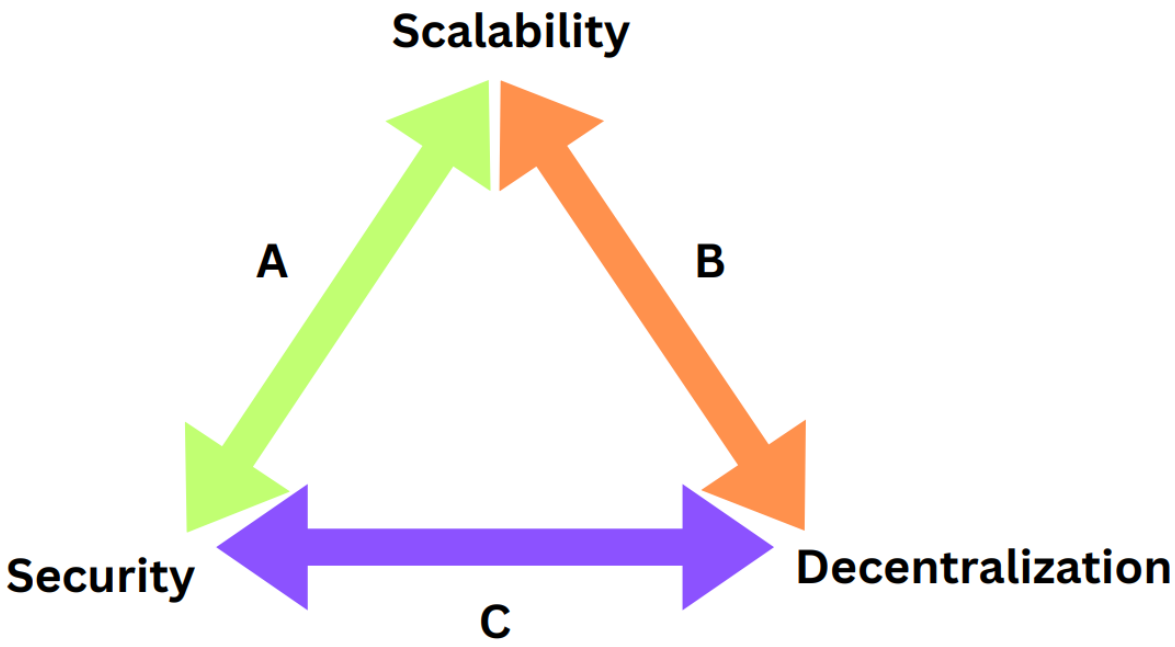
desirable for users and developers. In addition, a scalable blockchain provides a smoother experience for users which would otherwise lead to frustration and loss of trust.

In the case of many blockchain projects thus far, there have been some huge breakthroughs in scalability. The question is whether this is at the expense of security or decentralization or both? It's a huge challenge and in many cases you will find there are trade-offs as per the Trilemma. One interesting aspect to consider is as blockchains increase performance to tens of thousands of transactions per second, this can lower the fee rewards per block. Usually fees are higher when a blockchain struggles to scale as users need to pay higher fees for their transactions to be included in blocks. The competition for block space is higher, therefore fees are higher. Ethereum is the most classic example of this challenge as users over the years faced exorbitant gas fees. If scalability improves, this all sounds very well because fees usually will decrease as there is less congestion of transactions in a given block, but if adoption doesn't increase by roughly the same amount, this means the fee revenue for miner or validator nodes decreases. A 10x increase in performance is a great improvement, but unless adoption (in simple terms the transaction volume) increases 10x, the fee revenue has now decreased. This could lead to nodes dropping off the network to seek opportunities elsewhere or even encourage collusion. Less nodes or collusion increases centralization.

One way to handle this is by implementing a new innovative approach called **Automatic Transaction Rebroadcasting (ATR)** where transactions are pruned from the blockchain to make it more manageable. However, pruning is only the first step and many blockchains already do this, but there is a second step that only a rare few do. This is the process of automatically rebroadcasting any previously pruned transactions with a fee cost higher than the average market rate. This way the higher value transactions are constantly kept and refreshed where the lower value ones are pruned. The key part to all this is that it also maintains the fee revenue for nodes in the network to keep them honest and incentivized. More on this topic of ATR is discussed throughout the book.

Although it's not strictly the case most blockchains tend to tackle the facets as a priority in the order just described being security, decentralization and scalability. There is no point having maximum scalability if the platform is not secure! Of course, that is not say that scalability is developed last in the roadmap as all three facets would likely be worked on and designed in parallel. However, in blockchain and most technologies in general, scalability tends to be the last (and often most challenging) issue addressed as usually in the early stages of a project the priority is to get a product working and a proof of concept demonstrated. A blockchain platform can still go into production without high scalability, as long as it scales enough to work. This has clearly been the case as the vast majority of blockchains have few security issues but currently most cannot scale very well.

The following illustrates the Trilemma in terms of the three facets for a blockchain to function:



With the Trilemma, there can only be two pillars optimized before diminishing another, and therefore it's a design choice which two are enhanced and which one is compromised. In the diagram above it shows that two points of the triangle can be chosen with the other being diminished. For example, if one wants high security and decentralization, as per the triangle one may need to compromise on scalability. Another example is choosing (or enhancing) decentralization and scalability results in a compromise on security, the reason for which is described in the next section where increased scalability generally requires compromising security. Essentially, the Trilemma triangle diagram shows that choosing two points of the triangle moves away from the other point (hence a compromise). The challenge of course, is to maximize all three with no compromise.

Some blockchains claim to have solved the Trilemma by ensuring that increasing one pillar doesn't diminish another. However, this is still subject to much debate and cannot be totally proved until those chains have achieved mass adoption. Some have achieved it on their test networks (called a testnet) but it remains to be seen if this is still the case in a production environment (called a mainnet). Those blockchains that have claimed to have solved the Trilemma are featured in this book and are notably Kaspa, Kadena, Saito and others.

Why does enhancing one pillar compromise another?

Now that we understand the three main pillars, let's look into why enhancing one pillar diminishes another, from a technological viewpoint. For example, increasing security usually results in diminishing scalability and this is not a blockchain specific issue, but also a technological issue across the whole space. Anyone who has worked in software development in other applications would have experienced similar issues and certainly the Internet itself showcases this on many fronts.

The security and scalability trade-off

In cryptography (Note, this technology existed way before blockchain, and is used in many applications for securing connections to web servers for example) data is encrypted and decrypted. There are two types of encryptions:

- **Symmetric encryption**: Uses the same digital key for encrypting and decrypting data, the simple analogy being like a single padlock key for a treasure chest.
- **Asymmetric encryption**: A different key is used for encryption and decryption being a private key and a public key. These two keys are linked mathematically. Therefore, an owner with the public key that mathematically corresponds to the private key can decrypt the message. In the context of blockchain it decrypts a message to reveal a hash to prove authenticity of the sender who signed the transaction.

Asymmetric encryption is more secure and in the context of blockchain, it's used for encrypting digital signatures whereby any node with the public key can verify the authenticity of the message sender, because only the message sender has the private key. The public key can decrypt messages without any knowledge of the private key, thus there is no intrusion. This is why it is suitable for blockchain technology. If the same keys were used (as would be the case with symmetric encryption), then any node with the key could encrypt and verify the signatures. This would be a problem because there would be no way to verify the authenticity of the sender as anyone could sign the message, for example in the context of a transaction. In blockchain the public key is sent as part of a transaction and that key can only verify a digital signature (signed by the initiator of the transaction) for the corresponding private key (that the initiator owns). Essentially the private key proves ownership and enables spending of funds and the public key verifies the legitimacy of the digital signature.

In the context of asymmetric encryption however, the private and public key are mathematically related using some clever and beautiful mathematics trickery of prime numbers. In essence, this means that whoever has the public key (related to the private key) cannot determine the private key, and of course this is critical because, after all, the private key is meant to be private! It should only be known the by sender (in the blockchain context, the initiator of the transaction to send Bitcoin for example). Without going into detail, it's based on the concept in mathematics that given two prime numbers, if you multiply both these numbers to get a result, it is basically impossible to determine the two prime numbers used to get the result. Now applying this for a private and public key, it's impossible to determine the private key from the public key because they are mathematically related using prime numbers. This is a simple explanation as there is more involved than just prime numbers, but this gives you an idea of how complex mathematics is used to relate the keys.

Now that asymmetric encryption has been explained, let's use an analogy for the Lehmann to understand why increasing security diminishes scalability. It's clear that complex mathematics is used for encryption and decryption and even more so for asymmetric encryption due to the mathematical binding of the public and private keys. This requires more intense mathematics, complexity and therefore computer instructions. Therefore, in very simple terms, imagine for your house you have many different locks for door and windows where each has a separate key. This is more secure because there is no master key to unlock all doors and windows! You may also require two different keys to unlock a single door. Although this is more secure, it's also slower as accessing your house each time requires an extra key to unlock the door. Much like apartment blocks have a key to unlock access to the building and another to unlock your own door. Therefore, someone with only one key cannot access your house as they need both. It's clear that the time consumed to unlock access to the house is slower as more and more keys are required, albeit more secure. This is an extremely simple analogy, but helps to translate to the context of software which is essentially not very different. The more secure and complex encryption algorithms used the slower the execution. This addresses the balance and trade-off required between security and scalability, which are two facets of the Trilemma. The use of public and private keys in the context of a transaction is described more in the Bitcoin chapter.

To be clear, the example just described is not an issue as such with blockchains because all blockchains use asymmetric (public, private key) encryption for digital signatures, but still showcases how extra security can hinder scalability. Many blockchains use other modules and measures to increase security overall for their respective ecosystem (outside of digital signatures), but this can affect scalability. It all depends on what the blockchain sets out to achieve and thus needs to align its design with its goals. This will become clearer upon delving into the blockchain chapters later in this book.

As an example of a blockchain where a push for scalability has come at some expense of security is Solana which has achieved a high number of transactions per second. However, this has not been without some pain as Solana has suffered some vulnerabilities resulting in funds lost. There are two ways to view this where one could say this is a no pain no gain approach as Solana has pushed the boundaries and this can be seen as a positive where the result is very high scalability. On the other hand, funds have been lost and it's not a good perception for the Solana blockchain. Bugs in software are normal, but blockchain bugs can have catastrophic consequences due to huge amount of TVL (total value locked) on chain.

The decentralization and scalability trade-off

As described earlier a decentralized network does not rely on a centralized entity and therefore, for this to work there needs to be many nodes involved in consensus when a new block of transactions is proposed. Since there are many nodes, this increases chatter between those nodes and it takes time for all the nodes to synchronize.

There are different consensus mechanisms such as Proof of Stake and Proof of Work where the latter requires intensive resources to finalize and agree between the nodes. There is more detail on these consensus approaches later in the book. However, as the number of nodes increases the time required to reach consensus also tends to increase, which leads to higher latency and therefore a decrease in the number of transactions processed per second.

In a decentralized network, each node must validate transactions and in the case of Proof of Work this requires significant computational power and even more when the network grows. The more nodes there are the more time it can take for transaction data to propagate across the network, which then results in delays.

Many of the blockchains featured in this book have solutions to mitigate the effect of delays, latency and chatter between nodes. For example, Proof of Stake (increasingly used by many blockchains) requires a 2/3 majority vote of a set of nodes on a rotation basis to reach consensus, unlike Proof of Work. This means not all nodes are required to vote in a given window of time and so helps to make the network more efficient. There are many other solutions to the decentralization and scalability side of the Trilemma, and these are described later in this book in the various blockchain chapters.

However, one solution that some blockchains use to increase scalability is to dramatically decrease the number of nodes required to participate in consensus. BNB smart chain (BSC) has only 21 validator nodes for reaching consensus which although makes the chain very efficient, it also makes it very centralized. This is an example of a trade-off made in this part of the

Trilemma. BSC has in the past paused the whole blockchain as a result of a security issue and this can be seen from two different angles. The first angle is that with a small set of validators it makes it easy to coordinate between them and pause the chain to resolve problems. On the other hand, this means there is centralized control and this goes against the principle of blockchain in the first place. Pausing the whole blockchain doesn't seem an ideal approach each time there are issues and is tantamount to the idea that other controls could be put in place at the disadvantage of the users.

As a final note, increasing centralization also tends to lead to decreased security because nodes can collude together more easily in a malicious manner that could negatively affect the blockchain. This is known as a 51% attack and is discussed later in the book. This would be an example of another trade-off in the decentralization and security part of the Trilemma, whereby increasing centralization decreases security.

A modular approach to solving the Trilemma

Blockchains are generally built using a monolithic approach where nodes carry out many tasks such as validating transactions, ordering transactions, verifying disputes and finalizing blocks. A blockchain is a decentralized network of nodes that collaborate to validate the order and legitimacy of transactions within a block and the blocks themselves. A modular approach, such as that taken by Celestia, separates these functions.

The four core functions of blockchains

There are four main core functions of a blockchain and these are laid out below:

- **Execution Layer**: This is where applications built on the blockchain reside and transactions are therefore executed with the state of the chain updated.
- **Consensus Layer**: This is where ordering and validity of transactions occur.
- **Data Availability Layer**: This layer is responsible for the record of transactions and published transaction data being available for download.
- **Settlement Layer**: This layer finalizes transactions in blocks after being submitted by the execution layer for disputes and proofs.

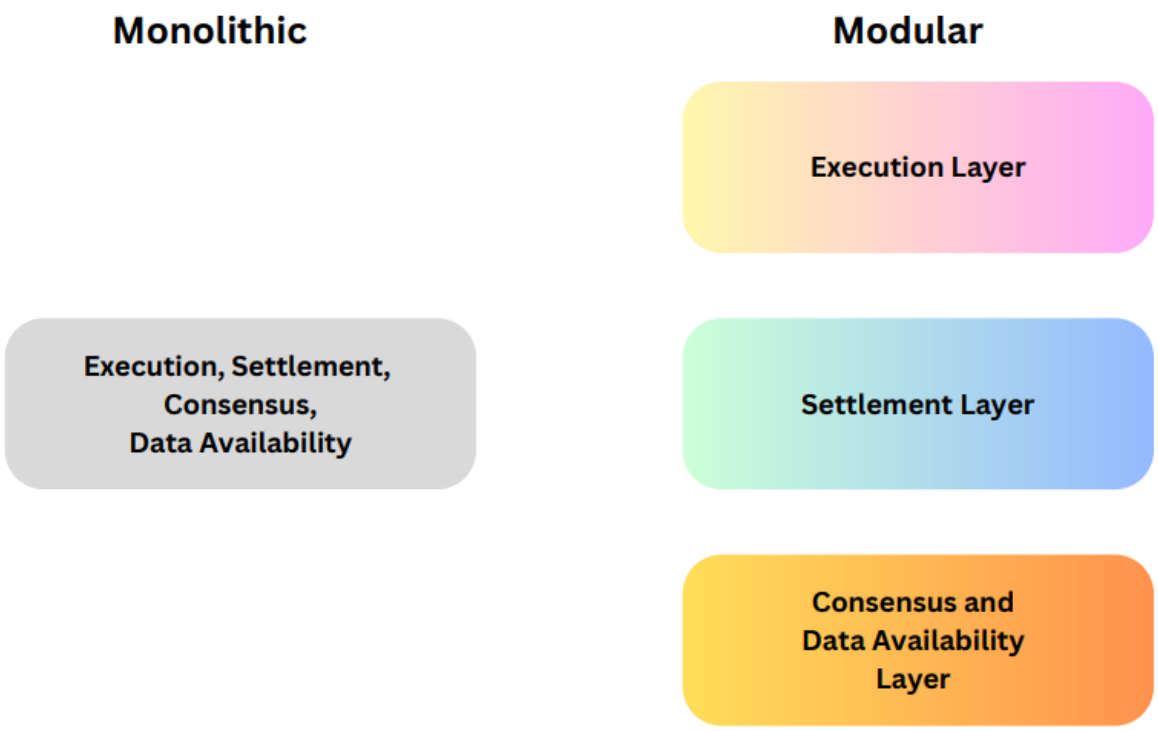
Typically, a monolithic blockchain has all the four functions built into one layer. Some blockchains take the execution function to a separate layer called a Layer 2 where computations are performed off chain and uploaded as a batch (rolled up with the results) to the layer 1 main blockchain later. This helps to reduce the load on the Layer 1 chain. However, the main problem is by combining all the functions in a single layer requires a collective responsibility for nodes to reach consensus, ensure data availability and execute transactions. As a result, attempts to optimize one function consequently leads to the demise of another because a node focusing on one task or function takes the focus off another, hence a trade-off is needed. This is because execution, consensus, availability and settlement are all competing for the same resources and leads to inefficient execution, bloating of states and high gas fees. As per the Trilemma, this is a trade-off to the scalability pillar.

Blockchains such as Celestia decouple execution functionality from consensus ensuring that the consensus layer is just required to order transactions and guarantee data availability.

Decoupling Execution from Consensus

Celestia decouples the layers so that there are now separate modules for the execution layer, settlement layer and consensus with data availability layer (note the latter is in one module). With this approach each layer specializes to perform its own function in an optimal way to allow increased scalability for example.

The following illustrates the approach taken by modular blockchains like Celestia of separating execution, settlement and consensus:



The solution that Celestia use consists of different types of nodes to carry out the separate functions. It uses light nodes where these nodes don't contain full blocks but instead, they use a technique called **data availability sampling**. This enables them to check random parts of a block to make sure that data is available without requiring the whole block to be downloaded. This allows the light nodes to detect invalid blocks such as where data has been withheld by block producers. This is an example of performing verification of a block required for the consensus function (one of the four core functions of a blockchain).

Those more familiar with Bitcoin may ask "How is this different to Bitcoin or other chains that also use light nodes?". The main point is that Bitcoin light nodes (called Simple Payment Verification nodes), although also don't contain full blocks, just check if a transaction is in a block, but don't verify the validity of the transactions or detect invalid blocks. This extra check requires the full nodes (that have a full copy of the blockchain) to perform this validation.

Over the course of this book, the modular approach isn't described much further, but highlights the fact that there are many different ways to tackle the Trilemma. The modular approach that Celestia uses is also relatively new. The general approach taken by most blockchains currently is to separate functionality using either a Layer 2 or a sidechain. This doesn't separate all four core functions outlined earlier, but Layer 2s for example decouple the execution function from

consensus to increase scalability. Many of the blockchain platforms discussed in this book use a Layer 2 or sidechain to address the challenges with the Trilemma. Although Layer 2s and sidechains are mostly outside the scope of this book, they are highlighted in some of the chapters to illustrate how this approach helps with scalability (such as the Bitcoin chapter). A large part of many solutions to the Trilemma are addressed at the layer 1 itself, and since this book is about layer 1 blockchain platforms, that will be the main focus.

Is the Trilemma a technological or economic problem?

The Trilemma is largely seen by the community as a technological issue. This is not unique to the blockchain space as anyone involved in software development knows very well that scalability usually comes at the expense of other areas such as security for example. Scalability is always a challenge for any technology project, but before blockchain it was able to be solved more readily as most solutions used a centralized model and therefore there was no need to consider decentralization as compromising scalability. With the advent of blockchain technology however, there were now two large factors hindering scalability, being the level of security and decentralization. In summary, less security and more centralization allowed scope for more scalability.

However, a blockchain project called Saito (featured later in the book) claims that the Trilemma is not a technological issue as such, but an economic one. This is largely because security for almost every blockchain is intact due to the economic energy used by nodes (validators and miners) to secure the chain. These nodes are rewarded for securing the blockchain and thus provides an incentive to maintain security. In most blockchains though, there is usually no reward or economic incentive to keep the respective chain decentralized or scale highly. This results, unsurprisingly, in these two pillars (decentralization and scalability) suffering somewhat.

The Saito solution is both economic and technological where the technology model supports the blockchain economy to reward efforts for all pillars of the Trilemma, not just the security pillar. The nodes contributing to decentralization and scalability also get rewarded. The details are described in the Saito chapter, but it's certainly a unique way of approaching the problem. As a result, Saito essentially claims there is no such thing as a Trilemma as such, but rather it's related to a lack of incentives for decentralization and scalability. If users were incentivized for their efforts for these pillars, the overall issue wouldn't exist.

Is there some confusion about decentralization and the trade-offs in the Trilemma?

In many blockchains there are large entities hosting nodes on cloud based services mainly for convenience as most users don't want to run their own nodes. For example, full nodes and light client nodes don't get financial rewards, only validator or mining nodes. Therefore, users typically delegate this to a cloud based service. In many cases, even validator nodes are run on cloud based services as users don't want to do the heavy lifting of all the setup and cost for hardware, so entities such as Infura and AWS take care of this.

Some blockchains as a result have many nodes but since a large portion are hosted by cloud based services this results in centralization and negates the idea of decentralization because it's no longer the case that the more nodes, the more decentralized! The real problem that decentralization solves is that of excludability. A cloud based service and minimum stake amounts for validators create a form of exclusion where nodes that don't have a minimum stake amount to partake in consensus are excluded. Those that don't want to run their own hardware have little option but to use a cloud service. This creates a system that is not open and therefore excludable, where the large entities or big players now have control. The principle of blockchain in the first place was to create a system that was open, where anyone could join. If users were incentivized financially to run their own full nodes or light nodes, this would result in more decentralization and also more scalability as the hardware costs would be compensated for, allowing new hardware to be affordable for upgrade.

This is the problem that Proof of Routing Work consensus mechanisms solve as is the case with Saito. As the reader, don't worry if this is beyond you at this point. This will all become clearer as you work through the book, especially towards the latter end where Proof of Routing Work is discussed and how Saito implements this.

In summary, based on this approach it could be said that the Trilemma **is** a trade-off between:

1. Security
2. Scalability
3. Decentralization

It is **not** a trade-off between:

1. Security
2. Scalability
3. Non-excludability

Routing Work pays nodes for the work they do when transactions are routed which incentivizes participation from all node types, thus creating a self-sustaining economic loop as node operators are rewarded in proportion to their contribution. It's clear that more node operators promote decentralization as there are fewer central points of failure, but the main difference here is that scalability is also supported because the increased number of nodes allows the network's capacity to handle larger volumes of transactions. This may be the case with many

blockchains, but in this case it's special because nodes can afford to upgrade their hardware which will likely be necessary as hardware specifications need to increase to cope with the higher demand. Otherwise the result will be to delegate it to cloud services which leads to centralization. It's this openness (or non-excludability) as the key principle to uphold as the network's funding mechanism supports this growth allowing all three elements to coexist without any compromise.

Consensus Mechanisms

Consensus Mechanisms Overview

There are many consensus mechanisms, the main two being Proof of Stake and Proof of Work. Proof of Stake has gained popularity due to its energy efficiency. However, it may be a surprise to many that Proof of Work in Bitcoin can potentially help the environment. This is described in more detail in the Proof of Work section. There are other Consensus Mechanisms such as Proof of Routing Work used only by Saito (described in the Saito section later in this book) which is a very unique approach to consensus. Solana uses Proof of History (described in the Solana section). For this section only Proof of Work and Proof of Stake are described with some interesting discussion points as a warmup and to reinforce knowledge you may already have on this topic.

There is a lot of documentation on this topic so, to minimize the bloat of this book I will touch briefly on certain aspects so that the focus can be on the newly emerging blockchains and other insights. Therefore, the description of PoW and PoS are more a recap and suitable for beginners in the space. The pros and cons of both Proof of Stake and Proof of Work are described later in this section.

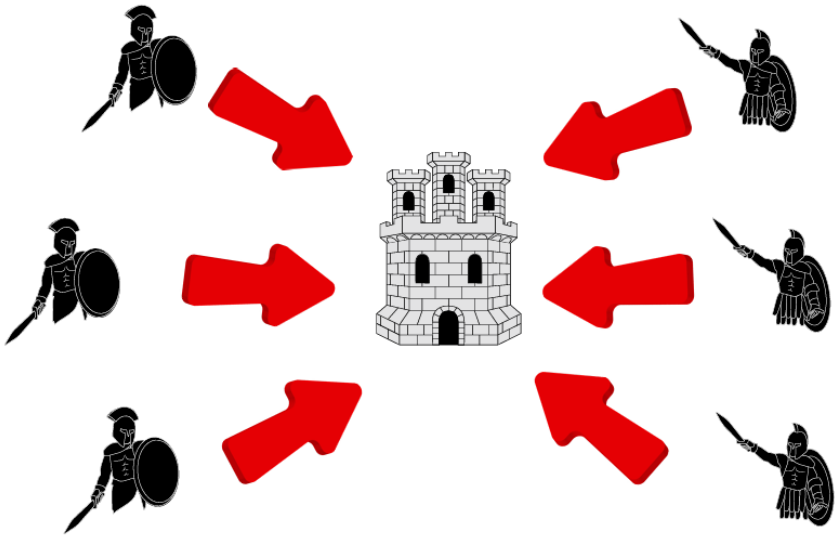
Proof of Work

Proof of Work is the consensus mechanism used by Bitcoin, Kadena, Kaspa and various other blockchains. Proof of Work is a solution to the Byzantine Generals' problem (and Proof of Stake is a different solution) of how to reach a consensus across a decentralized and distributed computing environment.

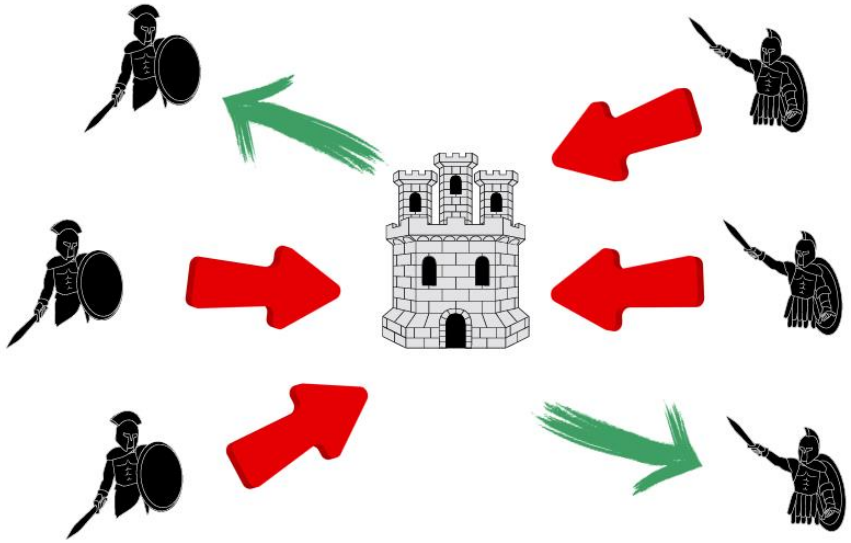
First, let's understand the Byzantine Generals' problem. The problem relates to a group of Byzantine generals, each commanding a portion of an army, who are around a city they intend to attack or retreat from. The generals need to reach a consensus on whether to attack or retreat, and they must do so in the presence of potential traitors among them. The challenge is to develop a consensus algorithm that allows the loyal generals to reach an agreement even when some of the generals may be traitors, where they may send contradictory messages to confuse the outcome.

The problem in a simple summary, is how do we coordinate people (in the case of software, nodes) across a large region, when those people (or nodes) are spread out globally without the honest people (or nodes) having to trust every other person (or node), where some may be dishonest in the either nearby but more so, remote locations?

The following depicts this problem. In a distributed computing world, think of the coordinated attack as an agreement by consensus for those nodes to achieve a certain goal as per the rules of a program and think of the traitors as misbehaving computer nodes who may not follow the programmed rules. This is a coordinated attack resulting in victory:



This is an uncoordinated attack resulting in defeat – the green arrows indicating those who betrayed:



So how can we avoid the situation in a distributed computing world where traitors (misbehaving computer nodes) compromise a goal to reach consensus? One such way to reach consensus is Proof of Work which requires an expensive computer calculation, and this is the process known as mining. Mining must be performed to create trustless transactions on the blockchain.

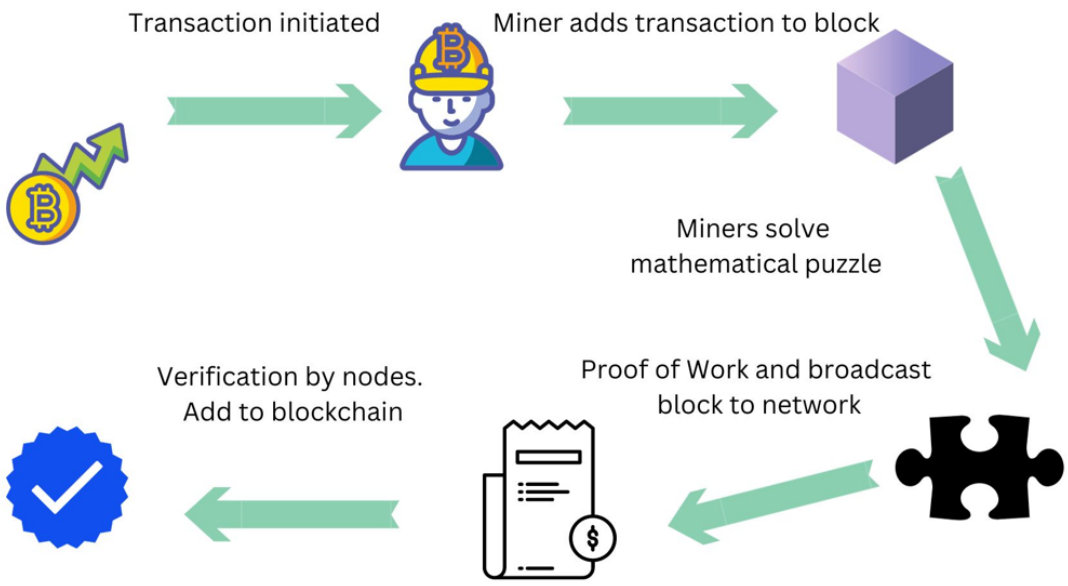
A consensus mechanism such as PoW that solves the Byzantine Generals Problem thus achieves Byzantine Fault Tolerance (BFT). This addresses the challenge by ensuring that the distributed system can withstand a certain level of faulty or malicious nodes without compromising its correctness and reliability.

PoW in summary works as the following:

- Transactions are broadcast to the network by nodes, then the nodes in the network verify these transactions to check they conform to the rules of the blockchain.
- Transactions are bundled up together in the form of a block by a mining node. These transactions are selected from a pool of unconfirmed transactions, with those with the highest fees selected as higher priority.
- Miners (mining nodes) then verify transactions for each block, checking to see if they are valid.

- Miners solve a mathematical puzzle known as proof of work. The puzzle is a difficult mathematical problem where a hash needs to be calculated with a certain number of leading zeros. The network has a predefined difficulty level that determines how hard it is to find a solution. Miners need to find a solution that, when combined with the block's data, produces a hash that meets the difficulty criteria.
- If the puzzle is solved, the miner broadcasts the block to the network for other nodes to validate, that is the block and its contents.
- Once all validated, the block is then stored on the blockchain where that block is linked to the previous block. Of course, this block contains the verified transactions.
- The miner is rewarded with a block reward and fees for those transactions to be included in the block.

The following illustration depicts this:



The question that most new to the blockchain tend to ask is “Why do the nodes have to solve such a complicated mathematical problem as a base for consensus, surely that’s not good for the environment?”. The answer is, it doesn't have to be this way, because there are other consensus mechanisms such as Proof of Stake that reach consensus without having to solve this problem. However, Proof of Work is a very strong solution to the Byzantine Generals Problem because it's very difficult to game. It’s a way of proving that a significant amount of work has been done, and the node (or pool of nodes) that has produced the most amount of work is rewarded which helps to secure the blockchain as there is an incentive to keep the node(s)

honest. As an analogy, think of the guy at the gym that has worked hard lifting weights and leaves the gym all ripped with huge muscles. This is very difficult to disprove, the evidence is clear he did the work! Likewise, a computer mining rig can prove this by solving a complex mathematical problem. This makes the network very difficult to hack because a node(s) would need to do even more work (using computational power, thus electricity) to create a block with malicious transactions.

To expand on this and clarify a little more, the benefits and some advantages of Proof of Work (with respect to Proof of Stake) are listed below:

Financial Incentives

Miners are economically motivated to act honestly. PoW has a huge cost because it requires significant power, therefore electricity, due to solving complex mathematical problems. Miners are therefore rewarded with cryptocurrency coins, known as the issuance, and the transaction fees in the block. This creates an incentive which aligns with the security of the network, because miners have an interest in remaining honest by following the rules if they want to receive rewards.

Decentralization

PoW allows anyone with the required computational power (hardware and electricity) to add new blocks to the blockchain. This open participation ensures that no single miner or group can control the network promoting a more decentralized distribution of computational resources.

PoW encourages a competitive nature that ensures that miners are always competing to solve complex mathematical puzzles to add new blocks. This competition prevents the dominance of any single miner or miners because other miners are always trying to participate and earn rewards.

Note that this is the idea and concept to achieve decentralization, but in practice there are large mining pools that control large portions of the computational power. However, so long as the incentive structure is good enough this negates a possible 51% attack because this attack would likely result in the miners losing more than the rewards they receive, due to the electricity costs and decline in the price of the cryptocurrency they are being rewarded in. This point is expanded on in the Mining and Staking Pools section later in this chapter.

Immutability of Transactions

Once a block is added to the blockchain, modifying it becomes very difficult. For an attacker to modify a block, they would need to modify the block concerned and recalculate the proof of work for that block and all previous blocks linked to it. This level of computational effort makes the blockchain highly resilient to modification.

Security and Prevention Against Double Spending Attacks

PoW helps to prevent double-spending attacks (spending the same amount of money twice) by providing a way for reaching consensus on the order of transactions in the blockchain. Once a block is added to the blockchain, it is linked to a chain of blocks. The cumulative proof of work makes it computationally extremely difficult to create a different chain with conflicting transactions.

Network Difficulty Adjustment

The network's difficulty level adjusts dynamically based on the total computational power (hash rate) of the network. This adjustment makes sure that, in general, a new block is added to the blockchain at a stable rate. If there is an increase of miners joining the network, the difficulty increases, and if there is a decrease, the difficulty decreases. This helps to maintain the security of the network and the stability of the network.

Hash power, often referred to as computational power or hash rate (measured in hashes per second), is the measure of how much computational work a miner or a group of miners can perform per second. It's a measure of the processing power required to solve mathematical problems being cryptographic puzzles and hashing algorithms.

The higher the hash power, the more computational work can be achieved, therefore, the higher the probability of successfully mining a new block.

How PoW in Bitcoin can help the environment

Although Proof of Stake has been popular due to its energy efficiency, I thought it worth discussing how Proof of Work in Bitcoin can help the environment. I think it's clear Bitcoin mining uses lots of energy (although there is a significant portion of clean renewable energy as miners seek to reduce costs) and there is an abundance of information out there on this, so this book won't rehash that. However, there is less information on how it can benefit the environment. For Proof of Stake, it's a no brainer (because it doesn't use huge computational resources), so that's why Bitcoin's PoW has been emphasized here. It's a little outside the scope

of this book, so this section is very brief, but again I think it's an interesting insight to stimulate your thoughts.

Essentially, there is an intermittent nature to solar and wind energy, because they only produce energy when the sun shines and the wind blows. A lot of this energy is generated when demand is low, so there is an oversupply of energy. Now, if this energy is not stored in batteries, it is basically wasted. The miners can purchase the excess energy from solar and wind farms, which of course increases the revenue of those renewable companies, and it also prevents taxpayers from subsidizing all that extra energy.

In summary, Bitcoin mining rigs can be switched on where there is an oversupply of energy, thus absorbing the energy that would be otherwise wasted and thereby redirecting the electricity from the grid to the mining rigs to mine Bitcoin. The rigs can then be switched off when there is an undersupply of energy and so redirect the energy back to the grid.

As a final note, in addition to renewables, Bitcoin miners can also help stranded methane (which is a potent greenhouse gas) which is not economically efficient to bring to market. Methane comes from gas and oil operations and landfills. The key to all this is that Bitcoin miners can operate in any geographical location where they can turn the stranded methane into electricity, then use that to mine bitcoin producing an environmental and monetary benefit.

Proof of Stake

There are many variants to Proof of Stake (PoS), but those are tackled later in the book as those blockchains that use PPoS or DPoS for example are described. This will give more relevance and context rather than describing every PoS variant in advance.

PoS is a consensus mechanism which is a variant of how to solve the Byzantine Generals Problem to PoW. It's used to achieve agreement on the state of the blockchain where it determines the creator of a new block based on the amount of cryptocurrency a participant has staked as collateral.

These participants, known as validators, are required to lock up a certain amount of cryptocurrency as collateral, known as their "stake." This stake serves as a commitment to the network and is held as validators perform their duties.

Validators take turns proposing and creating new blocks in proportion to the amount of cryptocurrency they have staked. The more the validator has staked, the higher probability they have of being selected to create a new block.

There are different methods for selecting the validator to create a new block and this varies among different PoS implementations. Some use a randomization process based on the stake, while others use the length of time the cryptocurrency has been staked (coin age).

It’s more energy-efficient compared to PoW because it doesn't need the same amount of computation, therefore electricity consumption. Validators are chosen based on their stake rather than their ability to solve complex mathematical puzzles.

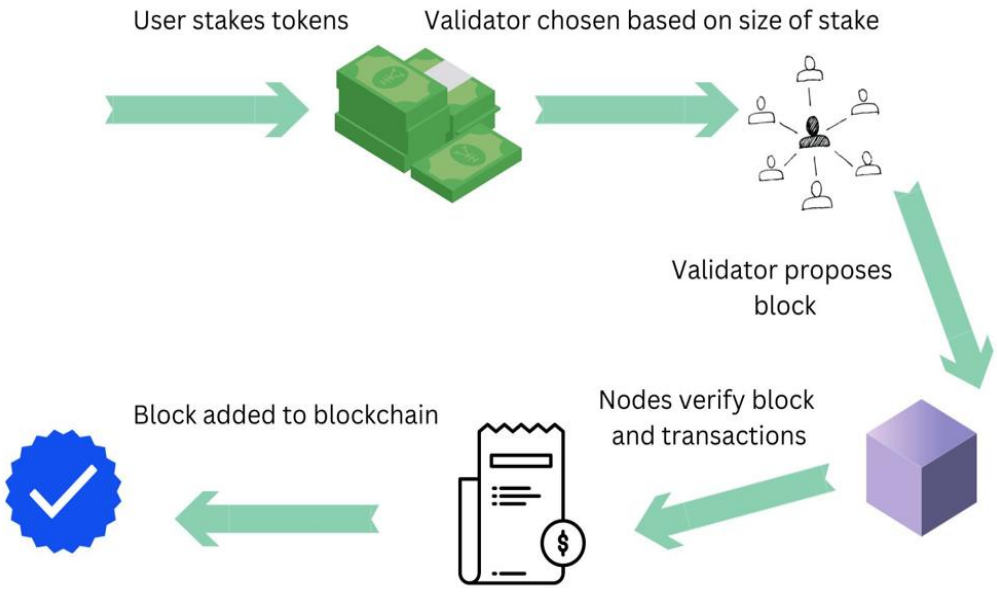
PoS networks implement security measures to deter malicious behavior. For example, validators may be punished and lose a portion of their stake if they are caught acting dishonestly.

The PoS mechanism in summary works as per the following:

- A user places a stake of tokens into a staking pool, thus locking them up as collateral.
- The next validator is chosen based on their stake size in that the probability being chosen is proportional to the number of tokens they have staked. There may also be another element in that length of time they have been staked for has an effect (coin age). This also increases their chances of being chosen. However, the system may reset this periodically to prevent a validator dominating the network by always proposing blocks.
- The chosen validator proposes a block consisting of transactions in it.

- Other participants get to approve and verify the proposed transactions in the block. If there is misbehavior such as double spending or proposing invalid blocks the validator may be punished and have their stake (all or part) slashed as a result.
- A new block is then linked into the blockchain.
- The validator earns a transaction fee (Note that in Ethereum this will also be a block reward, known as the issuance, and also MEV rewards. More detail on this in the Ethereum chapter).

The following illustration depicts this:



In the Ethereum chapter and other blockchain chapters that use PoS, the details of aspects such as voting, penalties, checkpoints and fork choice rules are fleshed out in more detail. However, as a high-level summary consensus can be achieved with no issues if less than one third of the validators are dishonest. What this means can be summarized in two points:

- If more than 1/3 of the validators are dishonest then the chain could be paused. If this happens the dishonest party may choose to not partake any further, meaning the rest of the validators will not be able to uphold a 2/3 majority (which is needed for consensus). This leads to no transactions occurring at all.
- If more than 2/3 of the validators are dishonest then collusion and bribing could happen. This situation is extremely undesirable because the blockchain now suffers

from things such as double spending of transactions and other attacks because the 2/3 majority has control.

Of course, this leads to making sure there is a huge set of nodes in the network to lessen the chance of collusion, but this is at odds with scalability because this now requires lots of chatter (and therefore delays) to achieve consensus across the large set of nodes in the network. Some methods to avoid such scenarios are rotating validator committees (providing a subset of validators selected from the main validator set), Slashing and Penalties, and Proposer Builder Separation (PBS) among others which are discussed later in the book.

Further to all this, the benefits and some advantages of Proof of Stake (with respect to PoW) are listed below:

Security

Validators have a financial stake in the network because they need to lock up tokens as collateral to receive rewards such as fees, MEV or a block issuance. This discourages malicious behavior, as validators risk losing their funds and rewards if they act dishonestly.

Energy Efficiency

PoS is generally more efficient than PoW in terms of energy consumption. PoW requires miners to solve complex mathematical puzzles thus using lots of energy and resources, but PoS relies on validators to create new blocks where they are chosen based on the amount of cryptocurrency they hold.

Cost Reduction

PoS reduces the need for expensive hardware and electricity. This makes it more cost-effective for participating nodes. Those in PoS blockchains don't require powerful mining rigs, and therefore ASICs, GPUs or any specialized hardware, which lowers operational costs.

Scalability

PoS is considered more scalable than PoW because it doesn't have the same level of computational limitations, which leads to being able to handle a larger number of transactions.

Therefore, with fewer computational hurdles, PoS can potentially offer quicker finality for transactions. However, as this book reveals, some more advanced PoW approaches using DAGs can achieve higher scalability.

Reduced Risk of Centralization

Although this is highly debatable one may say that since PoS doesn't rely on computational work, there is a lesser risk of centralization. This is because in PoW, in regions where electricity is cheap and mining hardware is more affordable, there is an increased tendency to become more centralized. Also, only those with deep pockets and expertise in mining hardware have the ability. PoS opens to the wider masses because all that is needed is the ownership of some cryptocurrency tokens, albeit often a large number. Either way, money is required, either to stake or buy mining hardware. However, with PoS one doesn't need the same level of expertise setting up large mining rigs.

Proof of Work vs Proof of Stake

The following illustrates PoW and PoS side by side for an at-a-glance comparison:

	PoW	**PoS**
Mechanism for Consensus	Mining	Validating
Rewards	The node that mines a block receives a reward.	The node that is chosen to validate and propose a block gets a reward.
Security	Provided by hashing to solve a mathematical problem to propose a block.	Staking collateral to propose a block.
Malicious Activity	No explicit rules for punishing bad actors.	Activity by bad actors results in the stake being slashed and penalties.
Efficiency	Less efficient. Renewable clean energy is encouraged for cost saving.	Energy efficient.
51% attack approach	Must control 51% of the hash power.	Must control 51% of the staked collateral.
Equipment	ASICs and GPUs	Standard server grade devices

In both PoS and PoW, they (miners and validators) earn transaction fees for the transactions they include in a block.

However, for Ethereum 2.0, in addition to these rewards, there is an issuance block reward like Bitcoin (although it was reduced since Ethereum 2.0 where it moved from PoW to PoS) and validators can also earn MEV (Maximal Extractable Value), which is the amount of value that can be extracted from a block by reordering transactions. MEV is earned by validators who can extract value from the transactions they include in a block. More on this in the Ethereum chapter.

There has been much detail so far on some advantages of PoW, and this was emphasized somewhat more because it appears less obvious to many, so the objective of this was to provide some stimulating insights. However, the following describes advantages and dis-advantages for PoW and PoS:

	Advantages	**Disadvantages**
Proof of Work	Can encourage use of renewable energy Proven over time	Expensive equipment required High energy usage Slower transaction speeds (longer time to finality)
Proof of Stake	Energy efficient Less expensive equipment required Less expertise regarding equipment	Unproven at large scales or long periods Can require high investment to stake

Note that it's mentioned that PoS is unproven at large scales or long periods, especially given that Ethereum recently moved to PoS and can't dramatically scale thus far. After five more years this situation may change. Other blockchains may well be able to scale, but don't yet have the level of adoption like Ethereum and are still relatively new.

Regarding scalability for PoW, this situation is also changing as solutions using DAGs and parallel chains are achieving higher speeds. However, in general the time to finalize blocks (time to finality or TTF) tends to be longer in some PoW chains, especially Bitcoin. Finality is discussed in more detail in later chapters, but essentially it's the time taken for a transaction to become immutable in the blockchain's ledger, making it irreversible. Some other PoW chains such as Kaspa, Alephium and Kadena have much faster TTF than Bitcoin.

Mining Pools vs Staking Pools

Mining pools are groups of miners who combine their computational power to mine blocks where the rewards are paid in proportion to their contribution. So even if a miner's individual computational power is low, mining pools are useful because they allow miners to receive more consistent rewards.

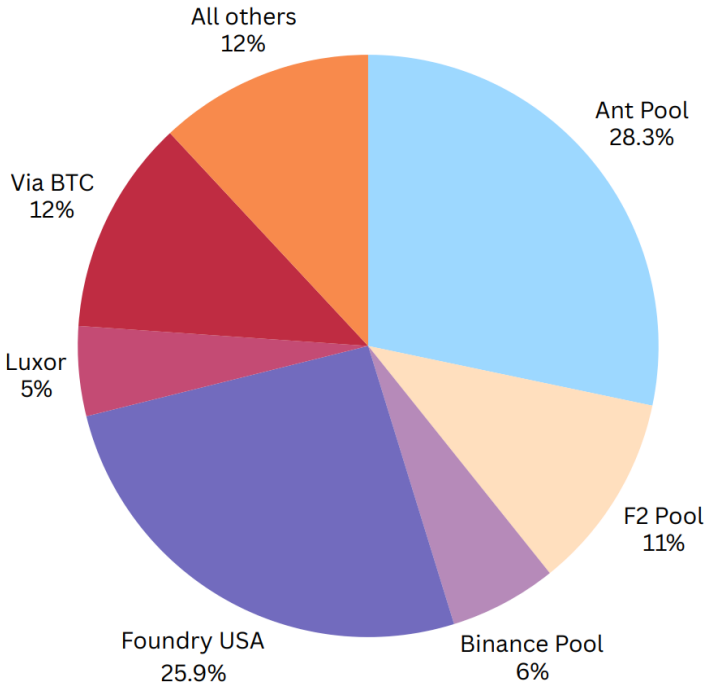
Staking pools are similar to mining pools, but they pool together the staked tokens of multiple users which increases the probability of earning rewards rather than using computational resources. Staking pools have a lower barrier to entry because one doesn't have to stake large amounts of tokens or run a validator node.

However, just to be clear, mining and staking pools, although similar, are not the same thing. With mining pools, they don't take custody of the ASICs, the miner itself owns and takes custody of the ASIC hardware. Therefore, a miner can simply switch their hash rate (computational power) and point somewhere else within seconds.

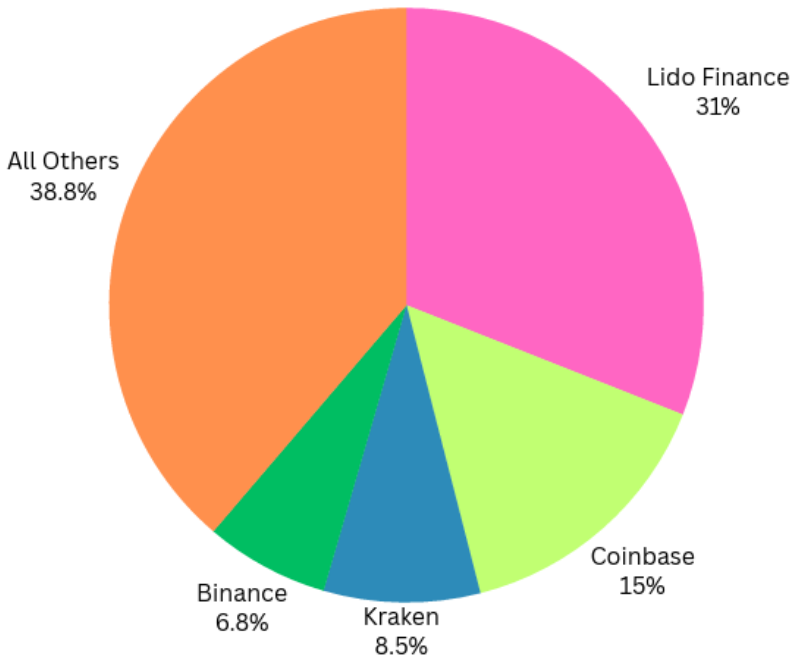
With staking pools, they in many cases take custody of the validators. This means that if a pool is controlled by a centralized entity (for example an exchange), you may not be able to switch and point to a different pool quickly and therefore, can be quite sticky. Many have limits and it can be days to wait before one can switch to a different pool. Payouts also can be delayed and in extreme situations if the centralized entity has financial issues, it's theoretically possible that one may not get paid rewards at all because the centralized entity has control. Of course, in the case where there is no centralized entity this isn't an issue, but currently it's quite common that an exchange is involved.

However, mining pools in Bitcoin for example, do have a level of centralization as also do staking pools in Ethereum. Centralization in this context is not ideal, although some would argue that because the incentive structure in Bitcoin is very sound and predictable, there is little or no motivation for a 51% attack, thus the miners and mining pools are kept honest, or could be known as benevolent whales. The incentive structure in Ethereum is also a sound one, although perhaps less predictable because since Ethereum 2.0 there are also MEV rewards which introduce a variable APY, and a reduced block issuance reward. If one considers the cost of an attack especially in Proof of Work whereby there is an electricity cost, but also the decline in the price of the cryptocurrency (as the market finds out about the attack), most calculations reveal that the cost is simply too high. Markets tend to react very quickly, so it's likely that a selloff will happen so quickly, that the attacker may not be able to realize their profit before the selloff occurs. Either way, it's a huge risk. But in both cases, the security budget and therefore, the reward structure seems sound enough to deter any motivation for a 51% attack. I will leave this an open question for you to ponder.

The following shows the current distribution for Bitcoin mining pools:



The following shows the distribution for Ethereum staking pools:



Note that this may have changed for both Bitcoin and Ethereum since the time of writing, but it shows that it's possible for large entities to have ownership control. Whether it's actually a bad thing or not depends partly on whether you think the incentive structure is good enough to deter any malicious attack such as a 51% attack. If the incentives are good enough, then we have benevolent whales, or so it appears. Let's just say, for the sake of not going down a rabbit hole, I will leave it at that!

However, a rough calculation of what a 51% attack looks like for Bitcoin is interesting, nonetheless.

Let's assume a hash rate of around 200 EH/s (exahashes per second), highly subject to change since the time of writing of course.

The attacker plans to execute a 51% attack for one hour.

The hash rate required is:

51% of 200 EH/s = 0.51 * 200 EH/s = 102 EH/s

The cost of acquiring this hash rate:

Acquiring ASIC mining hardware: Assume the cost per terahash (TH/s) is $50 per TH/s.

Cost = (102 EH/s / 1,000,000 TH/s) * $50/TH/s = $5100 per second, which is $18,360,000 for one hour!

This is a simplified calculation and does not factor in other costs like electricity and maintenance.

Calculate the potential earnings during the attack:

Bitcoin block reward is 6.25 BTC per block. With a block time of 10 minutes:

Possible earnings = (6 blocks in the hour) * 6.25 BTC + fees (but not hugely significant).

You can quite clearly see that unless the BTC price makes a significant leap, the cost of the attack far outweighs the rewards! This is exacerbated by the fact that likely the BTC price would also drop once confidence in Bitcoin subsides, so that attacker would need to liquidate quickly.

Physical Power vs Non-Physical Power

Another interesting aspect as a point for discussion is the concept of physical power and non-physical power. Physical power applies to PoW and non-physical power to PoS. The issue with non-physical power in PoS is that given there are so many rules, it yields the possibility that it can be gamed, manipulated and corrupted. In a sense, think of powerful people and governments where essentially now there is a digital government. It's hard to argue this for physical power since a physical fight cannot really be manipulated except in the rare event someone loses on purpose! If someone wins a fight, it's clear and evident. It's certainly less susceptible to manipulation.

Bitcoin created something never done before, where essentially you could now have Physical Power extending across the globe, not just being limited locally!

Some would say that PoS could be more susceptible to manipulation because where there are more rules, it creates room for manipulation.

It really depends on whether you think any rules for PoS can be gamed in some manner. One such theoretical example is that in PoW a 51% dishonest majority of miners could be overthrown by the 49% honest minority of miners because energy is basically infinite. They (the honest minority) can always get more physical power. But in PoS this is perhaps not the case. Let's say you have a 51% dishonest majority of validators that were able to collude together and a 49% honest minority. In theory, there could be a situation where this honest minority may never be able to relinquish power because it's based on the number of tokens staked. In Ethereum for example, there isn't an infinite amount of ETH. What if the honest minority were not physically able to get more ETH on the market to stake? Perhaps it won't happen in practice and even if it did the community could instigate a fork of the chain, but obviously it's not a desirable solution.

This is all very theoretical but it's an open question for you to think about.

Proof of Routing Work

This is a newly emerging consensus mechanism of which there is no official term, but is commonly referred to as Proof of Routing Work or simply, Routing Work. Saito is a blockchain that uses this consensus and it was also the first, hence sometimes the term is referred to as Saito Consensus. This consensus mechanism is at the forefront of blockchain and consensus design and therefore very few projects have adopted this approach. It has not been battle tested to the extent of PoS and PoW, but it's an eye opening method that as the reader will find, is a paradigm shift in terms of how consensus can be reached to agree on transactions and blocks.

Routing Work is completely different to other consensus mechanisms and for that reason, this section will be a light introduction as it is covered in more detail in the Saito chapter. There are also various mentions throughout the book where it's compared to other consensus designs.

As already mentioned in the Introduction and Trilemma sections, it is heavily driven from an economic incentives standpoint. This is achieved by rewarding nodes for routing transactions where the rewards are distributed to the node that performed the highest amount of Routing Work, hence the term "Proof of Routing Work". The rewards are in the form of fees collected from fee paying transactions in the network. To clarify, all node types are rewarded, not just the nodes that secure the network (as is the case with validators in PoS and miners in PoW). The important thing to note is that although other blockchains pay rewards in fees and a block issuance as well as other forms of payment, in Routing Work it is fees only.

This section describes the reasons as to why only fees are paid, why it's impossible for attackers to profit and how Routing Work increases scale without trading off decentralization or security. The details are fleshed out much more in the Saito chapter, but this is a teaser that demonstrates how Routing Work will make you think on a different level altogether.

Why should all node types be rewarded?

It's widely known that for almost every blockchain, nodes receive rewards as an incentive for keeping the network running. But which nodes receive the rewards generally? The miners (in PoW) and the validators (in PoS). What are the rewards for? Securing the network. The question then becomes why is only security rewarded and why aren't other nodes rewarded for their efforts? This is a common question that goes unanswered where most blindly accept that it must be for a very good reason. Of course, security is extremely important. However, the issue with only incentivizing validators and miners to secure the network is whilst security is intact it doesn't incentivize other nodes to perform other duties and tasks to encourage decentralization and scalability. Since all the budget is focused on security, this leaves

challenges for other tasks. After all, if a full node validates transactions and maintains a complete history of the blockchain why should it do this for free? The same applies for light nodes, albeit a lesser but still important task. This problem leads to degradation of the network and users either don't upgrade their hardware (due to no compensation) or they palm it off to larger centralized entities which leads to central points of failure. If a user running a node doesn't get compensated, why would they bother upgrading hardware, except in the case of volunteer provision? This can also lead to the paid nodes free-riding on the unpaid nodes, where the miners or validators use the services of the full nodes and light nodes for free, taking full advantage and pocketing all the profit once they fill a block with transactions that were validated by all these unpaid nodes. This is known as the **Free Rider Problem of Economics**. The Free Rider problem is discussed later in the book, but essentially this is a common issue where in PoW and PoS only the miners and validators profit but other node types don't. One of the main issues here is it affects scalability because with no incentive to upgrade hardware, scale will likely suffer.

In Routing Work, since all node types are eligible to receive fee rewards this means users are encouraged to run their own nodes instead of delegating it to a cloud service or other entity to handle, which leads to increased decentralization. Generally, in other consensus approaches, the cost of running and maintaining a node can be too much to bear so many choose to run it with a cloud based service which is prone to being a central point of failure and also leads to more centralization. How about if your node is compensated? This is specifically in the case of full or light nodes which in most blockchains receive no reward. This compensation would lead to more decentralization and scalability (explained later in this section) solving the Trilemma economically as already described in the Introduction of this book.

Why does Routing Work only pay fee rewards?

This is a very unique stance on incentives for blockchains where the approach encourages nodes to prioritize routing and sharing of transactions to make sure that the infrastructure is directly funded and sustainable. Note the emphasis here on "directly". Since fees are the incentive this helps to ensure nodes are relaying transactions with fees paid by users, where the more this happens, the more the nodes have eligibility to increase their routing work to obtain the fee reward. In this way, fee-only rewards ensure all node types are compensated at market rates which can also fund infrastructure as adoption increases maintaining scale. It's very likely with increased adoption and therefore more transactions, that hardware may need to be upgraded. A lack of fee rewards would discourage nodes to upgrade leading to degradation and centralization as only the larger money players can afford it.

All this being said, why not also incentivize with a block reward issuance like many other blockchains? One of the reasons is to prevent collusion and it also enhances auditability. This is because with fees in transactions, one can always track the source of the funds because a

transaction contains a sender and receiver address, so in Routing Work the funds can be traced. With a block issuance, this is not the case because this is a special transaction called a **coinbase transaction** with no source address. Therefore, in terms of auditability it's more difficult to trace the funds. One may ask: "why is this important?". Aside from possible siphoning of funds, it can also encourage collusion because with a block issuance nodes can be incentivized to collude due to the extra reward. Despite the reward ending up with a known recipient, it may not be easy to know who coordinated to generate the reward. This collusion to obtain the reward may be difficult to track allowing them to generate funds without revealing coordinated efforts. Where did the money come from? For example, hash power can be rented out from third-parties for those that don't want to maintain their own hardware. With no source address, this could be difficult to determine though. In general, this issue tends to be less of a problem in blockchains with high hash power and decentralization because it's more difficult to dominate, but the risk still remains. Most certainly, blockchains with less hash power or concentrated mining pools have a more prevalent problem because renting of hash power can tilt control.

The next point is the absence of a block reward avoids inflation and the Free Rider Problem. The validator and mining nodes that receive this block reward are now "free riding" on the unpaid nodes that perform a lot of work such as validating transactions, which is not desirable, especially as hardware demands grow as mentioned earlier.

Block rewards incentivize mining and staking to secure the network, but not data sharing because a mining node has an incentive to hoard or withhold a block, known as **Selfish Mining**. Typically, in Selfish Mining it's blocks that can be withheld and broadcast later at a time when it benefits the miner, but also transactions with lucrative fees can be withheld. In Routing Work, withholding data is punished because the lack of sharing means less relaying of data (transactions) and so less Routing Work is performed. The concept of Selfish Mining is explained more in the Bitcoin and Saito chapters.

The fee-only model ensures all nodes benefit from user activity, not just block producers, which fosters a high-bandwidth, decentralized network. Of course, while this is innovative it assumes adoption because without adoption the volume and therefore the fees will drop. However, any blockchain project needs adoption either way although perhaps Routing Work requires a little more than other mechanisms.

How does increasing decentralization increase scale without a trade-off?

In many blockchains, a node validates a transaction and if it makes it into a block, that transaction is validated again as part of the overall block validation. Although the two steps don't do exactly the same validation for the transaction, there is some overlap. For example, before a block is created by a miner, transactions are routed to their peers and they in turn validate that transaction for:

- Signature verification to ensure the signature is valid.
- Syntactic correctness to check formatting of the transaction and conform to rules within the protocol such as size limits.
- Input validity to check that funds are not double spent.

Many other checks are validated during transaction validation which don't exist in block validation and vice versa, but the points above are checked in both individually relayed transactions and those transactions once in a block. There are still some differences though, for example in checking double spent funds it is stricter in the block validation phase. There is some good justification for this in part because it filters out low quality transactions early and prevents wasting mining resources. It also ensures transactions are thoroughly vetted before and after they are confirmed. Nonetheless, overlap exists, so there is opportunity for optimization here.

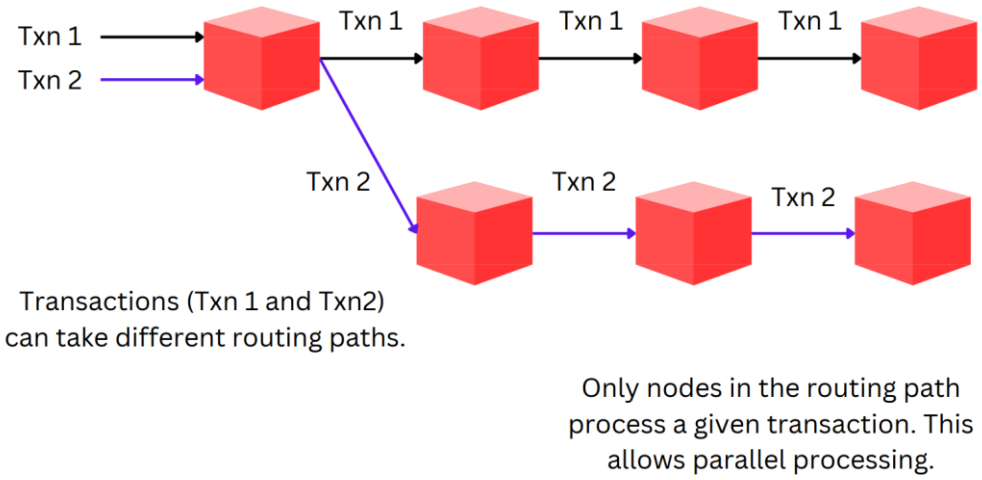
Essentially, all nodes need to validate all transactions and then again (albeit with some differences in the checks) once inside a block. With Routing Work this is not entirely the case. This is because Routing Work stems from transactions being relayed across a particular **Routing Path**. So, before a block is produced and broadcast, only the nodes in the routing path need to process and validate that transaction. In most consensus mechanisms, all full nodes need to process and validate all transactions, but in Routing Work only a subset do. It's a little like saying in today's Internet infrastructure "*not every node has to read the whole Internet to use it, just the parts they care about, but the core infrastructure keeps the web running*".

However, this is the case before a block is produced (the pre block production phase). Once a block is produced, all nodes still need to validate the transactions in the block, just as all blockchains do. Hence, there is still overlap in the validation of transactions before a block is produced and again for those transactions once in the block. The key difference here though, is only a subset of nodes (the ones in the routing path) that validate the transactions have this overlap. This is a quite a significant difference and thus allows more scale as nodes in different routing paths can process different transactions allowing parallel processing. This essentially also distributes the load of transactions being processed across many nodes and routing paths in the network. This in turn, means as more nodes are added to the blockchain (hence more decentralization), this assists to increase scale and performance! To clarify, as already mentioned, this optimization is for the phase before a block is produced.

This reduces bottlenecks and speeds up the inclusion of transactions in blocks and so this selective approach reduces the computational burden on nodes as they don't need to handle the entire transaction volume. This means that since not every node needs to reach consensus on each transaction before a block is produced, it allows faster propagation on its journey to final confirmation when a block is eventually created. It should be noted that although this makes it more resilient to congestion, it does rely on the availability and integrity of the nodes

in a given routing path. Suffice to say, in terms of integrity, you will find in this section that any attempt to compromise integrity is heavily disincentivized.

The following illustrates this:



In many consensus mechanisms, an issue arises whereby increasing nodes increases decentralization but this can limit scalability due to the need for coordination of all the nodes during validation. This again comes down to the Trilemma trade-off. Some blockchains however, mitigate this by distributing workloads such as using sharding or layer 2 solutions to offload transactions from the base layer. Some of these solutions are outlined in later chapters of this book. Proof of Routing Work consensus does not require layer 2 solutions because it's all built into the base layer. Therefore, it scales as more nodes are added to the network. This comes down to the economic solution to the Trilemma it uses, being a combination of incentivizing all node types as well as a technological solution to process transactions in the context of routing paths as just described.

How fee revenue is maintained at scale

As mentioned in the Trilemma section, there can be issues at scale where fee revenue for miners drop as performance increase leads to cheaper gas fees. The elegant solution is to use Automatic Transaction Rebroadcasting (ATR) where any user's wallet that contains a sufficient number of tokens automatically rebroadcasts a previously pruned transaction. The fee paid is higher than the average market rate which maintains consistent fee revenue for nodes in the network. This also helps to compensate the nodes for their service assisting them for funding hardware upgrades for example.

This removes the fear present in many blockchains where reduced block rewards in future leave fees as the main source of revenue, raising the question of whether this will be sufficient. In Routing Work, the pruning of data keeps the blockchain compact while transactions that need to persist are rebroadcast. The fee paid by users is initially a small fee rather than a single larger upfront fee to persist the transaction on-chain forever. Instead, the fees are smoothed over time where small amounts are paid in increments should the transaction be persisted. If there is insufficient balance in the user wallet to pay a rebroadcasting fee, then the pruned transaction is not rebroadcast. In many blockchains a fee is paid for a transaction to remain forever on-chain, but data for this transaction can be pruned. Why should users pay high fees for transactions up front especially if they are eventually pruned? Why should users suffer from performance issues due to blockchain bloat? Why should node operators suffer from this bloat? In ATR fees are market driven and users only pay fees when they want their data to persist, avoiding over-payment for short-lived transactions. These fees are paid to reward nodes that route transactions, not just miners, validators or centralized pools, thus incentivizing the collection and sharing of data. The more transactions routed efficiently the more the rewards for nodes that perform high Routing Work.

It should be noted that transactions are not deleted outright from the chain itself, but instead the associated UTXO (unspent transaction output) data. More detail on the technical side of how this works is explained in the Saito chapter.

How does Routing Work make it impossible for an attacker to profit?

One ultimate question that arises in Routing Work is the question of security. If there is no block issuance to incentivize nodes, how is the network secure? The very short answer is it's extremely secure and could be argued more so than other consensus mechanisms.

Although an attacker can attempt an attack it will never be profitable and thus 51% attacks will never succeed as is the case with Sybil attacks and in fact, any type of attempted attack. The details of this are outlined in the Saito chapter, but in essence the main reason is because fees fund the network and in turn this rewards nodes accordingly. An attacker has to spend his own fees routing his own transactions and even if the attacker is successful he receives the fees as rewards, but those were his own fees he spent! In the best scenario, he gets the money back but doesn't make any profit. In most scenarios, the situation is even more bleak, because the reward distribution mechanism allocates 50% of the fees in a block to the Routing Work winner and 50% to a miner that unlocked a **Golden Ticket** by solving a puzzle. The latter is what actually unlocks the funds. Therefore, an attacker likely gets only 50% of the fees back. Only in the case that the attacker also unlocked the Golden Ticket does he get all the fees back, but the costs of mining and hash power puts him at a loss. There are some ways an attacker can attempt to profit by only spending a portion of fees by making a certain percentage of transactions and letting other honest nodes put their fees in a block. The attacker could then

take the “honest” fees as well as his own fees. This still will not work because there is a payout cap that occurs if a huge spike in fees arises when an entity tries to attack. The algorithm is able to distinguish between a spike in fees from an attacker and a spike from honest nodes.

Another innovation is **Routing Signatures**, which are used to sign transactions that are relayed in a routing path from one node to another. An attacker trying to create fake transactions will need to know the private key of nodes in a routing path to create a signature and even if he knows that he will need the private key for each node in that path. Good luck with that! Even in the extremely hypothetical situation he is able to do that, there will need to be a fee paid for those transactions. Essentially, no matter how hard the attacker tries, he spends more and more money putting him at a larger loss, effectively directing the fees to honest nodes that ultimately will receive the reward.

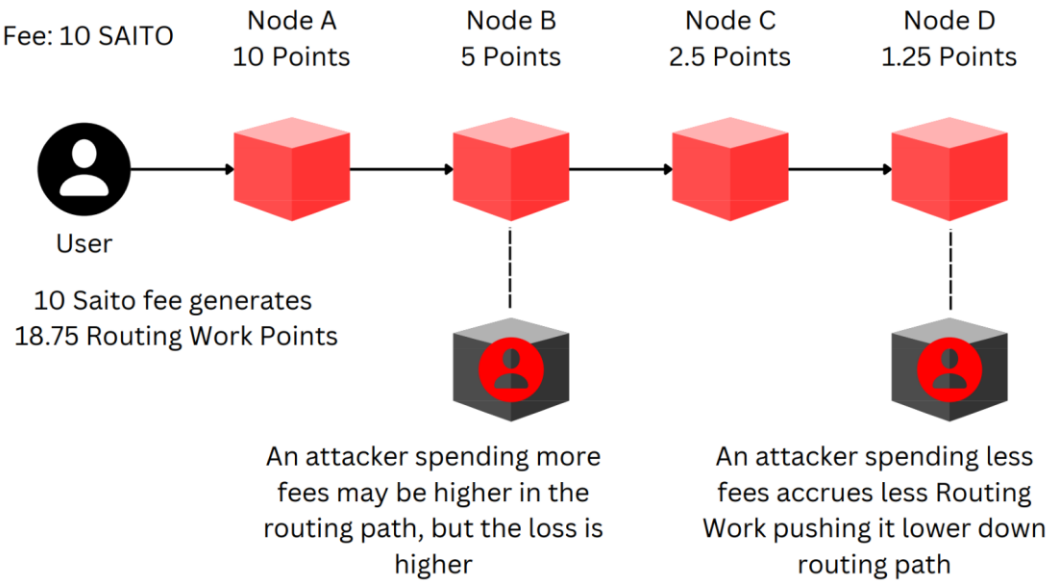
The Golden Ticket allocation is explained more in the Saito chapter, but one reason it’s so difficult to attack the network is because mining is divorced from block production. This is a very unique approach in blockchain consensus because usually mining is tied to producing blocks. In Routing Work, mining is **not** for block production, but only to unlock the funds. The block is produced by the node that has the highest routing work, not the mining node that solved the Golden Ticket puzzle. This is quite an incredible innovation to prevent attacks! This also solves fee recycling attacks where an attacker attempts to recycle fees to fund an attack for a following block. Routing Work only unlocks rewards for the previous block, which essentially can deter an attacker partly due to the delay of rewards, but also the divorcing of direct rewards from the current block.

If we take it back to Bitcoin or Ethereum, there is a cost whereby an attacker can profit in the same way an honest node can profit. This is called symmetric cost of attack. This essentially is down to the economic incentives in the network and the cost of attacking a blockchain compared to the possible rewards that can be gained. In Routing Work, this attack is an **asymmetric cost of attack** because honest nodes route transactions efficiently at a small cost, but attackers must burn their own money in fees and hash power, thus making an attack irrational in an economic sense. This is down to the brief description earlier that an attacker can only at best get 100% of the fees back, but more likely it will be less. In Bitcoin however, the incentive model is different and so the expenditure for an attacker (hash power) is almost equivalent to the expenditure for honest miners to produce blocks. For example, if an attacker gets 51% control of the network, hence hash power, they could in theory earn 51% of the block rewards. This means that the cost of attack is proportional to the gain in potential rewards. Although, in practice it may not be as simple as this (refer to a calculation for an attack on Bitcoin in the “Mining Pools vs Staking Pools” chapter) and there are other variables at play, this is still theoretically possible. In Routing Work, it is not even possible in theory once you do the math.

The fundamental reason as to why in Routing Work attackers have to burn fees is that the Routing Work points in a routing path are halved with each hop, which essentially halves the

fees. This is illustrated in the Saito chapter, but in summary the node highest up in the routing path has the highest eligibility to receive the fee rewards. Therefore, for an attacker to be profitable they need to lower costs for routing transactions to have any chance of making a profit. This requires performing less transactions with their own fees and also relying on some of the "honest fees" produced by honest nodes. If they spend all their own fees on their own transactions the cost is too high, which is why an attacker may be tempted to spend a little but hope that the fees generated by honest nodes will be sufficient to obtain some profit. However, this pushes them deeper in the routing path and since the Routing Work points are halved with each hop in the path, they now have less Routing Work points assigned to them. The other way an attacker can approach this is to compensate by paying higher fees initially to offset the Routing Work points being halved on each hop. This may push them higher up the routing path, but this also mathematically leads to the attacker losing as they have to spend more and more fees to maintain enough Routing Work. To emphasize again, the reason it's important to be at the top of the routing path (closest to the user) is because a node at the top of the routing path has a higher routing work score and therefore more eligibility to claim the fees in the block.

The following illustrates an attacker, at whichever point in the Routing Path can never succeed:



To clarify a little more, the notion of being "pushed" deeper down the routing path doesn't physically occur. This is just a metaphorical way of explaining it so the reader can visualize the situation more easily. The location of the node in the routing path for a given transaction doesn't change in a literal sense. However, this just helps the reader to picture in their head the idea that an attacker node essentially becomes a lower priority in the Routing Work points

rankings. The Routing Work for all nodes and their transactions is calculated before a block is produced where the node closest to the user has the highest probability of claiming the fee payout. Therefore, in general terms the node highest in the routing path has the highest potential claim and the deepest node has the lowest claim. An attacker can increase their routing work score if they route more transactions with lucrative fees despite being physically lower in the routing path and this may win them a payout, albeit at no profit as already mentioned. The metaphorical description of being "pushed" deeper in the routing path is from the view of the summary of all transactions that have occurred so far. You can view it as a routing path summary of all transactions in your head, or simply as a summary of routing work points accumulated across all transactions thus far (the latter being the logic actually used in Routing Work).

Usually though, as the network gossips transactions to other nodes, if they suspect suspicious activity such as a node dropping transactions (maybe the attacker wants to insert his own transactions to increase his routing work score) or not being efficient (maybe it's colluding), they mark it in a black list and red flag it. This results in the honest nodes routing around that suspicious node as they want to route transactions more efficiently to increase their routing work. The important point here is that the incentive is to route transactions as quickly as possible. This is what makes Routing Work very unique because it discourages collusion and inefficiency. Any time spent colluding is time spent not sharing transactions and time is money. This essentially means in a metaphorical sense the attacking node is "pushed" deeper down the routing path because it's ignored by many honest nodes. In a physical sense though, for a given transaction, it remains at the same point in the path, but the overall routing work score for the attacker decreases because it's collecting less transactions and therefore relaying less transactions to other nodes.

Symmetric vs Asymmetric attack

It's important to understand why symmetric forms of attack, such as in Bitcoin, mostly use no extra hash power compared to honest miners. Let's say an attacker wants to selfish mine by withholding blocks:

- The miner withholds block B1 and starts mining block B2 using their hash power.
- The difference is strategic because they are mining on their own private chain instead of the public chain.
- If the miner succeeds in mining B2, they can broadcast both blocks B1 and B2, where the network will adopt this chain if it's the accepted longest chain.
- This doesn't need additional hash power beyond what they mine normally. It's just a reallocation of effort.

The risk with this form of attack is than an honest miner may build a longer public chain accepted by the network before the selfish miner broadcasts, rendering the private chain obsolete.

The overall point here is the issue with symmetric attack is that malicious miners essentially mine at the same cost as honest miners. In Routing Work, as mentioned earlier, attackers spend more than honest nodes hence this is asymmetric, which keeps the network very secure.

Can a node be placed at the top of the routing path to control the network?

This is a very interesting thought experiment and a question many new to Routing Work ask. It's an obvious question with a very unobvious answer. This requires a fairly advanced understanding of Routing Work, so it's described more in the Saito chapter. However, in the spirit of sparking curiosity and interest, as the reader, this will help to be a fascinating thought to ponder as you work through the book.

The perceived problem with an operator strategically placing a node at the top of the routing path to gain control in a centralized manner to then control a majority of routing work appears to be a real issue. In practice however, this would be extremely difficult to pull off as there could be potentially hundreds or more routing paths as users initiate transactions across the network. Therefore, the node operator would have to be constantly placing their node in such a way that it's at the top of the routing path, being closest to the user for any given transaction. That would be practically impossible, especially with high adoption and volume of transactions! Though, the node at the top of a routing path could still make lots of their own transactions to garner more even more routing work, thus possibly maintaining its position and ranking in routing work points. Either way, let's imagine this were possible, essentially rendering this a hypothetical scenario.

As an example, consider a single central node controlling 80% of all blocks (as it has accumulated enough routing work points and therefore rewards consistently). What is this node incentivized to do? Remember the node that has built up enough routing work points is eligible to be a block producer and receive the fee rewards. It actually turns out that the network still benefits because the central node allows its peers to produce the remaining 20% of blocks. At first, it seems counter intuitive because one may ask why would it not just attempt even more control and swallow up even more control of blocks to receive more rewards? The reason this could be a bad idea is because this would likely leave most other nodes with no other option but to leave the network due to lack of fee revenue as they are receiving less than 20% of fee rewards. If they then leave the network this means fee revenue overall could decline as there are less nodes to process all the transactions, giving more load and strain to the controlling node. This results in a bottleneck as the dominant node struggles to handle the full load alone.

Remember, in Routing Work the incentive mechanism is all based on nodes collecting and sharing transactions efficiently. If this doesn't happen, poor network performance also leads to the users who make transactions to seek alternative networks. This lowers the pool of fees, which results in less rewards for the controlling node!

The other issue is if the controlling node has too large a share of the fee pool, it risks external nodes attacking it such as a DOS (Denial of Service) attack. This brings the node to a halt thus diverting all the transactions elsewhere so that other nodes can route transactions to receive a share of fee rewards.

At this point, this is a high level description but it goes much deeper in the Saito chapter. Essentially the controlling node is incentivized to be a benevolent actor by balancing overall network health with its share of the rewards. If it gets the balance wrong, it risks congestion, network degradation and less fee revenue for itself. This is a very surprising outcome as in most cases a central actor could act maliciously at the expense of network health and network rewards! In this case a central actor is actually incentivized to act in the overall interest of the network by allowing other nodes to participate and share the rewards. This is where the sharing of transactions is key in Routing Work and so the interest is a collective group interest rather than acting in individual interests. The more nodes that share transactions efficiently, the more the rewards for everyone involved. A central actor in other blockchain consensus mechanisms doesn't have the same incentive to act honestly because it's not based on the efficient routing of transactions directly. As you will find in the Bitcoin chapter and beyond, there are incentives to withhold lucrative transactions or blocks in Proof of Work especially, albeit still difficult to achieve successfully.

To emphasize again, this is a very hypothetical scenario and would likely never happen, but it shows the strength and resilience of this consensus mechanism.

Public Goods and how Routing Work funds them

A public good is a commodity that is non-excludable and non-rivalrous. For example, nobody can be excluded from street lights or clean air whether they pay for it or not. The non-rivalrous property is where one person's use of a good doesn't reduce the availability to others. If someone enjoys a firework display, it doesn't diminish the enjoyment of others watching.

The issue is public goods face challenges because users can free ride without paying, leading to degradation of the good or benefits extending to beyond those who pay which means individuals have less incentive to contribute.

Proof of Routing Work solves this issue because the protocol funds the public good, being the decentralized network infrastructure. This is because the fees and value flows to those who maintain the infrastructure rather than relying on third-parties or centralized entities to subsidize it. Transactions generating fees are distributed to nodes that provide valuable

services such as routing data or validating transactions. In other consensus mechanisms it's just for validation and even then, the full nodes performing validation are not compensated by the protocol, but rather have to be set up voluntarily or handled by a separate entity. Routing work aligns with a concept called quadratic funding where contributions are rewarded in a way that reflects their value to the network. This is key to maintaining a scalable and decentralized network as these two properties can then be paid for which essentially solves the Trilemma economically.

Public goods, private goods and collusion goods

A very interesting aspect from economics which applies to all blockchains is the concept of public, private and collusion goods. This brings three different types of utility to blockchains. It's an extremely important set of properties that Routing Work solves, where currently, many other blockchain consensus mechanisms find challenging. Let's describe why this is.

A **private good** is essentially the opposite to public goods where instead, a private good is rivalrous and excludable. Think fees paid for transactions to be included in a block, where in many cases a higher fee guarantees your transaction to go through more quickly. This higher fee essentially excludes others due to competition for fees.

A **collusion good**, while not a standard term in economics, is what a user receives by selling their fee privately. The meaning of "selling" a fee is specifically where users bypass the public fee market by privately negotiating with miners or validators to prioritize transactions for faster confirmation. The most classic example of this is MEV (Maximum Extractable Value) in Ethereum where users can get discounts on fees or kickbacks from a sale via private channels. This is carried out by users colluding with validators to prioritize their transactions, usually for profit by front-running or arbitrage. More about MEV and these terms are described in the Ethereum chapter.

This kind of collusion also unveils an issue known as free-riding where fewer fees then contribute to public goods such as network security (which is funded by burned fees in Routing Work). If more value flows into collusion goods then users and miners profit at the expense of fairness and decentralization of the blockchain, which are the public goods that the network should benefit from. The free riding specifically is where users benefit from these public goods while extracting private gains via collusion. Therefore, selling your transactions privately means your transaction isn't shared, at least not immediately, so the recipient receives more of the fee than they would otherwise. These private sales are funding these collusion goods, giving the miners or validators more profit and in turn, giving the user extra benefits.

Note that collusion can also occur between miners colluding with other miners via private channels or validators in the case of Proof of Stake consensus where there is potential to bribe other nodes for rewards. However, this doesn't fit under the bracket of a collusion good as this

provides no direct utility for users. The public, private and collusion goods are all utility for users making transactions in the network. Collusion and other forms of activity to benefit nodes at the expense of others are all challenges blockchains face. These are challenges that Routing Work readily solves with its consensus mechanism built based on economic incentives. These challenges are explained throughout the book, but it's separate to being a collusion good.

A **public good**, as already described is non-rivalrous and non-excludable. Since in Routing Work, fees need to be paid, it could be argued that any blockchain using Routing Work is not truly a public good. Only in the case of a fee-free blockchain is it perhaps strictly a public good as nobody is excluded. Therefore, it could be seen as a private good with public benefits as the network benefits as a whole because fees are shared via the pay split, for example, 50% to the Routing Work winner and 50% to the Golden ticket winner. A portion of the fees for routing work are also paid via Automatic Transaction Rebroadcasting where nodes rebroadcast previously pruned transactions. All node types are eligible for fee rewards, therefore, this means that the public good becomes a private good with public benefits. This is all somewhat debatable and is more an argument for economic purists. Suffice to say, Routing Work consensus itself can be viewed as a public good because having consensus pay routing nodes directly means nodes and users don't need to engage in collusion to pay for network infrastructure where money would eat away at the blockchain's public utility.

In Routing Work consensus it's the open and competitive aspects of blockchain provision which is the public good. For example, it's the open competition where any node can join and mine, thus elevating openness and decentralization where the non-excludable property is held intact. Nobody is excluded, thus it's the non-excludable form of utility that scales with publicly circulating transactions in a block, where all transactions are shared quickly. Essentially, this should be the case with all blockchains, but as you will find, there are many challenges such as collusion between users and nodes, collusion between nodes themselves and withholding of blocks in other consensus mechanisms that compromise this public good property.

The following illustrates the different types of goods and how they provide utility both on-chain (publicly) and off-chain (privately):

Public good

Economic security

Openness

Decentralization

Private good

Transactions in block

Collusion good

Kickback from sale

Fee discount

ON-CHAIN FEE

OFF-CHAIN FEE

Proof of Work Chains

Bitcoin

Brief History

Bitcoin was created by an anonymous person or team of individuals under the name Satoshi Nakamoto. A domain was registered in 2008 under bitcoin.org and a whitepaper was released later that year. The motivation behind Bitcoin was largely centered around the Global Financial Crisis and centralized control that many large corporations have. There was a hint based on this when Bitcoin went live on January 3, 2009, where the first message in the genesis block said, "*The Times 03/Jan/2009 Chancellor on brink of second bailout for banks*". This marked the first decentralized cryptocurrency, BTC, in response to a large distrust in legacy financial systems. No longer would someone need to completely depend on large powerful financial institutions, but rather one can now hold their own wealth in a wallet in a self-custodial fashion under their complete control. Nobody except the owner of that BTC can withdraw the funds from their wallet because only the owner knows the key to access it.

Bitcoin aimed to provide a secure, transparent, censorship resistant and ultimately decentralized alternative to the banking system, which is largely seen as a black box. No central authority controls Bitcoin, unlike banks and institutions, because the Bitcoin blockchain is a peer-to-peer network and therefore requires no trust of any institution or middleman.

Bitcoin has a total supply of 21 million BTC coins where this maximum supply is expected to be reached by the year 2140. BTC is created as a reward for miners who secure the network and validate transactions by solving complex mathematical puzzles. This reward halves every 4 years in the event called The Halving. This essentially lowers the inflation rate of BTC, contributing to its scarcity.

Since Bitcoin was the first blockchain, this chapter will describe some blockchain fundamentals that helps to lay the foundation for the rest of the book. It may make the read easier somewhat for beginners who have already watched a video on basic blockchain concepts.

What Problem is Bitcoin Solving?

There are some that say, "Bitcoin fixes everything!" or "Bitcoin solves the world's problems!". This is likely an exaggeration stemming from sheer bullishness and that's ok to some extent as people get excited, but it should be taken with a pinch of salt. Bitcoin solves many problems, but it's not going to fix the world! Moreover, it was created in response to many issues regarding fiat currency and the traditional financial system and is designed to solve many shortfalls. Some of the issues are relevant to many blockchains that were built after Bitcoin, and some are unique to Bitcoin, but some of these properties and solutions are subject to debate. As the reader you will likely have your own view. Suffice to say, let's explore these one by one as per the following:

Decentralization: Bitcoin operates on a decentralized network of computers rather than a central authority such as is the case with the banking system. This means no single entity controls the network which reduces the risk of corruption, censorship, and failure. The latter is quite significant as there is no central point of failure as nodes are distributed globally in a peer-to-peer fashion. No middleman is present in a Bitcoin transaction meaning that huge value can be transferred quickly and for a very cheap fee. Some critics question the value of Bitcoin, but it surely can be said that the ability to send money around at high speed (even if this requires a separate scalability layer) for a very cheap fee is solving something that traditional finance is not. This must be a valuable proposition in response to those critics.

Sound Money: Bitcoin has a finite supply of 21 million BTC coins unlike fiat currencies that can be printed without limit. This limited supply mimics the scarcity and durability of precious metals like gold, which have historically been sound forms of money. However, it could be said it's even better than gold because one can always find more gold, but nobody can discover more BTC because it's programmed in the code! It's also more divisible and portable than gold, but more on this in the later points in this section.

However, despite initially being a solution as peer-to-peer cash, Bitcoin has more emerged to be a store of value which makes it more comparable to gold than fiat currency. This is mainly because Bitcoin doesn't scale at the base layer as it currently only achieves 7 transactions per second, which isn't suitable as peer-to-peer cash. There are other scalability solutions such as Lightning and RSK that solve this, and these are discussed later in this chapter. The current narrative is a store of value, but it remains to be seen if it will become peer-to-peer cash with other scalability solutions or perhaps even both.

Immutability: Once a transaction is confirmed on the blockchain, it cannot be altered or removed. This is the property of immutability and ensures the integrity of transactions with a history going all the way back to the first transaction, making Bitcoin great for traceability and auditability.

Censorship Resistance: Transactions in BTC can be made by anyone globally without the need for permission from banks or governments. This is because a BTC wallet doesn't require any KYC (Know your customer) or ID checks which renders it difficult for authorities to censor transactions or freeze accounts. Of course, if the wallet is on an exchange, then we are essentially back to the banking system because these wallets are not self-custodial as they are owned by the owners of the exchange which means they are custodial (not self-custodial) wallets. These custodial wallets would require KYC and ID checks as part of exchange verification. However, Bitcoin was designed at its core to be peer-to-peer and self-custodial

meaning that a wallet is owned only by the wallet owner as only they have the key for the wallet to approve transactions and withdrawals.

Divisibility: Bitcoin is highly divisible, where each BTC coin is divisible into 100 million smaller units called Satoshis. This divisibility makes Bitcoin accessible for small payments for a very cheap fee, which is not always possible with traditional currencies. Again, as mentioned, due to scalability constraints small payments with BTC are not very practical but solutions such as Lightning, Tectum and other layers are working to address this.

Portability: BTC can be transferred rapidly and easily across borders, making it a global currency that isn't bound to a nation or any regulations. This can be compared to gold which is less so the case. It is true that large amounts of gold are not usually transported but rather there is a transfer of ownership in the form of a receipt. However, smaller amounts of gold, which are very valuable, are still at risk when transported and above a certain amount normally needs to be declared at airports. This is not the case with Bitcoin because all one must do is carry a digital wallet (be it a hot wallet or hardware wallet) which is extremely portable. On a final note, in simple terms, since the key for a wallet is a set of words, these can be memorized and stored in your head, meaning that your BTC is extremely portable!

Transparency: The blockchain is a public ledger, meaning all transactions are visible and verifiable by anyone. This transparency helps prevent fraud and things like money laundering. Critics argue that Bitcoin is used for money laundering, but this is a gross misunderstanding of how the technology works. Perhaps a tiny fraction is laundered, but any attempt is thwarted with huge challenges because Bitcoin (as with any blockchain) is a public ledger for all to see! The Bitcoin ledger does not induce these issues but rather solves them.

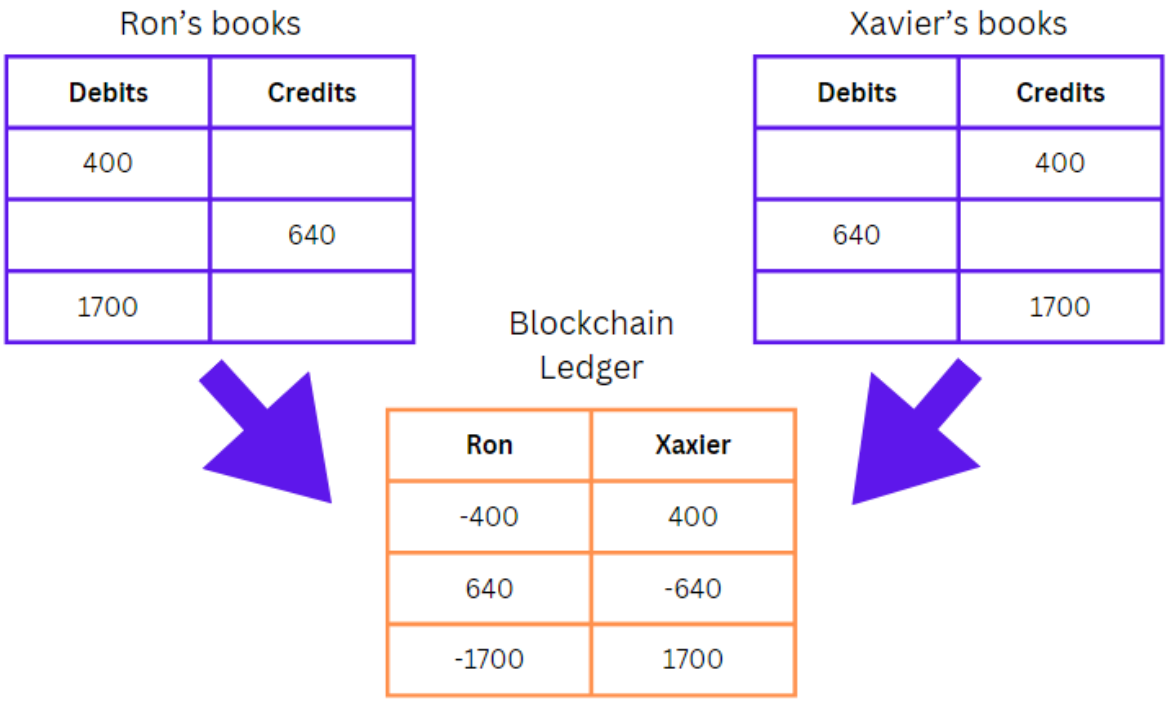
One such benefit that blockchain technology created (where Bitcoin was the first) was that of **triple entry accounting**. All subsequent blockchains of course benefit from this trait. To understand this, let’s first understand briefly what this is and how it arose, and what problem it solves.

Back in history, records of transactions were kept using single entry accounting where each financial transaction was tracked as a single entry to view the flow of money. This, however, didn’t provide a comprehensive view of the assets or liabilities for a business. Consequently, **double entry accounting** improved upon this by recording each transaction in two separate accounts, being a debit in one account and an equal credit in the other. This provided a balance sheet and resulted in a more systematic approach to bookkeeping to help detect errors and fraud. With the advent of blockchain technology, triple entry accounting adds a third entry which is secured cryptographically and serves as an immutable and verifiable record. This third

entry specifically is the unique digital signature that is signed for a transaction by all parties involved. A blockchain records every transaction in a block where the current block is linked to the previous block, creating a chain of blocks that is visible to all users. This makes transactions easier to trace because all records are linked in a chain going all the way back to the first (or genesis) block ever created. This can provide real-time verification of transactions and reduces financial manipulation.

The bonus with blockchain technology is that the transaction from one party to another is all transparent to the public, but in addition to that, the whole transaction from the sender to the receiver is viewable in a single place within a block. This is not the case with double entry accounting because there are two accounts, where to track the flow of money and see the full scope of a transaction, an accountant must check the debit account and the credit account to tie them both together as they are in separate places. So blockchain simplifies the approach because there is no need to cross-reference separate accounts to piece together transactions.

The following illustrates double entry accounting vs triple entry accounting:



It's clear to see that both books are essentially merged in the blockchain format. In the double entry approach each bookkeeper has their own view to maintain, but in the blockchain ledger triple entry approach all the credits and debits can be seen in one transaction (where each

transaction is a row in the ledger above). This allows smoother tracking and traceability of transactions.

Security: The Bitcoin network is secured by huge amounts of computational power, making it extremely resistant to attacks and fraud. Another level of security is based on cryptography, which is a technology that existed way before blockchain. In essence, cryptography allows transactions to be verified using a form of encryption where a pair of keys are used, being a **private key and a public key**. The sender of a transaction signs and approves a message with his private key to create a **digital signature**. Any other blockchain node can verify the digital signature (to check that the sender is legit) by using the public key that is mathematically related to the private key (hence the key pair).

Thus far, since inception the Bitcoin blockchain has never been compromised. The mechanics of this is described in the "Bitcoin's Proof of Work Consensus Mechanism" section.

Many of these properties just mentioned are arguable. For example, Bitcoin as sound money is subject to large debate in the community and beyond. Many believe other blockchains and cryptocurrencies are better as money. Likewise, decentralization is widely debated, not just for Bitcoin but for many other blockchains, because decentralization is a spectrum. How many nodes need to participate or vote in consensus to be decentralized? How much ownership of a token is required by many users or entities to be decentralized? There is no definite answer as such, and it is quite subjective. However, there are calculations such as the Nakamoto Coefficient that measure the level of decentralization for a blockchain from a technology perspective. In summary, this measure calculates the minimum number of entities that would need to collude before the blockchain security is compromised. However, some say this measure has some potential issue and there are other ways to measure it, but nothing completely definitive. The Nakamoto Coefficient is about the best we currently have.

Overall, it's clear that Bitcoin aims to offer a form of money that is decentralized, finite in supply, immutable, censorship resistant, divisible, portable, transparent, and secure. These properties address many shortcomings of the traditional finance system. Perhaps Bitcoin is not perfect because, for example, it currently doesn't scale very well at the base layer and the BTC price is very volatile. This makes it a challenge for things like micro-payments at scale and borrowing and lending. However, there are many solutions being built on other layers on top of Bitcoin to address these issues. These are described later in this chapter.

Bitcoin's Proof of Work Consensus Mechanism

As mentioned in the chapter on Proof of Work, this is a solution to the Byzantine Generals Problem. However, PoW solutions arose before Bitcoin and the advent of blockchain technology. One such solution was HashCash developed by Adam Back which was a PoW system to prevent spam and DoS attacks. With HashCash, in the effort to prevent spamming by sending a huge number of emails, the sender of an email needs to perform some computational work. This is very easy for an honest sender but not of course, for a spammer. Satoshi Nakamoto adapted this concept to then help Bitcoin secure the network and produce new blocks in the blockchain and verify transactions.

PoW also solves the Byzantine Generals Problem which is a scenario where many participants must come to consensus where there may be some rogue parties not cooperating properly. In a decentralized world this is very difficult to solve but PoW makes sure that even in the context of malicious behavior the true state of the chain is agreed upon by the honest nodes. This provides an effective way to reach consensus without requiring a third party because mining nodes need to expend energy and resources to mine and broadcast blocks which means it's not cost efficient to attempt fraudulent transactions. It should be noted that if a node attempts any such behavior, it's not explicitly punished, but rather it has punished itself by wasting lots of time and energy. Other nodes will simply reject the block as per consensus rules and it may even be red flagged by some nodes resulting in blocks from the malicious node possibly being ignored in the future. The steps in the consensus mechanism will help to understand this more in this section.

In summary, PoW is a fundamental approach to allow nodes to agree on the state of a blockchain in a decentralized manner to prevent fraudulent activity such as double spending transactions and therefore, maintaining the security and integrity of the blockchain without requiring a central authority.

As a precursor to describing how PoW consensus works, let's first understand the difference between mining nodes, full nodes and light nodes (SPV nodes).

Mining Nodes collect transactions from a mempool (a temporary store of transactions submitted by other nodes) and create new blocks. They select these transactions based on the fees as they want to collect as many fees as possible for rewards. Once it has produced a block by mining, it then broadcasts it to the network to then be linked on to the blockchain.

Full Nodes also collect transactions from the mempool and maintain a version of the mempool. They verify incoming transactions and propagate them to other nodes. They do not create blocks however, and therefore don't partake in mining blocks or expending lots of energy or computational resources. They make sure that only valid transactions are broadcast and so play a key role in the network's security.

Full Nodes maintain a full copy of the blockchain unlike **SPV (Simple Payment Verification) Nodes** which don't maintain a full copy but rather contain just the headers for each block, which is still very important for verification without requiring a full download of the whole blockchain (to conserve memory). This can be more useful for lighter clients or mobile devices and they can still verify that transactions in a block are included as the block header enables this verification.

The question becomes what is mining exactly and how are blocks produced and how is consensus reached? The following steps illustrate this and how PoW consensus works:

- **Collecting transactions**: Nodes (full or mining) process and verify transactions and submit these to a mempool, which is a temporary store. Nodes then collect transactions from this mempool.

- **Producing a new block:** Mining nodes start to form a block and so select transactions from the mempool and this is prioritized by many factors such as transactions fees for example.

- **Calculate Merkle Root:** A Merkle Tree of transactions is created which is a data structure that provides a summary of all transactions in a block. This is done by essentially hashing pairs of transactions. The **hash** at the top of the tree is called the **Merkle Root** and this is included in the header for the block. Hashing is described in more detail later in this section on Merkle Trees and Merkle Roots.

- **Prepare block header and start mining process:** The header for a block contains the version, previous block's hash, the Merkle Root, a difficulty target, a timestamp and a nonce. The **nonce** is a large 32-bit number that miners change during the mining process to generate a hash that solves the mathematical puzzle required to mine (or produce) a block. Miners then try to generate a valid hash value that meets the difficulty target for the network which consists of continuously hashing the block header while changing the nonce until a hash is found that is less than or equal to the target difficulty.

- **Target difficulty:** This is a number that adjusts every 2016 blocks (every 2 weeks) to make sure that the time between blocks is about 10 minutes. A lower target means

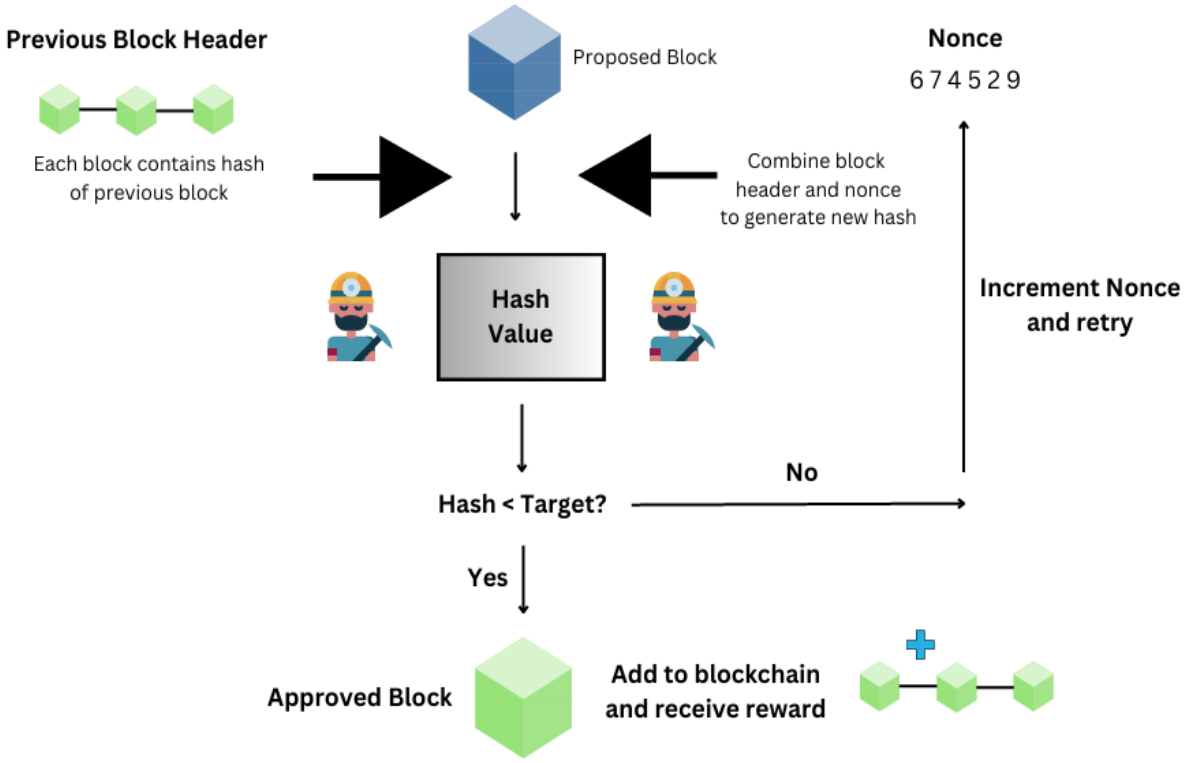
higher difficulty. The more miners in the network the higher the difficulty and vice versa. This is known as the **Difficulty Adjustment** in that as more miners contribute computational power then it's likely that a valid hash (solution to the mathematical problem) will be found quicker than the 10 minutes target. Therefore, this increase in **hash rate** renders a higher difficulty for the next adjustment.

- **Find valid hash:** Upon finding a valid hash, the miner has now successfully mined a block and thus has proved that it has performed the work required by the network, hence the name Proof of Work. Upon the next block mined after this current one, this hash is inserted into the field in the block header called "Previous Block Hash" so serves to be the hash that the next block mined will reference (to link the blocks in the chain).

- **Broadcast block to nodes:** The miner broadcasts this new block to the network and nodes receiving this block verify the transactions in the block, the block's hash and other information in the block such as the block size and that the block is syntactically valid as per the consensus rules.

- **Confirmation of the block:** If the block is deemed valid, then all other nodes link it into their copy of the blockchain. The miner is rewarded with a block reward in BTC and transaction fees from transactions within that block. The process repeats in the next 10-minute window for the next block where the hash of the confirmed block serves as the previous block hash for the next block header in future. This is what links each block together in the chain. Think of it like a pointer to the previous block where that previous block also points to its previous block going all the way back in history to the first ever block created.

The following illustrates PoW in action where miners continually try to generate a hash value that meets the difficulty target:

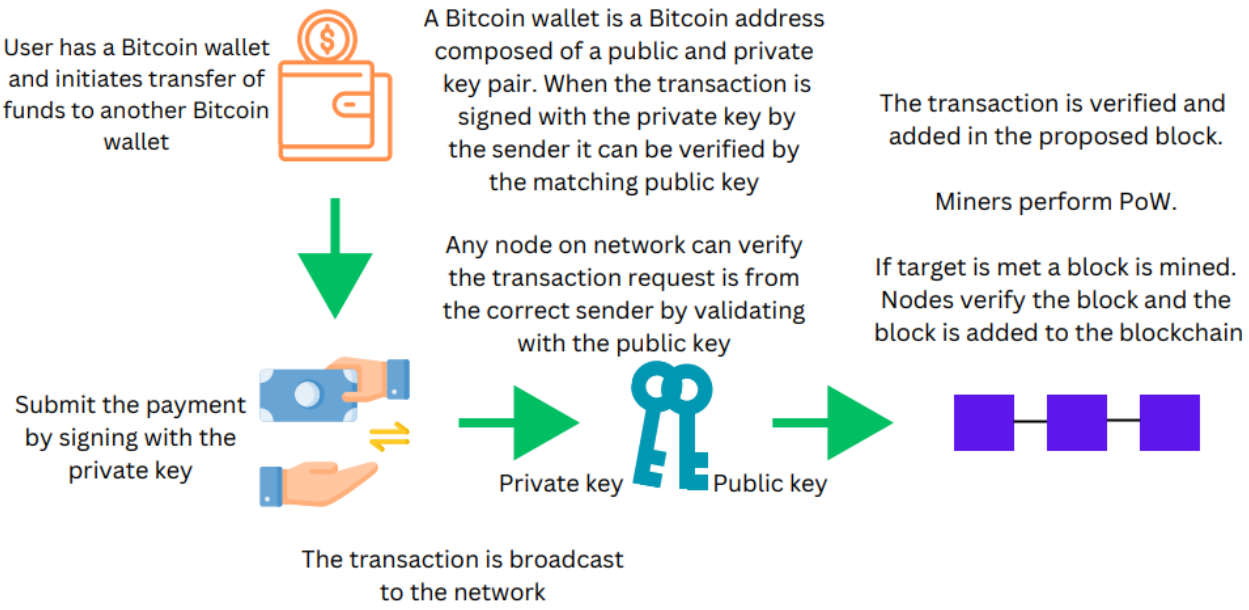


One may now ask how is Consensus reached? For example, in PoS it's basically a vote where a 2/3 majority deem the block as valid. In Bitcoin (and therefore PoW) it's based on the **longest chain rule**. The Longest chain rule in simple terms states that the valid chain is the one with the most accumulated PoW (which will be the longest chain, with the greatest number of blocks). After all, mining blocks requires work so more blocks mined means more work done, so the longest chain with the most proof of work is accepted as the valid chain by all nodes.

The system is designed such that as long as at least 50% of the hash rate of the network is controlled by honest miners then the network will converge on a single truthful version of the chain. However, if an entity controls 51% of the hash power it could be malicious and cause disruption and fraudulently double spend transactions. The decentralized nature of the network mitigates this and due to the huge computational power required to mine blocks it would be too expensive to garner the required amount of equipment and power to do this in practice. This is essentially why Bitcoin is deemed the most secure network in the world.

In summary, the PoW mechanism make sure that the network retains security by making it difficult to alter the blockchain because it's very time consuming and expensive to do so. This is because it requires a majority of the networks computing power to agree on the state of the blockchain which makes it more resistant to malicious behavior. The main point is that it's easy for other nodes to verify the work done but very resource intensive to produce that work.

Now that PoW has been outlined let's put all this together and show a Bitcoin transaction where some BTC is sent from one wallet to another. A block is proposed with the transaction added and PoW is performed to then add the block to the blockchain:



The PoW step towards the end is all detailed in the PoW diagram shown earlier. This puts all the pieces of the puzzle together. So, this is the overall summary in context of a transaction:

- Once the transaction is initiated and signed with the private key, it's broadcast to the network for other nodes to verify.
- A node uses the sender's public key to verify that the sender of the transaction is legitimate. It also does other verification to verify the transaction, such as verifying the transaction's hash, available balance and that the transaction hasn't been spent twice. The transaction is added to the mempool.
- Meanwhile miners are building a block to be proposed and are extracting transactions from the mempool and inserting them in the proposed block.
- The miners perform PoW by hashing the header of the proposed block. The header consists of the Merkle Root for all transactions in the block.
- If the miner's work satisfies the difficulty target (a hash with a certain number of leading zeros), it can broadcast this proposed block to the network.
- Once the block is verified, its linked on to the blockchain. The miner receives a block reward and the transaction fees in BTC.

The mechanics of the transaction verification is described in more detail in the next section on Merkle Trees and Merkle Roots.

Merkle Trees and Merkle Roots

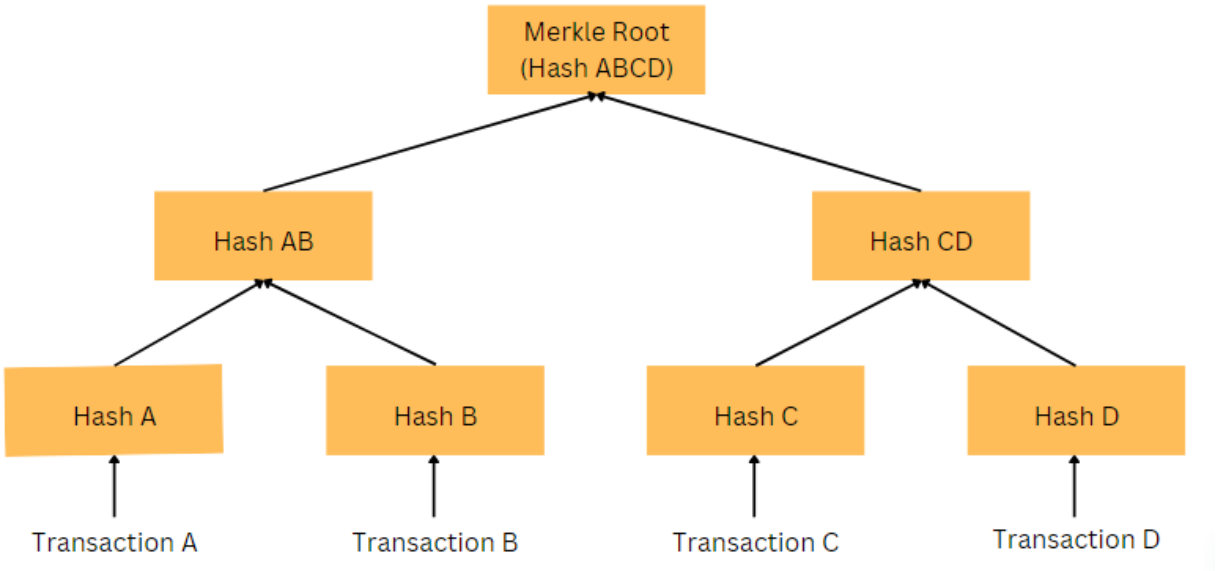
Merkle Trees and Merkle Roots are a fundamental part of blockchain technology and deserve more explanation as they have been mentioned so far and will be in other chapters in this book.

A Merkle Tree is a binary data structure that is used to verify the integrity of data effectively. Each part of the tree has a leaf node where each node represents a block of data, but rather than containing the data directly it contains the hash of the child nodes. The tree is built gradually by hashing pairs of child nodes until a single root hash is obtained, known as the Merkle Root.

Hashing and hash functions are essentially a mathematics concept that take data and apply it in a way such that once hashed the original contents cannot be obtained, making it a one-way function. The hash output is a condensed representation of that data and if the contents of the data were ever to be changed, the hash would change which invalidates the contents and anything (such as child nodes) connected to it. This is why hashing the data is good for verifying the integrity of data as a small change to the data results in a completely different hash output.

Due to the nature of hashing data this allows the verify the authenticity of the data without needing to check each part of the data itself. Once this is done, you now have a Merkle Proof whereby the integrity of the data can be verified for large datasets without the need for individual inspection, which is much more efficient.

The diagram below illustrates this for a set of transactions and shows the transactions, the hash of those transactions and the hash of those hashes (in pairs) until the Merkle Root is obtained. This is included in the block header and used for verification by other nodes that all transactions in the block are included and intact:



The Merkle Root represents the top of the whole tree and this is used to store block headers without requiring an entire block of information. This is also used for SPV nodes to verify data using the block header as it's more memory efficient.

The block header contains the Merkle Root (and other information such as the nonce and timestamp) and this is then all hashed (by continuously hashing the header via PoW) to form the block hash. This is then used as part of the blockchain where it points to the hash of the previous block, thus linking all the blocks together, hence the name blockchain. If a block hash changes (perhaps because data has been altered maliciously), it affects the hash of all other blocks which invalidates the chain. This is why hashing is such an efficient way to verify data integrity and to check that nothing has been tampered with.

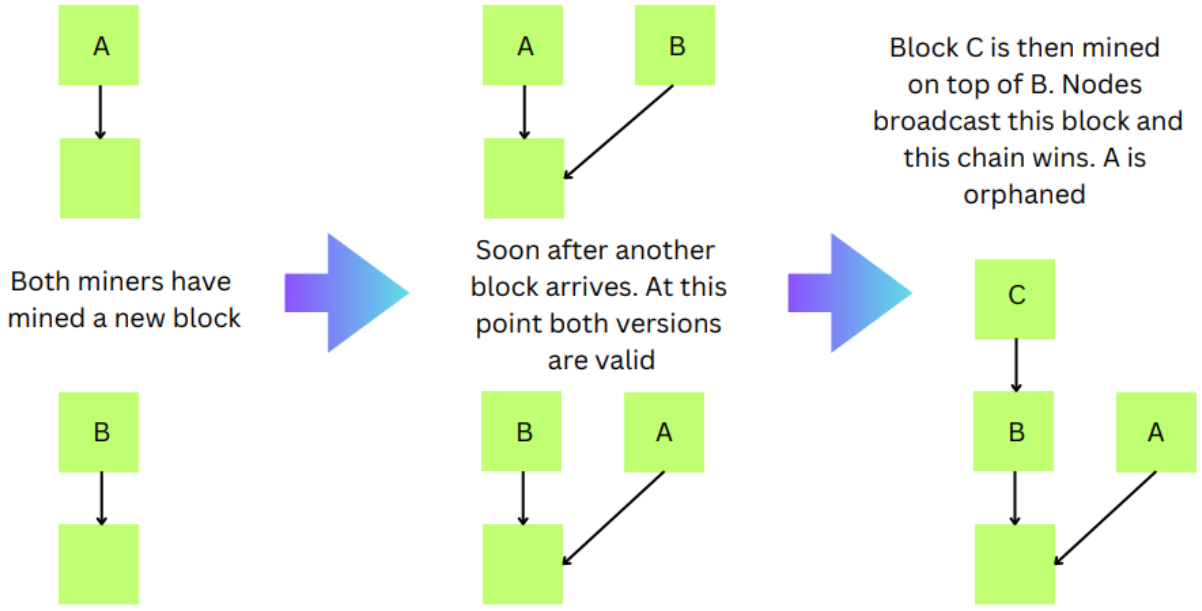
When two blocks are mined at the same time

There can be a case where two blocks are mined simultaneously. If this happens a temporary fork (or split) in the chain occurs. Let's say there are two blocks, block A and block B mined at the same time. This is what happens:

1. Block A is broadcast to the network and so is block B. Nodes accept and validate the first block they receive.
2. At this point the blockchain has diverged into two paths, one path with block A and another with block B, this is the point where the fork occurs.

3. Soon after nodes that received block A first, they may receive block B. However, it won't switch to block B unless it becomes part of the **longest chain**. This is a rule as part of Bitcoin's consensus mechanism.
4. Mining nodes whose view has block A at its tip of the chain will begin mining a block that extends block A. Likewise, those nodes who have block B at its tip will begin mining a block to extend block B. They are voting with their hash power. Now a race begins.
5. Let's say a node mines a new block, C, on top of B. This is propagated to the network and so nodes with blockchain containing block B at its tip receive a newly mined block, block C. This is deemed the longest chain, so blockchain B wins.
6. This is where a **reorg** (reorganization) occurs in that the nodes that had the block A blockchain now reorganize the blocks by discarding block A and linking block B, then block C to its chain. So, the nodes that were on the shorter chain, chain A, will switch to the longer chain, chain B.
7. Block A becomes an orphaned block.
8. Miners will now continue to mine blocks using block C as their parent (or previous block).
9. The transactions in block A may return to the mempool if not already in another block.

The following illustrates this situation whereby chain B becomes the winning chain with block C mined on top of block B:



Selfish Mining and Red Balloons

The **Red Balloons** concept is a type of game theory which refers to incentives to share information quickly whereby the faster that miners broadcast blocks the more likely they will earn the reward. If they hold on to the block too long, they risk another miner finding a longer chain and lose the reward. It also prevents delays and promotes faster finality of blocks.

In PoW blockchains there is a concept called **selfish mining** where a selfish miner or pool can gain an unfair advantage by choosing to withhold blocks and propagate them later. This maybe because there is a block with lucrative fees that the miner doesn't want to share and keep to himself or perhaps that miner colluded with another in a deal where one miner paid the other through a private channel.

It's difficult to succeed from doing this because the miner has to mine a block (by solving the mining puzzle) but not broadcast it immediately and instead they continue mining on their private chain. If they mine another block that extends their private chain, then as soon as the public chain catches up the miner publishes their longer chain forcing the honest miners to discard their work (as the longest chain wins). The attacker now has increased his share of the rewards. However, this takes a lot of hash power (maybe 30% of the total network at least) as essentially the miner has to mine two blocks ahead of the public chain.

There have been possible instances in the past where selfish mining may have taken place, for example in the Ghash.io pool in 2014 – 2015 where it controlled 50% of the hash power raising concerns of a 51% attack or selfish mining. There were also allegations this was occurring in the F2Pool in 2019 as they seemed to frequently win blocks suggesting manipulation, but there was no conclusive proof.

Despite no conclusive evidence, there are possibilities of selfish mining and fee sniping (another variant of selfish mining) in some blockchains and in the future it's still possible. The way to prevent this happening is to incentivize users to propagate blocks quickly and punish them for not doing so. In PoS blockchains there are penalties and slashing for doing so, which is why it's less of an issue for these types of blockchains. It's also less relevant because in PoS validators are chosen based on their stake, not computational power, so there's no race to solve puzzles, so withholding a block doesn't offer the same advantage.

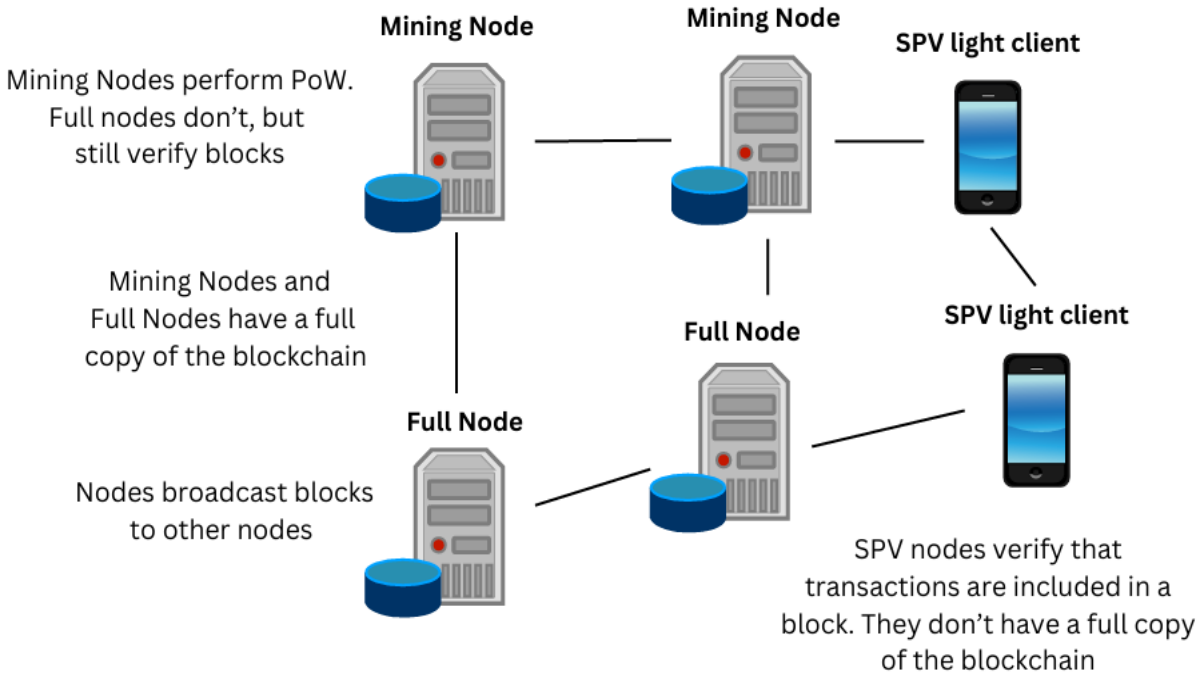
However, PoS aligns incentives differently and withholding blocks is still theoretically possible as validators can still collude to manipulate consensus especially if a significant portion of validators withhold their blocks at the same time. It's still very difficult to pull off and is less effective than in PoW blockchains, but there could still be a motivation if the fees in the block are looking juicy!

In Proof of Routing Work consensus mechanisms (discussed later in the book) there is no point to even attempt it because nodes are rewarded for routing transactions quickly whereby the Routing Work score for that node increases with the higher number of transactions that have

propagated through the network. Therefore, nodes are punished immediately because if they don't broadcast their transactions their Routing Work score will be lower, meaning they will most certainly lose the rewards (which in Routing Work are the fees in the block). This creates incentives that directly align with red balloons.

Bitcoin Network Architecture

The following illustrates the Bitcoin architecture that consists of mining nodes, full nodes and light (SPV) nodes:



There is no single blockchain but rather every node has its own copy of the blockchain. The blue disk in the image shows a copy of the blockchain. Think of a blockchain essentially as a database with a store of all the blocks containing transactions with all the required information.

The exception is light client SPV nodes that contain the block headers only. They can still do verification of transactions through Merkle Proofs which link individual transactions to the block's Merkle Root (as the Merkle Root is contained in the block header).

What prevents someone creating a different version of Bitcoin?

A common question that gets asked is what stops someone simply creating a new blockchain with a new token for example, called BTC2, with the view of making a ton of money?! The answer is there is nothing to stop somebody creating a new chain but it's unlikely to succeed. This is because if somebody creates a new version of Bitcoin today it's likely it will be heavily attacked and compromised and so result in double spending of transactions and fraudulent activity. Bitcoin had some luck because when it first launched nobody took it seriously so nobody attempted to hack it. Then as more miners joined the Bitcoin network this meant more miners did work to solve the mathematical puzzle to create a block of transactions. This secures the network and miners received a block reward in BTC for little power consumption at the time. This steadily increased the security of the network because the amount of computational power required to solve the mathematical puzzle (a hash value with a certain number of leading zeroes) increased as more miners joined the network, thus increasing the hash rate. The **hash rate** is the amount of computational power (measured in hashes per second) that miners contribute to secure the network.

As a result, there became a point where the difficulty and hash rate were so high that it was no longer cheap to mine BTC and certainly no longer cheap to attack it due to the amount of computational power required. Any attack nowadays is simply too expensive and ludicrous to attempt.

A new version of Bitcoin could be easily attacked because it will be so cheap to carry out as there will only be a few miners at the start so a 51% attack (essentially 51% control of the hash rate) will be much more feasible. This is partly why there has been no direct alternative to Bitcoin launched with any success. The only real direct competitors that are similar are the forked versions being Bitcoin Cash and Bitcoin Satoshi's Vision and these have a much lower hash rate. They still have a decent level of security however, but not to the extent that Bitcoin has.

Other factors are that Bitcoin has essentially become a brand name and it also has a strong network effect. Once a network effect is strong, it's very difficult to displace unless a solution arises that is significantly better. The word "significantly" should be emphasized here because some solutions that are marginally better may not result in a switch of user base. Usually, if a protocol is good enough, never been compromised and has stood the test of time few users will switch to another network or protocol that hasn't stood the test of time. A similar case existed for the Internet where once TCP/IP and SMTP protocols (for email) got the network effect, even though there were perhaps some better solutions, the user base was already content with TCP/IP and SMTP.